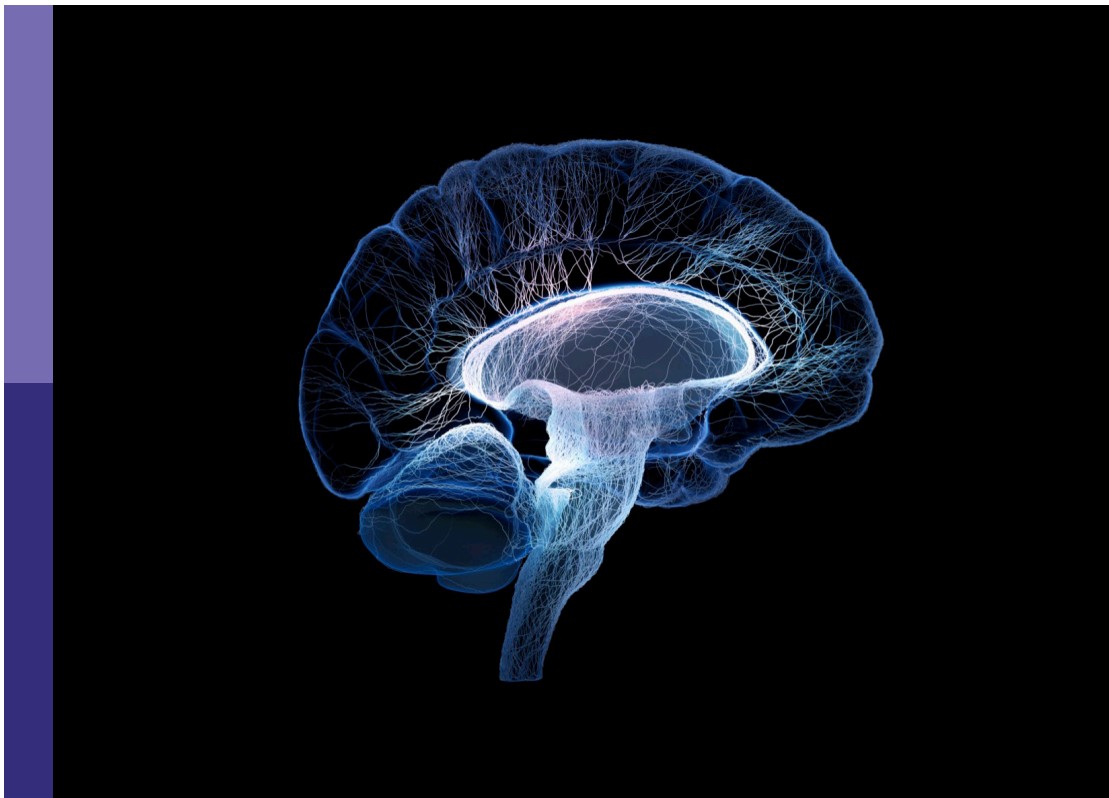
frontiers | Research Topics

Brain-inspired cognition and understanding for next-generation AI: Computational models, architectures and learning algorithms

Edited by
Chenwei Deng, Guang-Bin Huang and Yuqi Han

Published in
Frontiers in Neuroscience
Frontiers in Computational Neuroscience



ISSN 1664-8714
ISBN 978-2-83252-116-8
DOI 10.3389/978-2-83252-116-8

About Frontiers

Frontiers is more than just an open access publisher of scholarly articles: it is a pioneering approach to the world of academia, radically improving the way scholarly research is managed. The grand vision of Frontiers is a world where all people have an equal opportunity to seek, share and generate knowledge. Frontiers provides immediate and permanent online open access to all its publications, but this alone is not enough to realize our grand goals.

Frontiers journal series

The Frontiers journal series is a multi-tier and interdisciplinary set of open-access, online journals, promising a paradigm shift from the current review, selection and dissemination processes in academic publishing. All Frontiers journals are driven by researchers for researchers; therefore, they constitute a service to the scholarly community. At the same time, the *Frontiers journal series* operates on a revolutionary invention, the tiered publishing system, initially addressing specific communities of scholars, and gradually climbing up to broader public understanding, thus serving the interests of the lay society, too.

Dedication to quality

Each Frontiers article is a landmark of the highest quality, thanks to genuinely collaborative interactions between authors and review editors, who include some of the world's best academicians. Research must be certified by peers before entering a stream of knowledge that may eventually reach the public - and shape society; therefore, Frontiers only applies the most rigorous and unbiased reviews. Frontiers revolutionizes research publishing by freely delivering the most outstanding research, evaluated with no bias from both the academic and social point of view. By applying the most advanced information technologies, Frontiers is catapulting scholarly publishing into a new generation.

What are Frontiers Research Topics?

Frontiers Research Topics are very popular trademarks of the *Frontiers journals series*: they are collections of at least ten articles, all centered on a particular subject. With their unique mix of varied contributions from Original Research to Review Articles, Frontiers Research Topics unify the most influential researchers, the latest key findings and historical advances in a hot research area.

Find out more on how to host your own Frontiers Research Topic or contribute to one as an author by contacting the Frontiers editorial office: frontiersin.org/about/contact

Brain-inspired cognition and understanding for next-generation AI: Computational models, architectures and learning algorithms

Topic editors

Chenwei Deng — Beijing Institute of Technology, China

Guang-Bin Huang — Nanyang Technological University, Singapore

Yuqi Han — Tsinghua University, China

Citation

Deng, C., Huang, G.-B., Han, Y., eds. (2023). *Brain-inspired cognition and understanding for next-generation AI: Computational models, architectures and learning algorithms*. Lausanne: Frontiers Media SA. doi: 10.3389/978-2-83252-116-8

Table of contents

frontiers | Frontiers in Neuroscience

TYPE Editorial
PUBLISHED 24 March 2023
DOI 10.3389/fnins.2023.1169027

OPEN ACCESS

EDITED AND REVIEWED BY
Rufin VanRullen,
Centre National de la Recherche Scientifique (CNRS), France

*CORRESPONDENCE
Chenwei Deng
✉ cwdeng@bit.edu.cn

SPECIALTY SECTION
This article was submitted to Perception Science, a section of the journal Frontiers in Neuroscience

RECEIVED 18 February 2023
ACCEPTED 13 March 2023
PUBLISHED 24 March 2023

CITATION
Han Y, Deng C and Huang G-B (2023) Editorial: Brain-inspired cognition and understanding for next-generation AI: Computational models, architectures and learning algorithms. *Front. Neurosci.* 17:1169027. doi: 10.3389/fnins.2023.1169027

Editorial: Brain-inspired cognition and understanding for next-generation AI: Computational models, architectures and learning algorithms

Yuqi Han[1,2], Chenwei Deng[1]* and Guang-Bin Huang[3]

[1]School of Information and Electronics, Beijing Institute of Technology, Beijing, China, [2]Department of Computer Science and Technology, Tsinghua University, Beijing, China, [3]School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore, Singapore

KEYWORDS

perception science, brain-inspired cognition, artificial intelligence, multimodal perception, machine learning

Editorial on the Research Topic
Brain-inspired cognition and understanding for next-generation AI: Computational models, architectures and learning algorithms

1. Introduction

The human brain is probably the most complex thing in the universe. Apart from the human brain, no other system can automatically acquire new information and learn new skills, perform multimodal collaborative perception and information memory processing, make effective decisions in complex environments, and work stably with low power consumption. In this way, brain-inspired research can greatly advance the development of a new generation of artificial intelligence (AI) technologies.

Powered by new machine learning algorithms, effective large-scale labeled datasets, and superior computing power, AI programs have surpassed humans in speed and accuracy on certain tasks. However, most of the existing AI systems solve practical tasks from a computational perspective, eschewing most neuroscientific details, and tending to brute force optimization and large amounts of input data, making the implemented intelligent systems only suitable for solving specific types of problems. The long-term goal of brain-inspired intelligence research is to realize a general intelligent system. The main task is to integrate the understanding of multi-scale structure of the human brain and its information processing mechanisms, and build a cognitive brain computing model that attempt to simulate the cognitive function of the brain. In particular, attention needs to be paid to how the human brain cooperates with different computing components to accomplish different cognitive tasks such as perception, attention, learning, memorizing, knowledge representation, reasoning, decision-making, and judgment.

This special issue contains 14 research articles, which could be broadly classified into three classes: (1) three articles focus on investigating the spiking neural networks to explore the working mechanism for human brain, (2) three articles review several existing machine learning techniques and models by refering the working manner from human brain, (3) the remaining articles mainly focus on some practical applications such 3D modeling, robotics, speech recognition and image processing.

Specifically, Skatchkovsky et al. proposed a Bayesian learning framework for spiking neural networks (SNNs), which utilizes a Gaussian variational distribution for synaptic weights and a Bayesian single-task and continual learning rules with binary weights. Their study shows that the proposed framework has the ability to adapt to changing learning tasks and provides reliable quantification of uncertainty in the model's decisions. Hu and Liao proposed a membrane voltage slope-guided algorithm (VSG) that correlates delayed feedback signals with effective clues embedded in background spiking activity. This method finds potential points for emitting new spikes and the old spikes that need to be removed from the time derivative of membrane voltage, thereby avoiding the dilemma of failing to find adjustment points. Furthermore, it does not require iterative calculation to find the critical threshold. Zhao and Zeng proposed an intention prediction model for robots, which enables them to successfully predict user intentions through the spike-timing-dependent plasticity (STDP) mechanisms and simple feedback of right or wrong. Compared with the traditional Q-learning method, the proposed model significantly reduces training time.

Wingfield et al. proposed a deep artificial neural network model for speech processing that bears resemblance to patterns of activation in the human auditory cortex. This was achieved through a combination of spatio-temporal searchlight representational similarity analysis (ssRSA) and multimodal neuroimaging data. The study concludes that the low-dimensional bottleneck layer in the DNN could learn representations that characterize articulatory features of human speech. According to the study of Vaskevich and Torres, statistical learning is a highly dynamic and stochastic process that unfolds at different time scales, and evolves distinct learning strategies on demand. Their research reassesses how individuals dynamically learn predictive information in stable and unstable environments. Specifically, narrow-variance learners retain explicit knowledge of the regularity embedded in stimuli and use an error-correction strategy consistently in both stable and unstable environments. Broad-variance learners, on the other hand, emerge only in unstable environments. Lee et al. investigated brain-inspired predictive coding frameworks for machine challenging tasks (MCTs) and found that they have advantages in incremental learning, long-tailed recognition, and few-shot recognition. The study concludes that predictive coding learning is similar to the plasticity-stability property of the human brain, and mainly mimics the interaction between the hippocampus and prefrontal cortex.

For the practical application, Kumari et al. proposed an attentional search model for practical application in a 3D environment, utilizing two separate channels for object classification and location prediction. This enables the camera to accurately classify the target while focusing on it. Their model employs Elman and Jordan recurrence layers as well as JK-flip-flop recurrence layers instead of the traditional Long Short Term Memory (LSTM) to integrate temporal attention history into the network.

In the field of remote sensing, Shi et al. proposed an improved anchor-free SAR ship detection algorithm inspired by the brain's ability to effectively focus on target information and ignore interference from redundant information. The proposed model utilizes dense connection in the deep layer of the network and visual attention processing in the shallow layer to enhance feature extraction ability. Moreover, a wide height prediction constraint is applied to the target to further improve localization accuracy. Wang et al. proposed a knowledge-assisted neural network for millimeter wave radar object classification. This model injects two kinds of prior information containing spatial and physical understanding of objects for assistance. With the guidance of prior information, the network can learn object classification more akin to human brains and achieve superior performance. Tong et al. proposed an interpretable approach for automatic aesthetic assessment of remote sensing images. This method can highlight important areas of the image that influenced the model's decision, and provide visual explanations of the remote sensing aesthetic assessment.

Drawing inspiration from the way humans learn different object features based on the backgrounds and use historical appearances to aid in target positioning during tracking, Cui et al. proposed a novel tracking algorithm based on dynamic feature selection, aberrance repression, and a historical model retrieval module. By introducing dynamic feature-channel and aberrance repressed regularization into the loss function, the tracker can learn different feature weights according to the difference between the target and the background. The memory module, built by historical target samples, allows the tracker to learn a flexible representation that adapts to changes in object appearance during tracking. Similarily, Zheng et al. proposed a novel Gaussian prototype learning model to address tactile object recognition in open-set scenarios. Their unified framework integrates classification and class detection, consisting of two main components: a feature extractor and class prototypes. The feature extractor simulates the human perception system for transforming raw sensing data into abstract representations, and the prototypes for each category serve as abstract memories of the corresponding category in the brain. Experimental results validate that their model can not only correctly classify known tactile inputs but also effectively detect unknown tactile classes.

Current deep learning-based fundus image registration methods attempt to learn the geometric transformation or dense pixel-level displacement vector field directly between the test and reference images. However, the significant intra-class variability and small inter-class differences of fundus images pose a significant challenge for accurate keypoint matching. In response to this challenge, Xu et al. has proposed a spatially-varying adaptive pyramid context aggregation network to simultaneously match all the vessel crossing and branching points by taking advantage of the knowledge of contextual consistency.

Li et al. has proposed a kinship verification method based on face images that is relevant to real-life applications, such as missing children search, family photo classification, and kinship information mining. To enable the deep model to capture diverse

and abundant local features from different regions of the face, they have proposed an attention center learning guided multi-head attention mechanism. To address the issue of misclassification caused by single feature center loss, a family-level multi-center loss has been proposed to ensure a more appropriate intra/inter-class distance measurement for kinship verification.

These articles cover a wide variety of topics including encoding and decoding of spatial-temporal information, 3-D environment modeling, visual object detection and localization, speech recognition, and aesthetic assessment. From the perspective of brain-inspired intelligence, these researches enrich the corresponding research fields with insightful methodologies and techniques, and ultimately offering alternative solutions to effectively enhance the robustness, generalization ability, and interpret ability for related tasks.

We hope that our readers will have a delightful experience when reading these excellent articles.

Author contributions

YH prepared the orginal draft. CD and G-BH critically reviewed and edited the manuscript. All authors have reviewed and approved of the final manuscript.

Funding

This work was supported by National Natural Science Foundation of China (NSFC) under Grant 62171040 and the China Postdoctoral Science Foundation under Grant 2021TQ0177.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

frontiers | Frontiers in Neuroscience

TYPE Original Research
PUBLISHED 21 October 2022
DOI 10.3389/fnins.2022.1009237

OPEN ACCESS

EDITED BY
Yuqi Han,
Tsinghua University, China

REVIEWED BY
János Botzheim,
Eötvös Loránd University, Hungary
Joseph Thachil Francis,
University of Houston, United States

*CORRESPONDENCE
Yi Zeng
yi.zeng@ia.ac.cn

†These authors have contributed equally to this work

SPECIALTY SECTION
This article was submitted to Perception Science, a section of the journal Frontiers in Neuroscience

RECEIVED 01 August 2022
ACCEPTED 04 October 2022
PUBLISHED 21 October 2022

CITATION
Zhao Y and Zeng Y (2022) A brain-inspired intention prediction model and its applications to humanoid robot.
Front. Neurosci. 16:1009237.
doi: 10.3389/fnins.2022.1009237

A brain-inspired intention prediction model and its applications to humanoid robot

Yuxuan Zhao[1†] and Yi Zeng[1,2,3,4*†]

[1]Research Center for Brain-inspired Intelligence, Institute of Automation, Chinese Academy of Sciences, Beijing, China, [2]Center for Excellence in Brain Science and Intelligence Technology, Chinese Academy of Sciences, Beijing, China, [3]National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, China, [4]School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, China

With the development of artificial intelligence and robotic technology in recent years, robots are gradually integrated into human daily life. Most of the human-robot interaction technologies currently applied to home service robots are programmed by the manufacturer first, and then instruct the user to trigger the implementation through voice commands or gesture commands. Although these methods are simple and effective, they lack some flexibility, especially when the programming program is contrary to user habits, which will lead to a significant decline in user experience satisfaction. To make that robots can better serve human beings, adaptable, simple, and flexible human-robot interaction technology is essential. Based on the neural mechanism of reinforcement learning, we propose a brain-inspired intention prediction model to enable the robot to perform actions according to the user's intention. With the spike-timing-dependent plasticity (STDP) mechanisms and the simple feedback of right or wrong, the humanoid robot NAO could successfully predict the user's intentions in Human Intention Prediction Experiment and Trajectory Tracking Experiment. Compared with the traditional Q-learning method, the training times required by the proposed model are reduced by $(N^2 - N)/4$, where N is the number of intentions.

KEYWORDS

human-robot interaction, intention prediction, brain-inspired model, spiking neural networks, humanoid robot

1. Introduction

The research trend of the new generation of robots is to make robots participate in human life, and improve the naturalness and flexibility of interaction between humans and robots through human-robot interaction technology. Robots that can successfully predict user's intention and take appropriate actions according to the intention can effectively improve interaction efficiency and user experience. Researchers have made significant progress in user intention prediction modeling. These studies use a variety of frameworks or models to enable robots to predict users' intentions in specific human-robot interaction tasks. These frameworks or models use a variety of methods, such as probabilistic graphical models, deep learning techniques, and other methods that include extreme learning machine algorithms, etc.

There are many studies on the application of the probability graph model to human intention prediction. Song et al. (2013) proposes a probabilistic graphical model for predicting human manipulation intention from image sequences of human-object interaction. The model can enable the robot to successfully infer intention in a house-hold task which contains four intentions: hand-over, pouring, tool-use, and dish-washing. Vinanzi et al. (2019) proposes a novel artificial cognitive architecture to predict the intentions of a human partner. The architecture contains unsupervised dynamical clustering of human skeletal data and a hidden semi-Markov chain. With the architecture, the iCub robot can engage in cooperative behavior by performing intention reading based on the partner's physical clues. Besides that, Yu et al. (2021) proposes a Bayesian method for human motion intention in a human-robot collaborative task. Dermy et al. (2017) and Luo and Mai (2019) built models based on Probabilistic Movement Primitives for human intention prediction. Their models are verified in gaze guidance experiment or tabletop manipulation task.

Deep learning techniques, especially deep long short-term memory (LSTM) neural network, have also been used to predict human intentions. Yan et al. (2019) presents an LSTM neural network to recognize human intention. They designed a human-robot collaboration environment using a UR5 robot and a Kinect V2 depth camera. The experimental results show that the 2-layers deep LSTM network enables the robot to understand the human intentions even with only 40% of the motion sequences. Liu et al. (2019) presents a deep learning system combing convolutional neural network (CNN) and LSTM, and this system could accurately predict the motion intention of the human in a desktop disassembly task.

In addition, there are other methods for human intention recognition. Wang et al. (2021) proposes a teaching-learning-prediction (TLP) framework, which enables robots to learn and predict human hand-over intentions in collaborative tasks. The robot learns the human demonstrations *via* the extreme learning machine (ELM) algorithm, which realizes the robot's learning and prediction of human hand-over intentions in collaborative tasks. The experimental results show that the framework can enable the robot to effectively predict the human hand-over intention and complete the hand-over task. Since the framework enables robots to learn through human demonstrations, it can reduce human manual-programming efforts and improve the efficiency of human-robot collaboration. Lin et al. (2017) develops a human intention recognition framework in human-robot collaboration scenarios. The framework contains an inverse-reinforcement learning system to find the optimal reward function of the policy and a Markov-Decision process to model human intention. They use a coffee-making task and a pick-and-place task to verify the validity of the model and obtained the desired results. Li et al. (2020) proposes a task-based framework to enable robots to understand human intention from natural language dialogues. The framework includes a language semantics module for extracting instruction keywords, a visual object recognition module for identifying objects, and a similarity computation module for inferring intention based on the given task. With this framework, the robot could comprehend human intentions using visual semantics information.

It can be seen that most of the current studies use relatively complex methods to complete specific human-robot interaction tasks, and few studies use brain-inspired cognitive computational modeling methods to solve intention prediction tasks. Brain-inspired cognitive computational modeling is a method that draws on the results of neuroimaging studies on cognitive tasks, proposes feasible neural pathways and network structures, and conducts modeling based on the spiking neuron model.

Here, based on the neuroimaging studies of reinforcement learning, we propose a brain-inspired intention prediction model to enable the robot to perform actions according to the user's intention. Based on the brain-inspired network structure, the humanoid robot NAO could successfully predict the user's intentions in Human Intention Prediction Experiment and Trajectory Tracking Experiment only by using the spike-timing-dependent plasticity (STDP) mechanisms and the simple feedback of right or wrong.

2. Materials and methods

2.1. Architecture of the brain-inspired intention prediction model

The architecture of the brain-inspired intention prediction model is shown in Figure 1.

The dorsolateral prefrontal cortex (DLPFC) is responsible for representing state information (Barbey et al., 2013). In our computational model, the DLPFC receives input from visual cortex and abstractly represents the visual information.

Popular theories implicate that the basal ganglia (BG) are responsible for action selection (Stocco et al., 2010; Friend and Kravitz, 2014). The striatum D1 (StrD1) and striatum D2 (StrD2) are components of BG (Villagrasa et al., 2018). In our computational model, the BG is used for intention prediction, that is, BG selects the actions that conform to the user's intention according to the visual information represented by DLPFC.

The thalamus is generally considered to be a relay station, transmitting information between different cerebral cortex (Hwang et al., 2017). In our computational model, the thalamus acts as a relay station to transmit information from BG to PMC and OFC.

The primary motor cortex (PMC) is a critical area for controlling the execution of movement (Kakei et al., 1999). In our computational model, the PMC is used to control the actions of the robot.

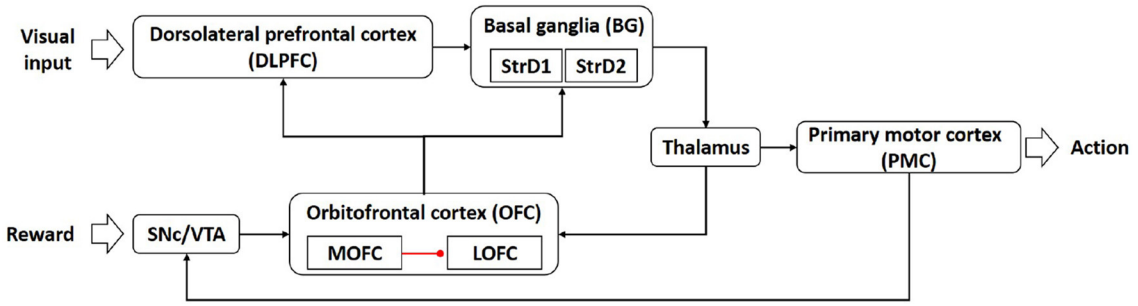


FIGURE 1
The architecture of the brain-inspired intention prediction model.

The substantia nigra pars compacta and ventral tegmental area (SNc/VTA) play important roles in reward cognition (Haber and Knutson, 2010; Zhao et al., 2018). In our computational model, the SNc/VTA receives the user's feedback and determines the pathway of information transmission. When the feedback information is positive, SNc/VTA combines the information from PMC and transmits the stimulation to OFC_2 (a sub-region in orbitofrontal cortex). When the feedback information is negative, the stimulation of SNc/VTA is 0.

The orbitofrontal cortex (OFC) is considered as a critical frontal region for memory formation (Frey and Petrides, 2002). The sub-region medial orbitofrontal cortex (MOFC) and lateral orbitofrontal cortex (LOFC) respond to positive reward (O'Doherty et al., 2001) and negative reward (Kringelbach, 2005). In our computational model, the OFC contains OFC_1 and OFC_2, MOFC and LOFC. The OFC_1 and OFC_2 are used to receive and store information from the thalamus and SNc/VTA. When the feedback information is positive, the MOFC receives stimulation from OFC_1, and the LOFC receives stimulation from OFC_2 and is inhibited by MOFC at the same time. When the feedback information is negative, only the LOFC receives the stimulus from OFC_1 and OFC_2. Then the MOFC transmits the information to DLPFC and StrD1 in BG, and the LOFC transmits the information to DLPFC and StrD2 in BG.

2.2. Model implementation

The concrete neural network architecture of the model is shown as Figure 2.

In order to describe one training process of the model more directly, in Figure 2, we use orange neurons, blue neurons and green neurons to represent the neurons be activated in one training process. **1. The intention prediction process:** (a) The visual information of image category *1* is input into DLPFC, and all neurons representing this category in DLPFC are activated (orange neurons in DLPFC); (b) After the synaptic weight matrix calculation between DLFPC and BG, the neuron representing intention *1* in BG is activated (orange neuron in BG); (c) Thalamus receives the results of BG (orange neuron in Thalamus) and passes the information to PMC to control motor generation (orange neuron in PMC). **2. The positive reward process**, if the user gives *positive reward* into SNc/VTA: (a) The OFC_1 receives the stimulation form Thalamus (blue neuron in OFC_1). The SNc/VTA combines the information from PMC and transmits the stimulation to OFC_2. And all neurons representing this action are activated (green neurons in OFC_2). (b) Then the stimulation of OFC_1 and OFC_2 are transmitted to MOFC (blue neuron in MOFC) and LOFC respectively, and LOFC is simultaneously inhibited by MOFC (green neurons in LOFC). (c) MOFC transmits the information to BG *via* StrD1 and to DLPFC at the same time. LOFC transmits the information to BG *via* StrD2 and to DLPFC at the same time. (d) The synaptic weight between DLPFC and BG is updated according to the time difference between the neurons firing. ***In short, the connection between image category 1 and intention 1 is strengthened, and the connections between image category 1 and other intentions are weakened.*** **3. The negative reward process**, if the user gives *negative reward* into SNc/VTA: (a) The OFC_1 receives the stimulation form Thalamus, then transmits the information to LOFC. (b) LOFC transmits the information to BG *via* StrD2 and to DLPFC at the same time. (c) The synaptic weight between DLPFC and BG is updated according to the time difference between the neurons firing. ***In short, the connection between image category 1 and intention 1 weakened, while the connections between image category 1 and other intentions remained unchanged.***

We use the Izhikevich neuron model to build the computational model. The Izhikevich neuron model achieves a good balance in biologically plausible and computational efficiency (Izhikevich, 2003). The neuron model is described as Equations (1) and (2). The variable v represents the membrane potential of the neuron and u represents a membrane recovery variable. And *a*, *b*, *c*, and *d* are dimensionless parameters. If the membrane voltage v is greater than 30 mV, the membrane

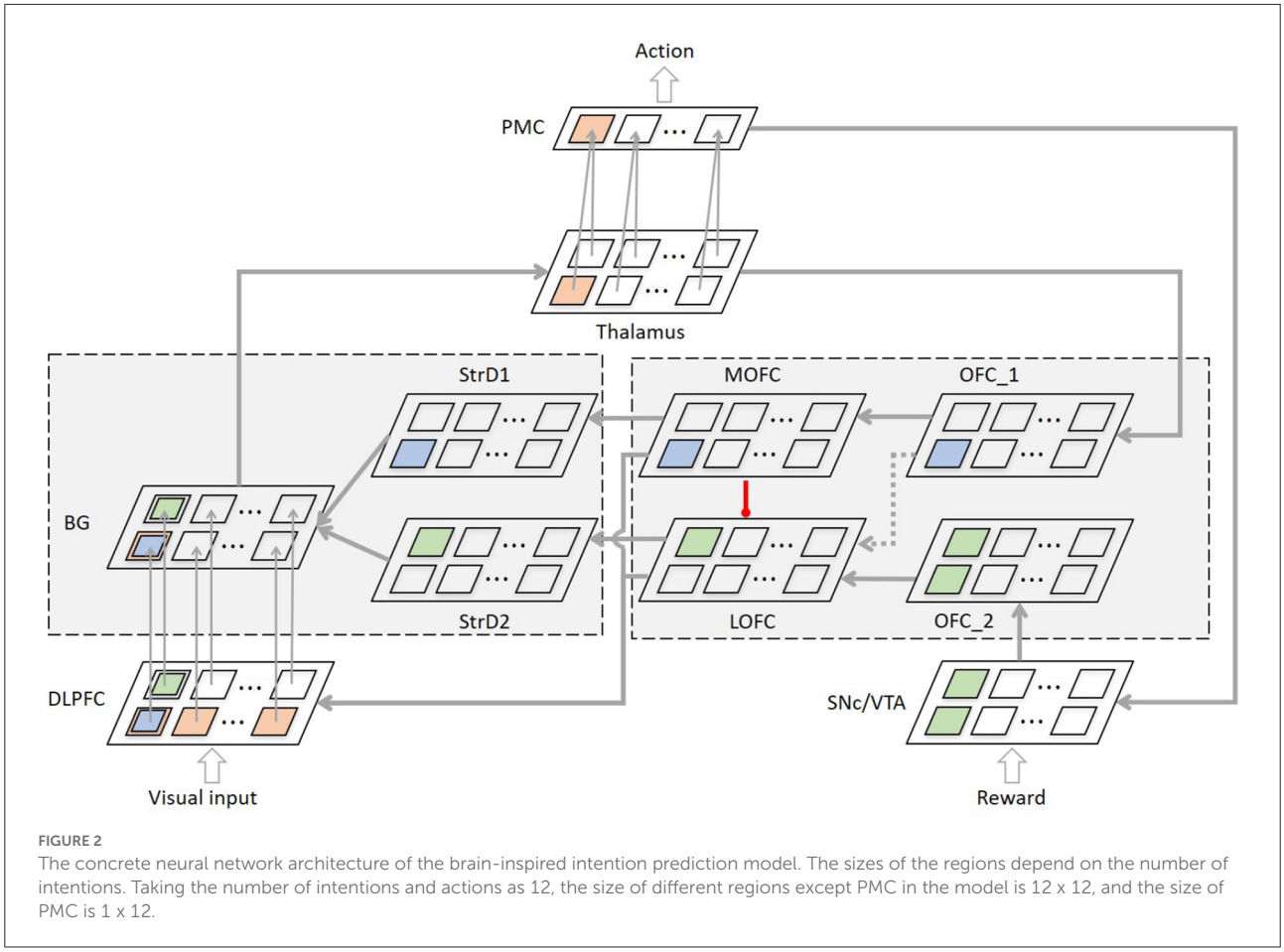


FIGURE 2
The concrete neural network architecture of the brain-inspired intention prediction model. The sizes of the regions depend on the number of intentions. Taking the number of intentions and actions as 12, the size of different regions except PMC in the model is 12 x 12, and the size of PMC is 1 x 12.

voltage and the recovery variable are reset according to the Equation (2). I is input, calculated by Equation (3). The W_{ij} is the synaptic weight between presynaptic neuron and postsynaptic neuron, and the O_{ij} is the output of presynaptic neuron. If multiple neurons fire at the same time, the neuron with the largest membrane voltage will inhibit other neuronal firings. The dimensionless parameters c and d are the same in different areas, they are $c = -65$ and $d = 8$. And the dimensionless parameters a and b of the neurons in StrD1, StrD2 and other areas are set as $a = 0.01$, $b = 0.01$; $a = 0.1$, $b = 0.5$; $a = 0.02$, $b = 0.6$, respectively.

$$\begin{aligned} v' &= 0.04v^2 + 5v + 140 - u + I \\ u' &= a(bv - u) \end{aligned} \tag{1}$$

$$if v \geq 30mV, then \begin{cases} v & \leftarrow c \\ u & \leftarrow u + d \end{cases} \tag{2}$$

$$\begin{aligned} I_{ij} &= W_{ij} \times O_{ij} \\ O_{ij} &= \begin{cases} 1 & if\ v_{ij} \geq 30 \\ 0 & otherwise \end{cases} \end{aligned} \tag{3}$$

Spike Timing Dependent Plasticity (STDP) is an important learning mechanism in the biological brain, which updates the synaptic weight between presynaptic and postsynaptic neurons according to the time difference between their firing (Bi and Poo, 2001). And the STDP mechanism is widely used in our previous work on brain-inspired cognitive computing modeling (Zeng et al., 2016, 2017, 2020; Zhao et al., 2021b). Within a millisecond time window, if the postsynaptic neurons are fired later than the presynaptic neurons, the synaptic weights between them increase, exhibit the long-term potentiation (LTP) mechanism; if the postsynaptic neurons are fired earlier than the presynaptic neurons, the synaptic weights between them decrease, exhibit the long-term depression (LTD) mechanism. The mathematical description of the STDP mechanism is shown in Equation (4). A_+ and A_- are the learning rates under the LTP mechanism and the LTD mechanism, respectively. τ_+ and τ_- are time constants for synaptic updates under the LTP mechanism and the LTD mechanism, respectively. To ensure the biologically plausible of the computational model, according to the results of the biological neuron fitting (Bi and Poo, 2001), we set A_+=0.777, A_-=-0.237, τ_+=16.8, and τ_-=-33.7 ms. To enable the robot to learn the user's flexible intentions more quickly, the synapse

weights are updated according to the ratio based on the current weight, as shown in Equation (5). In the computational model, synaptic plasticity occurs between DLPFC and BG, and the synaptic weight is fixed between other brain areas.

$$\Delta w = \begin{cases} A_+ \times e^{(\Delta t/\tau_+)} \ if \ \Delta t < 0 \\ A_- \times e^{(\Delta t/\tau_-)} \ if \ \Delta t \geq 0 \end{cases} \quad (4)$$
$$\Delta t = t_{\mathrm{DLPFC}} - t_{\mathrm{BG}}$$

$$W(t+1)_{ij} = W(t)_{ij} + W(t)_{ij} \times \Delta w \quad (5)$$

The stimulation transmitted from MOFC to BG and DLPFC exists as follows: the neurons in DLPFC fired first, and the neurons in StrD1 fired later, the Δt is less than 0. And the synaptic weight between DLPFC and BG increased, exhibiting the LTP mechanism. The stimulation transmitted from LOFC to BG and DLPFC exists as follows: the neurons in StrD2 fired first, and the neurons in DLPFC fired later, the Δt is greater than 0. And the synaptic weight between DLPFC and BG decreased, exhibiting the LTD mechanism. Therefore, when the user gives the right feedback, the weight of intention options selected by the model increases, while the weight of other candidate intention options decreases. When the user gives wrong feedback, the weight of intention options selected by the model decreases, while the weights of other candidate intention options are unchanged.

3. Results

We deploy the model on the humanoid robot NAO, and verify the effectiveness of the model through Human Intention Prediction Experiment and Trajectory Tracking Experiment.

3.1. Human intention prediction experiment

3.1.1. Experimental settings

The Human Intention Prediction Experiment allows the robot to predict human intentions through human gestures (the intention refers to the action that human expects the robot to perform), and to learn new intentions when human intentions change. After 12 gestures and 12 intentions are defined, the user can define the gesture-intention corresponding rules in his mind. The user makes gestures and the robot recognizes the gesture. Then the robot predicts the user's intention and performs the corresponding actions according to the proposed model. The user gives the right or wrong feedback according to whether the robot's action complies with his intentions. The robot can successfully predict the user's intention through multiple interactions. If some of the user's gesture-intention rules change, the robot can continue to learn those changed rules through interaction based on the learned model, and the unchanged rules are not affected, that is, the robot does not need to relearn all the rules.

The predefined 12 gestures are shown in Figure 3. Gesture A, both hands close to the body; Gesture B, single hand away from the body; Gesture C, single hand moves to the left; Gesture D, single hand moves to the right; Gesture E, single hand moves up; Gesture F, single hand moves down; Gesture G, both hands move down; Gesture H, single hand above the left shoulder; Gesture I, single hand above the right shoulder; Gesture J, both hands above both shoulders; Gesture K, left hand and left shoulder overlap; Gesture L, right hand and right shoulder overlap.

The predefined 12 intentions are shown in Figure 4. The intentions can be roughly divided into three categories: movement intentions (move forward, move backward, turn left, turn right, stand up, squat down and sit down), interaction intentions (clap the left palm, clap the right palm and clap both palms), and service intentions (beat the left shoulder and beat the right shoulder).

To make the experiment more intuitive, the initially defined gesture-intention correspondence rules are shown in Figure 5.

After learning the gesture-intention corresponding rules, the user modified the corresponding rules (as shown in Figure 6) to verify the flexibility of the model, and then the robot continued learning through interaction.

3.1.2. Experimental results

Considering that gesture recognition is not the focus of the proposed model, to recognize the user's gestures more simply, we used a Kinect camera for image acquisition. The Kinect camera can capture 25 user's joints and record their three-dimensional space coordinate. We defined 20 neurons to represent the movement direction of the left and right hands (upward movement, downward movement, left movement, right movement, close to the body and away from the body), as well as the position information of the left and right hands compared with the left and right shoulders (overlap with the left shoulder, overlap with the right shoulder, higher than the left shoulder and higher than the right shoulder). These neurons determine whether to fire based on the three-dimensional coordinate information of the joint.

After obtaining the gesture features, we use an unsupervised learning algorithm based on the STDP mechanism for gesture recognition. When a gesture is detected, the correlation coefficient of neurons firing pattern between the detected gesture and the learned gesture is calculated, and the activated target neuron is determined according to the correlation coefficient. If the correlation coefficient is very small, the gesture is determined as a new gesture, and a new target neuron is activated. The synaptic weights update between the new gesture and the new target are based on the STDP mechanism.

FIGURE 3
Predefined 12 gestures. From left to right are: Gesture A, Gesture B, Gesture C, Gesture D, Gesture E, Gesture F, Gesture G, Gesture H, Gesture I, Gesture J, Gesture K and Gesture L.

FIGURE 4
Predefined 12 intentions. From left to right are: Move forward, Move backward, Turn left, Turn right, Stand up, Squat down, Sit down, Clap the left palm, Clap the right palm, Clap both palms, Beat the left shoulder and Beat the right shoulder.

The method is an online learning method, and the recognition accuracy increases with the increase of the number of training samples. We define a trial training set that contains 12 types of gestures, each of the gestures is performed once. After the previous trial training ends, the next trial continues learning on the trained model. Test at the end of each trial. The test set consisted of 12 types of gestures performed 30 times each, with a total of 360 samples. When the training of the sixth trial is completed, the gesture recognition accuracy is 98.33%.

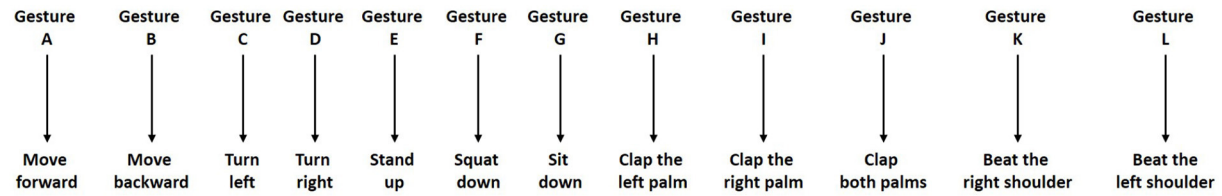


FIGURE 5
Gesture-intention corresponding rules.

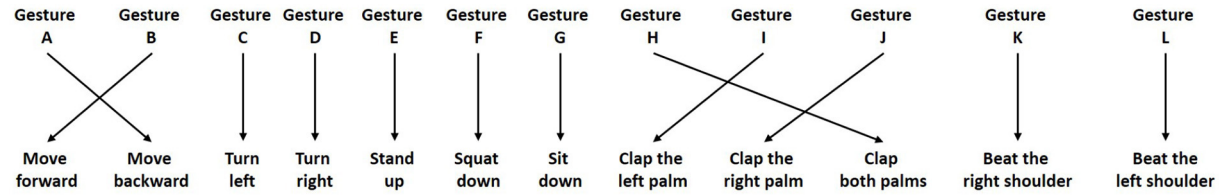


FIGURE 6
Changed gesture-intention corresponding rules.

The method is an online learning method, and the recognition accuracy increases with the increase of the number of training samples. The training set consists of repeated batch training sets. We define a batch training set that contains 12 types of gestures, each of the gestures is performed once. That is, there are 12 samples in a batch training set, which are different types of gestures. A batch training indicates that the model is trained on the batch training set. The online learning method of the model is realized in the following ways: after the previous batch training ends, the next batch continues learning on the trained model. Test at the end of each batch training. The test set consists of 12 types of gestures, each of which is executed 30 times. The test set includes 360 samples. When the training of the sixth batch is completed (that is, from the initial training, a total of six batches of training were carried out, each batch containing 12 gestures), the gesture recognition accuracy is 98.33%.

The user makes gestures randomly, and the robot predicts the user's intention and performs the corresponding action through the proposed model. Then, the user gives right or wrong feedback according to the robot's action and the initially defined gesture intention correspondence rules. The experiment is repeated many times until the robot could successfully predict the user's intention. The synaptic weights between DLPFC and BG are shown in Figure 7. In general, the number of interactions required to complete the training ranges from 12 (the robot successfully predicts the intention of the gesture each time) to 78 times (the robot tries all possible intentions until the last time to successfully predict the intention). In most cases,

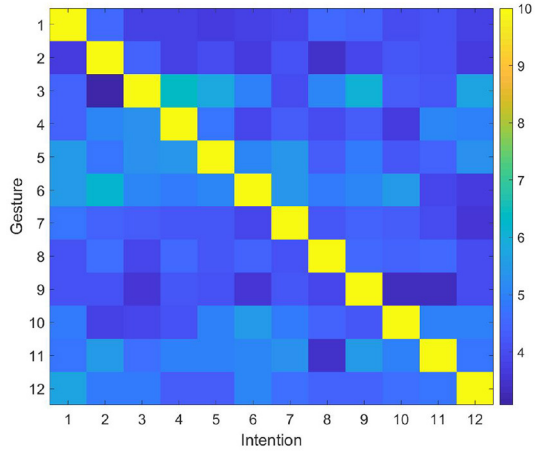


FIGURE 7
Synaptic weights between DLPFC and BG.

the robot needs 45 interactions to complete the training. Then the user modifies the corresponding rules, and gives feedback according to the modified rule. After modifying the rules, the synaptic weights between DLPFC and BG are shown in Figure 8. Since the synapse weights are updated according to the ratio based on the current weight, when the rules change, the robot can quickly forget the old rules. In general, after two interactions, the robot can forget the old rules and start learning the new ones.

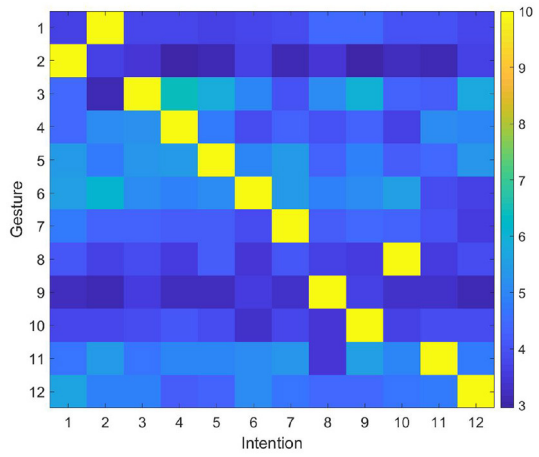


FIGURE 8
Synaptic weights between DLPFC and BG (Rules changed).

3.2. Trajectory tracking experiment

3.2.1. Experimental settings

The Trajectory Tracking Experiment can make the robot learn to walk along the track only through the right and wrong feedback of the remote control.

The training scenario and test scenario are shown in Figures 9A,B, respectively. In the training scenario, the robot makes behavioral decisions based on the collected image information and the proposed model, such as move forward, move backward, move left, move right, turn left and turn right. Then the user gives right or wrong feedback based on the robot's behavior, and gradually makes the robot learn to walk along the black track. The upper right corner of Figure 9A is the collected image information by the robot. Compared with the training scenario, the test scenario includes turn left and turn right behaviors in a tracking experiment Figure 10.

3.2.2. Experimental results

We detect the trajectory in the image through traditional image processing methods such as image binarization, edge detection, and Hough transform, and classify the trajectory according to its image characteristics. Finally, we use six neurons to implement an abstract representation of the results. The detection method is simple and effective, which can ensure the robot identifies the trajectory with high accuracy.

In the training scenario, the robot completed the training by walking two times along the trajectory clockwise and two times counterclockwise. In the test scenario, the robot can successfully complete the trajectory tracking experiment.

3.3. Compared with Q-learning method

First, we test the training times required by the Q-learning method (more details can be found in Supplementary material) and the proposed model under different number of intentions (form 1 to 9). All the intention-action corresponding rules are considered in the test process, and the number of rules corresponding to different intentions is shown in Table 1. The number of rules refers to the number of the intention-action corresponding rules, which is the total number of permutations of intention-action corresponding rules. For the intention-action corresponding rules under the same intention number, the training times required by the Q-learning method are fixed, while the training times required by the proposed method is slightly different according to different rules. In order to better compare the performance of the proposed method, the mode of training times [The Proposed Model (Mode)], minimum training times [The Proposed Model (Min)], and maximum training times [The Proposed Model (Max)] required by the proposed method under different rules are selected for comparison.

The result of detailed comparison between Q-learning method and the proposed model is shown in Figure 11. From Figure 11 and Table 1, it is easy to see that compared with the Q-learning method, the number of training times required by the proposed model decreases significantly with the increase of the number of intentions. Taking the number of intentions as 6 as an example, the number of all intention-action corresponding rules is 720. The Q-learning method requires 21 training times to complete the training, while the proposed model requires at least 6 times and at most 21 times under different rules. The mean of mode is 13.5 times. In general, the proposed model needs 13.5 times to complete training, which is 7.5 times less than Q learning method.

As can be seen from Figure 11, the training times of the proposed model under different rules are symmetrically distributed, so its mode is equal to the average value. Therefore, when the number of intentions is N, the improvement effect ($Train_{improve}$) of the proposed model on training times can be calculated by Equation (6). The $Train_Q$ is the training times required by Q-learning method, and the $Train_{BIIP}$ is the mode of training times required by the proposed model under the given intention numbers.

$$\begin{aligned} Train_{improve} &= Train_Q - Train_{BIIP} \\ &= (1+N)*N/2 - (N+(1+N)*N/2)/2 \\ &= (N^2-N)/4 \end{aligned} \quad (6)$$

Finally, we compared the training times required by the two methods when the number of intentions is 1–50. The experimental result is shown in Figure 12.

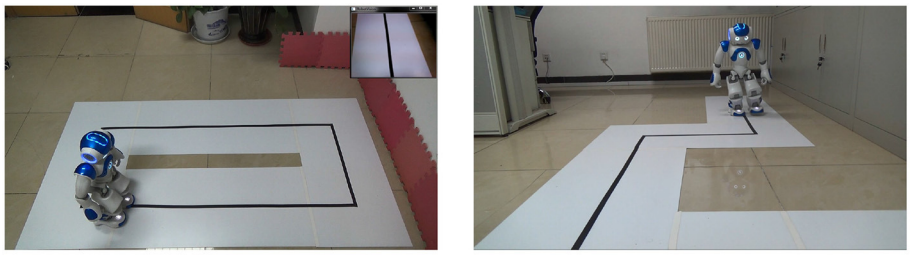


FIGURE 9
(A) Training scenario. **(B)** Test scenario.

FIGURE 10
The actions that the user expects the robot to perform according to different images. From left to right are: Move forward (the black line is in the center of the visual field), Move backward (no black line detected), Move left (the black line is on the left side of the visual field), Move right (the black line is on the right side of the visual field), Turn left (the black line turns left) and Turn right (the black line turns right).

TABLE 1 Number of rules under different intention numbers, and the comparison results of Q-learning method and the proposed model.

Number of intentions	1	2	3	4	5	6	7	8	9
Number of rules	1	2	6	24	120	720	5,040	40,320	362,880
Q-learning method	1	3	6	10	15	21	28	36	45
The proposed model (Mode)	1	2.5	4.5	7	10	13.5	17.5	22	27
The proposed model (Min)	1	2	3	4	5	6	7	8	9
The proposed model (Max)	1	3	6	10	15	21	28	36	45

4. Discussion

Based on the neural mechanism of reinforcement learning, we propose a brain-inspired intention prediction spiking neural network model to enable the robot to perform actions according to the user's intention. With the STDP mechanisms and the simple feedback of right or wrong, the humanoid robot NAO could successfully predict the user's intentions in Human Intention Prediction Experiment and Trajectory Tracking Experiment. Compared with the traditional Q-learning method, the training times required by the proposed model are reduced by $(N^2 - N)/4$, where N is the number of intentions.

Reinforcement learning, supervised learning and unsupervised learning are considered as the three basic machine learning paradigms. It has been successfully applied to different robotic tasks, such as navigation, manipulation, decision-making in human robot interaction. The Q-learning method is a widely used and very effective reinforcement learning method. Compared with the Q-learning method, our model has two characteristics: biologically plausible and requires fewer training times under the same task.

The biologically plausible of the model helps to reveal the neural mechanism of reinforcement learning in the brain from a computational perspective. We ensured the biologically plausible of the model from three aspects: the network structure, the neuron model and the learning mechanism. The network structure refers to the neural mechanism of reinforcement learning, including the relevant brain regions, the functions of these brain regions and the pathways between these brain regions. The neuron model is Izhikevich neuron model which achieves a good balance in biologically plausible and computational efficiency. The learning mechanism uses the most important STDP mechanism in the biological brain, and the results of the biological neuron fitting are used as the parameters of the computational model.

Compared with Q-learning method, the direct reason that our model needs fewer training times is the inhibition of LOFC by MOFC in the process of positive reward processing. The positive reward process indicates that the robot successfully predicted the user's intention. MOFC transmits the information to BG *via* StrD1 and to DLPFC at the same time. This pathway is used to strengthen the synaptic weight between the current

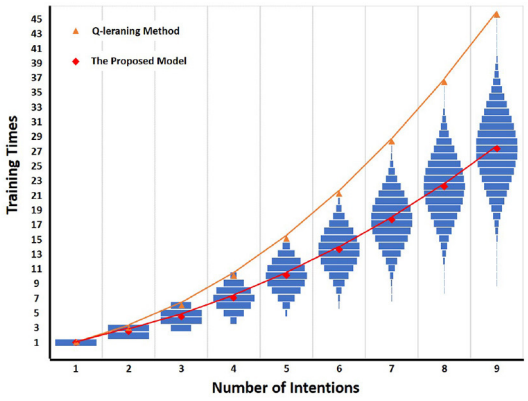


FIGURE 11
Detailed comparison between Q-learning and the proposed model. The horizontal axis is the number of intentions, and the vertical axis is the number of training times. The blue bar is the training times of the proposed model under different intention-action corresponding rules. The red diamond is the mode of training times required by all intention-action corresponding rules under different intention numbers. If there are multiple modes, the mean value is taken. The orange triangle is the training times of Q-learning method under different intention numbers.

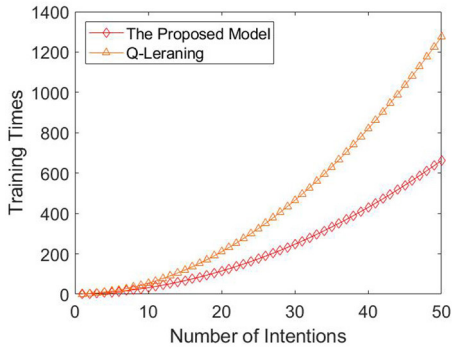


FIGURE 12
Comparison between Q-learning and the proposed model. The horizontal axis is the number of intentions, and the vertical axis is the number of training times. The red diamond is the mode of training times required by all intention-action corresponding rules under different intention numbers. If there are multiple modes, the mean value is taken. The orange triangle is the training times of Q-learning method under different intention numbers.

visual input (such as $Visual_1$) and the prediction intention (such as $Intention_1$) to ensure that the user's intention can be correctly predicted when the same visual input is received in the future. Meanwhile, MOFC inhibits LOFC, then LOFC transmits the information to BG *via* StrD2 and to DLPFC at the same time. This pathway is used to reduce the synaptic weight between the future visual inputs ($Visual_{others}$) and the currently predicted intentions ($Intention_1$), avoid new visual inputs to choose the intentions that have been learned ($Intention_1$), and promote new visual inputs to select other unlearned intentions ($Intention_{others}$).

5. Conclusion

We propose a brain-inspired intention prediction model based on the neural mechanism of reinforcement learning. We deploy the model on the humanoid robot NAO, and verified the effectiveness of the model through Human Intention Prediction Experiment and Trajectory Tracking Experiment. The experimental results show that the robot could successfully predict the user's intentions only through the simple feedback of right or wrong. In this way, the robot can quickly learn new rules without interfering with the learned and unchanged intention rules. The proposed model is simple and effective, which can effectively improve the flexibility and simplicity of human-robot interaction.

In our future work, we will combine our previous work in affective states recognition (Zhao et al., 2020, 2021a) to explore the potential of the proposed model in affective interaction tasks and improve the naturalness and flexibility of human-robot interaction.

Data availability statement

The python scripts can be downloaded from the GitHub repository of the brain-inspired cognitive intelligence engine at Research Center for Brain-inspired Intelligence, Institute of Automation, Chinese Academy of Sciences: https://github.com/BrainCog-X/Brain-Cog/tree/main/examples/Social_Cognition/Intention_Prediction. The script is based on the brain-inspired cognitive intelligence engine (BrainCog), more details could be found at https://github.com/BrainCog-X/Brain-Cog. Further inquiries should be directed to the corresponding author.

Author contributions

All authors conceived the initial idea, designed the model, carried out the experiments, and wrote the manuscript.

Funding

This work is supported by the Strategic Priority Research Program of the Chinese Academy of Sciences (Grant No. XDB32070100), the new generation of artificial intelligence major project of the Ministry of Science and Technology of the People's Republic of China (Grant No. 2020AAA0104305), the Beijing Municipal Commission of Science and Technology (Grant No. Z181100001518006), the Key Research Program of Frontier Sciences, CAS (Grant No. ZDBS-LY-JSC013).

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

Supplementary material

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fnins.2022.1009237/full#supplementary-material

References

Barbey, A. K., Koenigs, M., and Grafman, J. (2013). Dorsolateral prefrontal contributions to human working memory. *Cortex* 49, 1195–1205. doi: 10.1016/j.cortex.2012.05.022

Bi, G., and Poo, M. (2001). Synaptic modification by correlated activity: Hebb's postulate revisited. *Annu. Rev. Neurosci.* 24, 139–166. doi: 10.1146/annurev.neuro.24.1.139

Dermy, O., Charpillet, F., and Ivaldi, S. (2017). "Multi-Modal intention prediction with probabilistic movement primitives," in *10th International Workshop on Human-Friendly Robotics* (Napoli), 1–15.

Frey, S., and Petrides, M. (2002). Orbitofrontal cortex and memory formation. *Neuron* 36, 171–176. doi: 10.1016/S0896-6273(02)00901-7

Friend, D. M., and Kravitz, A. V. (2014). Working together: basal ganglia pathways in action selection. *Trends Neurosci.* 37, 301–303. doi: 10.1016/j.tins.2014.04.004

Haber, S. N., and Knutson, B. (2010). The reward circuit: linking primate anatomy and human imaging. *Neuropsychopharmacology* 35, 4–26. doi: 10.1038/npp.2009.129

Hwang, K., Bertolero, M. A., Liu, W. B., and D'Esposito, M. (2017). The human thalamus is an integrative hub for functional brain networks. *J. Neurosci.* 37, 5594–5607. doi: 10.1523/JNEUROSCI.0067-17.2017

Izhikevich, E. M. (2003). Simple model of spiking neurons. *IEEE Trans. Neural Netw.* 14, 1569–1572. doi: 10.1109/TNN.2003.820440

Kakei, S., Hoffman, D. S., and Strick, P. L. (1999). Muscle and movement representations in the primary motor cortex. *Science* 285, 2136–2139. doi: 10.1126/science.285.5436.2136

Kringelbach, M. L. (2005). The human orbitofrontal cortex: linking reward to hedonic experience. *Nat. Rev. Neurosci.* 6, 691–702. doi: 10.1038/nrn1747

Li, Z., Mu, Y., Sun, Z., Song, S., Su, J., and Zhang, J. (2020). Intention understanding in human-robot interaction based on visual-nlp semantics. *Front. Neurorobot.* 14, 610139. doi: 10.3389/fnbot.2020.610139

Lin, H.-I., Nguyen, X.-A., and Chen, W.-K. (2017). Active intention inference for robot-human collaboration. *Int. J. Comput. Methods Exp. Meas.* 6, 772–784. doi: 10.2495/CMEM-V6-N4-772-784

Liu, Z., Liu, Q., Xu, W., Liu, Z., Zhou, Z., and Chen, J. (2019). Deep learning-based human motion prediction considering context awareness for human-robot collaboration in manufacturing. *Procedia CIRP* 83, 272–278. doi: 10.1016/j.procir.2019.04.080

Luo, R. C., and Mai, L. (2019). "Human intention inference and on-line human hand motion prediction for human-robot collaboration," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Macau: IEEE), 5958–5964.

O'Doherty, J., Kringelbach, M. L., Rolls, E. T., Hornak, J., and Andrews, C. (2001). Abstract reward and punishment representations in the human orbitofrontal cortex. *Nat. Neurosci.* 4, 95–102. doi: 10.1038/82959

Song, D., Kyriazis, N., Oikonomidis, I., Papazov, C., Argyros, A., Burschka, D., et al. (2013). "Predicting human intention in visual observations of hand/object interactions," in *2013 IEEE International Conference on Robotics and Automation* (Karlsruhe: IEEE), 1608–1615.

Stocco, A., Lebiere, C., and Anderson, J. R. (2010). Conditional routing of information to the cortex: a model of the basal ganglia's role in cognitive coordination. *Psychol. Rev.* 117, 541–574. doi: 10.1037/a0019077

Villagrasa, F., Baladron, J., Vitay, J., Schroll, H., Antzoulatos, E. G., Miller, E. K., et al. (2018). On the role of cortex-basal ganglia interactions for category learning: a neurocomputational approach. *J. Neurosci.* 38, 9551–9562. doi: 10.1523/JNEUROSCI.0874-18.2018

Vinanzi, S., Goerick, C., and Cangelosi, A. (2019). "Mindreading for robots: predicting intentions via dynamical clustering of human postures," in *2019 Joint IEEE 9th International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob)* (Oslo: IEEE), 272–277.

Wang, W., Li, R., Chen, Y., Sun, Y., and Jia, Y. (2021). Predicting human intentions in human-robot hand-over tasks through multimodal learning. *IEEE Trans. Automat. Sci. Eng.*19, 2339–2353. doi: 10.1109/TASE.2021.3074873

Yan, L., Gao, X., Zhang, X., and Chang, S. (2019). "Human-robot collaboration by intention recognition using deep lstm neural network," in *2019 IEEE 8th International Conference on Fluid Power and Mechatronics (FPM)* (Wuhan: IEEE), 1390–1396.

Yu, X., He, W., Li, Y., Xue, C., Li, J., Zou, J., et al. (2021). Bayesian estimation of human impedance and motion intention for human-robot collaboration. *IEEE Trans. Cybern.* 51, 1822–1834. doi: 10.1109/TCYB.2019.2940276

Zeng, Y., Zhao, Y., and Bai, J. (2016). "Towards robot self-consciousness (i): brain-inspired robot mirror neuron system model and its application in mirror self-recognition," in *International Conference on Brain Inspired Cognitive Systems* (Beijing), 11–21.

Zeng, Y., Zhao, Y., Bai, J., and Xu, B. (2017). Toward robot self-consciousness (ii): brain-inspired robot bodily self model for self-recognition. *Cognit. Comput.* 10, 307–20. doi: 10.1007/s12559-017-9505-1

Zeng, Y., Zhao, Y., Zhang, T., Zhao, D., Zhao, F., and Lu, E. (2020). A brain-inspired model of theory of mind. *Front. Neurorobot.* 14, 60. doi: 10.3389/fnbot.2020.00060

Zhao, F., Zeng, Y., and Xu, B. (2018). A brain-inspired decision-making spiking neural network and its application in unmanned aerial vehicle. *Front. Neurorobot.* 12, 56. doi: 10.3389/fnbot.2018.00056

Zhao, Y., Cao, X., Lin, J., Yu, D., and Cao, X. (2021a). Multimodal affective states recognition based on multiscale cnns and biologically inspired decision fusion model. *IEEE Trans. Affect. Comput.* 1–14. doi: 10.1109/TAFFC.2021.3093923

Zhao, Y., Yang, J., Lin, J., Yu, D., and Cao, X. (2020). "A 3D convolutional neural network for emotion recognition based on eeg signals," in *2020 International Joint Conference on Neural Networks (IJCNN)* (Glasgow: IEEE), 1–6.

Zhao, Y., Zeng, Y., and Qiao, G. (2021b). Brain-inspired classical conditioning model. *iScience* 24, 101980. doi: 10.1016/j.isci.2020.101980

frontiers | Frontiers in Neuroscience

TYPE Original Research
PUBLISHED 08 November 2022
DOI 10.3389/fnins.2022.1033776

OPEN ACCESS

EDITED BY
Yuqi Han,
Tsinghua University, China

REVIEWED BY
Karl Friston,
University College London,
United Kingdom
Dimitrije Marković,
Technical University Dresden, Germany

*CORRESPONDENCE
Elizabeth B. Torres
ebtorres@psych.rutgers.edu

†These authors have contributed equally to this work

SPECIALTY SECTION
This article was submitted to Perception Science, a section of the journal Frontiers in Neuroscience

RECEIVED 31 August 2022
ACCEPTED 12 October 2022
PUBLISHED 08 November 2022

CITATION
Vaskevich A and Torres EB (2022) Rethinking statistical learning as a continuous dynamic stochastic process, from the motor systems perspective.
Front. Neurosci. 16:1033776.
doi: 10.3389/fnins.2022.1033776

Rethinking statistical learning as a continuous dynamic stochastic process, from the motor systems perspective

Anna Vaskevich[1†] and Elizabeth B. Torres[1,2,3*†]

[1]Sensory Motor Integration Lab, Department of Psychology, Rutgers, The State University of New Jersey, New Brunswick, NJ, United States, [2]Rutgers Center for Cognitive Science, Piscataway, NJ, United States, [3]Rutgers Computer Science Department, Computational Biomedicine Imaging and Modeling Center, Piscataway, NJ, United States

The brain integrates streams of sensory input and builds accurate predictions, while arriving at stable percepts under disparate time scales. This stochastic process bears different unfolding dynamics for different people, yet statistical learning (SL) currently averages out, as noise, individual fluctuations in data streams registered from the brain as the person learns. We here adopt a new analytical approach that instead of averaging out fluctuations in continuous electroencephalographic (EEG)-based data streams, takes these gross data as the important signals. Our new approach reassesses how individuals dynamically learn predictive information in stable and unstable environments. We find neural correlates for two types of learners in a visuomotor task: narrow-variance learners, who retain explicit knowledge of the regularity embedded in the stimuli. They seem to use an *error-correction* strategy steadily present in both stable and unstable environments. This strategy can be captured by current optimization-based computational frameworks. In contrast, broad-variance learners emerge only in the unstable environment. Local analyses of the moment-by-moment fluctuations, naïve to the overall outcome, reveal an initial period of memoryless learning, well characterized by a *continuous* gamma process starting out exponentially distributed whereby all future events are equally probable, with high signal (mean) to noise (variance) ratio. The empirically derived continuous Gamma process smoothly converges to predictive Gaussian signatures comparable to those observed for the error-corrective mode that is captured by current optimization-driven computational models. We coin this initially seemingly purposeless stage *exploratory*. Globally, we examine a posteriori the fluctuations in distributions' shapes over the empirically estimated stochastic signatures. We then confirm that the exploratory mode of those learners, free of expectation, random and memoryless, but with high signal, precedes the acquisition of the error-correction mode boasting smooth transition from exponential to symmetric distributions' shapes. This early naïve phase of the learning process has been overlooked by current models

driven by expected, predictive information and error-based learning. Our work demonstrates that (statistical) learning is a highly dynamic and stochastic process, unfolding at different time scales, and evolving distinct learning strategies on demand.

KEYWORDS

statistical learning, dynamic learning, exploration, stochastic process, error correction, active inference learning, reinforcement learning

Introduction

At the start of life, human babies gradually become aware of their bodies in motion and as they understand it, they come to own the consequences of impending movements that make up all their purposeful actions. Seemingly *purposelessly*, neonates explore their surroundings as they expand their limbs with antigravity motions and eventually learn to reach out to their immediate space in a well-controlled, *purposeful,* and intended manner. The type of highly dynamic, spontaneous, exploratory learning that is at first driven by surprise and curiosity, has no initial goal or desired target. At this early stage of learning, all future events are equally probable to the cognitive system. The learning is merely a wondering process, "*what happens if I do this?*", perhaps a guess, "*if I do this, then this (consequence) will ensue, otherwise, this other (consequence) will happen. . .*". The current work offers evidence to suggest that this endogenous and dynamic type of learning in early life may scaffold how we learn in general. That is, that before realizing that certain regularities are present in the environment we collect information spontaneously, without relying on prior knowledge, committing to some stimuli salient feature, or using referencing goals. This stage, that has so far been overlooked, is not well described by traditional models of error correction learning. These models rely on expectation and surprise minimization. However, there are situations whereby the system does not yet have referencing information to generate a prediction error or expected prediction error code.

Research about learning, whether in the perceptual, the motor, or the cognitive domain, is primarily based on error-correction schemas (Censor et al., 2012; Hasson, 2017; Frost et al., 2019). These schemas are aimed at reducing the difference between a desired configuration or goal to be learned, and the current learning state (Hasson, 2017). Such goals tend to be exogenous in nature, but implicit in them are rules that the system must find. Somehow the spontaneous self-discovery process that we relied on as babies, to learn about sensing our body in the world and sensing the world in our body, tends to fade away from our behavioral research. Indeed, curious exploration seldom enters our experimental paradigms in explicit ways (Frost et al., 2019). Some animal models of exploratory behavior (Drai and Golani, 2001) have nevertheless been successfully extended to characterize exploration in human infants as excursions that separate segments of movements' development from lingering episodes (Frostig et al., 2020). This recent research suggests behavioral homology across species and prompted us to hypothesize that at a finer temporal learning scale, a wondering, exploratory code may hide embedded in the fluctuations of our performance. We tend to average out such fluctuations as superfluous noise, often referred to as gross data. Certainly, when favoring *a priori* imposed theoretical means under assumptions of normality and stationarity in the data registered during the learning process, we miss the opportunity to know what possible information lies in the gross data.

The exploratory code discussed above is not to be confused with the exploration mode that is commonly addressed in models of exploration *vs*. exploitation in reinforcement learning (RL) (Sutton, 1992; Dayan and Balleine, 2002). Within this computational framework, learning depends on a reward, which is either intrinsically obtained, or extrinsically provided. However, for both exploration and exploitation, the learning is best described by error correction, as the system considers information and aims to descent optimally along the gradient of some implicit objective function, minimizing the error towards a desirable configuration. The RL framework does not explain how the objective (target) of the objective function is determined neither does it say how the value of the target self-emerges in different contexts. This includes more recent work on intrinsically motivated RL, where "Curiosity thus seems to be a matter of finding the right balance so that the agent is constantly maximizing the rate of reducing the prediction errors" (Dubey and Griffiths, 2020). Indeed, RL solves a different problem than that of self-discovering the perceptual goal or objective of a given situation.

We here focus precisely on how the system comes to self-discover the task-goal or purpose by firstly opening information channels welcoming surprise. More specifically, we isolate the spontaneous exploratory mode of learning. This mode without expectations, or referencing signals, leads to the self-discovery of the goal or objective. To that end, we focus on the cognitive processes known as implicit or statistical learning (SL). While we recognize other influential computational frameworks such as active inference and Bayesian RL contribute to our understanding of learning in general (Friston et al., 2016, 2017),

SL is ideal for the present study as it involves embedding and manipulating the predictability of specific regularities within the perceptual input, so that the emergence of expectations and transitions between different learning modes can be tracked online. We return to the relevance and implications of our results on other computational frameworks that rely on optimization and error-correction in the "Discussion" section.

Implicit SL describes the ability of the brain to extract (largely beneath awareness) regularities from the environment (Hasson, 2017; Frost et al., 2019; Conway, 2020). Such capacity has long been known to support a wide range of basic human skills such as discrimination, categorization, and segmentation of continuous information (Saffran et al., 1996; Romberg and Saffran, 2010; Christiansen, 2019) and predictive aspects of social interactions (Torres et al., 2013a; Sinha, 2014; Crivello et al., 2018). Previous research has consistently shown that regardless of the nature of the embedded regularity (motor, perceptual or both), SL involves motor control systems, so that when participants are required to respond, the presence of predictive information modulates both response preparation and response execution processes (Kunar et al., 2007; Schwarb and Schumacher, 2012; Vaskevich et al., 2021). Yet work to addresses the stochastic motor signatures of SL during motor decisions communicating a preferred stimulus is sparse (Torres et al., 2013a), particularly those involving different levels of neuromotor control (Torres, 2011).

In this work, we reevaluate SL from the standpoint of sensory-motor systems. We reasoned that the motor percept that emerges from the sensations of our own endogenously generated biorhythmic motions could serve to support the type of SL that other perceptual systems would experience to gain behavioral control. More specifically, we propose to reframe the SL problem using recent advances in developmental research of neuromotor control (Torres et al., 2016) that focuses on time series of biorhythmic signals like those derived from electroencephalographic (EEG) signals (Ryu et al., 2021). We track the dynamic changes in stochastic signature of the learning process, continuously evaluating an EEG signal recorded while participants perform in a learning task that contained predictive information (i.e., regularities).

To uncover the continuous dynamics of SL, we consider multiple time scales (**Figure 1A**) within the context of a visual search task (**Figure 1B**) whereby learning takes place across millisecond, minutes, and hours. Furthermore, we view the stochastic phenomena at a local and at a global level (**Figure 1C**). At the local level, we start naïve, without empirical knowledge of the stochastic process at hand. We do not make theoretical assumptions about this process (e.g., that is Gaussian, stationary, linear, etc.). Instead, we obtain moment by moment, the stochastic signatures of data parameters (e.g., signals' amplitude and timing) and track how they evolve over time, as the learning unfolds. At the global level, we then examine *a posteriori*, the fluctuations in those stochastic signatures that we empirically estimated, to gain insight into the overall dynamics of the SL process that took place. For example, we track the evolution of the empirically estimated probability distributions' shapes.

We analyze fluctuations of a continuous EEG signal, recorded during the visual search task. While we leverage the precise time stamping of the events in the data acquisition system and the use of stable and unstable implicit-learning environments (Vaskevich et al., 2021), we empirically estimate anew, moment by moment, the probability distribution function (PDF) that best fits fluctuations in the data and obtain the continuous family of PDFs describing the overall learning process. We let these fluctuations that are often discarded as gross data, reveal the primordial way of curious, exploratory learning, preceding the self-discovery of regularities conducive of a goal and eventually defining the error in the error-correction mode. We reframe SL from the point of view of a developing, nascent motor system that spontaneously transitions from purposeless to purposeful behavior.

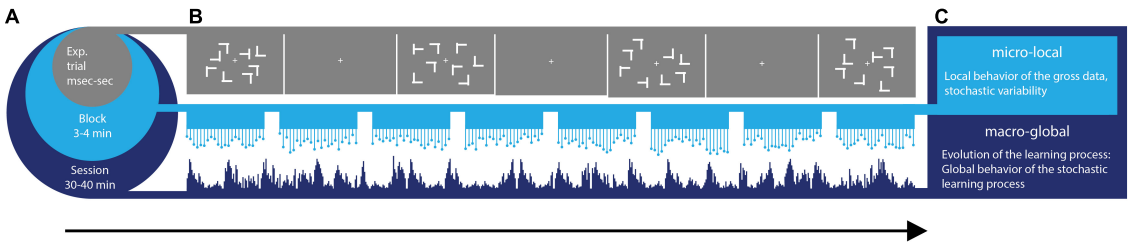


FIGURE 1
Dynamic statistical learning. **(A)** Different time scales of learning are accompanied by different types of learning supporting each level. From a level at sub-second time scales, to the scale of 40 min, different levels of granularity in the data afford different levels of precision to describe learning phenomena. Averaging out fluctuations in the system's responses may eliminate gross data containing important information on learning mechanisms. These may be varying from trial to trial and from block to block at each level. **(B)** Visual search task: the target was a letter T rotated either left or right that appeared among rotated Ls (distractors). Across trials, the spatial configurations of target and distractors (i.e., layouts) could repeat (correlated group), be generated randomly (random group) or repeat on half of the trials (mixed group). **(C)** Micro-Local vs. Macro-Global signatures of variability are extracted from fluctuations in EEG signals recorded while participants searched for the target and pressed the corresponding response key as fast as possible.

Materials and methods

This study involving human participants was reviewed and approved by the Institutional Review Board of Tel Aviv University. The participants provided their written informed consent to participate in this study. Behavioral and ERP analyses of these data were previously published (Vaskevich et al., 2021). Here we focus on the continuous EEG signal, without taking data epochs and averaging data parameters under theoretical assumptions of normality, linearity, and stationarity. Instead, we empirically estimate the continuous family of PDFs that in a maximum likelihood (MLE) sense, best fits what is traditionally discarded as superfluous gross data. This novel approach enabled us to isolate phenomena that cannot be observed when data is analyzed with conventional methods, leading to the uncovering of entirely new results.

Participants

Data from 70 participants (48 female, mean age, 23.7) was analyzed in this study: 24 in the random group, 23 in the correlated, and 23 in the mixed groups. There were no differences in age or gender between the three experimental groups. Two participants (one in the mixed group and one in the correlated group) were removed from the analyses due to incomplete data: their EEG recording started late, missing the first few trials. As we focus here on continuous data analyses of the full learning experience, these two subjects were excluded.

Stimuli and procedure

All participants gave informed consent following the procedures of a protocol approved by the Ethics Committee at the Tel Aviv University. The EEG signal was recorded during the visual search task. This task was followed by an explicit memory test during which EEG was not recorded. A more detailed account of the procedure can be found in Vaskevich et al. (2021).

Stimuli in the visual search task and the explicit memory test were white T's and L's (**Figure 1B**). All stimuli were made up of two lines of equal length (forming either an L or a T). From a viewing distance of approximately 60 cm, each item in the display subtended 1.5° × 1.5° of visual angle. All items appeared within an imaginary rectangle (20° × 15°) on a gray background with a white fixation cross in the middle of the screen (0.4° × 0.4°). Targets appeared with equal probability on the right or left side of the screen.

Visual search task

Participants searched for a rotated T (target) among heterogeneously rotated L's (distractors) while keeping their eyes on the fixation cross. Each trial began with the presentation of a fixation cross for 2,100, 2,200, or 2,300 ms (randomly jittered) followed by an array of one of two possible targets (left or right rotated T) among seven distractors. Participants were instructed to press a response key corresponding to the appropriate target as fast as possible -i.e., the goal of the task was to be accurate as fast as possible. Each participant was randomly assigned to one of three groups, with the degree of regularity in the task varying along a gradient. At one extreme the participants searched for the target within a highly predictable environment where predefined spatial configurations of target and distractors (layouts) were repeated from trial to trial (correlated group). Presumably, the embedded regularity can be easily and systematically confirmed by the system. At the other extreme, participants experienced the least amount of regularity, as from trial to trial, the layouts of the display were generated randomly (random Group). For the third group, consistent and random layouts were mixed throughout the task (mixed group). Any regularity cumulatively built from random guesses and confirmations, thus creating the ground for self-emergence of the overall goal or purpose of the task. This task is ideal to investigate the dynamic progression of SL. The gradient of predictability enables to examine, moment by moment, stochastic variations in learning between environments that differ in the reliability of predicting and confirming a guessed regularity. Depending on the group, the visual search contained the consistent mapping condition (correlated group), the random mapping condition (random group), or both (mixed group).

In summary, the three groups corresponded to predictable predictability (consistent group), predictable unpredictability (random group) and unpredictable predictability (mixed group). We were particularly interested in learning in the mixed group relative to the other two (predictable) groups.

For the consistent mapping condition, spatial configurations of targets and distractors were randomly generated for each participant (8 layouts for the mixed group and 16 layouts for the correlated group). In the random mapping condition targets and distractors appeared in random locations throughout the task. The order of layouts was randomized every 16 trials (in the case of the mixed group 16 trials correspond to eight consistent and eight random trials presented in a random order). The identity of the target (left or right rotation) was chosen randomly on each trial and did not correlate with the spatial regularity. Participants completed 512 trials in the experiment. Only correct trials were included in the analysis.

Explicit memory test

Participants were not informed of the regularity in the visual search task. Upon completing the task, participants in the mixed and correlated groups (when the task contained regularity) completed an explicit memory test, designed to reveal explicit knowledge of the regularity: participants saw the layouts that

were presented to them during the search task mixed with new, randomly generated layouts. For each layout participants had to indicate whether they have seen the layout during the visual search task or not. We then computed an Explicit Memory Test (ET) score (hit rate/false alarm rate) that is considered to reflect each participant's explicit knowledge of the regularity, so that higher scores correspond to better explicit knowledge (Vaskevich et al., 2021).

EEG recording

Electroencephalographic signals were recorded inside a shielded Faraday cage, with a Biosemi Active Two system (Biosemi B.V., Amsterdam, Netherlands), from 32 scalp electrodes at a subset of locations from the extended 10–20 system. The single-ended voltage was recorded between each electrode site and a common mode sense electrode (CMS/DRL). Data was digitized at 256 Hz (for a more detailed account see Vaskevich et al., 2021). We rely on continuous recordings, without averaging epochs of the data. In this work, we focus on the electrodes that do not reflect strong eye muscle activity either through blinking or the jaw movement. The analyzed subset Fp1, Fp2, AF3, AF4, F3, F4, F7, F8, Fz, FCz, C3, C4, Cz, T7, T8, P1, P2, P3, P4, P5, P6, P7, P8, Pz, PO3, PO4, PO7, PO8, POz, O1, O2, and Oz), includes all the electrodes that were previously analyzed (P7, P8, PO3, PO4, PO7, PO8, C3, C4). We use the EEGLAB PREP pipeline (Bigdely-Shamlo et al., 2015) to clean the EEG signals.

Cross-coherence analyses and network representation

The statistical analyses described in the next sections were done for a hub channel, chosen continuously for each time window (5 s of recording) with 50% overlap of the sliding window. Here we describe the process by which these hub channels were selected. Based on previous work with the same approach we bandpass filtered the data at 13–100 Hz using IIR filter at 20th order (Ryu et al., 2021). Two sample leads, taken pairwise across all sensors of the EEG cap were then used to instantiate the analyses. We used cross-coherence to quantify the similarity between any two leads (Phinyomark et al., 2012). For each pair, the maximal cross-coherence was obtained, with corresponding phase and frequency values at which the maximum was attained. The maximal cross-coherence matrix was used as an adjacency matrix to build a weighted undirected graph representation of a network (**Supplementary Figure 1**). Next, network connectivity analyses were used to obtain the maximum clustering coefficient representing the hub within each time window at the selected frequency band. The stochastic signatures of the moment-by-moment fluctuations in neural activity were then tracked in each overlapping window for the identified hub.

New data type: The micro-movement spikes

The analysis that is at the heart of the current work relies on the micro movements (MMS) spikes. This type of data and analytical platform, developed in the Torres lab (Torres et al., 2013a), and patented by the US Patent office (Torres, 2018a), was used in the current work to examine the change in stochastic variations of an EEG signal over time. To obtain the MMS of the EEG-hub biorhythmic signal, for each participant we take the peaks of the original EEG-hub waveform, derive the empirical distribution of the peaks and using the empirically estimated mean, we obtain the absolute deviation of each time point in the EEG-hub time series, from the empirically estimated mean. In the present data, the continuous Gamma family of probability distributions best fitted the peaks data, in an MLE sense. The Gamma family has well defined moments. We used the empirically estimated mean amplitude (μV) in our computations, to track the moment-by-moment fluctuations away from the empirically estimated mean. This builds a time series of micro-movements' spikes (MMS) which consists of periods of activity away from the mean interspersed with quiet period of mean activity. Importantly, we retained the original times where those fluctuation peaks occurred and built normalized spike trains using the deviations from the mean amplitude using equation (1). An example is shown in **Figures 2A,B**.

Equation 1 scales out allometric effects owing to anatomical differences (Lleonart et al., 2000). Each local peak (max) of these series of fluctuations is divided by the sum of its value and the averaged values of points between the two local minima surrounding it

$$MMS = \frac{\max}{\max + avg_{\min -to- \min}} \quad (1)$$

The result is then plotted, reflecting the unitless standardized MMS (**Figure 2C**), which describe the minute fluctuations in the original waveform (the EEG-hub), away from the empirically estimated mean (**Figure 2C**). Sweeping through the MMS trains, the values of the peaks (ranging now between 0 and 1) are gathered into frequency histograms for windows of 5 s with 50% overlap between each two consecutive windows (**Figure 2D** shows the corresponding histograms from the sampled blocks and windows in **Figure 2C**). We explored between 1- and 5-s-long windows (with 50% overlap) and settled on 5 s as the minimal time unit that gave us acceptable 95% confidence intervals in the empirical estimation process requiring 100 peaks or more.

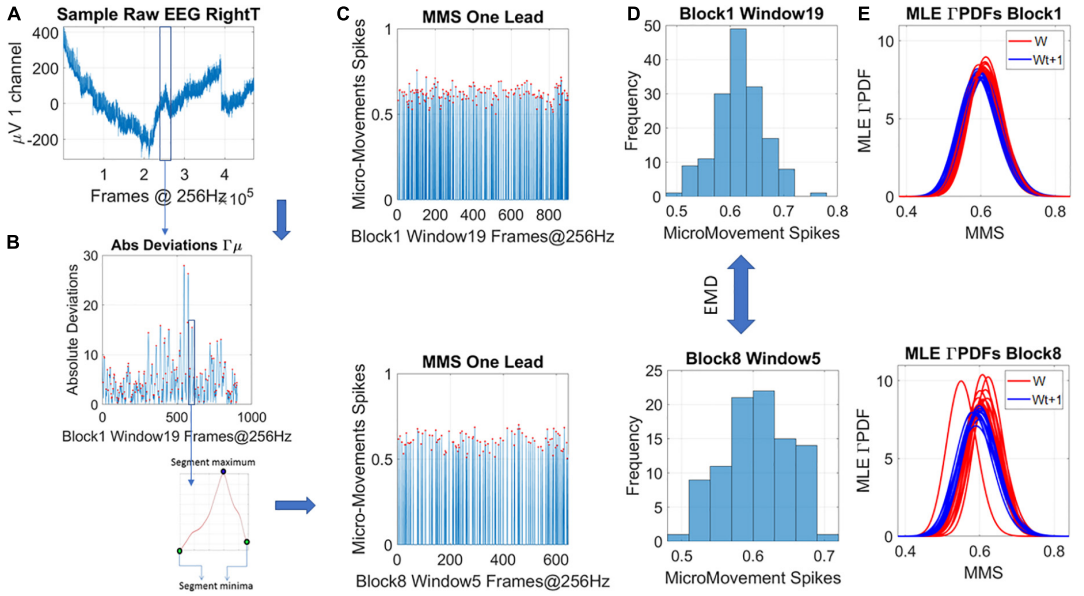


FIGURE 2
Transforming continuous analog signals to digital spikes: micro-movement spikes MMS. **(A)** Sample electroencephalographic signal from one hub channel determined through network connectivity analyses, zooming into one segment. Sweeping through the signal, windows of 5 s with 50% overlap are taken to scale each peak value deviated from the empirically estimated mean (μV). **(B)** For each participant, the original peaks are used to empirically estimate the mean amplitude across the session, and obtain, for each point in the time series, the absolute deviation from the mean. This series of fluctuations are then used to scale out possible allometric effects from e.g., anatomical head differences, using equation 1 in the methods. **(C)** The unitless, standardized MMS are plotted as time series conserving the original peaks' timing, shown here for two sample states in some window of blocks 1 and 8. **(D)** The peaks (red dots) are gathered into a frequency histogram to obtain the histogram's difference, from window to window (block by block), using the earth movers' distance, a similarity metric used in transport problems. We then obtain the amount of effort that it takes to transform one frequency histogram into the other. **(E)** Using maximum likelihood estimation (MLE) the best continuous family of probability distributions fitting the frequency histogram is obtained, shown here for different time windows.

A similarity metric for abstract probability spaces

The Earth Mover's Distance, EMD (Monge, 1781; Rubner et al., 1998) was used to obtain the scalar difference from moment to moment between the frequency histograms. This built a time series of such scalar quantity and enabled quantification of the dynamically changing stochastic trajectories. **Figure 2D** shows two sample histograms that can serve as input to the EMD metric expressing this (abstract) distance notion in probability space. **Figure 2E** shows an example of the empirically estimated Gamma PDFs across windows, contrasting blocks 1 and 8 for two quadrants of the Gamma parameter plane where these points are to be represented (see next section).

Local analyses: Empirical estimation of gamma scale and shape parameters

Upon deriving the MMS, we proceed to sweep through them using 5-s-long windows of MMS activity, with 50% overlap. This gives us a local estimation (at each window) of the stochastic process. Using MLE, we empirically estimate the shape and scale of the best PDF in an MLE sense. Examples of frequency histograms are shown in **Figure 2D** for different sample blocks and windows. Examples of PDFs are shown in **Figure 2E**. We found that the continuous Gamma family of PDFs were the best MLE fit for these windows of normalized MMS activity. Among distributions that we tested were the Lognormal, the normal, the exponential, the Gamma and the Weibull.

The Gamma was the best continuous family fitting the MMS in a MLE sense. The Gamma (a) shape and (b) scale parameters were then plotted on the Gamma parameter plane (**Figures 3A,B**). The Gamma family choice confirms previous work, as it has been found to be the optimal for representing MMS derived from human biorhythmic data registered from the face, eyes, whole body, heart, EEG, fMRI signals (e.g., Torres et al., 2013a; Ryu et al., 2021). This section is dedicated to explaining the empirical meaning of the Gamma parameter plane. We note here that at this level of analyses we are naïve as to the overall stochastic process and are empirically estimating its moment-by-moment evolution according to our unit of time (5 s window) chosen to yield tight confidence intervals.

The continuous Gamma family spans distributions of different shapes and different scales. Prior research has

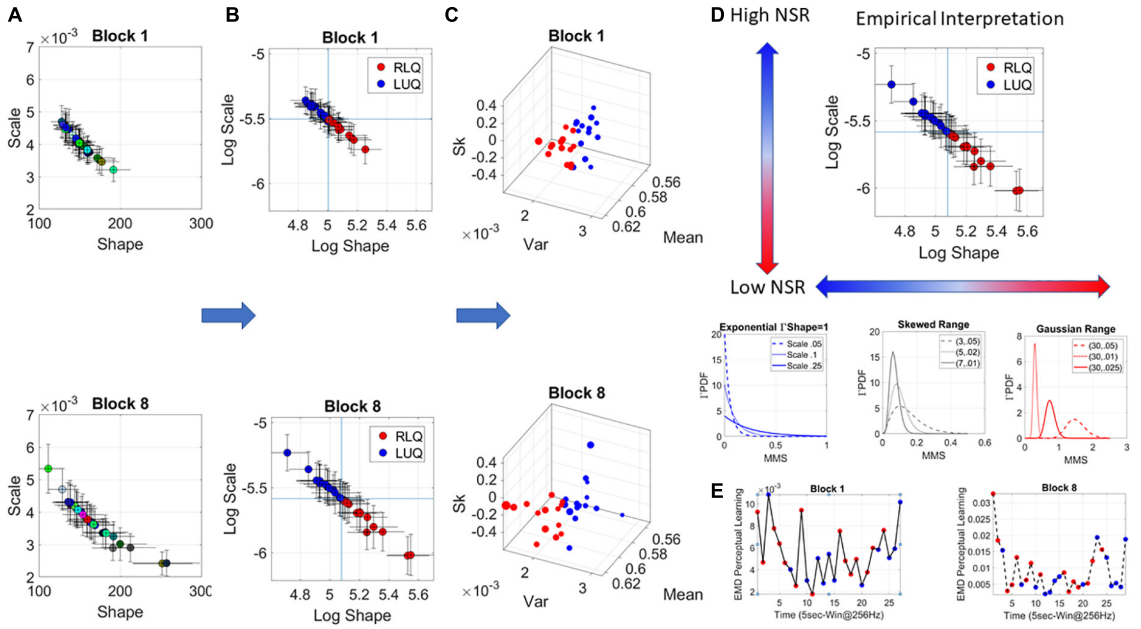


FIGURE 3
Stochastic analyses of the MMS derived from hub's activities. **(A)** Upon determination of the lead the MMS are obtained and MLE used to determine the parameters of the best continuous family of probability distribution functions (PDFs) describing their fluctuations. In this case the Gamma family. The Gamma shape and scale parameters thus estimated, are then plotted with 95% confidence intervals, on the Gamma parameter plane. **(A)** Each point represents the signatures of a 5-s window with 50% overlap. Colors represent arbitrary order. **(B)** The log–log Gamma parameter plane is obtained to track points according to the quadrants spanned by the median shape and median scale, taken across each block. The Right Lower Quadrant (RLQ) contrasts with the Left Upper Quadrant (LUQ). **(C)** The Gamma moments are obtained to visualize the points in **(B)** on a parameter space whereby the Gamma mean is represented along *x-axis*, the variance along the *y-axis*, the skewness along the *z-axis* and the size of the marker is proportional to the kurtosis. The color corresponds to the direction of the shift, where the point lands, red is from the LUQ to the RLQ, or from the RLQ to itself, whereas blue is from the RLQ to the LUQ, or from the LUQ to itself. **(D)** Empirical interpretation of the Gamma plane and the quadrants. Along the shape axis, the distributions change from the shape $a = 1$ memoryless exponential to the Gaussian range, with skewed distributions with heavy tails in between. **(E)** The EMD is used to track the magnitude of the shift from each estimated PDF in windows at t and t + 1, while the direction is tracked by the quadrant landing. This curve represents the evolution of the stochastic process and serves to determine, e.g., critical points of transitions for each block of the session.

empirically characterized maturation of human neuromotor development, showing over the human lifespan a tightly linear relationship between the log shape and log scale of this family (Torres et al., 2013a; Ryu et al., 2021). As humans mature, distributions of the fluctuations in biorhythmic activities measured from their central and peripheral nervous systems grow more symmetric while the scale (dispersion) decreases. This characterization has reduced the parameters of interest to one (the shape or the scale) since knowing one, we can infer the other with high certainty. Focusing here then on the ranges of PDF shapes, we track the SL evolution. These parameters reflect different degrees of randomness and different levels of noise to signal ratio NSR (which in the Gamma family is given by the scale parameter of (equation 2).

$$NSR = \frac{\Gamma\sigma}{\Gamma\mu} = \frac{a \cdot b^2}{a \cdot b} = b \qquad (2)$$

We will use in our descriptions 1/NSR = SNR and will refer to it as the signal (empirically estimated mean over empirically estimated variance). Figure 3A shows the Gamma parameters estimated for each window in blocks 1 and 8, while Figure 3B shows the log-log Gamma parameter plane with a division into quadrants that reflect different empirical properties of the stochastic process. We take the median of the shape values and the median of the scale values and draw a line across each axis (Figure 3B), to break the Gamma parameter plane into quadrants that shift from window to window. Quadrants reflect the evolution of the stochastic process. Figure 3C shows the corresponding Gamma moments space following the color-code of Figure 3B whereby points that fall on the right lower quadrant (RLQ) are those representing symmetric distributions with low NSR (low dispersion), while those in the left upper quadrant (LUQ) represent distributions closer to the exponential range and having high NSR.

As an example, in Figure 3D, we summarize these results for empirical interpretation and inference in block 8. Generally, at the leftmost extreme, when the Gamma shape is 1, we have the special case of the memoryless exponential distribution (no points appear in this range for this example). This is the case of having a random process whereby events in the past do not

inform more about events in the future than current events would. All future events are equally probable. The information is coming from the *here and now*. At this level of randomness, prior research has shown corresponding highest levels of NSR (We note that the signal to noise ratio SNR = 1/NSR will be used henceforth). Such distributions are typical to the motor code at the start of neurodevelopment (Torres et al., 2013a, 2016). Around 4–5 years of age, when the system is (on average) mature enough to start schooling, receive instructions, and sustain longer attention spans, a transition into heavy tailed distributions is observed. By college age these distributions are tending to Gaussian, so that the shape parameter is at the other extreme of the shape axis on the Gamma parameter plane and the SNR is at its highest value (**Figure 3D**).

Prior work has also revealed that in systems where maturation is compromised (e.g., autism across the lifespan) these global signatures remain in the exponential range, randomly relying on the here and now and manifesting very low SNR. In this case, the system does not progress into acquiring a predictive code (Torres et al., 2013a).

For each Gamma PDF derived from the MMS in each window, the shape and scale parameters are plotted with 95% confidence intervals as points along a stochastic trajectory, on the Gamma parameter plane. **Figure 3E** makes use of the EMD to quantify the stochastic shifts from moment to moment in each learning block, as points transition from quadrant to quadrant.

Dynamically tracking the stochastic signatures of the data

As the stochastic signatures *(a,b)* shift quadrants from moment to moment, they describe probability-positions over time (the dynamics of the stochastic process) on the Gamma parameter plane. Differentiation of this probabilistic positional trajectory yields an abstract velocity field whereby each velocity vector tangent to the trajectory, expresses the direction and the magnitude of the stochastic shift. To track the direction, we use the location of the landing point on the quadrants (the LUQ or the RLQ). The shift may leave the process in the same quadrant, or it may shift it away to the other quadrant. As shown in **Figure 2D**, to track the magnitude of the shift, we use the EMD scalar quantity representing the difference between the frequency histograms of amplitude fluctuations (MMS) derived from the EEG-hub channel activity. This is shown in **Figure 3E** for one participant's activity in blocks 1 and 8. That is, the EMD value on the *y*-axis represents the difference between the histogram at time t and the histogram at time t + 1, taken at consecutive windows of activity. Notice that this is not physical distance. It is abstract distance in probability space. Likewise, this is not physical time, but time that depends on the length of the window and the overlapping % of the sliding window process.

Global analyses

As we accumulate the above discussed stochastic trajectories, we are locally tracking the shapes of the PDFs over the empirically estimated Gamma parameters. We use EMD to trace the moment-by-moment evolution of the stochastic Gamma process, as it unfolds across all trials and blocks. But initially we are naïve to the fluctuations in this process. It is then as we contemplate the full stochastic profile, *a posteriori*, that we can track the spikes of the EMD at a global time scale, i.e., across the entire session. This is shown in **Figures 4A–D** using the MMS and Gamma process once again, this time, the empirically estimation is on the fluctuations of the Gamma shape parameter representing the stochastic shifts of the distributions of the Gamma shape.

The general formula for the PDF of the Gamma distribution is shown below (equation 3), where a is the shape parameter and b is the scale parameter.

$$f(x) = \frac{1}{\Gamma(a)\, b^a} x^{a-1} e^{-\frac{x}{b}} \tag{3}$$

Although the continuous Gamma family of PDFs can be parameterized with two parameters (a shape and b scale parameter), we can also obtain its statistical moments. We will use this alternative description of the distributions later to help visualize the results. The moments (μ, σ, skewness, kurtosis) are $a \cdot b, a \cdot b^2, 2/\sqrt{a}, 6/k$ respectively.

Results

Behavioral results

The results from the analyses of the behavior and averaged potentials were previously reported (Vaskevich and Luria, 2018; Vaskevich et al., 2021). For completeness we summarized here the main behavioral result. Participants in the mixed group reached significantly slower reaction times than participants in both the correlated and random groups, even though the task contained a potentially beneficial regularity on half of the trials. This result replicated previous findings and highlights the crucial issue of validity: when the regularity is valid, applying this statistical information results in facilitation to both the search and response processes (correlated group). However, when the regularity is mixed with random trials, thus appearing within a relatively unreliable and unstable environment, a global interference effect emerges, so that the reliance on all prior information is attenuated. Previously proposed theoretical interpretation for these highly counterintuitive results were

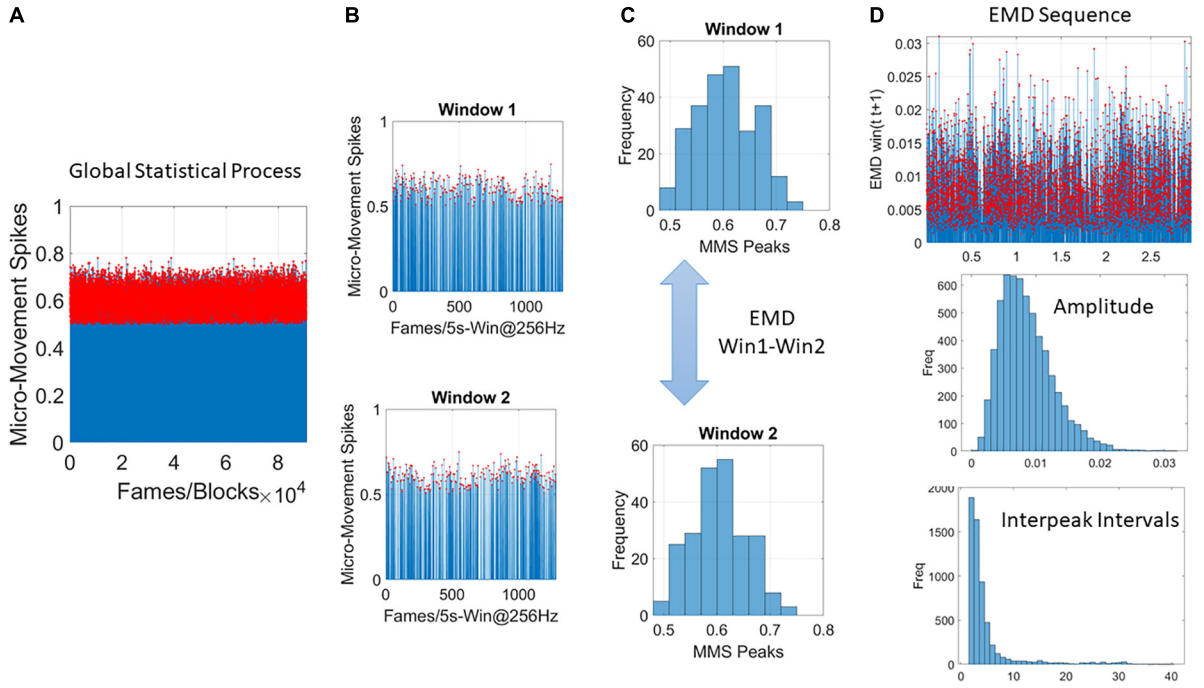


FIGURE 4
Global analyses **(A)** performed by pooling the MMS across trials and blocks and taking 5-s-long windows with 50% overlap **(B)** to obtain frequency histograms that can be compared using the EMD metric **(C)**. **(D)** Sweeping through the full trajectory of a condition gives the EMD sequence to obtain the peaks in red and gather them into a frequency histogram tracking the fluctuations in amplitude of the EMD variation (i.e., how the distribution change shape and dispersion) and the rate at which these changes occur as the inter peak interval intervals measuring the distances as well across peaks representing the PDF transitions. These histograms are used in MLE estimation of the distribution parameters best describing this global process.

reported in Vaskevich and Luria (2018, 2019) and Vaskevich et al. (2021).

Explicit memory test

In the mixed group, participants correctly classified previously seen layouts as familiar on 57% of the trials (hit rate), and incorrectly classified new layouts as familiar on 50% of the trials (false alarm rate). In the correlated group, participants correctly classified previously seen layouts as familiar on 55% of the trials (hit rate), and incorrectly classified new layouts as familiar on 48% of the trials (false alarm rate). For both the correlated and the mixed groups the differences between hit rate and false alarm were not significant, $F < 1$. The random group did not complete the explicit memory test as there was nothing to test for- there was no regularity in the task.

To assign a memory score (ET) we calculated the ratio between hit rate and false alarm rate for each participant. Higher scores correspond to better explicit memory of the visual layouts presented during the search task. The Overall memory scores of the correlated group ($M = 1.37$, $SD = 0.9$) and the mixed group ($M = 1.25$, $SD = 0.7$) were not significantly different, $F < 1$.

Local level of the stochastic process

For all three groups (correlated, random, mixed) we isolated the MMS from the continuous EEG data. We converted the fluctuations in the EEG amplitude (peaks μV) and inter-peak-interval timing (ms) to unitless, standardized MMS trains that were then analyzed using a sliding window of 5 s with 50% overlap (see section Methods). The window-by-window analyses for each participant revealed two subgroups in the mixed group. On the Gamma moments parameter space, along the Gamma variance dimension, one subgroup of learners (subgroup A of *broad-variance learners*) expressed higher variance of the fluctuations in the MMS amplitudes at the start of the experiment. This departure from the other subgroup (B of *narrow-variance learners*) can be appreciated individually for each participant over the entire experiment in **Figure 5**.

The fluctuations in the empirically estimated Gamma variance were then unfolded over blocks for each participant (**Figures 6A,B**). After the second block of trials, the levels of variance derived from the MMS-amplitude in subgroup A systematically decreased, eventually converging to the much lower level of the subgroup B. As such, subgroup A, with the initially much higher variance, expressed a higher bandwidth of

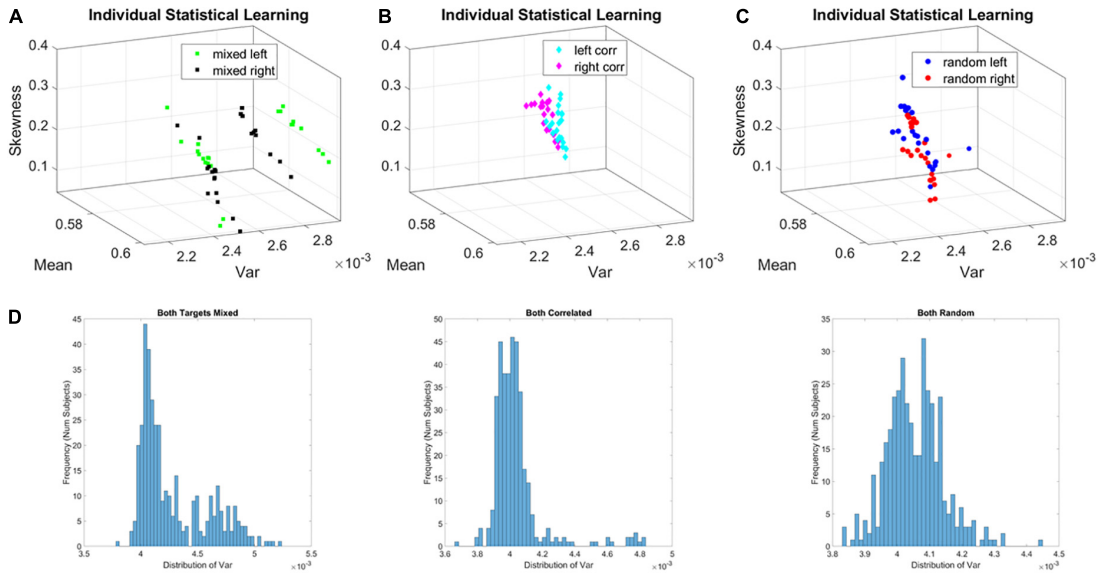


FIGURE 5
Local learning evolution captures two classes of learners in the unstable environment (i.e., mixed group). Empirically estimated Gamma moments span a parameter space whereby each participant represents a point by the moments of the probability distribution. The coordinates are the mean (*x-axis*), the variance (*y-axis*), the skewness (*z-axis*). The color represents the target orientation (left or right). **(A)** Mixed case (i.e., group) whereby trials intermix random and correlated conditions, spanning a relatively unstable learning environment. In this group two self-emerging distinct subgroups of participants. **(B)** Correlated group, for which layouts are consistent from trial to trial, spanning a stable learning environment. **(C)** Random group, for which layouts are generated randomly from trial to trial, spanning a stable learning environment where no regularity is present. **(D)** Corresponding frequency histograms of the distribution of the variance across trials, target types and participants.

overall variance values than subgroup B, which started out with much lower variance and remained in that regime throughout the eight blocks of the experiment. This was the case for both target types. Furthermore, this low range of variance in subgroup B was comparable to the ranges of variance observed in the random and correlated groups. This can be appreciated in **Figures 6C,D** for the random case and **Figures 6E,F** for the correlated case.

To show the overall differences in stochastic signatures of each case, we pooled the Gamma variance data from all blocks and for each mixed, correlated, and random group respectively (**Figure 5D**). The mixed group is indeed significantly non-unimodal, according to the Hartigan dip test of unimodality, $p < 0.01$ (Hartigan and Hartigan, 1985). The PDF derived from the MMS amplitude of the mixed group significantly differed from those in the random and correlated groups, according to the Kolmogorov Smirnov test for two empirical distributions ($p < 0.01$).

Relationship between behavioral outcomes and stochastic results

The two subgroups broad-variance A and narrow-variance B of the mixed group did not differ in reaction times or accuracy, suggesting that all participants were able to reach the same level of online performance. Instead, they were differentiated by their explicit knowledge of the regularity imbedded in the task, as reflected by their memory scores in the explicit memory test: 10 subjects in the broader variance subgroup A, $M = 0.94$, $SD = 0.4$ *vs.* 13 subjects in the narrow variance group subgroup B, $M = 1.52$, $SD = 0.75$, $p < 0.01$ non-parametric Wilcoxon ranksum test (**Figure 7A**). The subgroup A with broader bandwidth of variability showed low test scores, thus exhibiting less explicit knowledge of the regularity. In contrast, the subgroup B with the narrow, steady bandwidth of variability, gained a higher level of explicit knowledge, as reflected in higher explicit memory test scores (**Figure 7B**). We coined the process showing higher variance with low explicit memory score (subgroup A) "*exploratory mode.*" In contrast, we called the process showing lower variance and high explicit memory score "*error-correction mode*" (subgroup B). Here the mode refers to learning mode or phase and in the next results, we provide a stochastic characterization of these two fundamentally different modes of learning which, nevertheless, converged in block 8 to a similar variance range.

For completeness, the memory scores of the correlated group were also examined. Overall, memory scores ($M = 1.37$, $SD = 0.9$) were like the scores observed in subgroup B of the mixed group. This result is consistent with the similar stochastic learning signatures of the correlated group and this high memory subgroup (observed in the variance trajectories of

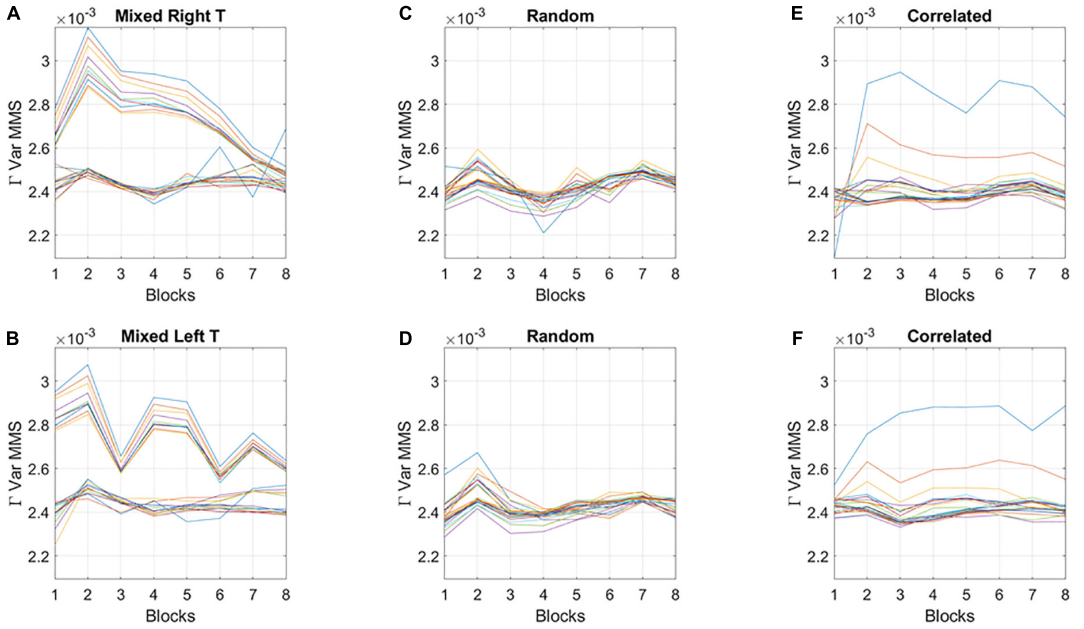


FIGURE 6
Broad- and narrow-variance groups according to the empirically estimated Gamma variance parameter block by block. **(A,B)** Two subgroups in the mixed group are revealed for right- and left-oriented targets (each curve represents the trajectory of a participant within the group). The subgroup with lower variance and narrower bandwidth of values throughout the experimental session separate from those in the subgroup with high variance and broader bandwidth of values. However, both subgroups converge to similar variance levels toward the 8th block of learning. Target types show different trajectories but similar convergence trend. **(C,D)** Random group shows similar levels of variance and stable learning throughout the experimental session, as does the correlated group **(E,F)** (with two outliers).

Figure 6). We here infer that as the regularity in the correlated group was highly reliable, with layouts repeating on all trials, it seems that all participants reached some minimal level of explicit knowledge, therefore no subgroups emerged.

Global a posteriori stochastic analyses of distribution shapes

Analyses of the stochastic signatures derived from pooling all trials, block by block, across all participants allowed us to examine the evolution of the distribution of the empirically estimated Gamma shape parameter, i.e., as the system experienced the learning and the PDFs shifted shape. The moment-by-moment fluctuations in the shape parameter provide insights into the dynamics of the stochastic process. Notice here that in our local computation (i.e., the MMS distributions at each window), we were naïve to the global dynamic nature of the stochastic Gamma process, as we were locally estimating the Gamma parameters (shape and scale) and the Gamma moments. Upon estimation of the full stochastic trajectory across the entire session, trial by trial and block by block, we are no longer naïve to the process. As such, we can make a global statement at the time scale of the entire session.

Among the moments of the distributions of the shape parameter, the variance of the evolving Gamma PDF shape parameter revealed the separation of the mixed group from the random and from the correlated groups (Figure 8A). Furthermore, a distinction is also observed for the mean parameter of the distribution of Gamma shapes (Figure 8B). As such, the SNR shows the highest signal content for the mixed group (Figure 8C). For both the correlated and random groups, the distribution shape has an increasing trend, consistent in both cases for the right- and left-oriented targets. However, in the mixed group, there is an initial increase in the shape that decreases and stabilizes by the 4th to 5th block, at much lower values of the variance, so that the SNR of the mixed group is much higher than that of the random or correlated groups. This elevated SNR indicates that the mixed environment is much more effective for learning than environments that contain purely random or purely correlated trials alone. Its information content is higher.

Unfolding the gamma process for each learning mode

We show the stochastic shifts of each of the error correction (lower Gamma shape variance and higher explicit

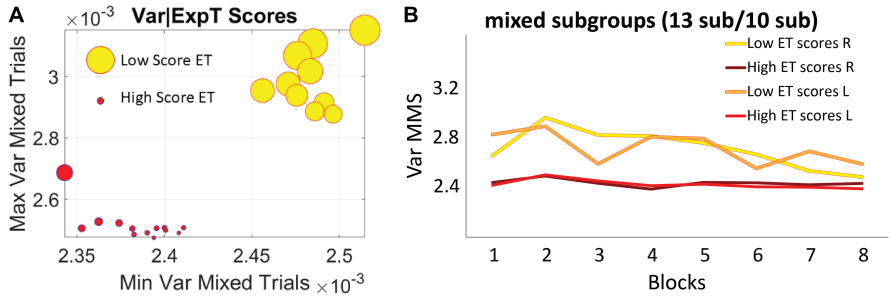


FIGURE 7
Self-emerging subgroups in the mixed group are differentiated by the scores of the explicit memory test. **(A)** The horizontal axis comprises the minimum value of the variance, while the vertical axis comprises the maximum value of the variance for each participant. Thus, the graph depicts the full range of variance values. The size of the marker is proportional to the explicit memory test score and the color represents the subgroup, with no overlapping between the two sets of participants. **(B)** Empirically estimated Gamma variance parameter unfolded block by block as in **Figures 6A,B**, for the two subgroups of the mixed condition. The group with less explicit knowledge [lower scores on the explicit memory test (ET score M = 0.94, SD = 0.4)] starts out with higher variance of the fluctuations of the MMS amplitudes (broad-variance group A), eventually converging to the much lower variance level of the subgroup that showed higher explicit knowledge of the regularity (ET score M = 1.52, SD = 0.75) (narrow-variance group B).

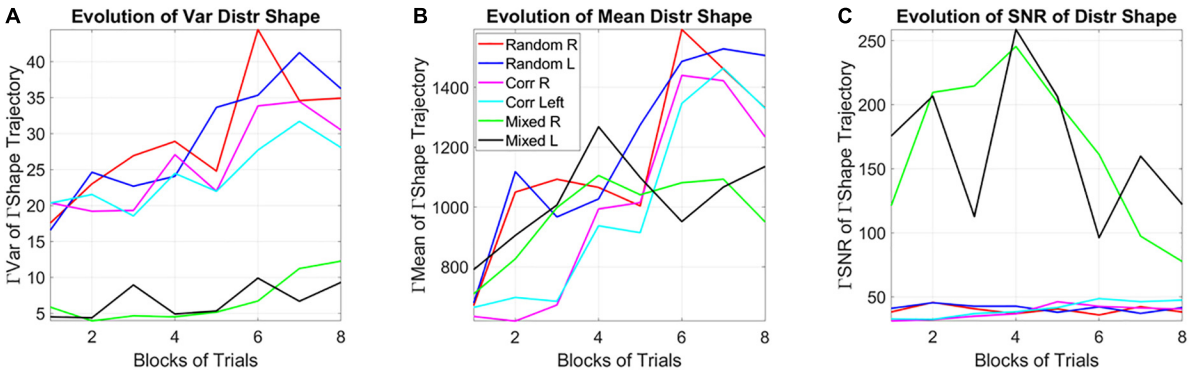


FIGURE 8
Learning evolution taken globally across participants and full session, shows the unstable environment (mixed group) to provide the most efficient conditions for learning, as indicated by the highest SNR. **(A)** Tracking, block by block, the empirically estimated variance of the distribution of gamma shape values obtained from the fluctuations in MMS amplitudes for each type of stimulus and target. Correlated and random groups trend upward with a steeper rate for correlated, while the mixed group stabilizes after ½ the session. The variance separates the correlated and random groups from the mixed group, with a marked reduction on the variability of distribution shapes and an overall trend to increase the variability in distribution shape toward the final blocks. **(B)** Tracking, block by block, the empirically estimated mean value of the distribution of shape values from the fluctuations in MMS amplitudes. **(C)** The signal to noise ratio (mean/variance) then shows the highest signal for the mixed trials, with a downward tendency after ½ the total session.

memory test score) and exploratory (higher Gamma shape variance and lower explicit memory test score), as they unfold across the blocks.

The empirically estimated Gamma family shape parameters of the subgroup with high explicit memory scores (subgroup B) starts in the symmetric Gaussian range but trends down and converges towards the skewed, heavy tailed distributions, shown in **Figure 9A** for the mean Gamma shape and in **Figure 9B** for the variance Gamma shape of the two types of learners [the SNR (mean/var ratio) for the two subgroups is shown in **Figure 9C**]. The trajectory on the Gamma parameter plane (**Figure 9D**) confirms the departure from a memoryless random state (i.e., when the Gamma shape value is 1). To better visualize these processes, we zoom in and unfold the two types of learning modes of **Figure 9D**. **Figure 9E** focuses on the exploratory process. As time progresses, the learning generally evolves from memoryless (Gamma shape 1) towards skewed, heavy tailed distributions and more symmetric distributions of the shape. **Figure 9F** focuses on the error correction process. Here we see the opposite trend whereby initially the distributions have symmetric shape (in the Gaussian range of the Gamma family) but as time progresses, the distribution shapes approach values closer to those observed for the exploratory process: skewed, heavy tailed distributions.

Notice here that we are capturing the distribution of the fluctuations in the estimated Gamma shape parameter with

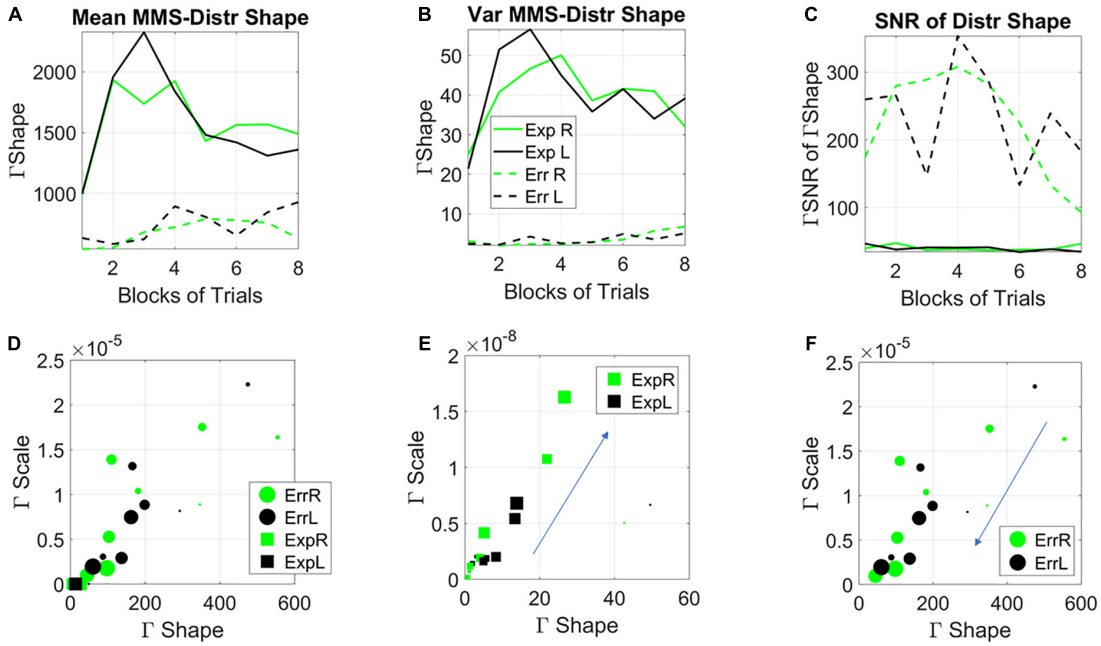


FIGURE 9
Stochastic characterization of exploratory *vs*. error correction modes across blocks by subgroups. **(A)** The evolution of the empirically estimated mean based on the distribution of Gamma shape values extracted from the MMS. **(B)** The evolution of the empirically estimated variance of the distribution of Gamma shape parameters. **(C)** The SNR (mean/var ratio) for the exploratory and error-correction subgroups. **(D)** Block by block evolution of the empirically estimated shape and scale parameters of the continuous Gamma family of probability distributions. Block number is proportional to the marker size, with earlier blocks having smaller size and later blocks increasing in size. The exploratory mode is confined to the gamma shapes close to the memoryless exponential distribution, while the error corrective mode evolves from higher to lower values of the Gaussian regime of the Gamma family. Unfolding each case [exploratory **(E)** and error corrective **(F)**] shows their convergence to a regime away from the memoryless exponential and tendency to more Gaussian like distributions. This convergent *global* behavior is congruent with the convergent *local* behavior of Figure 3.

a Gamma process as well. We are referring to the Gamma shape and Gamma scale parameters of the distributions derived (globally a posteriori) from the fluctuations in Gamma shape of the MMS derived from the EEG hub channels. On this Gamma parameter plane, the dispersion (Gamma scale of the fluctuations in Gamma shape value of MMS) along the *y*-axis, is larger as learning occurs, broadening the bandwidth of distribution shapes as learning takes place. The switch from exponential to heavy tailed to Gaussian distributions reflects the more systematic confirmation of a regularity in the stimuli. Initially, all future stimuli are equally probable (exponential regime), but in time, correct prediction of futures events increases, consistent with the transition from a detected regularity to a systematic goal. Once a goal is in place, error correction is the learning regime reflecting Gaussian predictive process embedded in this overall Gamma process. Here is where we see a tendency to symmetric shapes approached by both modes along the horizontal axis of the Gamma parameter plane. One mode (the exploratory) approaching it from the left, away from the memoryless exponential. The other approaching it from the right.

The stochastic transition depicted in Figures 9E,F confirms the separation between two fundamentally different learning styles with initially different stochastic regimes. It also highlights a phase transition approximately midway of the learning progression. Notwithstanding the initial differences, these regimes converged to similar signatures in the end. This transition from memoryless exploration (exponential) to predictive error-correction (heavy-tailed to Gaussian) surfaces in correspondence to midway of the session, blocks 3–4. Likely the regularity then self-emerges and eventually, through guess and systematic confirmation, transitions to a steady goal, one that serves to compute an error from.

In Figures 9E,F we see the system transitioning from an initial purposeless search to a search that then acquires a clear purpose, i.e., self-discovery of a task goal that was not instructed to the system. Our results suggest that this transition from memoryless into error correction-based learning depends on some minimum level of explicit knowledge. Examining this global process, we presume that in one subgroup enough explicit knowledge to trigger this transition was acquired much earlier than in the other subgroup. The group boasting an initial exploratory mode, for which the search was in the here and now, did not acquire distributions of the shape parameter away from the exponential range until around blocks 3–4. This was when the system shifted to a Gaussian mode

(Figure 9E larger markers) and when locally the variance of the MMS shrunk (Figures 6A,B), thus spiking (globally) the SNR of the fluctuations in shape parameter (Figure 8C). In this exploratory scenario, the system does not immediately progress into acquiring a predictive code. In other words, because of not yet committing to regularities in the perceptual input, the predictive processing that underwrites exploitative or goal-directed behavior is initially precluded in favor of broadening the bandwidth of information that enables surprise and self-referencing towards the self-discovery of a goal. Only then, does the system transitions into an error-corrective regime.

Dynamic statistical learning

At a global timescale (i.e., stochastic trajectory of the empirically estimated parameters examined *a posteriori*, across the entire experimental session) we assessed the change in stochastic variations of the signals over time. To do so, we examined the evolution of the fluctuations in the change of Gamma distributions' shapes using the Earth Movers Distance (EMD) metric (see trajectories in the Supplementary Figures 2–4). We compared from trial to trial, and block to block, across participants, the fluctuations in the amplitude of the change in distributions of the Gamma shape parameter (as measured by the EMD). We also assessed the rate of the change in peaks (inter peak intervals related to the physical timing of the overall global process by our unit of time, 5-s windows with 50% overlap). These parameters are analogous to a kinematic "speed temporal profile" of the PDFs' shape trajectory (Torres and Lande, 2015; Torres et al., 2016). As the Gamma process shifts stochastic signatures per unit time on the Gamma parameter plane, we obtain enough MMS peaks and estimate the Gamma parameter of each window with tight 95% CI. The EMD scalar profile over time, measuring how the histograms used in the estimation process change from window to window, reflect the dynamic nature of the stochastic shifts that occur as the participants perform the task and learn in exploratory, or in error correction mode, converging toward the signatures of the latter at the end of the learning process.

The analyses revealed that the system clearly distinguishes the rates at which the distributions change shape from the random to the correlated groups and between those and the mixed group. Figure 10A shows this on the log-log Gamma parameter plane where each point with 95% confidence intervals, represents the performance for the right target (left not shown for simplicity but has similar patterns, see Supplementary Figure 5). The corresponding PDFs for both right and left oriented targets are shown in Figure 10B. We can appreciate that the mixed case yields the most toward-Gaussian-predictive shifts in distribution change, with the highest shape value. This is accompanied by the highest SNR (i.e., at the lowest Gamma scale value). Furthermore, these rates of change in the two subgroups of the mixed case, clearly distinguish the left from the right oriented targets, with comparable rates of shifts in distribution shape for the exploratory and the error corrective subtypes. These are shown in Figure 10C (estimated Gamma parameters) and Figure 10D (corresponding Gamma PDFs). Different neural correlates of the learning process are shown in Supplementary Figure 6. These comparable shifts in distribution dynamics for exploratory and error correction stochastic regimes, hint at a smooth process whether the system is curiously wondering in exploratory mode, or aiming for a task goal, in error corrective mode.

Discussion

This study evaluated online dynamics of SL using a new data type and analytical approach. This new platform relies on the moment-by-moment fluctuations in the signal of interest, which are traditionally discarded as gross data. Within the context of a visual search paradigm that manipulated, trial by trial, the reliability of stimulus regularities, while registering EEG signals, we examined the continuous stochastic process reflecting SL. We first isolated the EEG hub lead, maximally connected to other leads, and then proceeded to apply our new statistical analyses to this continuous data stream.

We found that SL is a highly dynamic and stochastic process, sensitive to the reliability of the incoming information. Moreover, we discovered that embedded in the gross data, traditionally discarded as superfluous noise under assumptions of normality, lies a code that describes different modes of learning. Based on our stochastic characterization of the learning phenomena at different local *vs.* global scales, we equate this distinction with two fundamentally different types of learning processes. These are the commonly studied error correction mode linked to stimulus regularity, and the newly characterized exploratory mode. This exploratory mode, stochastically characterized here for the first time, is likely reflecting surprising contextual variations that lead the system to eventually self-discover the purpose of the task with (i) the self-discovery of a goal through self-referencing and (ii) transitioning to the error-correction mode. Eventually the latter can lead to fast and accurate performance. To aid interpreting these results, we leverage prior research on the broad characterization of human biorhythmic activity (Torres et al., 2013a; Ryu et al., 2021) and reframe SL from the vantage point of neuromotor control, where spontaneous (seemingly purposeless) and deliberate (highly purposeful) motions coexist in any natural behavior from the start of life (Torres, 2011; Torres et al., 2016).

Two main results emerged from the current analyses. First, we show that unstable environmental conditions (i.e., mixing reliable and unreliable stimulus regularities) provide the most

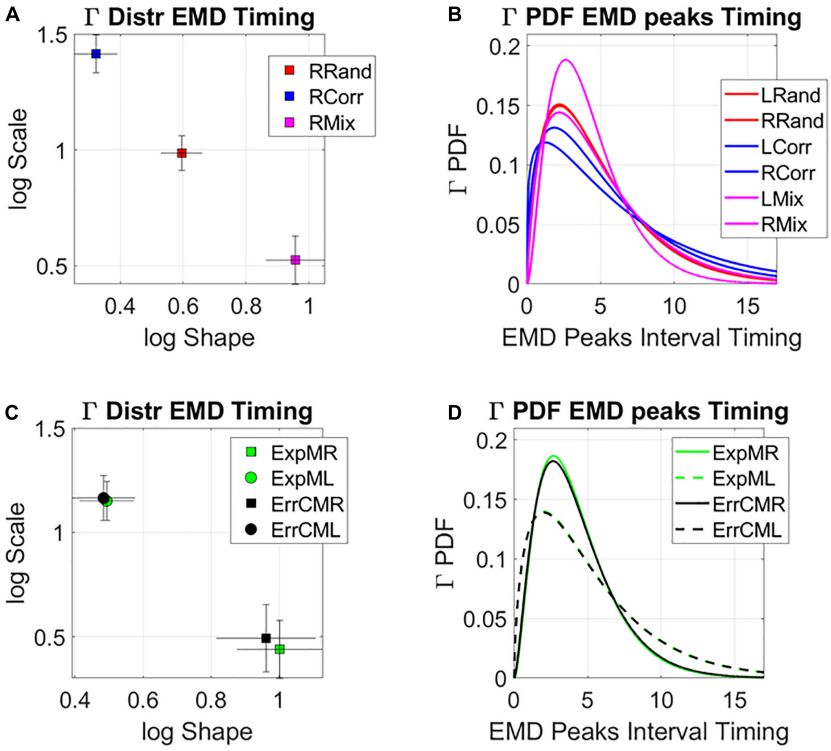


FIGURE 10
Global statistical learning dynamics. Unfolding the global rate of change in distribution shapes, as the system transitions from PDF to PDF, using the EMD to ascertain distribution differences from moment to moment. **(A)** Right target case is shown for the three groups with 95% confidence intervals for the empirically estimated Gamma shape and scale parameters. Each point represents a different distribution. Here the mixed group shows the maximal values of log shape (Gaussian) and SNR (1/log scale). **(B)** The PDFs corresponding to the maximum likelihood estimation (MLE) distributions in **(A)**. **(C)** Investigating the differentiation between targets for the two subgroups of the mixed condition at the global level reveals similar rate of change in the interpeak intervals, suggesting smooth transitions in both exploratory and error corrective cases. **(D)** Corresponding PDFs for **(C)**.

opportunity for learning, as characterized by higher SNR on both the global and local levels of analyses. Next, we show that on an individual basis, this unstable environment may give rise to different learning profiles: within this mixed group, two subgroups of participants self-emerged from the analyses. For one subgroup-B, coined error correction mode, the learning profile shows narrow variance in the MMS from start to finish and higher explicit memory test scores, reflecting better recall of the regularity. However, for the second subgroup-A, coined exploratory mode, the learning profile reflected an early stage of broad variance and memoryless learning which later converged into the signatures of the error correction mode. Crucially, this subgroup showed lower scores in the explicit memory test, as they did not recall the regularity with the degree of accuracy of the other subgroup. In their initial learning performance, all future events were equally probable, without a bias towards a particular regularity being reliably noted or recalled. We now turn to discussing each of these results in detail, while considering their implications on our understanding of SL in general.

Unpredictable environments provide more opportunity for learning, corresponding to a more efficient learning process than predictable environments

When comparing the stochastic signature of learning within an unstable environment mixing the stimulus regularity between random and correlated trails (mixed group) with stable conditions providing reliable regularity (correlated and random groups), the process proved to be less stationary, more predictable in nature, and was characterized by higher SNR. These characteristics suggest that complex environments provide higher opportunity to learn than reliable environments. Moreover, within our theoretical framework, higher SNR corresponds to more efficient learning. These results are consistent with neuroimaging studies that have identified brain systems that track uncertainty in a curvilinear U-shaped function (Nastase et al., 2014; Hasson, 2017). Within these systems, full randomness or full regularity are alike in terms of

informativeness and provide less information than the mixed case. As such, these systems seem to be especially sensitive to tracking relatively unreliable information in the environment.

Given that the real world is indeed complex, with our cognitive system continuously bombarded with variable regularities, it seems natural that we should be more attuned to learning under relatively unreliable (yet richer in information) conditions. However, suggesting that learning under such conditions is more efficient may seem to contradict the behavioral pattern previously observed in these data: participants in the mixed group reached slower RTs than both in the random and correlated groups (for a detailed account see Vaskevich et al., 2021). To resolve this issue, one must bear in mind that efficiency of learning is not necessarily manifested in online performance. That is, more complex learning conditions may hinder online reactions, but be beneficial for the long term. We propose that to gain further insight on SL, future studies should combine the methods introduced in the current work with experimental designs that involve changing regularities online and considering multiple sessions of learning. Indeed, such designs are becoming common within the field (Makovski and Jiang, 2010; Zellin et al., 2013; Vaskevich and Luria, 2019). However, so far, they lack the perspective of evaluating the dynamic and stochastic online evolution of the learning process, which is enabled by the methods used in the current work.

Learning dynamics at multiple time scales

Within the SL domain, focusing on the dynamics of the learning process itself, with the specific consideration of multiple time scales, has been recently suggested as the next necessary step in SL research (Hasson, 2017; Frost et al., 2019; Conway, 2020). Experts in the field agree that to understand the neural substrates underlying behavior it is necessary to view it, and to measure it, as a continuous process, evaluating learning trajectories of its stochastic variations and learning stability. However, so far, this direction has not matured into meaningful research, largely due to limitations of the standard analytical techniques. To date, several measurements, such as rhythmic EEG entrainment (Batterink et al., 2019; Moser et al., 2021), functional connectivity (FC) analysis (Toth et al., 2017), and divergences in EEG activity in the beta-band (Bogaerts et al., 2020) have been used to assess the online signature of SL. Collectively, these studies show that during different tasks with embedded regularities the EEG signal changes over time to reflect SL. They provide insight into the mechanisms that are going through a transition during SL, such as task automaticity (Toth et al., 2017), and word representation (Batterink et al., 2019), thus complementing behavioral measures that rely on reaction times and accuracy. In the context of the present work, they provide solid justification for the choice of EEG recordings as the data used to assess the stochastic profile of SL. However, none of the previously proposed measurements are informative regarding the ongoing dynamics of the learning process itself, as in all the above-mentioned studies the signal is segmented into periods, with the relevant measurement averaged across many trials for each period, under the assumption of normality.

The present work goes beyond assumptions of normality, linearity and stationarity in the data and exploits the moment-by-moment fluctuations that prior work discards as gross data. Embedded in that gross data we uncovered the phase transitions in probability space that distinguished two fundamentally different modes of learning and revealed one (memoryless exponential) that converges to the other (predictive Gaussian). Both modes are well characterized by the continuous Gamma family of PDF s at the local level, when we are naïve to the upcoming moment-by-moment distribution, and at the global level, when *a posteriori*, we can see the fluctuations in the (Gamma) distribution shape unfolded through the Gamma process itself.

Exploratory versus error correction modes differentiated by explicit knowledge of the embedded regularity

For a cohort of participants, the unstable environment (mixed group) triggered an initial stage of memoryless exploratory learning. During this stage, the stochastic signature of the process reflected a type of learning whereby initially all future events were equally probable. The stochastic signature unveiled in this initial period of learning for the broad-variance subgroup A of this cohort, suggests that the system was not relying on prior knowledge but was instead gathering as much information as possible from the "*here and now*." Presumably, this exploratory stage was elicited by the high levels of surprise in an environment that contained rules that were not followed consistently over time. Crucially, this subgroup A also exhibited low scores in the explicit memory test. We posit that for participants in the narrow-variance subgroup B showing higher level of explicit knowledge, the stochastic signature reflected an error correction mode of learning throughout, from the beginning to the end of the task.

The behavioral differentiation between subgroups A and B, suggests that the transition from exploratory behavior into error correction depends on some minimal level of explicit knowledge that needs to be obtained. This conclusion contradicts the current assumption that both explicit and implicit SL always reflects error correction (Hasson, 2017; Frost et al., 2019). For instance, within theories arguing that both explicit and implicit learning systems operate simultaneously (i.e., dual-system approach), it has been suggested that during a learning episode, implicit associative learning occurs initially, which leads to the formulation of predictive "wagers" that steadily become more

correct, leading to explicit awareness of the learned patterns (Dale et al., 2012). The initial stage of exploratory, memoryless sampling from the perceptual input that has emerged from our analyses has so far been overlooked.

The new methodology introduced in this work is grounded on deliberate *vs.* spontaneous movement classes (Torres, 2011), with different classes of temporal dynamics. Framed in this way, the error correction code would correspond to the deliberate movements intended to a goal. Such movements are well characterized by paths that can be traversed with different temporal dynamics and remain impervious to changes in speed (Atkeson and Hollerbach, 1985; Nishikawa et al., 1999; Torres and Zipser, 2004; Torres and Andersen, 2006). Within such learning, the path to the goal is independent of how long it takes to attain it and remains stable despite the moment-by-moment temporal structure of the stimuli, which must be learned and transformed into physical, motoric action (Torres and Zipser, 2002). This invariance is akin to timescale invariance in models of temporal learning, strongly supported by empirical data (Gallistel and Gibbon, 2000). In contrast, exploratory learning, would correspond to the class of spontaneous movements, i.e., highly sensitive to contextually driven variations in temporal dynamics of the stimuli (Torres, 2011; Brincker and Torres, 2018). These different dynamics can be distinguished in the variance profile of the learners in the mixed group of **Figure 6A**. They respond dynamically different across blocks, depending on target type. In this sense, exploratory trajectories with higher variance, lower explicit memory scores and fundamentally different target responses, are contextually more informative than error correcting trajectories. According to their initial exponential distribution signature, during this exploratory mode, all events are equally probable. The system samples without restriction. This mode may increase the chances of surprise, grabbing the system's attention to some context-relevant events, perhaps self-discovering (through guess and confirmation of the regularity) the transition toward a consistent, ever more systematic state that may eventually result in a desirable, stable task-goal. At this point the system seems to enter and guide the error correction mode under a Gaussian regime. Such smooth transition across memoryless exponential, heavy tailed, skewed distributions to Gaussian modes are evident in the convergence of the variance profiles of the two subgroups in the mixed group to a common regime (locally obtained for the MMS Gamma variance in **Figure 6A** and globally computed in **Figures 9E,F** for the Gamma family of fluctuations in Gamma distribution shapes). Their smoothly evolving transition dynamics were also unveiled in the stochastic signatures of their rates of change (**Figure 10**).

We propose to trace back the newly characterized exploratory mode to the neonatal stages of learning. Such stages appear prior to the maturation of perceptual systems and are guided by endogenous bodily fluctuations that the infant senses from self-generated movements (likely heavily involving central pattern generators already operating at birth; Grillner and El Manira, 2020). To that end, we cite how neonates learn, perhaps supporting our idea that humans' mental strategies and the different learning modes discovered here, are embodied, grounded on the types of learning that we ontogenetically transitioned through during early infancy, when seemingly purposeless movements preceded intentional ones (Thelen, 2001).

Studies of infants exploring an environment where the mother serves as an anchoring reference place, find that the babies explore using interleaving segments of progressive movements with lingering episodes (Frostig et al., 2020). They confirm that such exploratory behavior is homologous across species and situations (Drai et al., 2000; Frostig et al., 2020). Furthermore, a recent study from the SL domain demonstrated that infants prefer to attend to events that are neither highly unpredictable nor highly predictable (Kidd et al., 2012). The authors suggest that this effect is a characteristic of immature members of any species, that must be highly selective in sampling information from their environment to learn efficiently. We add to these interpretations a concrete stochastic model and suggest that infants attend to relatively unpredictable environments because these are ideal for the exploratory behavior that dominates early stages of surprise- and curiosity-driven motor learning in neonates (Torres et al., 2016) and infants (Torres et al., 2013a). Across early stages of life, when altricial mammals generally mature their somatic-sensory-motor systems (More and Donelan, 2018), human infants acquire a stable motor percept. As they undergo motor milestones (myelination, acquisition of motor, and sensory maps, etc.), the families of PDFs that are empirically estimated from their bodily biorhythmic motions, transition from spontaneously purposeless, memoryless exponential to intentionally purposeful, highly predictive Gaussian (Torres et al., 2013a).

Given our results, it appears that the exploratory type of learning is preserved throughout adulthood, and that there are conditions in which this exploratory, memoryless learning with high SNR, emerges on demand, and is likely extremely advantageous. An open question is, when is this type of learning beneficial? One possibility is that it supports flexibility within the system, as it provides it with a broader range of information that would have been missed by a premature systematic biasing toward a regularity, without allowing/evoking wondering behavior. That is, in changing, unstable environments, it may be best to initially gather as much information as possible, before committing to an error correction, goal-targeted mode. This direction, which is beyond the scope of the present work, may be tested by examining whether exploratory periods emerge during processes that require flexibly extending an existing solution to new context, known in motor control as transfer and generalization (Krakauer et al., 2006; Torres et al., 2013b; Wu and Smith, 2013; Tanaka and Sejnowski, 2015), but such studies

are rare. This research may bear important implications for clinical programs that are currently grounded in animal models of conditional reinforcement that do not address the possible benefits of an exploratory mode of learning, whereby the value of a reward self-emerges internally from the self-discovery process, rather than externally given and *a priori* set by an external agent.

Related to these proposed processes, are recent models of human and machine learning that emphasize the role of curiosity within the learning system (Pathak et al., 2017; Dubey and Griffiths, 2020). These models suggest that the causal environment determines when curiosity is driven by novelty or by prediction errors. In an environment where the past and future occurrences of stimuli are independent of each other, the optimal solution for gaining a future reward is to explore novel stimuli. This novelty mode, that has been referred to as novelty-error-based (Dubey and Griffiths, 2020), and the standard prediction-error-based approaches have at their heart the same computational problem: optimize by minimization of an error that depends on a given targeted goal, while using prior information. Though also fueled in part by curiosity, the exploratory mode suggested in our present results is computationally different from the error correction mode. As explained, in our exploratory mode initially, all future events are equally probable, the SNR of the stochastic process is high, and the system does not yet operate with a goal in mind. In fact, it must self-discover it, gathering as much information as possible in a memoryless way, without yet committing to an objective function, a value function, a policy, or a reward. In this case, opposite to RL, Bayesian Reinforcement leaning and active inference, the system does not minimize surprise.

We argue that to characterize learning properly, this additional type of endogenous, curious *unexpected* exploration should be incorporated into future models of inference and learning. Indeed, intrinsic motivation and curiosity has become a dominant theme in machine learning and artificial intelligence over the past years (Daw et al., 2006; Baranes and Oudeyer, 2009; Schmidhuber, 2010; Still and Precup, 2012; Little and Sommer, 2013; Friston et al., 2017; Schwartenbeck et al., 2019). Perhaps the best example of this is active inference and learning (Friston et al., 2011, 2016). Active inference provides an account of optimal behavior in terms of maximizing the evidence for forward, world or generative models of engagement with the world. In other words, instead of learning to maximize reward, agents maximize model evidence or marginal likelihood (as scored with evidence bounds or variational free energy; Winn and Bishop, 2005).

In active inference, behaviors are chosen to maximize both expected value and expected information gain (i.e., expected free energy) (Parr and Friston, 2019). Statistically speaking, this ensures that behavior complies with both the principles of optimum Bayesian decision theory (Berger, 1993) and Bayesian design (MacKay, 2003; Parr and Friston, 2019). This leads naturally to an initial phase of exploratory behavior driven by expected information gain (a.k.a. expected Bayesian supplies, intrinsic value, epistemic affordance, etc.), which then gives way to exploitative behavior driven by expected value (a.k.a., prior preferences, extrinsic value pragmatic affordance, etc.). Our results speak of a different facet of this transition, namely one where *the system has no expectation whatsoever*. Instead, all future events are equally probable and signal information is at its highest, maximizing surprise. There is at this point, no gradient direction pointing the system towards descending error. During this initial naïve learning phase, the system casts a broad net over all incoming information that enhances the chance for a surprising event, before committing to any salience or regularity. This is precisely opposite to (complementary of) the minimization of predictive error or the consequences of predictive error. Crucially, the fact that the transition between the memoryless exploration mode and the error correction mode could be predicted from an independent assessment of behavioral data (i.e., explicit knowledge) lends a predictive validity to our analysis of the neuronal correlates of a new aspect of learning. Only after a goal self-emerges it can be incorporated into an objective function or model, transitioning from trial-and-error model-free, to error-correction model-based learning, as an objective function gets defined. At that stage, minimizing expected surprise, as in active inference, fits well with the error-correction phase that all participants eventually converged to. However, active inference, as other learning frameworks, will need to be modeled differently from its current conceptualizations of optimal expectation-driven exploration to include the newly discovered spontaneous and memoryless stage of learning.

Through the motor control lens, we posit that the new (expectation-free) exploratory mode described here, scaffolds the emergence of what we have coined *spontaneous autonomy* (Torres, 2018b), different from deliberate autonomy (i.e., derived from targeted error-correction). It will be critical to include random-memoryless, expectation-free exploratory learning with high signal content, in the future design of autonomous robots/agents. This type of autonomy can be realized through the self-referenced discovery of the relationships between actions and their consequences. The latter leads to the sense of action ownership and to the volitional control of physical acts that are congruent with one's own mental intent (Torres et al., 2013b). We posit that only then, after acquiring this selectively adapted balance between autonomous and controlled acts, will others understand one's intent and contribute, through co-adaptation, to the person's agency.

We have in summary shown that using new analytical techniques, we can get a precise characterization of the dynamic nature of SL, the rich stochastic signal embedded in fluctuations that are traditionally treated as gross data and the differential nature of contrasting learning modes. Investigation is warranted on whether these results generalize to other SL paradigms,

and to the acquisition of predictive information in learning in general. Of particular interest, are questions of individual differences, and the degree to which the exploratory and error correction learning modes may be differently recruited on demand by the same learner under different contexts. We here offer methods that allow to investigate these and many new questions in future SL research from the perspective of the nascent, developing motor systems and their richly layered dynamic and stochastic motor percepts.

Data availability statement

The raw data supporting the conclusions of this article will be made available by the authors, without undue reservation.

Ethics statement

The studies involving human participants were reviewed and approved by Institutional Review Board of Tel Aviv University. The patients/participants provided their written informed consent to participate in this study.

Author contributions

AV ran the experiment, recorded the original EEG data, and analyzed the behavioral results. ET developed the novel analytical tools, analyzed the EEG data, and provided the statistical inference/interpretation from prior empirical and computational work. Both authors contributed to the conceptualization of this work, interpretation of the results, and the preparation of the manuscript. This work merges the motor control dynamic and stochastic perspective, brought in by ET, with the study of statistical learning, brought by AV.

Funding

This work was funded by the New Jersey Governor's Council for the Research and Treatments of Autism CAUTI18ACE.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

Supplementary material

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fnins.2022.1033776/full#supplementary-material

References

Atkeson, C. G., and Hollerbach, J. M. (1985). Kinematics Features of unrestrained vertical arm movements. *J. Neurosci.* 5, 2318–2330. doi: 10.1523/JNEUROSCI.05-09-02318.1985

Baranes, A., and Oudeyer, P. Y. (2009). Robust intrinsically motivated exploration and active learning. *IEEE Trans. Auton. Ment. Dev.* 1, 155–169. doi: 10.1109/TAMD.2009.2037513

Batterink, L. J., Paller, K. A., and Reber, P. J. (2019). Understanding the neural bases of implicit and statistical learning. *Top. Cogn. Sci.* 11, 482–503. doi: 10.1111/tops.12420

Berger, J. O. (1993). *Statistical decision theory and Bayesian analysis*. New York, NY: Springer-Verlag.

Bigdely-Shamlo, N., Mullen, T., Kothe, C., Su, K. M., and Robbins, K. A. (2015). The PREP pipeline: Standardized preprocessing for large-scale EEG analysis. *Front. Neuroinform.* 9:16. doi: 10.3389/fninf.2015.00016

Bogaerts, L., Richter, C. G., Landau, A. N., and Frost, R. (2020). Beta-band activity is a signature of statistical learning. *J. Neurosci.* 40, 7523–7530. doi: 10.1523/JNEUROSCI.0771-20.2020

Brincker, M., and Torres, E. B. (2018). "Chapter 1- Why study movement variability in autism," in *Autism : The movement sensing perspective*, eds E. B. Torres and C. Whyatt (Boca Raton, FL: CRC Press), 386. doi: 10.1201/9781315372518-2

Censor, N., Sagi, D., and Cohen, L. G. (2012). Common mechanisms of human perceptual and motor learning. *Nat. Rev. Neurosci.* 13, 658–664. doi: 10.1038/nrn3315

Christiansen, M. H. (2019). Implicit statistical learning: A tale of two literatures. *Top. Cogn. Sci.* 11, 468–481. doi: 10.1111/tops.12332

Conway, C. M. (2020). How does the brain learn environmental structure? Ten core principles for understanding the neurocognitive mechanisms of statistical learning. *Neurosci. Biobehav. Rev.* 112, 279–299. doi: 10.1016/j.neubiorev.2020.01.032

Crivello, C., Phillips, S., and Poulin-Dubois, D. (2018). Selective social learning in infancy: Looking for mechanisms. *Dev. Sci.* 21:e12592. doi: 10.1111/desc.12592

Dale, R., Duran, N., and Morehead, R. (2012). Prediction during statistical learning, and implications for the implicit/explicit divide. *Adv. Cogn. Psychol.* 8, 196–209. doi: 10.5709/acp-0115-z

Daw, N. D., O'Doherty, J. P., Dayan, P., Seymour, B., and Dolan, R. J. (2006). Cortical substrates for exploratory decisions in humans. *Nature* 441, 876–879. doi: 10.1038/nature04766

Dayan, P., and Balleine, B. W. (2002). Reward, motivation, and reinforcement learning. *Neuron* 36, 285–298. doi: 10.1016/S0896-6273(02)00963-7

Drai, D., Benjamini, Y., and Golani, I. (2000). Statistical discrimination of natural modes of motion in rat exploratory behavior. *J. Neurosci. Methods* 96, 119–131. doi: 10.1016/S0165-0270(99)00194-6

Drai, D., and Golani, I. (2001). SEE: A tool for the visualization and analysis of rodent exploratory behavior. *Neurosci. Biobehav. Rev.* 25, 409–426. doi: 10.1016/S0149-7634(01)00022-7

Dubey, R., and Griffiths, T. L. (2020). Understanding exploration in humans and machines by formalizing the function of curiosity. *Curr. Opin. Behav. Sci.* 35, 118–124. doi: 10.1016/j.cobeha.2020.07.008

Friston, K., FitzGerald, T., Rigoli, F., and Schwartenbeck, P. O Doherty J., and Pezzulo, G. (2016). Active inference and learning. *Neurosci. Biobehav. Rev.* 68, 862–879. doi: 10.1016/j.neubiorev.2016.06.022

Friston, K., Mattout, J., and Kilner, J. (2011). Action understanding and active inference. *Biol. Cybern.* 104, 137–160. doi: 10.1007/s00422-011-0424-z

Friston, K. J., Lin, M., Frith, C. D., Pezzulo, G., Hobson, J. A., and Ondobaka, S. (2017). Active Inference, curiosity and insight. *Neural Comput.* 29, 2633–2683. doi: 10.1162/neco_a_00999

Frost, R., Armstrong, B. C., and Christiansen, M. H. (2019). Statistical learning research: A critical review and possible new directions. *Psychol. Bull.* 145, 1128–1153. doi: 10.1037/bul0000210

Frostig, T., Alonim, H., Scheingesicht, G., Benjamini, Y., and Golani, I. (2020). Exploration in the presence of mother in typically and non-typically developing pre-walking human infants. *Front. Behav. Neurosci.* 14:580972. doi: 10.3389/fnbeh.2020.580972

Gallistel, C. R., and Gibbon, J. (2000). Time, rate, and conditioning. *Psychol. Rev.* 107, 289–344. doi: 10.1037/0033-295X.107.2.289

Grillner, S., and El Manira, A. (2020). Current principles of motor control, with special reference to vertebrate locomotion. *Physiol. Rev.* 100, 271–320. doi: 10.1152/physrev.00015.2019

Hartigan, J. A., and Hartigan, P. M. (1985). The dip test of unimodality. *Ann. Stat.* 13, 70–84. doi: 10.1214/aos/1176346577

Hasson, U. (2017). The neurobiology of uncertainty: Implications for statistical learning. *Philos. Trans. R. Soc. Lond. B Biol. Sci.* 372:20160048. doi: 10.1098/rstb.2016.0048

Kidd, C., Piantadosi, S. T., and Aslin, R. N. (2012). The Goldilocks effect: Human infants allocate attention to visual sequences that are neither too simple nor too complex. *PLoS One* 7:e36399. doi: 10.1371/journal.pone.0036399

Krakauer, J. W., Mazzoni, P., Ghazizadeh, A., Ravindran, R., and Shadmehr, R. (2006). Generalization of motor learning depends on the history of prior action. *PLoS Biol.* 4:e316. doi: 10.1371/journal.pbio.0040316

Kunar, M. A., Flusberg, S. J., Horowitz, T. S., and Wolfe, J. M. (2007). Does contextual cuing guide the deployment of attention? *J. Exp. Psychol.* 33, 816–828. doi: 10.1037/0096-1523.33.4.816

Little, D. Y., and Sommer, F. T. (2013). Learning and exploration in action-perception loops. *Front. Neural Circuits* 7:37. doi: 10.3389/fncir.2013.00037

Lleonart, J., Salat, J., and Torres, G. J. (2000). Removing allometric effects of body size in morphological analysis. *J. Theor. Biol.* 205, 85–93. doi: 10.1006/jtbi.2000.2043

MacKay, D. J. C. (2003). *Information theory, inference, and learning algorithms*. Cambridge: Cambridge University Press.

Makovski, T., and Jiang, Y. V. (2010). Contextual cost: When a visual-search target is not where it should be. *Q. J. Exp. Psychol.* 63, 216–225. doi: 10.1080/17470210903281590

Monge, G. (1781). *Memoire sur la theorie des deblais et des remblais., Histoire de l' Academie Royale des Science; avec les Memoires de Mathematique et de Physique*. Paris: De L'imprimerie Royale.

More, H. L., and Donelan, J. M. (2018). Scaling of sensorimotor delays in terrestrial mammals. *Proc. Biol. Sci.* 285:20180613. doi: 10.1098/rspb.2018.0613

Moser, J., Batterink, L., Li Hegner, Y., Schleger, F., Braun, C., Paller, K. A., et al. (2021). Dynamics of nonlinguistic statistical learning: From neural entrainment to the emergence of explicit knowledge. *Neuroimage* 240:118378. doi: 10.1016/j.neuroimage.2021.118378

Nastase, S., Iacovella, V., and Hasson, U. (2014). Uncertainty in visual and auditory series is coded by modality-general and modality-specific neural systems. *Hum. Brain Mapp.* 35, 1111–1128. doi: 10.1002/hbm.22238

Nishikawa, K., Murray, S., and Flanders, M. (1999). Do arm postures vary with the speed of reaching? *J. Neurophysiol.* 81, 2582–2586. doi: 10.1152/jn.1999.81.5.2582

Parr, T., and Friston, K. J. (2019). Generalised free energy and active inference. *Biol. Cybern.* 113, 495–513. doi: 10.1007/s00422-019-00805-w

Pathak, D., Agrawal, P., Efros, A. A., and Darrell, T. (2017). "Curiosity-driven exploration by self-supervised prediction," in *Proceedings of the 34th international conference on machine learning*, ed. Y. W. T. D. Precup (Sydney: MLResearch Press), 2778–2787. doi: 10.1109/CVPRW.2017.70

Phinyomark, A., Thongpanja, S., Hu, H., Phukpattaranont, P., and Limsakul, C. (2012). "The usefulness of mean and median frequencies in electromyography analysis," in *Computational intelligence in electromyography analysis-A perspective on current applications and future challenges*, ed. G. R. Naik (Rijeka: InTech), 195–220. doi: 10.5772/50639

Romberg, A. R., and Saffran, J. R. (2010). Statistical learning and language acquisition. *Wiley Interdiscip. Rev. Cogn. Sci.* 1, 906–914. doi: 10.1002/wcs.78

Rubner, Y., Tomasi, C., and Guibas, L. J. (1998). "Metric for distributions with applications to image databases," in *Proceedings of the ICCV*, Bombay.

Ryu, J., Bar-Shalita, T., Granovsky, Y., Weissman-Fogel, I., and Torres, E. B. (2021). Personalized biometrics of physical pain agree with psychophysics by participants with sensory over responsivity. *J. Pers. Med.* 11:93. doi: 10.3390/jpm11020093

Saffran, J. R., Aslin, R. N., and Newport, E. L. (1996). Statistical learning by 8-month-old infants. *Science* 274, 1926–1928. doi: 10.1126/science.274.5294.1926

Schmidhuber, J. (2010). Formal theory of creativity, fun, and intrinsic motivation (1990-2010). *IEEE Trans. Auton. Ment. Dev.* 2, 230–247. doi: 10.1109/TAMD.2010.2056368

Schwarb, H., and Schumacher, E. H. (2012). Generalized lessons about sequence learning from the study of the serial reaction time task. *Adv. Cogn. Psychol.* 8, 165–178. doi: 10.5709/acp-0113-1

Schwartenbeck, P., Passecker, J., Hauser, T. U., FitzGerald, T. H., Kronbichler, M., and Friston, K. J. (2019). Computational mechanisms of curiosity and goal-directed exploration. *eLife* 8:e41703. doi: 10.7554/eLife.41703

Sinha, P. (2014). Autism as a disorder of prediction. *Proc. Natl. Acad. Sci. U.S.A.* 42, 15220–15225. doi: 10.1073/pnas.1416797111

Still, S., and Precup, D. (2012). An information-theoretic approach to curiosity-driven reinforcement learning. *Theory Biosci.* 131, 139–148. doi: 10.1007/s12064-011-0142-z

Sutton, R. S. (1992). *Reinforcement learning*. Boston: Kluwer Academic Publishers. doi: 10.1007/978-1-4615-3618-5

Tanaka, H., and Sejnowski, T. J. (2015). Motor adaptation and generalization of reaching movements using motor primitives based on spatial coordinates. *J. Neurophysiol.* 113, 1217–1233. doi: 10.1152/jn.00002.2014

Thelen, E. (ed.) (2001). *Mechanisms of cognitive development: Behavioral and neural perspectives*. Pittsburg, PA: Lawrence Erlbaum Associates Inc.

Torres, E., and Andersen, R. (2006). Space-time separation during obstacle-avoidance learning in monkeys. *J. Neurophysiol.* 96, 2613–2632. doi: 10.1152/jn.00188.2006

Torres, E. B. (2011). Two classes of movements in motor control. *Exp. Brain Res.* 215, 269–283. doi: 10.1007/s00221-011-2892-8

Torres, E. B. (2018a). *Methods for the diagnosis and treatment of neurological disorders*. New Brunswick, NJ: Rutgers State University of New Jersey.

Torres, E. B. (2018b). *Objective biometric methods for the diagnosis and treatment of nervous system disorders*. London: Academic Press.

Torres, E. B., Brincker, M., Isenhower, R. W., Yanovich, P., Stigler, K. A., Nurnberger, J. I., et al. (2013a). Autism: The micro-movement perspective. *Front. Integr. Neurosci.* 7:32. doi: 10.3389/fnint.2013.00032

Torres, E. B., Yanovich, P., and Metaxas, D. N. (2013b). Give spontaneity and self-discovery a chance in ASD: Spontaneous peripheral limb variability as a proxy to evoke centrally driven intentional acts. *Front. Integr. Neurosci.* 7:46. doi: 10.3389/fnint.2013.00046

Torres, E. B., and Lande, B. (2015). Objective and personalized longitudinal assessment of a pregnant patient with post severe brain trauma. *Front. Hum. Neurosci.* 9:128. doi: 10.3389/fnhum.2015.00128

Torres, E. B., Smith, B., Mistry, S., Brincker, M., and Whyatt, C. (2016). Neonatal diagnostics: Toward dynamic growth charts of neuromotor control. *Front. Pediatr.* 4:121. doi: 10.3389/fped.2016.00121

Torres, E. B., and Zipser, D. (2002). Reaching to grasp with a multi-jointed arm (I): A computational model. *J. Neurophysiol.* 88, 1–13. doi: 10.1152/jn.00030.2002

Torres, E. B., and Zipser, D. (2004). Simultaneous control of hand displacements and rotations in orientation-matching experiments. *J. Appl. Physiol.* 96, 1978–1987. doi: 10.1152/japplphysiol.00872.2003

Toth, B., Janacsek, K., Takacs, A., Kobor, A., Zavecz, Z., and Nemeth, D. (2017). Dynamics of EEG functional connectivity during statistical learning. *Neurobiol. Learn. Mem.* 144, 216–229. doi: 10.1016/j.nlm.2017.07.015

Vaskevich, A., and Luria, R. (2018). Adding statistical regularity results in a global slowdown in visual search. *Cognition* 174, 19–27. doi: 10.1016/j.cognition.2018.01.010

Vaskevich, A., and Luria, R. (2019). Statistical learning in visual search is easier after experience with noise than overcoming previous learning. *Vis. Cogn.* 27, 537–550. doi: 10.1080/13506285.2019.1615022

Vaskevich, A., Nishry, A., Smilansky, Y., and Luria, R. (2021). Neural evidence suggests both interference and facilitation from embedding regularity into visual search. *J. Cogn. Neurosci.* 33, 622–634. doi: 10.1162/jocn_a_01667

Winn, J., and Bishop, C. M. (2005). Variational message passing. *J. Mach. Learn. Res.* 6, 661–694.

Wu, H. G., and Smith, M. A. (2013). The generalization of visuomotor learning to untrained movements and movement sequences based on movement vector and goal location remapping. *J. Neurosci.* 33, 10772–10789. doi: 10.1523/JNEUROSCI.3761-12.2013

Zellin, M., Conci, M., von Mühlenen, A., and Müller, H. J. (2013). Here today, gone tomorrow–adaptation to change in memory-guided visual search. *PLoS One* 8:e59466. doi: 10.1371/journal.pone.0059466

frontiers | Frontiers in Neuroscience

TYPE Original Research
PUBLISHED 10 November 2022
DOI 10.3389/fnins.2022.1012964

OPEN ACCESS

EDITED BY
Chenwei Deng,
Beijing Institute of Technology, China

REVIEWED BY
Mingyuan Meng,
The University of Sydney, Australia
Xu Yang,
Beijing Institute of Technology, China

*CORRESPONDENCE
Xin Liao
11767883@qq.com

SPECIALTY SECTION
This article was submitted to Perception Science, a section of the journal Frontiers in Neuroscience

RECEIVED 06 August 2022
ACCEPTED 25 October 2022
PUBLISHED 10 November 2022

CITATION
Hu L and Liao X (2022) Voltage slope guided learning in spiking neural networks.
Front. Neurosci. 16:1012964.
doi: 10.3389/fnins.2022.1012964

Voltage slope guided learning in spiking neural networks

Lvhui Hu[1] and Xin Liao[2]*

[1]School of Intelligent Medicine, Chengdu University of Traditional Chinese Medicine, Chengdu, China, [2]Information Center, Hospital of Chengdu University of Traditional Chinese Medicine, Chengdu, China

A thorny problem in machine learning is how to extract useful clues related to delayed feedback signals from the clutter of input activity, known as the temporal credit-assignment problem. The aggregate-label learning algorithms make an explicit representation of this problem by training spiking neurons to assign the aggregate feedback signal to potentially effective clues. However, earlier aggregate-label learning algorithms suffered from inefficiencies due to the large amount of computation, while recent algorithms that have solved this problem may fail to learn due to the inability to find adjustment points. Therefore, we propose a membrane voltage slope guided algorithm (VSG) to further cope with this limitation. Direct dependence on the membrane voltage when finding the key point of weight adjustment makes VSG avoid intensive calculation, but more importantly, the membrane voltage that always exists makes it impossible to lose the adjustment point. Experimental results show that the proposed algorithm can correlate delayed feedback signals with the effective clues embedded in background spiking activity, and also achieves excellent performance on real medical classification datasets and speech classification datasets. The superior performance makes it a meaningful reference for aggregate-label learning on spiking neural networks.

KEYWORDS

spiking neural networks, spiking neurons, aggregate-label learning, temporal credit-assignment, synaptic adjustment

1. Introduction

The birth and development of artificial intelligence are deeply inspired by the sophisticated biological brain, such as the striking deep learning represented by the artificial neural network (ANNs), which has attracted considerable attention in the past decade (LeCun et al., 2015). ANNs highly abstract biological neurons, and obtains the analog outputs by the weighted sum of the analog inputs through activation function. This conversion process is somewhat consistent with the biological spiking process, and the analog inputs and outputs are also regarded as equivalent to the firing rates of biological neurons (Rueckauer et al., 2017). However, ANNs still lack biological realism compared to physiological neural networks that utilize binary spikes for information transfer (Bengio et al., 2015).

Then, spiking neural networks (SNNs) offer a new computing paradigm with theoretical advantages in computational efficiency and power consumption due to the adoption of the binary spiking mechanism. However, these advantages have not been fully exploited, and the results are far from achieving the desired impact. One of the

major reasons is the lack of efficient learning algorithms, so research on SNN algorithms remains attractive. Nevertheless, many valuable works have emerged. Among them, depending on the presence of additional teaching signals, existing SNN algorithms can be roughly divided into supervised and unsupervised.

Neurophysiological studies have shown that the long-term potentiation (LTP) and depression (LTD) of synaptic transmission are ubiquitous phenomena existing in almost every excitatory synapse in the mammalian brain (Malenka and Bear, 2004). Spike-timing dependent plasticity (STDP) rule (Bi and Poo, 1998), which combines these two phenomena, becomes a feasible unsupervised learning rule benefiting by its definite biological basis. Then STDP intrigues the research of local learning rules that imitate the neuroscience mechanisms (Masquelier et al., 2007; Diehl and Matthew, 2015; Tavanaei and Maida, 2017a). For example, STDP rules have been applied to an SNN architecture that simulates visual function to promote neurons show the selectivity of orientation and disparity (Barbier et al., 2021), to shallow convolutional SNNs to realize near-real-time processing of events collected from neuromorphic vision sensors (She and Mukhopadhyay, 2021), and to weight-quantized SNNs to complete online learning (Hu et al., 2021), etc. In addition, variants of STDP have also been embedded into Inception-like SNNs for highly parallel feature extraction (Meng et al., 2021) or ensemble convolutional SNNs for object recognition (Fu and Dong, 2021). This biologically inspired learning do not require regulatory signals and is easy to execute, making it attractive to hardware implementation of emerging memory devices (Burr et al., 2016; Zhou et al., 2022). However, such local learning rules are more suitable for small-scale pattern recognition tasks, and it is difficult for them to be directly applied in complex tasks due to the lack of global information related to convergence for large models (Mozafari et al., 2018).

On the other hand, there is also documented evidence supporting the existence of instruction-based learning in the central nervous system (Knudsen, 1994; Thach, 1996). Over the years, a growing number of supervised learning algorithms of SNN have been proposed (Ponulak and Kasiński, 2010; Florian, 2012; Mohemmed et al., 2012; Xu et al., 2013b; Memmesheimer et al., 2014; Zhang et al., 2018a,b, 2019; Luo et al., 2022), and some of them obtained comparable accuracies to that of ANNs in large-scale applications. SpikeProp (Bohte et al., 2000) is a classical supervised learning method of SNNs, which is derived from the gradient descent algorithm of ANNs. While the application of this algorithm is limited by the fact that each neuron can only fire once, so the multi-spike version of it are proposed to improve performance (Ghosh-Dastidar and Adeli, 2009; Xu et al., 2013a). As for the critical dilemma of non-differentiable discrete spikes in SNNs, Spikeprop uses a linear assumption of membrane potential at these time instants to bypass it. The other way proposed in SLAYER (Shrestha and Orchard, 2018) to handle it is to replace the derivatives of these non-differentiable moments with approximate functions, SuperSpike (Zenke and Ganguli, 2018) algorithm uses the surrogate gradients, and DSR (Meng et al., 2022) uses gradients of sub-differentiable mappings. These algorithms and some others (Wu et al., 2018c, 2019) almost all follow the idea of back-propagation through time (BPTT), which makes full use of information on both time and space scales, but it also means quite a bit of computing and storage requirements.

Beyond these, there are some situations where the guidance signals are ambiguous. For example, animal survival behavior to identify whether small clues in the environment represent danger or opportunity involves detecting relationships between multiple clues and ambiguous long-delayed feedback signals. Multi-Spike Tempotron (MST) (Gütig, 2016), an aggregate-label learning algorithm, is proposed to train a detector to automatically respond wherever a valid clue appears, given only the number of desired spikes. It uses the distance between the true threshold and the a critical threshold (under which a specific number of spikes can be fired) as the error signal for weight adjustment, enabling it to obtain robust and powerful learning capabilities. Then TDP1 (Yu et al., 2018) is proposed to simplify the iteration calculation in MST and improve the learning efficiency. However, they are still computationally expensive due to the need to calculate the critical threshold. Therefore, MPD-AL (Zhang et al., 2019) directly adjusts the weight from the membrane voltage, which greatly reduces the computing requirements. However, the disadvantage of this method is that there is a possibility that the tunable point cannot be found.

Inspired by MPD-AL, we propose an voltage slope-guided algorithm (VSG). When the number of spikes emitted by the output neuron is not equal to its desired spike number, an appropriate point is selected to adjust the weight according to the slope of the membrane voltage, so that the neuron can emit more spikes or remove redundant spikes. The proposed method avoids the dilemma of failing to find the adjustment points, and does not need iterative calculation to find the critical threshold. The comparative experiments with MPD-AL, MST, and TDP1 verify its superiority, and the classification results on realistic datasets further proves its practical performance.

The rest of the article is organized as follows: In Section 2, we introduce the proposed algorithm and compare it with several other algorithms. In Section 3, we conduct a series of experiments to verify the performance of the algorithm. Finally, the algorithm is analyzed and discussed in Section 4.

2. Neuron model and learning algorithm

In this section, the neuron model employed will be first introduced, followed by the proposed VSG algorithm, and

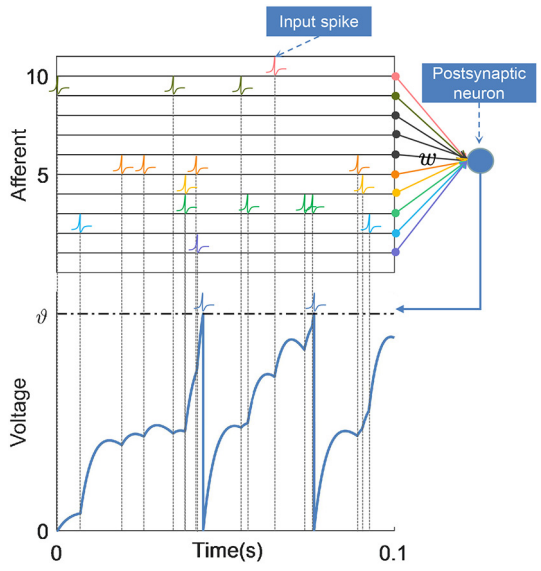


FIGURE 1
Schematic of neuronal structure and membrane voltage dynamics. The postsynaptic neuron integrates spikes from afferent neurons over time, and each incoming spike contributes to the membrane voltage together with the corresponding synaptic weights. When the membrane voltage of the postsynaptic neuron crosses the firing threshold ϑ, it emits a spike and the voltage is reset.

finally this algorithm will be compared with its counterparts, highlighting their differences.

2.1. Neuron model

The leaky integrate-and-fire neuron model (Maass and Bishop, 1999; Gütig, 2016) is one of the most widely used spiking neuron models, benefiting from its computational simplicity and modest biological reliability. So we also adopt it in this article.

The postsynaptic neuron receives spikes transmitted from its N presynaptic neurons through synapses, which induce postsynaptic potentials (PSPs) on the postsynaptic neuron, resulting in changes in its membrane voltage $V(t)$, as shown in Figure 1. Thus, the membrane voltage of the postsynaptic neuron gradually rises from the resting state $V_{rest} = 0$. When the membrane voltage crosses the threshold ϑ, the neuron fires a spike, and the membrane voltage quickly resets to the resting potential, then it enters a refractory period. This process can be expressed as:

$$V(t) = V_{rest} + \sum_{i=1}^{N} w_i \sum_{t_i^j < t} K(t - t_i^j) - \sum_{t_s^j < t} \eta\left(t - t_s^j\right), \quad (1)$$

where w_i is the weight of the synapse established with the i-th afferent neuron, and t_i^j denotes the time of the j-th spike from the afferent neuron. t_s^j denotes the time of j-th spike emitted by this postsynaptic neuron. $K(\cdot)$ and $\eta(\cdot)$ characterize the normalized PSP kernel and refractory period, respectively, which are defined as

$$K(x) = V_0\left[\exp\left(-\frac{x}{\tau_m}\right) - \exp\left(-\frac{x}{\tau_s}\right)\right], \quad x > 0, \quad (2)$$

and

$$\eta(x) = \vartheta \cdot \exp\left(-\frac{x}{\tau_m}\right), \quad x > 0, \quad (3)$$

where τ_m and τ_s are the membrane time constant and the synaptic time constant, which together control the shape of the PSP. V_0 is a coefficient that normalizes the PSP. These two kernels only make sense when $x > 0$, since a spike only takes effect at the time after its occurrence.

2.2. Voltage slope guided learning

Unlike algorithms that generate an exact desired spike train, VSG aims to generate a desired number of spikes in response to an input pattern. When the actual spike count N_o is more or less than the desired count N_d, the network parameters are adjusted:

1. $N_o < N_d$: When the actual spikes are insufficient, the network parameters are strengthened so that more spikes can be delivered. Thus, the time instant with the largest membrane voltage slope (except the existing spike times) is selected as the critical time t^*. The membrane voltage $V(t^*)$ at this moment has the strongest upward trend. Adjusting the membrane voltage at this point will be more efficient compared to other locations.
2. $N_o > N_d$: When more spikes are fired than the expectation, the redundant spikes should be removed by weakening the network parameters. Therefore, the critical moment t^* will be selected from the existing spike times. On the contrary, among these moments, the point with the weakest upward trend of membrane voltage crossing the threshold is chosen. Because it can be removed with less effort than other spikes.

As shown in Figure 3A (left), the red arrows and green arrows, respectively, represent the critical points if more or less spikes need to be emitted in the case that there are already five output spikes.

For these two cases, we construct error function based on the distance between the critical membrane voltage $V(t^*)$ and its target membrane voltage V_{tar}. In the case of $N_o < N_d$, it is obvious that the target voltage should be equal to the threshold in order to emit more spikes. While in the case of $N_o > N_d$, the critical membrane voltage should be lowered in order to remove

the spike, so the target voltage can be set as the resting potential V_{rest}:

$$E = \frac{1}{2}\left(V\left(t^*\right) - V_{tar}\right)^2, \tag{4}$$

where

$$V_{tar} = \begin{cases} \vartheta, & N_o < N_d, \\ V_{rest}, & N_o > N_d. \end{cases} \tag{5}$$

Then the gradient descent method is applied to obtain the weight updating rule:

$$\Delta\omega_i = -\eta\frac{\mathrm{d}E}{\mathrm{d}w_i} = -\eta\left(V\left(t^*\right) - V_{tar}\right)\frac{\mathrm{d}V(t^*)}{\mathrm{d}w_i}, \tag{6}$$

η is the learning rate which define the update magnitude of the synaptic weights. In fact, $\pm\eta \cdot \mathrm{d}V/\mathrm{d}w$ can also be used directly to enhance/weaken weights during the experiment without considering the error function, which has a learning efficiency similar to Equation (6), as shown in Figure 6B.

Without loss of generality, suppose that there is a fully connected network with L ($L \geq 2$) layers. For a neuron s in layer L (the output layer), if the output spike count is not equal to its desired number, all synaptic weights that contribute to its firing will be adjusted. Assuming that the critical spike time of the neuron is t^*, and the corresponding membrane voltage is $V(t^*)$. Then according to Equation (6), all we need to do is to calculate $\mathrm{d}V/\mathrm{d}w$:

2.2.1. Output layer

According to Equation (1), $V(t^*)$ is not only affected by the input spikes from the previous layer, but also by the previous spikes $t_s^f < t^*$ ($f = 1, 2, ..., F$) excited by the neuron itself, therefore,

$$\frac{\mathrm{d}V(t^*)}{\mathrm{d}w_{is}^L} = \frac{\partial V(t^*)}{\partial w_{is}^L} + \sum_{f=1}^{F}\frac{\partial V(t^*)}{\partial t_s^f}\frac{\partial t_s^f}{\partial w_{is}^L}, \tag{7}$$

where w_{is}^L is the synaptic weight between i-th neuron in the layer $L-1$ and s-th neuron in the layer L.

From Equation (1), the first term of Equation (7) can be expressed as

$$\frac{\partial V(t_x)}{\partial w_{is}^L} = \sum_{t_i^j < t^x} K\left(t_x - t_i^j\right), \tag{8}$$

where $t_x \in \left\{t_s^1, t_s^2, \cdots, t_s^F, t^*\right\}$, t_i^j is the j-th spike of the i-th neuron in layer $L-1$. While for the second term of Equation (7), we have

$$\frac{\partial V(t^*)}{\partial t_s^f} = -\frac{\vartheta}{\tau_m}\exp\left(-\frac{t^* - t_s^f}{\tau_m}\right), \tag{9}$$

and

$$\frac{\partial t_s^f}{\partial w_{is}^L} = \frac{\partial t_s^f}{\partial V(t_s^f)}\frac{\partial V(t_s^f)}{\partial w_{is}^L}, \tag{10}$$

where $\partial V(t_s^f)/\partial w_{is}^L$ can be calculated by Equation (8). Suppose n^l is the number of neurons in the l-th layer. Then following the linear hypothesis for the voltage crossing threshold in Bohte et al. (2002) and Yu et al. (2018), we get

$$\frac{\partial t_s^f}{\partial V(t_s^f)} = -\left(\frac{\partial V(t_s^f)}{\partial t_s^f}\right)^{-1} = -\left(\frac{\partial V(t)}{\partial t}\Big|_{t=t_s^{f-}}\right)^{-1}, \tag{11}$$

where

$$\frac{\partial V(t)}{\partial t} = \sum_{i=1}^{n^L} w_{is}^L \sum_{t_i^j < t}\kappa(t - t_i^j) + \sum_{t_s^f < t}\frac{\eta(t - t_s^f)}{\tau_m}, \tag{12}$$

$$\begin{aligned}\kappa(t - t_i^j) = \frac{\partial K(t - t_i^j)}{\partial t} &= \frac{V_0}{\tau_s}\exp\left(-\frac{t - t_i^j}{\tau_s}\right) \\ &- \frac{V_0}{\tau_m}\exp\left(-\frac{t - t_i^j}{\tau_m}\right).\end{aligned} \tag{13}$$

2.2.2. Hidden layers

Suppose w_{ij}^l is the synaptic weight between i-th neuron in the layer $l-1$ and j-th neuron in the layer l. It has an impact on the spike time $t_j^{m,l}$, i.e., the m-th ($m = 1, 2, \cdots$) spike time of the neuron j in layer l, and then affect the spike time of neurons in all the subsequent layers through $t_j^{m,l}$. Therefore, the derivative of $V(t^*)$ with respect to w_{ih}^l ($1 \leq l \leq L-1$) is

$$\frac{\mathrm{d}V(t^*)}{\mathrm{d}w_{ij}^l} = \sum_{t_j^{m,l} < t^*}\frac{\partial V(t^*)}{\partial t_j^{m,l}}\frac{\partial t_j^{m,l}}{\partial w_{ij}^l}, \tag{14}$$

where $\partial t_j^{m,l}/\partial w_{ij}^l$ can be calculated just like Equation (10). $\partial V(t^*)/\partial t_j^{m,l}$, the key term for error propagation between layers, is denoted as $\delta_j^{m,l}$.

For $1 \leq l < L-1$,

$$\begin{aligned}\delta_j^{m,l} \triangleq \frac{\partial V(t^*)}{\partial t_j^{m,l}} &= \sum_{k=1}^{n^{l+1}}\sum_{t_k^{f,l+1}}\frac{\partial V(t^*)}{\partial t_k^{f,l+1}}\frac{\partial t_k^{f,l+1}}{\partial t_j^{m,l}} \\ &= \sum_{k=1}^{n^{l+1}}\sum_{t_k^{f,l+1}}\delta_k^{f,l+1}\cdot\frac{\partial t_k^{f,l+1}}{\partial t_j^{m,l}}, \quad t_j^{m,l} < t_k^{f,l+1} < t^*.\end{aligned} \tag{15}$$

And for $l = L-1$,

$$\delta_j^{m,l} = \frac{\partial V(t^*)}{\partial t_j^{m,l}} + \sum_{t_j^{m,l} < t_s^{f,L} < t^*}\frac{\partial V(t^*)}{\partial t_s^{f,L}}\frac{\partial t_s^{f,L}}{\partial t_j^{m,l}}. \tag{16}$$

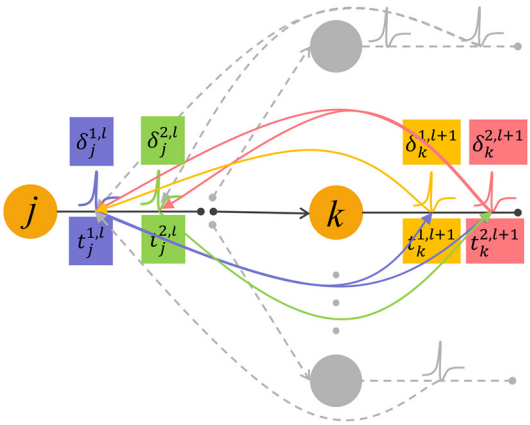


FIGURE 2
The feedforward propagation of spikes and backpropagation of errors. The neuron j in the l-th layer emits two spikes $t_j^{1,l}$ and $t_j^{2,l}$, they affect the generation of spikes in the next layer after them (purple and green curved arrows). The neuron k in the layer $l+1$ also emits two spikes denoted as $t_k^{1,l+1}$ and $t_k^{2,l+1}$. In the feedback process, a spike generated by neuron k transmit the error signal δ to input spikes that contribute to it (pink and yellow curved arrows).

Noted that the error backpropagation is performed based on spikes, and Equation (15) involves complex spike time relationships when δ propagate back between adjacent layers. As shown in Figure 2, the spike $t_j^{2,l}$ has an effect on the later spike $t_k^{2,l+1}$ emitted by the downstream neuron (green arrow), but has no effect on the earlier spike $t_k^{1,l+1}$. Therefore, when the error signal $\delta_k^{1,l+1}$ corresponds to the spike $t_k^{1,l+1}$ is backpropagated, it will only transmit to the earlier spike $t_j^{1,l}$ that contribute to it (yellow arrow).

From Equation (1), the first term of Equation (16), i.e., the derivative of the membrane voltage with respect to the input spike coming from its presynaptic neuron is calculated as below

$$\frac{\partial V(t^*)}{\partial t_j^{m,L-1}} = -w_{js}^L \cdot \kappa\left(t^* - t_j^{m,L-1}\right), \tag{17}$$

and $\partial V(t^*)/\partial t_s^{f,L}$ can be calculated by Equation (9). And for $1 \le l \le L-1$,

$$\begin{aligned}\frac{\partial t_k^{f,l+1}}{\partial t_j^{m,l}} &= \frac{\partial t_k^{f,l+1}}{\partial V(t_k^{f,l+1})}\frac{\partial V(t_k^{f,l+1})}{\partial t_j^{m,l}} \\ &= \left(\frac{\partial V(t_k^{f,l+1})}{\partial t_k^{f,l+1}}\right)^{-1} w_{jk}^{l+1}\kappa\left(t_k^{f,l+1} - t_j^{m,l}\right).\end{aligned} \tag{18}$$

Thereupon, the whole learning process of the VSG is summarized in Algorithm 1.

Input: T: time duration; Δt: time step; N_d: desired spike number; η: learning rate; S: input spike pattern; ϑ: firing threshold; V_{rest}: resting potential; μ: mean of the Gaussian distribution; σ: standard deviation of the Gaussian distribution;

Initialize: synaptic weights $\boldsymbol{w} \sim N(\mu, \sigma)$, actual spike number $N_o = 0$;
while $N_a \neq N_d$ **do**
  $t_o = \varnothing$, $N_o = 0$;
  for $t = 0 : \Delta t : T$ **do**
    calculate membrane voltage $V(t)$ in response to the input pattern S by Equation (1) and $\partial V(t)/\partial t$ by Equation (12);
    if $V(t) \geq \theta$ **then**
      $t_o \leftarrow t_o \cup \{t\}$;
      $N_o \leftarrow N_o + 1$;
      $V(t) \leftarrow V_{rest}$;
  if $N_o \neq N_d$ **then**
    if $N_o < N_d$ **then**
      $t^* = \arg\max_{t \notin t_o}\{\partial V(t)/\partial t\}$, $V_{tar} = \vartheta$;
    else
      $t^* = \arg\min_{t \in t_o}\{\partial V(t)/\partial t\}$, $V_{tar} = V_{rest}$;
    calculate gradient $\mathrm{d}V(t^*)/\mathrm{d}\boldsymbol{w}$ by Equation (14);
    $\Delta \boldsymbol{w} = -\eta\left(V(t^*) - V_{tar}\right) \cdot \mathrm{d}V(t^*)/\mathrm{d}\boldsymbol{w}$;
    $\boldsymbol{w} \leftarrow \boldsymbol{w} + \Delta \boldsymbol{w}$;
return $\boldsymbol{w}$;

Algorithm 1. Learning algorithm of the VSG.

2.3. Comparison with other aggregate-label learning algorithms

Existing aggregate-label learning works can be divided into threshold-driven methods, such as MST, TDP1, and membrane voltage-driven methods, such as MPD-AL. The threshold-driven method searches for a critical threshold ϑ^* that can increase/decrease the number of spikes by one, then the distance between the critical threshold and the actual firing threshold ϑ is used as the error to update the synaptic weights. However, ϑ^* cannot be solved analytically, it can only be obtained by performing dichotomy in the interval where it may appear. Such a search process must be executed for each update iteration, which is quite time-consuming. As for the membrane voltage-driven method MPD-AL, when more spikes are needed, the time of the maximum peak of membrane voltage (below the

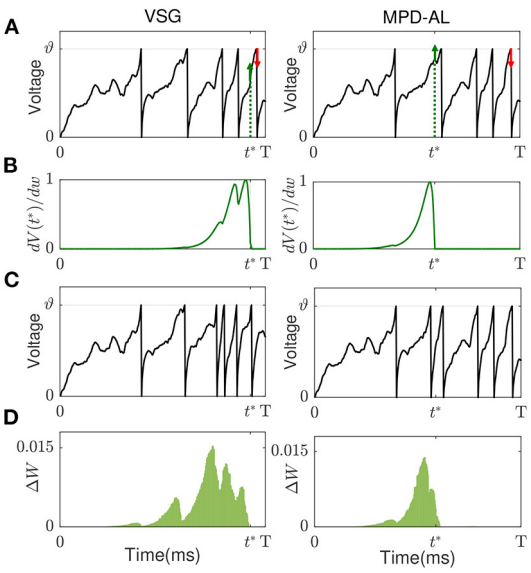


FIGURE 3
The comparison of learning between VSG and MPD-AL. **(A)** The membrane potential traces before learning. t^* indicates the key time point selected when a spike needs to be added. **(B)** Learning curves depicting the spike-timing dependence of the contribution of different synapses to $V(t^*)$. **(C)** The membrane potential traces after learning (with one spike added). **(D)** The amount of weight change before and after learning. For the convenience of observation, each presynaptic neuron sends only one spike, and the synapses in **(B,C)** are arranged in the order of their corresponding spike time.

threshold) is taken as the critical time for enhancing the weights, and when fewer spikes are needed, the last spike time is used as the critical time to weaken the weights, as shown in Figure 3A (right).

Inspired by MPD-AL, we choose the point with the strongest rising trend of membrane voltage at non-spike time and the weakest rising trend of membrane voltage at spike time as the key point for enhancement and weakening, respectively. As shown in Figure 3, taking the addition of a spike as an example, the two algorithms have different choices for t^*, resulting in different learning curves (Figure 3B), thus adding a new spike in different places (Figure 3C).

Neither VSG nor MPD-AL require the complicated process of finding ϑ^*, which makes them more efficient than threshold-driven algorithms. However, when a new spike is required, MPD-AL needs to find all local maxima of the membrane voltage below the threshold and then select the largest one. But sometimes such a point does not exist, especially when there are already many spikes, as shown in Figure 4. In this case, MPD-AL can no longer add spikes and the learning stalls. While VSG does not have this problem, because the point with the largest slope must exist, and it is likely to be raised to the threshold quickly, since a large slope means a large upward trend. Similarly, among the firing spike, the point with the lowest slope means that it has less power to cross the threshold, and when a spike needs to be removed, it takes less effort to eliminate it. We will verify the rationality of this selection of adjustment point through experiments in the next section.

On the other hand, VSG seems to be a little more computationally expensive compared to MPD-AL, because it requires additional computation of the time derivative (slope) of membrane voltage. But this calculation can be integrated into the calculation of membrane voltage, since they use exactly the same intermediate variables (Equations 12 and 1). In this way, as shown in Figure 6A, it takes almost no more time for VSG to calculate the membrane voltage than MPD-AL, with a total time increase of <0.01 s for 1,000 calculations [the average time for one trial is too small, and the device is Intel(R) Core(TM) i5-8400 CPU @ 2.80, 2.81 GHz]. However, MPD-AL spends about three times as long as VSG in finding the adjustment point. Because it needs to find all the local peak of membrane voltages and then perform the maximum operation, while VSG only needs to perform the maximum operation on the membrane voltage slope. Overall, the computational cost of finding adjustment points for VSG is low.

3. Experimental results

Various experiments are carried out to examine the performance of the proposed VSG learning algorithm. We first investigate the learning efficiency of the VSG, and then apply it to learn predictive clues. Several practical classification tasks are performed thereafter to further evaluate its capability.

3.1. Learning of desired number of spikes

In this section, we first investigate the ability of a single neuron to learn to deliver a fixed number of spikes through training of VSG algorithm, and then verify the plausibility of its way of finding adjustment points. Finally, it is compared with several competitive aggregation-label learning algorithms to further evaluate its learning efficiency.

In this first experiment, the learning neuron receives spikes from 500 presynaptic neurons and are trained to deliver 10 spikes over a period of 500 ms. To observe the learning under different input conditions, input spikes are generated by the Poisson distribution at 4 and 20 Hz, respectively, while the synaptic weights are initialized by the same Gaussian distribution $N(0.01, 0.01)$. Figures 5A,B depicts the membrane voltage traces and synaptic weights of this output neuron before (blue) and after (black) learning when the input spike is 4 Hz. The sparse input caused the neuron not to fire initially, after learning, many synaptic weights are enhanced so that the neuron fires 10 spikes. Figures 5C,D shows the situation of neurons

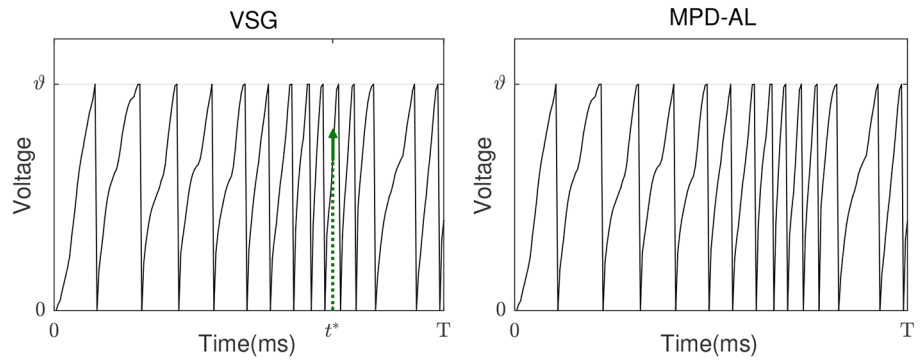


FIGURE 4
Pain point of MPD-AL. When the membrane voltage rapidly accumulates and frequently emits spikes, there may be no local maximum membrane voltage below the threshold. In this case, MPD-AL cannot find t^* if another spike is required **(right)**. However, VSG can find the point where the membrane voltage increases the fastest, namely its t^* **(left)**.

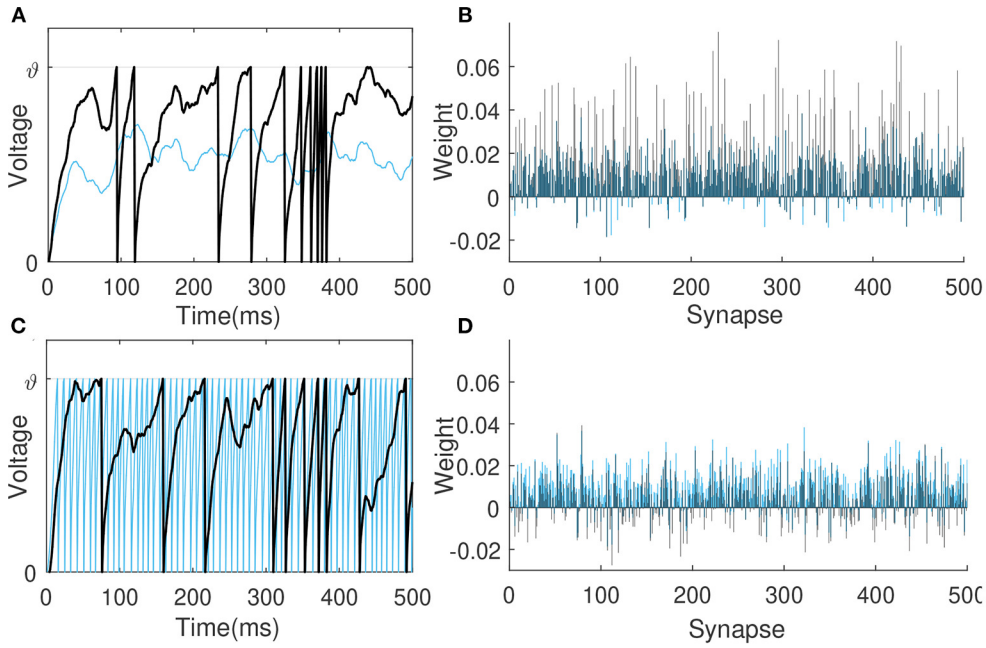


FIGURE 5
Membrane voltage traces and synaptic weights before (blue) and after learning (black). The learning neuron receives 4 Hz **(A,B)** and 20 Hz **(C,D)** spikes from 500 presynaptic neurons, respectively, and are trained to emit 10 spikes in 500 ms.

before and after learning when the input spikes is 20 Hz. Before learning, too dense input causes neurons to emit a lot of spikes, and the VSG algorithm weakens the synaptic weights as a whole, so that neurons only emit 10 spikes at the end.

Then we verify the rationality of the way the VSG finds adjustment points. We choose different combinations of ways to find adjustment points to test the efficiency of training neurons to emit a specified number of spikes. The firing rate of input is 4–10 Hz, which allows the initial spike count to be more or less than the desired count. Other experimental conditions remain unchanged. The average times over 20 trials for several combinations at each desired spike count are reported. If the neuron does not successfully trigger the corresponding number of spikes until 2,000-th iterations, record the time it took to run 2,000 iterations. As shown in Figure 6B, when the desired number of spikes is small, it is more effective to add a new spike at the maximum peak of the subthreshold membrane voltage. But when the desired number of spikes is large, learning may fail due to the inability to find an adjustable point, and the required time will increase greatly, as shown by the combinations of a and

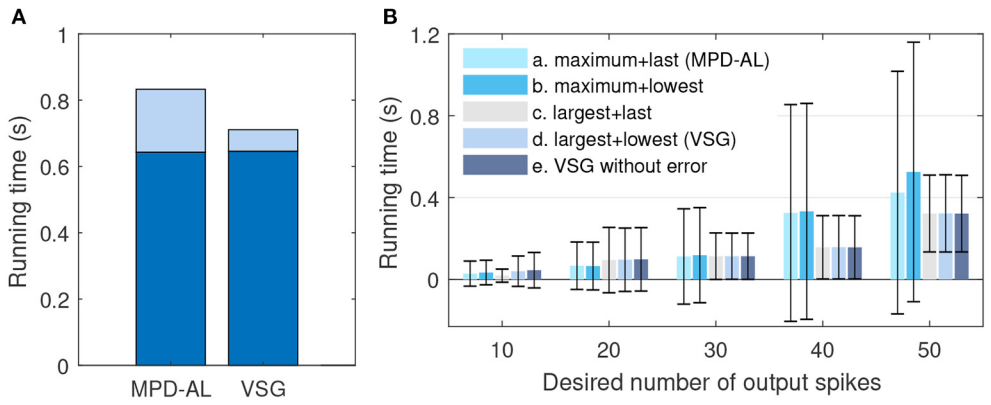


FIGURE 6
The comparison of efficiency between VSG and MPD-AL. **(A)** The total time to calculate the membrane voltage (dark blue) and find the corresponding adjustment point (light blue) for 1,000 trials. **(B)** The average time required to learn the corresponding number of spikes over 20 trials (up to 2,000 iterations each). For cases where one spike needs to be added and removed, several combinations of methods for finding the adjustment point are tested: (a) maximum peak of subthreshold membrane voltage + last spike (MPD-AL), (b) maximum peak of subthreshold membrane voltage + spike with the lowest slope, (c) non-firing point with the largest slope + last spike, (d) non-firing point with the largest slope + spike with the lowest slope (VSG). In addition, the VSG method without considering the error function (e) is also tested.

b. While the method of selecting the point with the largest slope to add a new spike is stable, as shown by the combination of *c*, *d*, and *e*. In addition, by comparing the combination *a* and *b* (or *c* and *d*), it can be found that selecting the spike with the lowest slope or the last spike as the removed spike makes little difference. Therefore, in a nutshell, the way of VSG to find the adjustment point strikes a good balance between efficiency and stability.

Furthermore, we conduct experiments to compare the learning efficiency of VSG and other aggregate-label algorithms. To this end, we test the time required for each algorithm to learn successfully when the desired output count ranges from 10 to 80, with an interval of 10. The firing rate of input is fixed at 4 Hz. Other experimental conditions are the same as above. Figure 7A shows the number of times each algorithm successfully delivered the desired number of spikes over 20 trials. It can be found that when the desired count is greater than or equal to 40, MPD-AL cannot successfully learn every time, because sometimes it cannot find t^*. While the other three algorithms can learn successfully, even when the number of desired spikes is very large. Figure 7B shows the time required to successfully fire the target number of spikes. The time required for different algorithms almost increases with the increase of the desired spike count, especially MST. The time required for TDP1 is relatively less, but also much more than the proposed algorithm. MPD-AL can learn very quickly only when the required number of spikes is small ($\leq$30). When the desired spike count is large, the average time it consumes increases significantly due to several failed learning. In short, the learning efficiency of the proposed algorithm is better than other aggregate-label algorithms.

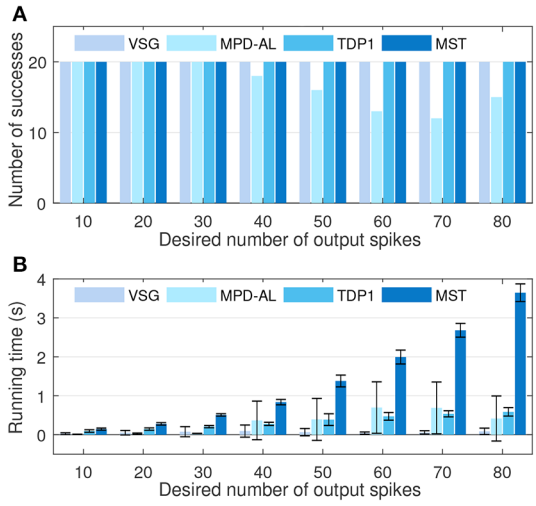


FIGURE 7
The comparison among VSG, MPD-AL, TDP1, and MST algorithms in terms of learning efficiency. **(A)** The number of successes of learning within 2,000 iterations over 20 trials. **(B)** The required learning time.

3.2. Detection of predictive clues

The task of detecting clues is to simulate the predictive behavior of animals in response to small changes in the environment as they survive in nature. For example, prey may recognize danger by the sound of breaking twigs among many natural noises and flee before predator attacks. Therefore, detecting predictive clues are to identify effective clues hidden

within distracting streams of unrelated sensory activity. In addition, there is also a difficulty in how to correlate clues with long-delayed feedback signals, which is called the "temporal-credit assignment problem" (Gütig, 2016). In this section, we will demonstrate the ability of VSG to solve this task.

Similar to Gütig (2016) and Zhang et al. (2019), 10 short (50 ms) spiking patterns with firing rate of 4 Hz are generated from 500 afferents to simulate clues, where effective clues and distracting clues are randomly set as required. These clues are then randomly embedded into the background spiking pattern (with duration T_b), as shown in Figure 8A, and the number of occurrences of each cue follows a Poisson distribution with mean P_m. The firing rate of the background pattern is 0∼4 Hz, with an average of 2 Hz, simulating the complex variability of the environment. The single neuron takes the long synthetic spike patterns containing clues and backgrounds as input, and detects effective clues through training, that is, it emits a specified number of spikes at the position where the effective clues appear, while remaining silent where other distracting clues and background patterns appear. During training, a total of 100 training samples are generated for neurons learning, T_b and P_m are set to 500 ms and 0.5, respectively. While in testing phase, in order to make all clues fully exposed, they are set to 2,200 ms and 0.8.

We set up different experiments to detect different kinds of clues. Assuming that d_i spikes are expected to be fired in response to the appearance of clue i, and the number of times that clue i occurs in a certain sample is c_i. Then for this sample, the desired spike count of the learning neuron is $N_d = \sum_1^{10} c_i d_i$, of which $d_i = 0$ for distracting clues. During the learning process, if the actual spike count is not equal to N_d, the synaptic weight is strengthened or weakened according to the VSG algorithm. We first trained the neuron to detect a single kind of clue, and the remaining nine kinds of clues are distractors. After training, the neuron not only fires the correct number of spikes, but also fires only where the effective clue appears, and remains silent elsewhere. Further, no matter whether d_i corresponding to this effective clue is 1 or 5, the neuron can learn successfully, as shown in Figures 8B,C. Then, we train the neuron to detect five different clues under the conditions that their corresponding spike counts are {1, 1, 1, 1, 1} and {1, 2, 3, 4, 5}, respectively. These involve more complicated temporal-credit assignments. But surprisingly, the neuron can automatically learn effective clues and assign them the corresponding number of spikes based only on the feedback signal of the total number of output spikes, as shown in Figures 8D,E. The experimental results show the capabilities of the VSG algorithm to decompose the delayed output signal and detect effective clues.

3.3. Classification of medical datasets

In this section, we test the proposed method on three medical datasets from UCI machine learning repository (Dua and Graff, 2017) and compare with other algorithms.

3.3.1. Data encoding and output decoding

The data encoding refers to encoding real values into spike times. As in Shrestha and Song (2016), Wang et al. (2017), Taherkhani et al. (2018), and Luo et al. (2022), Gaussian receptive field population encoding is used to encode each feature in the original data separately. To encode a certain feature, K identically shaped Gaussian functions that overlap each other and cover the interval $[a, b]$ are created, where a, b are the maximum and minimum values of this feature, respectively. Feeding a real value x into these Gaussian functions yields the output value y_i ($i = 1, 2, \cdots, K$), and then inversely mapping y_i to $[0, T]$ to get the spike time. T is the time window of encoding (in this section, $T = 100$ ms). A large y_i corresponds to an early firing time, a small y_i corresponds to a late firing time, and spikes with time later than $0.9T$ are canceled. Thus, an original sample containing N features is encoded as an input pattern containing KN neurons, each with at most one spike time. More details about the encoding process can be found in Luo et al. (2022).

Here, for classification tasks, decoding the output refers to determining the category identified by the network from its output. In this section, the number of neurons in the output layer is set equal to the number of categories, and each neuron corresponds to a category. During training, the neuron corresponding to the sample's label is expected to fire N_d (= 5) spikes, while the other output neurons are expected to not fire. In the inference phase, the sample belongs to the class corresponding to the output neuron that emits the most spikes. If no output neuron fires, the sample belongs to the class of neuron with the largest membrane voltage.

3.3.2. Medical datasets and classification results

The Wisconsin Breast Cancer dataset (WBC) contains 699 pieces of data described by 9 features, excluding 16 pieces of data with missing values, 683 samples are used in our experiments. The BUPA Liver Disorders dataset contains 345 samples with six features, and the Pima Diabetes dataset contains 746 samples with eight features. Each of the three datasets has two categories. As in SpikeProp (Shrestha and Song, 2016), SWAT (Wade et al., 2010), SRESN (Dora et al., 2016), and FE-Learn (Luo et al., 2022), we divided the training set and test set in a 1:1 ratio. For data encoding, we use the same number of neurons as in SpikeTemp (Wang et al., 2017) and FE-Learn to encode each feature, shown in Table 1 (# Encoders). A single-layer network and a two-layer network with 360 hidden neurons are used to conduct experiments separately. 20 independent trials are carried out in each experiment, and each trial run 200 epochs. Table 2 reports the mean and standard deviation of the classification accuracy in 20 trials.

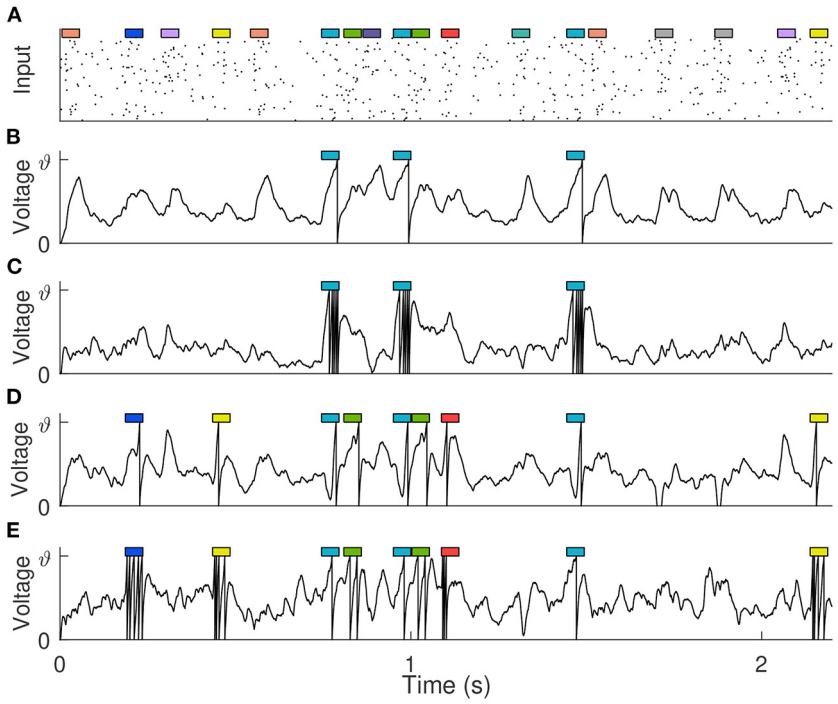


FIGURE 8
Detection of predictive clues. **(A)** The input spike pattern showing only 100 of the 500 synaptic afferents. 10 different cues (represented by colored rectangles, 50 ms each) are embedded in the background pattern. **(B,C)** The membrane voltage traces of the trained neuron when there is only one kind of effective clue, and it corresponds to 1 and 5 expected output spikes, respectively. **(D,E)** The membrane voltage traces of the trained neuron when there are five kinds of effective clues, and they correspond to {1, 1, 1, 1, 1} and {1, 2, 3, 4, 5} expected output spikes, respectively.

TABLE 1 Description of the dataset.

Dataset	WBC	Liver disorders	Pima diabetes
No. of instances	683	345	768
No. of categorizes	2	2	2
No. of features	9	6	8
No. of encoders	15	25	10
No. of training	341	172	384
No. of testing	342	173	384

As shown in Table 2, the performance of single-layer VSG is moderate, which is better than that of SWAT, Multilayer DL-ReSuMe (Taherkhani et al., 2018), and single-layer FE-Learn. The two-layer VSG performs better, further outperforming SRESN and two-layer FE-Learn compared to its single-layer counterpart. On the BUPA Liver Disorders dataset, it achieves the highest test accuracy of 65.1% together with SpikeProp, but a smaller standard deviation indicates that it is more stable than SpikeProp. Furthermore, it achieves sub-optimal accuracy on both the WBC and Pima Diabetes datasets. SpikeTemp achieved a state-of-the-art test accuracy of 98.3% on the WBC dataset, but it has a 2:1 ratio of training and test set, meaning it uses more training samples to train the model and fewer test samples to validate, which makes it more advantageous. The accuracy of SpikeProp on the Pima Diabetes dataset is much higher than other methods, but it requires a very large number of training epochs, and it is inferior to the proposed method on the WBC dataset. In conclusion, none of these algorithms can be absolutely dominant, and the performance of the proposed algorithm is relatively excellent.

3.4. Classification of speech datasets

In this section, we conduct experiment on speech recognition datasets. As mentioned earlier, VSG can detect useful clues in long spatiotemporal patterns, so it is also suitable for processing signals with rich temporal information like speech signals.

3.4.1. Data encoding and output decoding

The TIDIGITS corpus (Leonard and Doddington, 1993) is a common dataset widely used for speech recognition

TABLE 2 Comparison of classification performance on medical datasets.

Dataset	Breast cancer		Liver disorders		Pima diabetes	
	Architecture	Epochs	Architecture	Epochs	Architecture	Epochs
SpikeProp	55-15-2	1,000	37-15-2	3,000	55-20-2	3,000
SWAT	54-702-2	500	36-468-2	500	54-702-2	500
SRESN	54-(8-12)	306	36-(6-9)	715	54-(9-14)	254
SpikeTemp	135-306	/	150-226	/	80-431	/
Multi DL-ReSuMe	/	100	246-360-2	100	/	100
MPD-AL	135-2	200	150-2	200	80-2	200
FE-Learn	135-2	200	150-2	200	80-2	200
FE-Learn^2	135-360-2	200	150-360-2	200	80-360-2	200
VSG	135-2	200	150-2	200	Feb-80	200
VSG^2	135-360-2	200	150-360-2	200	80-360-2	200
	Train (%)	Test (%)	Train (%)	Test (%)	Train (%)	Test (%)
SpikeProp	97.3 ± 0.6	97.2 ± 0.6	71.5 ± 5.2	65.1 ± 4.7	78.6 ± 2.5	76.2 ± 1.8
SWAT	96.5 ± 0.5	95.8 ± 1.0	74.8 ± 2.1	60.9 ± 3.2	77.0 ± 2.1	72.1 ± 1.8
SRESN	97.7 ± 0.6	97.2 ± 0.7	60.4 ± 1.7	59.7 ± 1.7	70.5 ± 2.4	69.9 ± 2.1
SpikeTemp	99.1	98.3	93	58.3	77.5	67.6
Multi DL-ReSuMe	98.2	96.4	69.9	61.8	72.1	70.6
MPD-AL	99.9 ± 0.1	97.2 ± 0.6	92.7 ± 1.8	62.2 ± 3.6	71.4 ± 1.9	69.6 ± 1.3
FE-Learn	94.8 ± 0.9	94.3 ± 1.7	72.2 ± 5.0	61.2 ± 3.6	79.3 ± 1.2	71.2 ± 2.0
FE-Learn^2	100 ± 0.0	97.5 ± 0.5	96.6 ± 0.7	64.8 ± 2.3	90.6 ± 1.4	72.5 ± 1.5
VSG	99.2 ± 0.5	97.1 ± 0.7	74.7 ± 1.6	63.8 ± 2.0	77.4 ± 1.4	72.3 ± 1.5
VSG^2	99.3 ± 0.3	97.6 ± 0.6	96.3 ± 8.1	65.1 ± 1.9	91.8 ± 1.8	73.7 ± 1.7

(Wu et al., 2018a,b). It consists of 11 isolated spoken digit strings (from "0" to "9," and "oh") and speakers from 22 different dialectical regions. 2,464 and 2,486 speech utterances make up the standard training set and testing set. There is already a set of well-established and feasible encoding methods for this dataset: As shown in Figure 9, the raw speech waveform is first filtered by a Constant-Q-Transform (CQT) cochlear filter bank to extract spectral information, where the filter bank consists of 20 cochlear filters from 200 Hz to 8 kHz. Then the threshold coding mechanism (Gütig and Sompolinsky, 2009) is applied to convert the each frequency sub-band into a spike pattern of 31 neurons. Finally, the spike patterns obtained from all frequency bands are spliced into a complete spike pattern of 620 neurons. More details about the encoding process can be found in Pan et al. (2020).

There are also differences among samples of the same category in a dataset, especially for large and complex datasets, for which a fixed number of outputs is unreasonable. Therefore, we adopt the dynamic decoding (DD) strategy (Luo et al., 2019, 2022; Zhang et al., 2019) in this experiment. Instead of specifying a fixed number of output spikes, the dynamic decoding strategy decides whether to add a new spike based on the current sample. Here, we modify the strategy as follows to adapt to the proposed algorithm: If the actual spike count of an output neuron is $1 \leq N_o < N_d$, a new spike should be added, but unless the membrane voltage of the selected point reaches a given sub-threshold, i.e., $V(t^*) \geq \vartheta_s$, the new point will be discarded and no learning will be performed. This gives the output neuron a degree of freedom to respond to different inputs of the same class.

3.4.2. Network settings and results

The input layer of the network has 620 neurons and is responsible for feeding the encoded spike patterns into the network. The output layer contains $11M$ neurons, of which M neurons are a group, corresponding to a class in the dataset. For the group of neurons corresponding to the sample's label, $N_d = 5$, while the rest of the neurons are expected to not fire ($N_d = 0$). In the training phase, if the actual number of spikes emitted by a output neuron is not equal to N_d, the parameters are adjusted according to the DD strategy ($\vartheta_s = 0.8$) and the VSG algorithm, where the Adam optimizer (Kingma and Ba, 2015) is also used. During inference, the sample is classified into the class corresponding to the group of neurons with the largest number of output spikes. If all output neurons fail to fire, the sample is considered to belong to the class corresponding to the neuron with the largest membrane voltage. As in the previous section, we use a single-layer network ($620 - 11$) and a two-layer

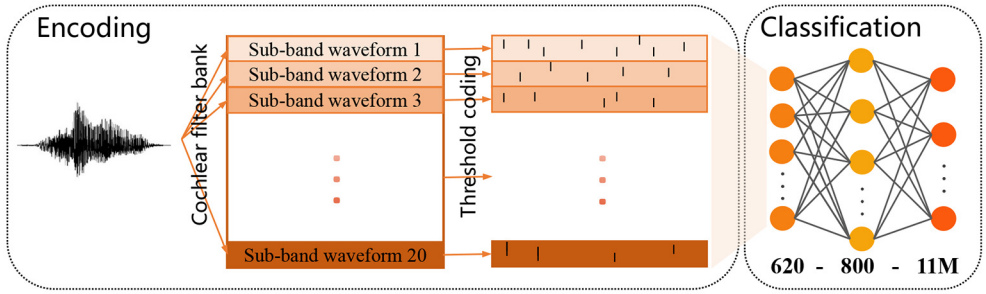


FIGURE 9
Schematic diagram of the encoding process of a speech sample **(left)** and the applied two-layer classification network **(right)**.

TABLE 3 Comparison of classification performance on TIDIGITS datasets*.

Model	Type	Layers	Accuracy
Tavanaei and Maida (2017b)	SNN+SVM	1	91.00%
Tavanaei and Maida (2017a)	Spiking CNN+HMM	3	96.00%
Neil and Liu (2016)	MFCC+RNN	4	96.10%
ETDP (Zhang et al., 2020)	SNN	2	95.80%
MPD-AL (Zhang et al., 2019)	SNN+DD	1	97.52%
FE-Learn (Luo et al., 2019)	SNN+DD	1	96.42%
FE-Learn^2 (Luo et al., 2022)	SNN+DD	2	98.10%
VSG ($M = 1$)	SNN+DD	1	96.34%
VSG ($M = 1$)	SNN+DD	2	98.23% (98.03%)
VSG ($M = 10$)	SNN+DD	2	98.47% (98.32%)

*DD, dynamic decoding. Values in parentheses are the average of 10 experiments.

network with 800 hidden neurons ($620-800-11M, M = 1, 10$) to conduct experiments separately.

Table 3 shows the highest test accuracies achieved by the proposed method and other baseline methods. A single-layer network trained with VSG can achieve a maximum accuracy of 96.34%. As a single-layer network with only 11 output neurons, it performs well, as the best performing MPD-AL (among single-layer network) has 110 output neurons. In addition, when there is only one set of output neurons ($M = 1$), the two-layer network trained by VSG outperforms the two-layer FE-Learn by a slight advantage. When the number of output neurons is increased ($M = 10$), the performance can be further improved, reaching the highest accuracy of 98.47% as against other baseline methods. However, since the proposed method has only a slight advantage over FE-Learn^2, it may not have statistical confidence. So we re-executed the proposed algorithm 10 times (500 epochs each) on the two-layer network and reported the average test accuracies (in parentheses). When $M = 10$, the average accuracy is 98.32%, which is also higher than the highest accuracy of FE-Learn^2. In addition, although the average accuracy when $M = 1$ is only 98.03%, the highest accuracy (98.23%) is higher than that of FE-Learn^2. We believe that this can demonstrate the superiority of the proposed algorithm.

4. Discussion and conclusion

Temporal-credit assignment problem is a non-trivial problem in machine learning, and the aggregate-label learning algorithm MST is an innovative SNN algorithm to solve this problem. Then TDP1 improves the computational efficiency of MST by modifying the formula for calculating the weight derivative. Subsequently, MDP-AL bypasses the procedure of iteratively finding critical thresholds in the MST and TDP1 by adjusting the weights directly from the membrane voltage, thus greatly reducing the computation time. But there is a drawback in MPD-AL, that is, it may not be able to find the critical time it needs, leading to the failure of learning.

In this paper, we propose to find the potential points for emitting a new spike and the old spike that need to be removed from the time derivative of membrane voltage, avoiding the dilemma of failing to find the adjustment points. Furthermore, on the one hand, the intermediate variables required to calculate this time derivative are also necessary in the calculation of membrane voltage and subsequent weight derivatives, so little additional computation is added. On the other hand, we choose the point with the fastest growth of the time derivative to add the spike, and select the point with the slowest growth of the derivative (among the existing pulses) to remove it, which is experimentally proven to achieve a good balance between efficiency and stability.

A single neuron trained with this algorithm can be used to tackle the challenging temporal-credit assignment problems. Specifically, it can detect valid clues embedded in distracting clues and background spiking activity, deconstruct aggregated delayed feedback signal and then assign them to valid clues. Further, unlike MST, TDP1, and MPD-AL, which is limited to the training of a single neuron or a single-layer network, the proposed algorithm is rooted in multi-layer SNNs for derivation, which further extends its performance. Its

application on UCI and speech classification datasets also proves its superiority.

Although the proposed algorithm is simple and efficient, it has drawbacks. Like MPD-AL, when learning predictive clues, if the clues in the training samples are too densely distributed, it will be difficult to learn, which may be an unavoidable problem caused by not calculating the precise critical threshold. In addition, as a multi-layer spike-driven SNN learning algorithm, the proposed learning rule suffers from common problems such as gradient exploding and dead neurons. These all require us to further optimize.

Data availability statement

The original contributions presented in the study are included in the article/supplementary material, further inquiries can be directed to the corresponding author.

Author contributions

Both authors listed have made a substantial, direct, and intellectual contribution to the work and approved it for publication.

Funding

This work was supported in part by the National Natural Science Foundation of China under Grant 82174236 and in part by the Project of Science & Technology Department of Sichuan Province under Grant SYZ202102.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

Barbier, T., Teulière, C., and Triesch, J. (2021). "Spike timing-based unsupervised learning of orientation, disparity, and motion representations in a spiking neural network," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops* (Nashville, TN: IEEE), 1377–1386. doi: 10.1109/CVPRW53098.2021.00152

Bengio, Y., Lee, D. -H., Bornschein, J., Mesnard, T., and Lin, Z. (2015). Towards biologically plausible deep learning. *arXiv preprint arXiv:1502.04156*. doi: 10.48550/arXiv.1502.04156

Bi, G. Q., and Poo, M. M. (1998). Synaptic modifications in cultured hippocampal neurons: dependence on spike timing, synaptic strength, and postsynaptic cell type. *J. Neurosci.* 18, 10464–10472. doi: 10.1523/JNEUROSCI.18-24-10464.1998

Bohte, S. M., Kok, J. N., and La Poutre, H. (2002). Error-backpropagation in temporally encoded networks of spiking neurons. *Neurocomputing* 48, 17–37. doi: 10.1016/S0925-2312(01)00658-0

Bohte, S. M., Poutre, J. A. L., and Kok, J. N. (2000). *Error-Backpropagation in Temporally Encoded Networks of Spiking Neurons*. CWI (Centre for Mathematics and Computer Science).

Burr, G. W., Shelby, R. M., Sebastian, A., Kim, S., Kim, S., Sidler, S., et al. (2016). Neuromorphic computing using non-volatile memory. *Adv. Phys. X* 2, 89–124. doi: 10.1080/23746149.2016.1259585

Diehl, P. U., and Matthew, C. (2015). Unsupervised learning of digit recognition using spike-timing-dependent plasticity. *Front. Comput. Neurosci.* 9:99. doi: 10.3389/fncom.2015.00099

Dora, S., Subramanian, K., Suresh, S., and Sundararajan, N. (2016). Development of a self regulating evolving spiking neural network for classification problem. *Neurocomputing* 171, 1216–1229. doi: 10.1016/j.neucom.2015.07.086

Dua, D., and Graff, C. (2017). *UCI Machine Learning Repository*. Available online at: http://archive.ics.uci.edu/m

Florian, R. V. (2012). The chronotron: a neuron that learns to fire temporally precise spike patterns. *PLoS ONE* 7:e40233. doi: 10.1371/journal.pone.0040233

Fu, Q., and Dong, H. (2021). An ensemble unsupervised spiking neural network for objective recognition. *Neurocomputing* 419, 47–58. doi: 10.1016/j.neucom.2020.07.109

Ghosh-Dastidar, S., and Adeli, H. (2009). A new supervised learning algorithm for multiple spiking neural networks with application in epilepsy and seizure detection. *Neural Netw.* 22, 1419–1431. doi: 10.1016/j.neunet.2009.04.003

Gütig, R. (2016). Spiking neurons can discover predictive features by aggregate-label learning. *Science* 351:aab4113. doi: 10.1126/science.aab4113

Gütig, R., and Sompolinsky, H. (2009). Time-warp-invariant neuronal processing. *PLoS Biol.* 7:e1000141. doi: 10.1371/journal.pbio.1000141

Hu, S. G., Qiao, G. C., Chen, T. P., Yu, Q., Liu, Y., and Rong, L. M. (2021). Quantized STDP-based online-learning spiking neural network. *Neural Comput. Appl.* 33, 12317–12332. doi: 10.1007/s00521-021-05832-y

Kingma, D. P., and Ba, J. (2015). "Adam: a method for stochastic optimization," in *International Conference on Learning Representations, abs/1412.6980* (San Diego, CA).

Knudsen, E. I. (1994). Supervised learning in the brain. *J. Neurosci.* 14, 3985–3997. doi: 10.1523/JNEUROSCI.14-07-03985.1994

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature* 521:436. doi: 10.1038/nature14539

Leonard, R. G., and Doddington, G. (1993). *TIDIGITS LDC93S10*. Philadelphia, PA: Linguistic Data Consortium. Available online at: https://catalog.ldc.upenn.edu/LDC93S10

Luo, X., Qu, H., Wang, Y., Yi, Z., Zhang, J., and Zhang, M. (2022). "Supervised learning in multilayer spiking neural networks with spike temporal error backpropagation," in *Early Access*, 1–13. doi: 10.1109/TNNLS.2022.3164930

Luo, X., Qu, H., Zhang, Y., and Chen, Y. (2019). First error-based supervised learning algorithm for spiking neural networks. *Front. Neurosci.* 13:559. doi: 10.3389/fnins.2019.00559

Maass, W., and Bishop, C. M. (1999). *Pulsed Neural Networks*. Cambridge: The MIT Press. doi: 10.7551/mitpress/5704.001.0001

Malenka, R. C., and Bear, M. F. (2004). LTP and LTD: an embarrassment of riches. *Neuron* 44, 5–21. doi: 10.1016/j.neuron.2004.09.012

Masquelier, T., Thorpe, S. J. (2007). Unsupervised learning of visual features through spike timing dependent plasticity. *PLoS Comput. Biol*. 3:e31. doi: 10.1371/journal.pcbi.0030031

Memmesheimer, R. -M., Rubin, R., Olveczky, B. P., and Sompolinsky, H. (2014). Learning precisely timed spikes. *Neuron* 82, 925–938. doi: 10.1016/j.neuron.2014.03.026

Meng, M., Yang, X., Bi, L., Kim, J., Xiao, S., and Yu, Z. (2021). High-parallelism inception-like spiking neural networks for unsupervised feature learning. *Neurocomputing* 441, 92–104. doi: 10.1016/j.neucom.2021.02.027

Meng, Q., Xiao, M., Yan, S., Wang, Y., Lin, Z., and Luo, Z. -Q. (2022). "Training high-performance low-latency spiking neural networks by differentiation on spike representation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (New Orleans, LA: IEEE), 12444–12453. doi: 10.1109/CVPR52688.2022.01212

Mohemmed, A., Schliebs, S., Matsuda, S., and Kasabov, N. (2012). Span: spike pattern association neuron for learning spatio-temporal spike patterns. *Int. J. Neural Syst*. 22:1250012. doi: 10.1142/S0129065712500128

Mozafari, M., Kheradpisheh, S. R., Masquelier, T., Nowzari-Dalini, A., and Ganjtabesh, M. (2018). First-spike-based visual categorization using reward-modulated stdp. *IEEE Trans. Neural Netw. Learn. Syst*. 29, 6178–6190. doi: 10.1109/TNNLS.2018.2826721

Neil, D., and Liu, S. -C. (2016). "Effective sensor fusion with event-based sensors and deep network architectures," in *IEEE International Symposium on Circuits & Systems* (Montreal, QC: IEEE), 2282–2285. doi: 10.1109/ISCAS.2016.7539039

Pan, Z., Chua, Y., Wu, J., Zhang, M., Li, H., and Ambikairajah, E. (2020). An efficient and perceptually motivated auditory neural encoding and decoding algorithm for spiking neural networks. *Front. Neurosci*. 13:1420. doi: 10.3389/fnins.2019.01420

Ponulak, F., and Kasi?ski, A. (2010). Supervised learning in spiking neural networks with resume: sequence learning, classification, and spike shifting. *Neural Comput*. 22, 467–510. doi: 10.1162/neco.2009.11-08-901

Rueckauer, B., Lungu, I. -A., Hu, Y., Pfeiffer, M., and Liu, S. -C. (2017). Conversion of continuous-valued deep networks to efficient event-driven networks for image classification. *Front. Neurosci*. 11:682. doi: 10.3389/fnins.2017.00682

She, X., and Mukhopadhyay, S. (2021). Speed: Spiking neural network with event-driven unsupervised learning and near-real-time inference for event-based vision. *IEEE Sensor. J*. 21, 20578–20588. doi: 10.1109/JSEN.2021.3098013

Shrestha, S. B., and Orchard, G. (2018). Slayer: Spike layer error reassignment in time. *Adv. Neural Inform. Process. Syst*. 31, 1419–1428. doi: 10.5555/3326943.3327073

Shrestha, S. B., and Song, Q. (2016). "Adaptive delay learning in spikeprop based on delay convergence analysis," in *International Joint Conference on Neural Networks* (Vancouver, BC), 277–284. doi: 10.1109/IJCNN.2016.7727209

Taherkhani, A., Belatreche, A., Li, Y., and Maguire, L. P. (2018). A supervised learning algorithm for learning precise timing of multiple spikes in multilayer spiking neural networks. *IEEE Trans. Neural Netw. Learn. Syst*. 1–14. doi: 10.1109/TNNLS.2018.2797801

Tavanaei, A., and Maida, A. (2017a). "Bio-inspired multi-layer spiking neural network extracts discriminative features from speech signals," in *International Conference on Neural Information Processing* (Long Beach, CA: Springer), 899–908. doi: 10.1007/978-3-319-70136-3_95

Tavanaei, A., and Maida, A. S. (2017b). A spiking network that learns to extract spike signatures from speech signals. *Neurocomputing* 240, 191–199. doi: 10.1016/j.neucom.2017.01.088

Thach, W. T. (1996). On the specific role of the cerebellum in motor learning and cognition: clues from pet activation and lesion studies in man. *Behav. Brain Sci*. 19, 411–431. doi: 10.1017/S0140525X00081504

Wade, J. J., Mcdaid, L. J., Santos, J. A., and Sayers, H. M. (2010). Swat: a spiking neural network training algorithm for classification problems. *IEEE Trans. Neural Netw*. 21, 1817–1830. doi: 10.1109/TNN.2010.2074212

Wang, J., Belatreche, A., Maguire, L. P., and Mcginnity, T. M. (2017). Spiketemp: an enhanced rank-order-based learning approach for spiking neural networks with adaptive structure. *IEEE Trans. Neural Netw. Learn. Syst*. 1–14. doi: 10.1109/TNNLS.2015.2501322

Wu, J., Chua, Y., and Li, H. (2018a). "A biologically plausible speech recognition framework based on spiking neural networks," in *2018 International Joint Conference on Neural Networks* (Rio de Janeiro), 1–8. doi: 10.1109/IJCNN.2018.8489535

Wu, J., Chua, Y., Zhang, M., and Li, H. (2018b). A spiking neural network framework for robust sound classification. *Front. Neurosci*. 12:836. doi: 10.3389/fnins.2018.00836

Wu, Y., Deng, L., Li, G., Zhu, J., Xie, Y., and Shi, L. (2019). "Direct training for spiking neural networks: faster, larger, better," in *Proceedings of the AAAI Conference on Artificial Intelligence* (Honolulu, HI: AAAI), 33. doi: 10.1609/aaai.v33i01.33011311

Wu, Y., Lei, D., Li, G., Zhu, J., and Shi, L. (2018c). Spatio-temporal backpropagation for training high-performance spiking neural networks. *Front. Neurosci*. 12:331. doi: 10.3389/fnins.2018.00331

Xu, Y., Zeng, X., Han, L., and Yang, J. (2013a). A supervised multi-spike learning algorithm based on gradient descent for spiking neural networks. *Neural Netw*. 43, 99–113. doi: 10.1016/j.neunet.2013.02.003

Xu, Y., Zeng, X., and Zhong, S. (2013b). A new supervised learning algorithm for spiking neurons. *Neural Comput*. 25, 1472–1511. doi: 10.1162/NECO_a_00450

Yu, Q., Li, H., and Tan, K. C. (2018). Spike timing or rate? neurons learn to make decisions for both through threshold-driven plasticity. *IEEE Trans. Cybern*. 49, 2178–2189. doi: 10.1109/TCYB.2018.2821692

Zenke, F., and Ganguli, S. (2018). Superspike: Supervised learning in multilayer spiking neural networks. *Neural Comput*. 30, 1514–1541. doi: 10.1162/neco_a_01086

Zhang, M., Luo, X., Wu, J., Chen, Y., Belatreche, A., Pan, Z., et al. (2020). An efficient threshold-driven aggregate-label learning algorithm for multimodal information processing. *IEEE J. Select. Top. Signal Process*. 14, 592–602. doi: 10.1109/JSTSP.2020.2983547

Zhang, M., Qu, H., Belatreche, A., Chen, Y., and Yi, Z. (2018a). A highly effective and robust membrane potential-driven supervised learning method for spiking neurons. *IEEE Trans. Neural Netw. Learn. Syst*. 30, 123–137. doi: 10.1109/TNNLS.2018.2833077

Zhang, M., Qu, H., Belatreche, A., and Xie, X. (2018b). EMPD: an efficient membrane potential driven supervised learning algorithm for spiking neurons. *IEEE Trans. Cogn. Dev. Syst*. 10, 151–162. doi: 10.1109/TCDS.2017.2651943

Zhang, M., Wu, J., Chua, Y., Luo, X., and Li, H. (2019). "MPD-Al: an efficient membrane potential driven aggregate-label learning algorithm for spiking neurons," in *The AAAI Conference on Artificial Intelligence, Vol. 33* (Honolulu, HI: AAAI), 1327–1334. doi: 10.1609/aaai.v33i01.33011327

Zhou, Z., Xiang, Y., Xu, H., Wang, Y., and Shi, D. (2022). Unsupervised learning for non-intrusive load monitoring in smart grid based on spiking deep neural network. *J. Modern Power Syst. Clean Energy* 10, 606–616. doi: 10.35833/MPCE.2020.000569

frontiers | Frontiers in Computational Neuroscience

TYPE Original Research
PUBLISHED 16 November 2022
DOI 10.3389/fncom.2022.1062678

OPEN ACCESS

EDITED BY
Yuqi Han,
Tsinghua University, China

REVIEWED BY
Yuxuan Zhao,
Institute of Automation (CAS), China
Shuangming Yang,
Tianjin University, China

*CORRESPONDENCE
Sungroh Yoon
sryoon@snu.ac.kr

RECEIVED 06 October 2022
ACCEPTED 28 October 2022
PUBLISHED 16 November 2022

CITATION
Lee J, Jo J, Lee B, Lee J-H and Yoon S (2022) Brain-inspired Predictive Coding Improves the Performance of Machine Challenging Tasks. *Front. Comput. Neurosci.* 16:1062678. doi: 10.3389/fncom.2022.1062678

Brain-inspired Predictive Coding Improves the Performance of Machine Challenging Tasks

Jangho Lee[1], Jeonghee Jo[2], Byounghwa Lee[3], Jung-Hoon Lee[3] and Sungroh Yoon[1,4]*

[1]Department of Electrical and Computer Engineering, Seoul National University, Seoul, South Korea, [2]Institute of New Media and Communications, Seoul National University, Seoul, South Korea, [3]CybreBrain Research Section, Electronics and Telecommunications Research Institute (ETRI), Daejeon, South Korea, [4]Interdisciplinary Program in Artificial Intelligence, Seoul National University, Seoul, South Korea

Backpropagation has been regarded as the most favorable algorithm for training artificial neural networks. However, it has been criticized for its biological implausibility because its learning mechanism contradicts the human brain. Although backpropagation has achieved super-human performance in various machine learning applications, it often shows limited performance in specific tasks. We collectively referred to such tasks as *machine-challenging tasks* (MCTs) and aimed to investigate methods to enhance machine learning for MCTs. Specifically, we start with a natural question: *Can a learning mechanism that mimics the human brain lead to the improvement of MCT performances?* We hypothesized that a learning mechanism replicating the human brain is effective for tasks where machine intelligence is difficult. Multiple experiments corresponding to specific types of MCTs where machine intelligence has room to improve performance were performed using predictive coding, a more biologically plausible learning algorithm than backpropagation. This study regarded incremental learning, long-tailed, and few-shot recognition as representative MCTs. With extensive experiments, we examined the effectiveness of predictive coding that robustly outperformed backpropagation-trained networks for the MCTs. We demonstrated that predictive coding-based incremental learning alleviates the effect of catastrophic forgetting. Next, predictive coding-based learning mitigates the classification bias in long-tailed recognition. Finally, we verified that the network trained with predictive coding could correctly predict corresponding targets with few samples. We analyzed the experimental result by drawing analogies between the properties of predictive coding networks and those of the human brain and discussing the potential of predictive coding networks in general machine learning.

KEYWORDS

brain-inspired learning, biologically plausible learning, deep learning, backpropagation, predictive coding

1. Introduction

The human brain has an intricate and heterogeneous structure that consists of a high recurrent and nonlinear neural network (Felleman and Van Essen, 1991; Friston, 2008; Bertolero et al., 2015). It is commonly understood that the learning system of the human brain operates on the synaptic plasticity mechanism (Hebb, 2005), wherein the modulation in synaptic weights varies according to the intrinsic or extrinsic stimuli (Power and Schlaggar, 2017). Specifically, neural plasticity regulates the process of synaptic transmission as a fundamental property of neurons (Citri and Malenka, 2008; Mateos-Aparicio and Rodríguez-Moreno, 2019). Based on this property, the neuronal responses to sensory stimuli enable the robust recognition (Ohayon et al., 2012; Denève et al., 2017; Geirhos et al., 2017; Wardle et al., 2020) and noise-resistance learning (Suzuki et al., 2015; Perez-Nieves et al., 2021) in human perception.

Based on the human brain architecture, artificial neural networks (ANNs) were suggested to simulate the pattern of the human decision-making process for recognition tasks. Rumelhart et al. (1986) introduced the backpropagation algorithm that adjusts the network parameters to achieve reliable performance. Backpropagation iteratively updates the network parameters relying on the error signal generated at the end of the network between the produced output and the desired output. In the last decade, with the benefits of backpropagation (Rumelhart et al., 1986), ANNs have exceeded human-level performance on classification, segmentation, and detection (He et al., 2016; Dosovitskiy et al., 2020). However, learning ANNs with backpropagation have been criticized for their biological implausibility, wherein its behavior conflicts with the activity of real neurons in the human brain (Akrout et al., 2019; Illing et al., 2021). First, the human brain operates according to *neural plasticity*, which indicates the capability for modifying neural circuit connectivity or degree of interaction (Neves et al., 2008). Second, global error-guided learning requires the forward weight matrices to propagate the error signal flow to the lower layer, that is *weight transport problem* (Grossberg, 1987). Multiple learning algorithms have been proposed to alleviate the previously mentioned issues based on strong constraints of backpropagation and reinforce its biological plausibility (Liao et al., 2016; Lillicrap et al., 2016; Whittington and Bogacz, 2017; Woo et al., 2021; Dellaferrera and Kreiman, 2022). This study explored the predictive coding network (Whittington and Bogacz, 2017) among the various biologically plausible learning and its characteristics.

A predictive coding network (Whittington and Bogacz, 2017) was introduced to resolve the biological limitations of backpropagation depending on the hierarchically organized visual cortex of the human brain (Rao and Ballard, 1999; Friston, 2008). With respect to biological plausibility, a predictive coding network concentrates on local and Hebbian plasticity by minimizing the prediction errors between expected and actual inputs (Rao and Ballard, 1999; Millidge et al., 2020). The learning mechanism of the predictive coding network is different from that of backpropagation, which updates the network parameters using only one error derived from the last layer (Rumelhart et al., 1986). Predictive coding is regarded as a local learning algorithm because its learning is performed with local error nodes and global error nodes. A learning network with predictive coding approximates the learning dynamics of backpropagation (Whittington and Bogacz, 2017) and can also be expanded to arbitrary computational graphs (Millidge et al., 2020). Multiple works (Han et al., 2018; Wen et al., 2018; Choksi et al., 2021) inspired by the property of prediction itself have been proposed, and some studies (Choksi et al., 2021; Salvatori et al., 2021) demonstrated that the potential of the predictive manner related to human perception.

However, despite the remarkable accomplishment of ANN architectures and their learning algorithms, there remains a performance gap between machine and human intelligence in some applications. We collectively refer to these tasks as *machine-challenging tasks* (MCTs); MCTs are difficult for machine intelligence while easy for human intelligence. This study considers the representative MCTs as incremental learning, long-tailed recognition, and few-shot learning (Hassabis et al., 2017). A more detailed definition and explanation of MCTs will be presented in Section 2.2. Humans progressively and ceaselessly acquire new knowledge and preserve it by virtue of the hippocampus (Preston and Eichenbaum, 2013). The primary function of the hippocampus is that it enables long-term memory of the spatial and sequential order from the human experience (Bird and Burgess, 2008; Davachi and DuBrow, 2015). This property makes the human intelligence exhibits robust and performs better than machine intelligence (Goodfellow et al., 2014; Zhou and Firestone, 2019; Liu et al., 2021). Meanwhile, ANNs trained with backpropagation tend to forget what it learned when it learns new information, that is *catastrophic forgetting* (McCloskey and Cohen, 1989; French, 1999; Goodfellow et al., 2013). As another example, machine intelligence shows unsatisfactory performance under limited or imperfect training data recognition (De Man et al., 2019; Liu et al., 2019a). When training ANNs for classification tasks in a long-tail scenario, the classifier can be easily biased toward the majority classes that contain the most data and show poor performance in minority classes (Johnson and Khoshgoftaar, 2019). These phenomena result from the fundamental differences in visual processing between the brain and ANNs (Xu and Vaziri-Pashkam, 2021). Inspired by Hassabis et al. (2017), we hypothesized that the closer the learning algorithm is to the human brain, the more effective it is for the MCTs.

Similar to our assumption on the MCTs, the learning algorithms inspired by the brain are consistently studied to reduce the performance gap between machine intelligence and

human intelligence based on human's various attributes. In terms of human learning mechanisms, a spiking neural network (SNN) is considered a promising solution to replicate the neural processing process of the brain. Yang et al. (2022c) proposed an SNN-based continual meta-learning framework and demonstrated that the suggested model improves the accuracy and robustness of the continual meta-learning tasks. Yang et al. (2022b) also established the ensemble framework with multiple spike-driven few-shot online learning and confirmed the effectiveness of the brain-inspired paradigm. On the other hand, recent studies reported that the neural network trained biologically plausible manner embodies specific memory functions in the human memory system. Salvatori et al. (2021) discovered that the network trained with predictive coding can naturally implement the associative memory function, such as reconstructing incomplete regions. Yang et al. (2022a) verified that the multicompartmental spiking neural network incorporates the working memory satisfying four essential components of brain-inspired mechanisms. Therefore, based on previous studies, we speculated that predictive coding has other latent properties. This study aimed to discover hidden properties and extend the scope of predictive coding to MCTs. Contrary to the conventional solutions for the MCTs, our study focused on the predictive coding algorithm itself employed for the optimization of the network parameters. In incremental learning, it is confirmed that predictive coding better reveals the plasticity-stability property and enables faster adaptation to new tasks than backpropagation. In long-tailed recognition, it reduces the classification bias problem of minority classes.

This paper is organized as follows: In Section 2, the predictive coding network is briefly reviewed. In Section 3, the experiments on incremental learning based on a predictive coding network are presented. In Section 4, the experiments on limited data recognition based on a predictive coding network, such as long-tailed recognition and few-shot learning, are described. In Section 5, we discuss why predictive coding network improves the performance of MCTs. In Section 6, related work to help understand our paper is presented. In Section 7, we conclude the paper with limitations and a summary.

Our contributions can be summarized as follows:

- The study characterized the MCTs, which are easy for human intelligence and difficult for machine intelligence, in machine learning fields and proposed a hypothesis that the brain-inspired learning algorithm improves the performance of MCTs.
- Predictive coding, a biologically plausible learning algorithm, was adopted for MCTs, such as incremental learning and limited data recognition. In addition, extensive experiments were performed by reimplementing the learning with backpropagation with predictive coding.
- The effect of learning algorithms close to brain learning on MCTs in terms of neuroscience was presented. Mainly, the experimental results were analyzed with respect to the plasticity-stability dilemma and interplay between the hippocampus and prefrontal cortex.

2. Related Work

2.1. Biologically Plausible Learning

The backpropagation algorithm (Rumelhart et al., 1986), which simulates the properties of the human brain, has achieved excellent progress in various machine learning tasks. The algorithm calculates the global error by comparing the predicted outputs and the actual targets at the network's end to achieve an objective. Then, it propagates the error signal to the front of the network to update parameters. Although backpropagation is the most popular learning algorithm for ANNs, it is often regarded as a biologically implausible algorithm from a neuroscience perspective. The main reason is that backpropagation does not operate following the local synaptic plasticity (Takesian and Hensch, 2013; Mateos-Aparicio and Rodríguez-Moreno, 2019) as a fundamental property of the nervous system. Synaptic plasticity refers to the ability to reorganize structures or connections by intrinsic or extrinsic stimuli. Another reason is that the backpropagation requires a copy of the weight matrices to transfer backward error signal (Grossberg, 1987). However, retaining synaptic weights on each neuron is impractical in the human brain. So, Lillicrap et al. (2016) replaced the backward weight matrices with fixed random weights to avoid those problems. Liao et al. (2016) reported that the signs of backward weight matrices were important, and when the signs between the forward and backward matrices were concordant, the same or better performance could be achieved. Furthermore, various learning algorithms have been proposed to reinforce biological plausibility while maintaining the classification performance (Lee et al., 2015; Whittington and Bogacz, 2017; Ahmad et al., 2020; Lindsey and Litwin-Kumar, 2020; Pogodin and Latham, 2020). Among them, predictive coding, based on the predictive process of the brain, was suggested to achieve better biologically plausible properties than the backpropagation algorithm and achieved comparable performance to the backpropagation on arbitrary computational graphs (Whittington and Bogacz, 2017).

2.2. Machine Challenging Tasks (MCTs)

ANNs have achieved comparable or superior performances to humans by backpropagation in

visual recognition (Russakovsky et al., 2015; Geirhos et al., 2017). However, ANNs have unsatisfactory performance in certain tasks regarded as simple and easy for human intelligence (Goodfellow et al., 2013; Snell et al., 2017; Cao et al., 2019). As detailed in Section 1, these types of tasks as MCTs (e.g., incremental learning, long-tailed recognition, and few-shot recognition).

Humans ceaselessly take new information from multiple sensory organs and reorganize it in the brain (Felleman and Van Essen, 1991; Denève et al., 2017). These processes proceed in a *lifelong manner* because knowledge construction is affected by previous experiences. In addition, humans can refine or transfer knowledge acquired from different types of previous tasks built in an incremental manner (Preston and Eichenbaum, 2013; Davachi and DuBrow, 2015). In contrast to human intelligence, ANNs have *catastrophic forgetting* in which the collected information is lost after training of subsequent tasks (Goodfellow et al., 2013). Moreover, the human visual system shows robust performances even in limited data recognition, such as long-tailed and few-shot visual recognition. Real-world data commonly follow long-tailed distribution wherein the majority classes occupy the significant part of the dataset and have an open-ended distribution (Liu et al., 2019b). The primary purpose of long-tailed recognition is to correctly classify the minority class samples to the corresponding targets, reducing the classification bias effect (Cao et al., 2019). Further, the classification of tail class samples can be regarded as a few-shot recognition problem as the degree of imbalance increases (Samuel et al., 2021).

The discrepancy in learning performances between humans and ANNs is closely related to the characteristics of the human brain. First, the human brain operates under two properties: plasticity and stability (Takesian and Hensch, 2013). Plasticity refers to the brain's change in connectivity and circuitry that enables humans to acquire knowledge, keep memories, and adapt to the external environment (Power and Schlaggar, 2017). Meanwhile, stability refers to the ability of long-term memory where stable memory is relevant to stable neuron connectivity (Susman et al., 2019). A balance between plasticity and stability is achieved with excitatory and inhibitory circuit activity in the visual cortex (Takesian and Hensch, 2013). Second, the brain engages the hippocampus and neocortex, as explained by the complementary learning system theory that characterizes learning in the brain (Preston and Eichenbaum, 2013). The hippocampus focuses on acquiring new knowledge, and knowledge is transferred and generalized to the neocortex via the memory consolidation process. Such mechanisms do not exist in backpropagation. However, they can be indirectly performed in learning predictive coding through the free-energy minimization process of predictive coding. As such, we assume that humans can achieve superior performance in MCTs.

3. Predictive Coding Networks

Most architectures in ANNs follow an L-layer structure wherein each layer consists of a set of neurons (Rumelhart et al., 1986). The training with the backpropagation algorithm can be explained to minimize a global error generated at the last layer of a network. In the backpropagation algorithm, an activation value of each layer is defined as follows:

$$\hat{v}_0 = x \tag{1}$$

$$\hat{v}_i = f(\hat{v}_{i-1}; \theta_i) \tag{2}$$

where i is the indices of layers, and θ_i is the parameters of i-th layer. The goal of backpropagation algorithm is to minimize a loss function $\mathcal{L}(\hat{y}, y)$ between the ground-truth target y and the prediction value $\hat{y}$. The final layer output is derived from the forward pass as follows:

$$\hat{y} = f(x; \theta) = \hat{v}_L \tag{3}$$

In the backward pass, the optimization of parameters is performed by the derivative of the loss function. The gradient of each layer is computed in reverse order as follows:

$$\delta_i = \delta_{i+1} \frac{\partial f_{i+1}(\hat{v}_i; \theta_{i+1})}{\partial \hat{v}_l} \tag{4}$$

and

$$d\theta_i = -\frac{\partial \mathcal{L}(\hat{y}, y)}{\partial \theta_i} \tag{5}$$

where δ_i and $d\theta_i$ are the error signal and the gradient from i-th layer, respectively.

Meanwhile, in the predictive coding algorithm, an error node is defined in every layer, and the goal of learning is to minimize the collective energy function (Friston, 2003; Bogacz, 2017; Buckley et al., 2017), which is the sum of prediction errors as illustrated in Figure 1. A predictive coding network assumes the network as a directed acyclic computational graph $\mathcal{G} = \{\mathcal{E}, \mathcal{V}\}$ to deliver an error from the last layer to the first layer. $\mathcal{E}$ and $\mathcal{V}$ are defined as a set of error nodes $e_i \in \mathcal{E}$ and a set of activation nodes $v_i \in \mathcal{V}$ at every layer.

By analogy to the cortical hierarchy in the human brain, predictive coding can be formulated as a variational inference algorithm (Friston, 2005; Buckley et al., 2017). Millidge et al. (2020) extended predictive coding to an arbitrary computational graph $\mathcal{G}$ considering its hierarchical and generative structure. Given a computational graph $\mathcal{G}$, the feedforward prediction is defined as $p(v_i) = \Pi_i^N p(v_i|\mathcal{P}_i))$ and variational posterior is derived as $Q(\{v_i\}) = \Pi_i^N Q(v_i)$, where $\mathcal{P}(x)$ indicates the set of parent nodes and $\mathcal{C}(x)$ denotes the set of child nodes for the given node x. Each activation node has the prediction $\hat{v}_i = f(\mathcal{P}(v_i); \theta_i) = f(\hat{v}_{i-1}; \theta_i)$ for i-th layer. Based

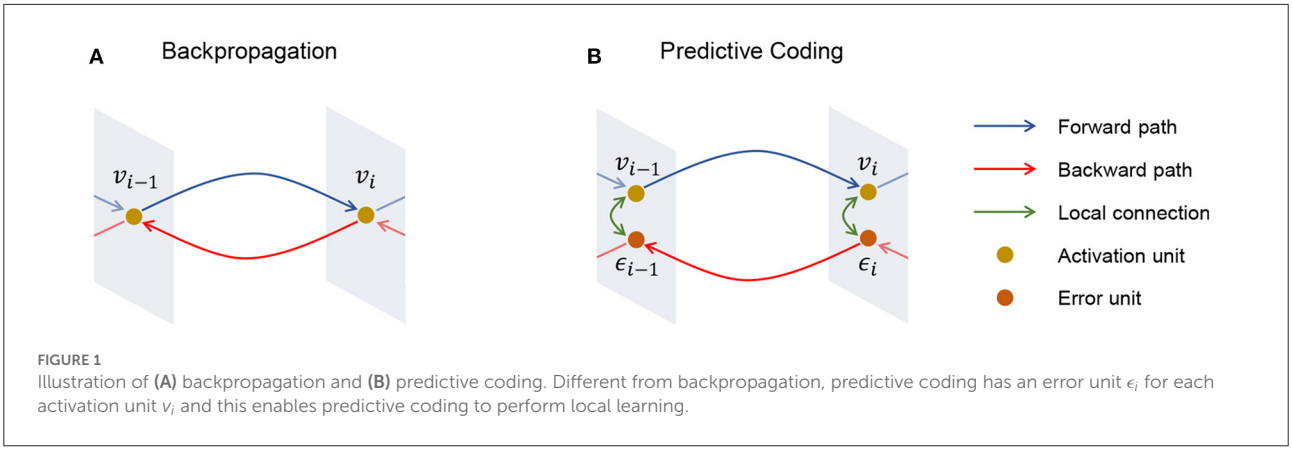


FIGURE 1
Illustration of **(A)** backpropagation and **(B)** predictive coding. Different from backpropagation, predictive coding has an error unit ϵ_i for each activation unit v_i and this enables predictive coding to perform local learning.

on this, Millidge et al. (2020) defined a objective function of predictive coding as the variational free energy $\mathcal{F}$ as follows (Friston, 2005; Buckley et al., 2017):

$$\mathcal{F} = KL[(Q(\{v_i\})||p(\{v_i\}))] \geq KL[Q(\{v_i\})||p(\{v_{1:N-1}|v_0, v_N\})] \approx \sum_{i=0}^{N} e_i^T e_i \tag{6}$$

where a prediction error of each layer e_i. The i-th error node e_i can be calculated as follows:

$$e_i = \hat{v}_{i-1} - v_i = f_i(v_{i-1}; \theta_i) - v_i \tag{7}$$

where v_{i-1} is the activation node value of the previous layer.

In the backward phase of predictive coding, network parameters θ containing activation nodes $\{v_i\}$ and error nodes $\{e_i\}$ are updated via gradient descent of each layer as follows:

$$dv_i = -\frac{\partial \mathcal{F}}{\partial v_i} = e_i - \sum_{j \in \mathcal{C}(v_i)} \frac{\partial \hat{v}_j}{\partial v_i}. \tag{8}$$

The learning is performed by minimizing the variational free energy $\mathcal{F}$ until converges as follows:

$$\theta_i = \theta_i + \eta d\theta_i \tag{9}$$

where η is the weight learning rate. Parameters are updated as follows:

$$d\theta_i = -\frac{\partial \mathcal{F}}{\partial \theta_i} = -e_i \frac{\partial f_i(v_{i-1}; \theta_i)}{\partial \theta_i} \tag{10}$$

The equation 10 indicates the local learning rule of the predictive coding where the parameters of i-th layer are only updated based on the e_i and v_{i-1}.

4. Incremental Learning with Predictive Coding

Based on previous studies (Hassabis et al., 2017; Perez-Nieves et al., 2021), our fundamental assumption is that the more biologically plausible the learning algorithm, closely replicating the learning mechanism of the brain, the more effective it will be for MCTs. Previous studies focused on confirming that the predictive coding network itself inherits the physiological characteristics of the brain. Salvatori et al. (2021) recently explored that predictive coding networks naturally implement associative memory, which plays a vital role in human intelligence (Colom et al., 2022). Motivated by the previous study, the current research assumed that predictive coding networks have a latent ability to consolidate the sequentially acquired knowledge in the human memory system. Therefore, we propose a predictive coding framework for incremental learning and verify the efficacy of MCTs. The task of incremental learning can be mainly categorized into two categories (Masana et al., 2020): class-incremental learning and task-incremental learning. The current study focused on the former. In class-incremental learning, the knowledge from previously seen classes is no longer available when a network learns the knowledge of unseen classes, and the learned network aims to achieve favorable classification accuracy for all tasks without forgetting. Multiple tasks were sequentially learned based on the pre-defined order to validate our assumption, and each task with its validation set finishing the training of the given task was evaluated. The algorithms are detailed in Algorithm 1.

4.1. Experimental Settings

A 3-layer predictive coding network with ReLU non-linearity, where the number of the hidden nodes was 800 for the simple dataset such as MNIST (LeCun et al., 1998) and FMNIST (Xiao et al., 2017), was employed. Similar to the study by Serra et al. (2018), a simplified Alexnet architecture (Krizhevsky et al., 2012) consisting of three convolutional layers was used for the complex dataset such as CIFAR-10 (Krizhevsky et al., 2009). The three convolutional layers comprised 64, 128, and 256 channels.

Input: Dataset $\mathcal{D}_{t=1}^{T}$, Computational Graph $\mathcal{G} = \{\mathcal{E}, \mathcal{V}\}$, inference learning rate η_v, weight learning rate η_θ

for all dataset for each task $\mathcal{D}_t \in \mathcal{D}$ **do** ▷ For each minibatch in the sequential tasks
  $\hat{v}_0 \leftarrow x_t$ ▷ Initialize the graph with inputs
  for all $\hat{v}_i \in \mathcal{V}$ **do** ▷ Forward phase: calculate predictions
    $\hat{v}_i \leftarrow f(\mathcal{P}(\hat{v}_i); \theta)$
  end for
  $\epsilon_L \leftarrow f_L(v_{L-1}; \theta_i) - v_L$ ▷ Compute output error
  while *not converged* **do** ▷ Backward phase: backward iteration
    for all $(v_i, \epsilon_i) \in \mathcal{G}$ **do**
      $\epsilon_i \leftarrow \hat{v}_{i-1} - v_i$ ▷ Compute prediction errors
      $v_i \leftarrow v_i + \eta_v \frac{d\mathcal{F}}{dv_i}$ ▷ Update the vertex values
    end for
  end while
end for
for all $\theta_i^t \in \mathcal{E}$ **do** ▷ Update weights at equilibrium
  $\theta_i^t \leftarrow \theta_i^{t+1} + \eta_\theta \frac{d\mathcal{F}}{d\theta_i}$
end for
  return θ^t

Algorithm 1. Predictive Coding for Incremental Learning.

We refined the data to formulate sequential incremental tasks. The data were divided into multiple portions following the representative incremental learning approaches (Lee et al., 2017; Sokar et al., 2021), and constructed four datasets: disjoint-MNIST, disjoint-FMNIST, split-MNIST, and split-CIFAR-10. Disjoint-MNIST and disjoint-FMNIST were organized by separating MNIST and FMNIST into two tasks. In addition, a more complex dataset, called split-MNIST and split-CIFAR-10, was also established, where all classes were separated into five tasks, and each task contained two categories. The details of the tasks on the multiple datasets are described in Tables 1 and 2. Finally, we evaluated incremental learning performance. We trained a network with sequential order and measured that the acquired knowledge was maintained after each task's training, same as Serra et al. (2018).

A learning rate of 0.05 was used, and the learning rate was divided by 1/3 to perform incremental learning, if there was no advancement in the validation loss for five consecutive epochs. In predictive coding, the weight learning rate was set as 0.1 while keeping the other hyperparameters. The minimum learning rate was set as $1e^{-4}$ and batch size as 64. All experiments were conducted using data split according to five different seeds. We provide the code to reproduce the results in the manuscript at https://github.com/jangho2001us/PredictiveCoding$_$IncrementalLearning.

4.2. Experiments on Incremental Learning

Incremental learning was performed on disjoint-MNIST and disjoint-FMNIST using the predictive coding framework to validate our hypothesis. To implement the incremental learning task in a predictive coding manner, we integrated the code of Serra et al. (2018) and Rosenbaum (2021) by replacing the network learning from the backpropagation with the predictive coding networks. The performance of each task was evaluated after completing the learning of each task in Tables 3 and 4. The performance in all tasks learned was evaluated using the best model of the last task. In this case, the best model refers to the model with the highest performance in the given task. Moreover, the other backpropagation-based incremental approaches containing SGD (Goodfellow et al., 2013), SGD-F (Goodfellow et al., 2013), EWC (Kirkpatrick et al., 2017), IMM (Lee et al., 2017), LFL (Jung et al., 2016), and LWF (Li and Hoiem, 2017) were evaluated to observe whether the predictive coding framework itself is effectual for preventing catastrophic forgetting. For all datasets, the average performance of the network trained with SGD based on the predictive coding manner outperformed the performance of the network trained with SGD based on backpropagation. Furthermore, learning with predictive coding exceeds strong competitor EWC (Kirkpatrick et al., 2017) on disjoint-MNIST and split-MNIST.

To make the challenging experimental settings, we combined two classes into one task and created five tasks using MNIST and CIFAR-10, similar to the study by Sokar et al. (2021). Incremental learning performance of backpropagation and predictive coding on split-MNIST and split-CIFAR-10 is shown in Tables 5 and 6. The performance of incremental learning based on predictive coding was also compared with that of conventional approaches (Goodfellow et al., 2013; Jung et al., 2016; Kirkpatrick et al., 2017; Lee et al., 2017; Li and Hoiem, 2017). To observe its ability to retain previously obtained knowledge, we visualized the average accuracy of trained tasks in Figure 2. Figure 2 and Table 5 are the experimental results from the same protocol (split-MNIST). After finishing every epoch, we evaluated the performance of all the tasks and drew Figure 2. While Table 5 shows the results of the average evaluation five times using the best model derived from each task. It was confirmed that catastrophic forgetting occurred in both learning algorithms, but the degree of forgetting was certainly more severe in the experimental results of backpropagation. Learning with predictive coding showed stable performance even when the learning task changed, in contrast to the pattern of backpropagation. In the backpropagation experiment, when the network acquired the

TABLE 1 Details of the tasks in the disjoint-MNIST and disjoint-FMNIST benchmarks.

Task id	MNIST classes	FMNIST classes	Training	Testing
1	[0, 1, 2, 3, 4]	[T-shirt/top, Trouser, Pullover, Dress, Coat]	25000	5000
2	[5, 6, 7, 8, 9]	[Sandal, Shirt, Sneaker, Bag, Ankle boot]	25000	5000

TABLE 2 Details of the tasks in the split-CIFAR-10 benchmark.

Task id	CIFAR-10 classes	Category	Training	Testing
1	[airplane, car]	vehicle	10000	2000
2	[bird, cat]	animal	10000	2000
3	[deer, dog]	animal	10000	2000
4	[frog, horse]	animal	10000	2000
5	[ship, truck]	vehicle	10000	2000

TABLE 3 Comparison of incremental learning performance (%) on disjoint-MNIST.

Algorithm	Method	Task1	Task2	Average
BP	SGD (Goodfellow et al., 2013)	88.19	98.99	93.59
	SGD-F (Goodfellow et al., 2013)	99.61	84.56	92.09
	EWC (Kirkpatrick et al., 2017)	92.29	98.99	95.64
	IMM-MEAN (Lee et al., 2017)	98.22	97.10	97.66
	IMM-MODE (Lee et al., 2017)	85.51	98.47	91.99
	LFL (Jung et al., 2016)	93.20	65.78	79.49
	LWF (Li and Hoiem, 2017)	99.43	98.84	99.13
PC	SGD (Goodfellow et al., 2013)	92.80	98.91	95.85

We denoted the learning with backpropagation as BP and learning with the predictive coding framework as PC. We used the five random seeds in the experiments and reported the average performance between task1 and task2.

knowledge of task 3, the knowledge of task 2 was forgotten. Further, when the network learned knowledge of task 5, it was confirmed that the discriminative information of tasks 1 and 2 was removed from the memories. These experimental results confirm that a biologically plausible learning algorithm reduces catastrophic forgetting in incremental learning and enhances the performance of incremental learning as one of MCTs.

We carried out additional experiments to demonstrate the advantages of learning with the brain-inspired algorithm. We implemented the predictive coding version of EWC (Kirkpatrick et al., 2017), IMM-MEAN (Lee et al., 2017), and IMM-MODE (Lee et al., 2017) algorithms and evaluated their performance on disjoint-MNIST. In the EWC algorithm, learning with predictive coding improves the average performance from 95.64% to 97.52%. In addition, learning with predictive coding enhances the average performance 0.21% and 5.42% in IMM-MEAN and IMM-MODE, respectively.

TABLE 4 Comparison of incremental learning performance (%) on disjoint-FMNIST.

Algorithm	Method	Task1	Task2	Average
BP	SGD (Goodfellow et al., 2013)	67.37	97.47	82.42
	SGD-F (Goodfellow et al., 2013)	91.87	82.06	86.96
	EWC (Kirkpatrick et al., 2017)	88.79	96.66	92.72
	IMM-MEAN (Lee et al., 2017)	85.70	95.46	87.78
	IMM-MODE (Lee et al., 2017)	64.15	96.33	80.24
	LFL (Jung et al., 2016)	79.00	83.01	81.00
	LWF (Li and Hoiem, 2017)	91.24	97.35	94.30
PC	SGD (Goodfellow et al., 2013)	75.68	97.11	86.40

We denoted the learning with backpropagation as BP and learning with the predictive coding framework as PC. We used the five random seeds in the experiments and reported the average performance between task1 and task2.

5. Limited Data Recognition with Predictive Coding

The potential of predictive coding networks for limited data recognition was then investigated. Specifically, the efficacy of predictive coding networks in long-tailed recognition and few-shot recognition type of MCTs was analyzed. First, real-world datasets are often highly imbalanced following long-tail distribution in which data category accounts for a significant portion of the overall data (Johnson and Khoshgoftaar, 2019; Liu et al., 2019b). Owing to the skewed class distribution of the dataset, the network trained with a class-imbalanced dataset may show a classification bias problem in which the samples of tail classes are predicted as head classes (Cao et al., 2019). In addition, managing few-shot samples in an open-world setting is crucial because it is similar to the situation in which the human recognition system can be encountered. Second, to achieve more human-like recognition performance, effectively managing few-shot examples in an open-world setting is crucial. Two experimental scenarios are significant because it is realistic situations that human recognition can encounter.

The cortical neuron in the human brain can learn with only a few repetitions owing to the local synaptic plasticity (Yger et al., 2015), and it is widely known that such plasticity contributes to the interactions between limited data (Wu et al., 2022). It has been demonstrated that the changes in synaptic connections assist in learning new information and long-term memory formation (Yang et al., 2009). Given the characteristics

TABLE 5 Comparison of incremental learning performance (%) on split-MNIST.

Algorithm	Method	Task1	Task2	Task3	Task4	Task5	Average
BP	SGD (Goodfellow et al., 2013)	98.52	74.06	93.74	96.43	99.61	92.47
	SGD-F (Goodfellow et al., 2013)	99.95	90.52	95.43	98.06	87.38	94.27
	EWC (Kirkpatrick et al., 2017)	99.41	75.24	94.21	96.34	99.60	92.96
	IMM-MEAN (Lee et al., 2017)	99.94	98.67	94.38	96.55	88.33	95.57
	IMM-MODE (Lee et al., 2017)	99.88	74.20	95.27	97.47	99.42	93.25
	LFL (Jung et al., 2016)	94.34	52.62	54.34	70.63	89.36	72.26
	LWF (Li and Hoiem, 2017)	99.95	99.10	99.77	99.83	99.76	99.68
PC	SGD (Goodfellow et al., 2013)	99.89	97.09	99.28	99.39	98.37	98.80

We denoted the learning with backpropagation as BP and learning with the predictive coding framework as PC. We used the five random seeds in the experiments and reported the average performance from task1 to task5.

TABLE 6 Comparison of incremental learning performance (%) on split-CIFAR-10.

Algorithm	Method	Task1	Task2	Task3	Task4	Task5	Average
BP	SGD (Goodfellow et al., 2013)	72.17	66.08	71.44	84.17	93.71	77.51
	SGD-F (Goodfellow et al., 2013)	95.72	67.96	60.03	69.97	77.38	74.15
	EWC (Kirkpatrick et al., 2017)	72.76	64.90	67.53	73.99	72.15	70.26
	IMM-MEAN (Lee et al., 2017)	89.71	78.35	78.51	74.73	78.91	80.04
	IMM-MODE (Lee et al., 2017)	76.14	67.07	73.63	84.79	93.87	79.10
	LFL (Jung et al., 2016)	71.50	59.30	71.71	84.47	84.85	74.37
	LWF (Li and Hoiem, 2017)	76.95	70.58	78.46	94.34	93.99	82.86
PC	SGD (Goodfellow et al., 2013)	70.42	74.27	80.70	87.21	90.96	80.71

We denoted the learning with backpropagation as BP and learning with the predictive coding framework as PC. We used the five random seeds in the experiments and reported the average performance from task1 to task5.

of synaptic plasticity, experiments with a predictive coding framework were performed on the class-imbalanced data, and the biologically plausible learning algorithm that helped limited data recognition was identified.

5.1. Experimental Settings

The same architecture used in the previous section consisting of three-layer MLP was used in long-tailed recognition. The number of hidden neurons was set as 800 with ReLU non-linearity and dropout. We used MNIST (LeCun et al., 1998) for our experiment and synthesized the long-tailed data with an imbalance ratio γ. The imbalance ratio was defined as the proportion of the samples of the highest number of classes to the lowest number of classes as $\frac{N_{max}}{N_{min}}$. Although it differed depending on the imbalance ratio, in general, N_{max} and N_{min} usually followed the relationship, $N_{max} \gg N_{min}$. Exponential distribution and the number of samples N_l in l-th class was defined as $N_l = N_{max} \cdot \gamma^{-\frac{l-1}{L-1}}$. The four types of imbalanced data distribution were then synthesized as previously described (Kim et al., 2020). To train a network, we set a batch size of 128 and optimized a model until 100 epochs. When backpropagation was used for learning, the learning rate was increased from 0.0001 to 0.5 by growing five times, and the best performance results among them were determined. When predictive coding was used for the optimization, a learning rate of 0.002 with a weight decay of $2e^{-4}$ was used. Additionally, the weight learning rate η was set as 0.1 and the number of iterations as 20 as hyperparameters for predictive coding networks. All the experiments with predictive coding were performed under the fixed prediction assumption. We provide the code to reproduce the results in the manuscript at https://github.com/jangho2001us/PredictiveCoding$_$LongTailedRecognition.

In few-shot recognition, the same experimental settings with those of Snell et al. (2017), which comprised four convolutional blocks with Batch normalization, ReLU, and MaxPool were used. Experiments on few-shot recognition were conducted with Omniglot (Lake et al., 2011) dataset containing 1623 categories of handwritten characters. The performance of few-shot recognition is commonly measured by N-way k-shot classification, where N implies the number of given classes and k indicates the number of samples in each category. The current study extended the experimental protocol of the original paper to 30-way k-shot experiment settings because those evaluation

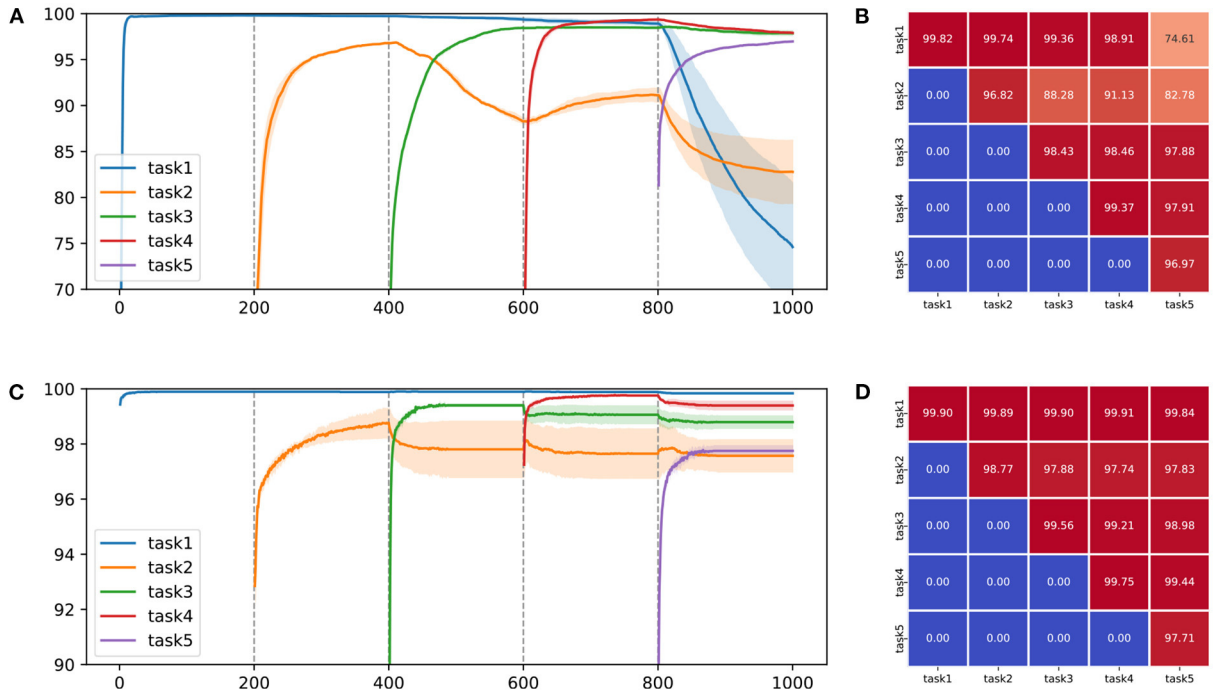


FIGURE 2
Qualitative and quantitative performance comparison on two learning schemes for **(A,B)** backpropagation and **(C,D)** predictive coding on split-MNIST. In **(A,C)**, the solid line indicates the average accuracy for each task and the transparent region represents the standard deviation on five random seeds. The vertical dashed line refers to the point at which the task to be learned changes. In **(B,D)**, each value indicates the performance each task measured by the final model.

protocols are more difficult because the number of classes for the candidate group increases. The learning rate was set to $1e^{-3}$ and then reduced by 1/10 every 20 epoch to train a network. For learning networks with a predictive coding framework, the same learning rate, weight decay, weight learning rate, and iterations were used. For more information, please refer to the original paper (Snell et al., 2017). We provide the code to reproduce the results in the manuscript at https://github.com/jangho2001us/PredictiveCoding$_$FewShotRecognition.

5.2. Experiments on Long-tailed Recognition

In Table 7, we compared the long-tailed recognition performance with Cross-Entropy (CE) loss, Mixup approach (Zhang et al., 2017), Focal loss (Lin et al., 2017), Class-Balanced Focal (CB Focal) loss (Cui et al., 2019), Label-Distribution-Aware-Margin (LDAM) loss (Cao et al., 2019), and Balanced Meta-Softmax (BALMS) loss (Ren et al., 2020). Further details on multiple learning objectives are provided in the Supplementary material. The experimental results showed the benefit of learning with predictive coding networks. First, the long-tailed recognition performance was higher by 4.45% in learning the network with a predictive coding framework than that in learning with CE loss under severe class imbalance of data distribution. Similar results in the following experiments were observed when the network was trained with other learning objectives such as Focal (Lin et al., 2017) and BALMS (Ren et al., 2020). In this experiment, the performance improvement is evaluated using the predictive coding framework rather than comparing performance between different learning objectives. The results shown in Table 7 indicate that the learning algorithm close to the human brain brings a positive effect on MCTs, confirming our assumption.

5.3. Experiments on Few-shot Recognition

The few-shot recognition performance trained with backpropagation and predictive coding framework is shown in Table 8, Learning with predictive coding enabled robust recognition under the various few-shot experimental protocols. Additionally, predictive coding networks showed their potential ability under challenging inference settings such as 20-way 1-shot and 30-way 1-shot rather than 20-way 5-shots and 30-way 5-shots. The experimental results confirmed our assumptions

TABLE 7 Comparison of classification performance (%) on MNIST under four different imbalance distributions.

Algorithm	Objective Function	Imbalance Ratio (γ)			
		200	100	50	10
BP	CE	68.78	78.06	89.63	97.17
	Mixup (Zhang et al., 2017)	67.60	76.69	86.97	96.15
	Focal (Lin et al., 2017)	70.92	79.42	90.89	97.31
	CB Focal (Cui et al., 2019)	69.93	79.72	91.26	97.09
	LDAM (Cao et al., 2019)	65.17	75.58	84.91	97.14
	BALMS (Ren et al., 2020)	72.25	81.34	92.50	97.23
PC	CE	73.23 (+4.45)	79.26 (+1.20)	90.10 (+0.47)	97.37 (+0.20)
	Mixup (Zhang et al., 2017)	67.77 (+0.17)	77.60 (+0.91)	88.26 (+1.29)	96.27 (+0.12)
	Focal (Lin et al., 2017)	71.99 (+1.07)	79.57 (+0.15)	91.18 (+0.29)	97.03 (-0.28)
	CB Focal (Cui et al., 2019)	70.19 (+0.26)	80.28 (+0.56)	91.40 (+0.14)	97.24 (+0.14)
	LDAM (Cao et al., 2019)	65.54 (+0.37)	76.05 (+0.47)	85.08 (+0.17)	97.20 (+0.06)
	BALMS (Ren et al., 2020)	74.22 (+1.97)	82.28 (+0.94)	93.50 (+1.00)	97.45 (+0.22)

Experiments are performed with five random seeds, and the average performance is reported. Relative variance is provided in the bracket. Increments are presented as red and decrements as blue.

TABLE 8 Experimental results on the low-shot recognition on the Omniglot dataset.

Algorithm	Method	5-way Acc.		10-way Acc.		20-way Acc.		30-way Acc.	
		1-shot	5-shot	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
BP	ProtoNet (Snell et al., 2017)	98.41	99.56	96.87	99.18	94.64	98.54	92.97	97.98
PC	ProtoNet (Snell et al., 2017)	98.46 (+0.05)	99.59 (+0.03)	96.98 (+0.11)	99.19 (+0.01)	94.88 (+0.24)	98.59 (+0.05)	93.14 (+0.17)	98.05 (+0.07)

Five random seeds are used in the experiment, and the average performance is reported. Relative variance is shown in the bracket. Increments are presented as red.

and supported that the brain-like learning algorithm was effective for MCTs.

6. Discussion

6.1. Analysis of Plasticity-stability Aspects

The plasticity-stability dilemma is a well-known problem widely studied in biology (Mateos-Aparicio and Rodríguez-Moreno, 2019). This phenomenon is related to the power of consolidation of new information without forgetting previously acquired information (Mermillod et al., 2013). Further, it is an essential issue in incremental learning with ANNs (Lin et al., 2022). The human brain is well-controlled to learn new information and to prevent the learned information from being overridden by the new information (Takesian and Hensch, 2013). However, ANNs naturally induce catastrophic forgetting and expose the trade-off between plasticity and stability (Kirkpatrick et al., 2017).

To confirm that predictive coding achieves a better plasticity-stability trade-off than backpropagation, we experimented with split-MNIST by controlling the stability of two learning mechanisms. Adjusting the learning rate is not directly related to stability, but it was used because it was considered as a factor that could adjust stability in our experiments. In Figure 3, we report the experimental results and compare the learning schemes by adjusting the learning rate of backpropagation and the weight learning rate of predictive coding. In backpropagation experiments, the

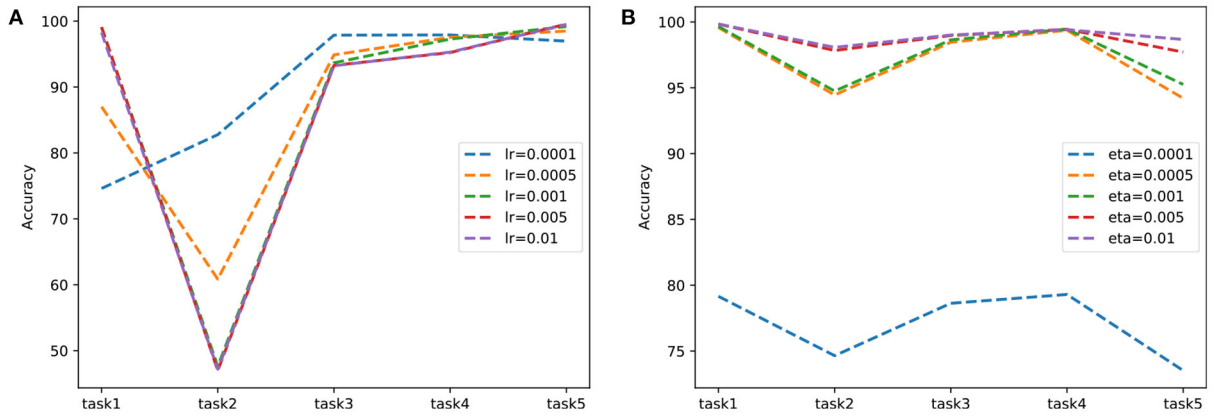


FIGURE 3
Comparison of learning with **(A)** backpropagation and **(B)** predictive coding on split-MNIST in two learning schemes. To adjust network stability, the learning rate of backpropagation and the weight learning rate of predictive coding are varied.

learning is reduced from 0.01 to 0.0001 to decrease forgetting of acquired knowledge. When the learning rate was 0.0001, the network forgot less information to perform task 2. However, it still showed limited performance in tasks 1 and 2. Thus, maintaining stability by reducing the learning rate may not be acceptable because it deteriorates the overall performance. Meanwhile, performance was consistently high for each task in predictive coding experiments. These results implied predictive coding had better plasticity properties than backpropagation while maintaining stability.

6.2. Interplay of Hippocampus and Prefrontal Cortex

The hippocampus plays an essential role in episodic memory at the top of the cortical processing hierarchy (Felleman and Van Essen, 1991). In incremental learning, the ability to regulate learned information and retrieve context-appropriate memories is essential. We can understand the effectiveness of predictive coding in incremental learning as the interaction between the hippocampus and the prefrontal cortex in the human brain (Eichenbaum, 2017; Barron et al., 2020). It is well known that the hippocampus can quickly encode new information, stabilize memory traces, and organize memory networks (Preston and Eichenbaum, 2013). In addition, this mechanism has been physiologically proven through functional magnetic resonance imaging studies (Hindy et al., 2019).

We have shown that the learning process of predictive coding networks is analogous to the interaction between the hippocampus and the prefrontal cortex in the human brain (Eichenbaum, 2017). As described in Algorithm 1, the learning process based on predictive coding networks can be divided into two phases: forward and backward pass. In the forward phase, the predictive coding network computes its predictions for every layer. In the backward phase, the predictive coding network minimizes the free-energy summation as a learning objective. The two-phase learning of predictive coding networks corresponds to acquiring and consolidating information in the hippocampus and prefrontal cortex. The predictive coding framework promotes the two processes and enables accurate inference when data containing information corresponding to the previously learned task are received.

6.3. Rationale for Selecting Predictive Coding

The reason why we selected predictive coding as a brain-inspired algorithm is as follows. As described in Section 2, predictive coding is potentially more biologically plausible because local learning rules perform parameter updates. This property is distinct from the update of backpropagation executed from the global error signal. It will be ideal if the parameter update is performed asynchronously in a different layer, such as the neural plasticity of the human brain. However, the parameter update of predictive coding occurs under the *fixed prediction assumption* (Millidge et al., 2020). The fixed prediction assumption implies that the parameters are updated based on the *fixed* predictions of the forward phase. Whittington and Bogacz (2017) demonstrated that a predictive coding network with a fixed prediction assumption performs the same parameter updates as backpropagation. Another limitation of predictive coding is the degree of convergence of variational free energy used as a learning objective. The convergence of the backward phase is achieved by setting a specific number of iterations (Rosenbaum, 2021). Depending on the number of backward iterations, learning with predictive coding may

converge or diverge. Although these two issues introduced earlier remain open questions, we conducted our experiments using predictive coding because we thought its advantages outweighed its disadvantages.

7. Conclusion

This study empirically demonstrated the potential effectiveness of predictive coding in MCTs. However, despite this, the predictive coding algorithm still has some limitations. First, predictive coding requires a longer training time than backpropagation because it executes backward iteration until the error nodes and activation nodes converge. Although we expanded our experiments for large networks such as VGGNet and ResNet (He et al., 2016; Krizhevsky et al., 2017), we could not perform the experiments on MCTs because of the excessive training time. Second, to conduct learning with predictive coding, the network should be an architecture composed of sequential layers. For example, if shortcut connections exist, it is challenging to implement them into a predictive coding layer. In this case, we set the block unit, which is the boundary of the shortcut, as the predictive coding layer. If predictive coding combines learning speed and scalability, there will be infinite opportunities for development as a learning algorithm that can replace backpropagation.

In summary, we extensively analyze the benefits of learning ANNs with predictive coding frameworks for MCTs. MCTs can be described as tasks that are easy for human intelligence while difficult for machine intelligence. Based on our hypothesis, we empirically demonstrate that brain-inspired predictive coding has advantages in incremental learning on MNIST and CIFAR, long-tailed recognition on MNIST, and few-shot recognition on Omniglot. In neuroscience, especially the intrinsic properties of the human brain, we discuss why training ANNs with a predictive coding framework improves the performance of MCTs. The study concludes that predictive coding learning is similar to the plasticity-stability property of the human brain and mainly mimics the interaction between the hippocampus and prefrontal cortex. Finally, it is an interesting avenue for future work to reduce the training time under the fixed prediction assumption and relax the constraint of predictive coding while maintaining the performance.

Data availability statement

The original contributions presented in the study are publicly available. The data can be found here: http://yann.lecun.com/exdb/mnist and https://www.cs.toronto.edu/~kriz/cifar.html.

Author contributions

JL contributed to the design of the study and performed the experiments. JL, JJ, BL, and J-HL developed the algorithm and performed the result analysis, and wrote the revised manuscript. SY conceived and supervised the project, checked ideas and terminology, performed result analysis, editing, and revision of the manuscript.

Funding

This work was supported by Electronics and Telecommunications Research Institute (ETRI) grant funded by the Korean government [22ZS1100, Core Technology Research for Self-Improving Integrated Artificial Intelligence System], and the BK21 FOUR program of the Education and Research Program for Future ICT Pioneers, Seoul National University in 2022.

Acknowledgments

The authors would like to thank Hyemi Jang and Bonggyun Kang for their contribution in the early stages of this study.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

Supplementary material

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fncom.2022.1062678/full#supplementary-material

References

Ahmad, N., van Gerven, M. A., and Ambrogioni, L. (2020). "Gait-prop: a biologically plausible learning rule derived from backpropagation of error," in *Advances in Neural Information Processing Systems 33*, 10913–10923.

Akrout, M., Wilson, C., Humphreys, P., Lillicrap, T., and Tweed, D. B. (2019). "Deep learning without weight transport," in *Advances in Neural Information Processing Systems 32*, NeurIPS 2019, Vancouver.

Barron, H. C., Auksztulewicz, R., and Friston, K. (2020). Prediction and memory: a predictive coding account. *Prog. Neurobiol.* 192, 101821. doi: 10.1016/j.pneurobio.2020.101821

Bertolero, M. A., Yeo, B. T., and DŠEsposito, M. (2015). The modular and integrative functional architecture of the human brain. *Proc. Natl. Acad. Sci. U.S.A.* 112, E6798ŰE6807. doi: 10.1073/pnas.1510619112

Bird, C. M., and Burgess, N. (2008). The hippocampus and memory: insights from spatial processing. *Nat. Rev. Neurosci.* 9, 182–194. doi: 10.1038/nrn2335

Bogacz, R. (2017). A tutorial on the free-energy framework for modelling perception and learning. *J. Math. Psychol.* 76, 198–211. doi: 10.1016/j.jmp.2015.11.003

Buckley, C. L., Kim, C. S., McGregor, S., and Seth, A. K. (2017). The free energy principle for action and perception: a mathematical review. *J. Math. Psychol.* 81, 55–79. doi: 10.1016/j.jmp.2017.09.004

Cao, K., Wei, C., Gaidon, A., Arechiga, N., and Ma, T. (2019). "Learning imbalanced datasets with label-distribution-aware margin loss," in *Advances in Neural Information Processing Systems 32*, NeurIPS 2019, Vancouver.

Choksi, B., Mozafari, M., Biggs O'May, C., Ador, B., Alamia, A., and VanRullen, R. (2021). Predify: "Augmenting deep neural networks with brain-inspired predictive coding dynamics," in *Advances in Neural Information Processing Systems 34*, NeurIPS 2021.

Citri, A., and Malenka, R. C. (2008). Synaptic plasticity: multiple forms, functions, and mechanisms. *Neuropsychopharmacology* 33, 18–41. doi: 10.1038/sj.npp.1301559

Colom, R., Karama, S., Jung, R. E., and Haier, R. J. (2022). Human intelligence and brain networks. *Dialogues Clin. Neurosci.* 12, 489–501. doi: 10.31887/DCNS.2010.12.4/rcolom

Cui, Y., Jia, M., Lin, T.-Y., Song, Y., and Belongie, S. (2019). "Class-balanced loss based on effective number of samples," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Long Beach, CA, 9268–9277. doi: 10.1109/CVPR.2019.00949

Davachi, L., and DuBrow, S. (2015). How the hippocampus preserves order: the role of prediction and context. *Trends Cogn. Sci.* 19, 92–99. doi: 10.1016/j.tics.2014.12.004

De Man, R., Gang, G. J., Li, X., and Wang, G. (2019). Comparison of deep learning and human observer performance for detection and characterization of simulated lesions. *J. Med. Imaging* 6, 025503. doi: 10.1117/1.JMI.6.2.025503

Dellaferrera, G., and Kreiman, G. (2022). "Error-driven input modulation: solving the credit assignment problem without a backward pass," in *Proceedings of the 39th International Conference on Machine Learning*, Baltimore, MA, 4937–4955.

Denéve, S., Alemi, A., and Bourdoukan, R. (2017). The brain as an efficient and robust adaptive learner. *Neuron* 94, 969–977. doi: 10.1016/j.neuron.2017.05.016

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., et al. (2020). An image is worth 16x16 words: transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*. doi: 10.48550/arXiv.2010.11929

Eichenbaum, H. (2017). Prefrontal-hippocampal interactions in episodic memory. *Nat. Rev. Neurosci.* 18, 547–558. doi: 10.1038/nrn.2017.74

Felleman, D. J., and Van Essen, D. C. (1991). Distributed hierarchical processing in the primate cerebral cortex. *Cereb. Cortex* 1, 1–47. doi: 10.1093/cercor/1.1.1

French, R. M. (1999). Catastrophic forgetting in connectionist networks. *Trends Cogn. Sci.* 3, 128–135. doi: 10.1016/S1364-6613(99)01294-2

Friston, K. (2003). Learning and inference in the brain. *Neural Netw.* 16, 1325–1352. doi: 10.1016/j.neunet.2003.06.005

Friston, K. (2005). A theory of cortical responses. *Philos. Trans. R. Soc. B Biol. Sci.* 360, 815–836. doi: 10.1098/rstb.2005.1622

Friston, K. (2008). Hierarchical models in the brain. *PLoS Comput. Biol.* 4, e1000211. doi: 10.1371/journal.pcbi.1000211

Geirhos, R., Janssen, D. H., Schütt, H. H., Rauber, J., Bethge, M., and Wichmann, F. A. (2017). Comparing deep neural networks against humans: object recognition when the signal gets weaker. *arXiv preprint arXiv:1706.06969*. doi: 10.48550/arXiv.1706.06969

Goodfellow, I. J., Mirza, M., Xiao, D., Courville, A., and Bengio, Y. (2013). An empirical investigation of catastrophic forgetting in gradient-based neural networks. *arXiv preprint arXiv:1312.6211*.

Goodfellow, I. J., Shlens, J., and Szegedy, C. (2014). Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*. doi: 10.48550/arXiv.1412.6572

Grossberg, S. (1987). Competitive learning: from interactive activation to adaptive resonance. *Cogn. Sci.* 11, 23–63. doi: 10.1111/j.1551-6708.1987.tb00862.x

Han, K., Wen, H., Zhang, Y., Fu, D., Culurciello, E., and Liu, Z. (2018). "Deep predictive coding network with local recurrent processing for object recognition," in *Advances in Neural Information Processing Systems 31*, NeurIPS 2018, Montreal.

Hassabis, D., Kumaran, D., Summerfield, C., and Botvinick, M. (2017). Neuroscience-inspired artificial intelligence. *Neuron* 95, 245–258. doi: 10.1016/j.neuron.2017.06.011

He, K., Zhang, X., Ren, S., and Sun, J. (2016). "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, GA, 770–778. doi: 10.1109/CVPR.2016.90

Hebb, D. O. (2005). *The Organization of Behavior: A Neuropsychological Theory*. Psychology Press.

Hindy, N. C., Avery, E. W., and Turk-Browne, N. B. (2019). Hippocampal-neocortical interactions sharpen over time for predictive actions. *Nat. Commun.* 10, 1–13. doi: 10.1038/s41467-019-12016-9

Illing, B., Ventura, J., Bellec, G., and Gerstner, W. (2021). "Local plasticity rules can learn deep representations using self-supervised contrastive predictions," in *Advances in Neural Information Processing Systems 34*.

Johnson, J. M., and Khoshgoftaar, T. M. (2019). Survey on deep learning with class imbalance. *J. Big Data* 6, 1–54. doi: 10.1186/s40537-019-0192-5

Jung, H., Ju, J., Jung, M., and Kim, J. (2016). Less-forgetting learning in deep neural networks. *arXiv preprint arXiv:1607.00122*. doi: 10.48550/arXiv.1607.00122

Kim, J., Hur, Y., Park, S., Yang, E., Hwang, S. J., and Shin, J. (2020). "Distribution aligning refinery of pseudo-label for imbalanced semi-supervised learning," in *Advances in Neural Information Processing Systems 33*, 14567–14579.

Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., et al. (2017). Overcoming catastrophic forgetting in neural networks. *Proc. Natl. Acad. Sci. U.S.A.* 114, 3521–3526. doi: 10.1073/pnas.1611835114

Krizhevsky, A., and Hinton, G. (2009). *Learning Multiple Layers of Features From Tiny Images*. Technical Report, University of Toronto, Toronto, ON, Canada.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). "ImageNet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, NeurIPS 2012, Lake Tahoe.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2017). ImageNet classification with deep convolutional neural networks. *Commun. ACM* 60, 84–90. doi: 10.1145/3065386

Lake, B., Salakhutdinov, R., Gross, J., and Tenenbaum, J. (2011). "One shot learning of simple visual concepts," in *Proceedings of the Annual Meeting of the Cognitive Science Society*.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proc. IEEE* 86, 2278–2324. doi: 10.1109/5.726791

Lee, D.-H., Zhang, S., Fischer, A., and Bengio, Y. (2015). "Difference target propagation," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases* (Proto: Springer), 498–515. doi: 10.1007/978-3-319-23528-8_31

Lee, S.-W., Kim, J.-H., Jun, J., Ha, J.-W., and Zhang, B.-T. (2017). "Overcoming catastrophic forgetting by incremental moment matching," in *Advances in Neural Information Processing Systems 30*, NeurIPS 2017, Long Beach, CA.

Li, Z., and Hoiem, D. (2017). Learning without forgetting. *IEEE Trans. Pattern Anal. Mach. Intell.* 40, 2935–2947. doi: 10.1109/TPAMI.2017.2773081

Liao, Q., Leibo, J., and Poggio, T. (2016). "How important is weight symmetry in backpropagation?" in *Proceedings of the AAAI Conference on Artificial Intelligence*, Phoenix, AZ, doi: 10.1609/aaai.v30i1.10279

Lillicrap, T. P., Cownden, D., Tweed, D. B., and Akerman, C. J. (2016). Random synaptic feedback weights support error backpropagation for deep learning. *Nat. Commun.* 7, 1–10. doi: 10.1038/ncomms13276

Lin, G., Chu, H., and Lai, H. (2022). "Towards better plasticity-stability trade-off in incremental learning: a simple linear connector," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, New Orleans, AK, 89–98. doi: 10.1109/CVPR52688.2022.00019

Lin, T.-Y., Goyal, P., Girshick, R., He, K., and Dollár, P. (2017). "Focal loss for dense object detection," in *Proceedings of the IEEE International Conference on Computer Vision*, Venice, 2980–2988. doi: 10.1109/ICCV.2017.324

Lindsey, J., and Litwin-Kumar, A. (2020). "Learning to learn with feedback and local plasticity," in *Advances in Neural Information Processing Systems 33*, 21213–21223.

Liu, K., Li, X., Zhou, Y., Guan, J., Lai, Y., Zhang, G., et al. (2021). Denoised internal models: a brain-inspired autoencoder against adversarial attacks. *arXiv preprint arXiv:2111.10844*. doi: 10.1007/s11633-022-1375-7

Liu, X., Faes, L., Kale, A. U., Wagner, S. K., Fu, D. J., Bruynseels, A., et al. (2019a). A comparison of deep learning performance against health-care professionals in detecting diseases from medical imaging: a systematic review and meta-analysis. *Lancet Digit. Health* 1, e271Ű e297. doi: 10.1016/S2589-7500(19)30123-2

Liu, Z., Miao, Z., Zhan, X., Wang, J., Gong, B., and Yu, S. X. (2019b). "Large-scale long-tailed recognition in an open world," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Long Beach, CA, 2537–2546. doi: 10.1109/CVPR.2019.00264

Masana, M., Liu, X., Twardowski, B., Menta, M., Bagdanov, A. D., and van de Weijer, J. (2020). Class-incremental learning: survey and performance evaluation on image classification. *arXiv preprint arXiv:2010.15277*. doi: 10.48550/arXiv.2010.15277

Mateos-Aparicio, P., and Rodriguez-Moreno, A. (2019). The impact of studying brain plasticity. *Front. Cell. Neurosci.* 13, 66. doi: 10.3389/fncel.2019.00066

McCloskey, M., and Cohen, N. J. (1989). "Catastrophic interference in connectionist networks: the sequential learning problem," in *Psychology of Learning and Motivation*, Vol. 24 (Elsevier), 109–165. doi: 10.1016/S0079-7421(08)60536-8

Mermillod, M., Bugaiska, A., and Bonin, P. (2013). The stability-plasticity dilemma: investigating the continuum from catastrophic forgetting to age-limited learning effects. *Front. Psychol.* 4, 504. doi: 10.3389/fpsyg.2013.00504

Millidge, B., Tschantz, A., and Buckley, C. L. (2020). Predictive coding approximates backprop along arbitrary computation graphs. *arXiv preprint arXiv:2006.04182*. doi: 10.48550/arXiv.2006.04182

Neves, G., Cooke, S. F., and Bliss, T. V. (2008). Synaptic plasticity, memory and the hippocampus: a neural network approach to causality. *Nat. Rev. Neurosci.* 9, 65–75. doi: 10.1038/nrn2303

Ohayon, S., Freiwald, W. A., and Tsao, D. Y. (2012). What makes a cell face selective? The importance of contrast. *Neuron* 74, 567–581. doi: 10.1016/j.neuron.2012.03.024

Perez-Nieves, N., Leung, V. C., Dragotti, P. L., and Goodman, D. F. (2021). Neural heterogeneity promotes robust learning. *Nat. Commun.* 12, 1–9. doi: 10.1038/s41467-021-26022-3

Pogodin, R., and Latham, P. (2020). "Kernelized information bottleneck leads to biologically plausible 3-factor Hebbian learning in deep networks," in *Advances in Neural Information Processing Systems 33*, 7296–7307.

Power, J. D., and Schlaggar, B. L. (2017). Neural plasticity across the lifespan. *Wiley Interdiscipl. Rev. Dev. Biol.* 6, e216. doi: 10.1002/wdev.216

Preston, A. R., and Eichenbaum, H. (2013). Interplay of hippocampus and prefrontal cortex in memory. *Curr. Biol.* 23, R764ŰR773. doi: 10.1016/j.cub.2013.05.041

Rao, R. P., and Ballard, D. H. (1999). Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. *Nat. Neurosci.* 2, 79–87. doi: 10.1038/4580

Ren, J., Yu, C., Ma, X., Zhao, H., Yi, S., et al. (2020). "Balanced meta-softmax for long-tailed visual recognition," in *Advances in Neural Information Processing Systems 33*, 4175–4186.

Rosenbaum, R. (2021). On the relationship between predictive coding and backpropagation. *arXiv preprint arXiv:2106.13082*. doi: 10.1371/journal.pone.0266102

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature* 323, 533–536. doi: 10.1038/323533a0

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., et al. (2015). Imagenet large scale visual recognition challenge. *Int. J. Comput. Vis.* 115, 211–252. doi: 10.1007/s11263-015-0816-y

Salvatori, T., Song, Y., Hong, Y., Sha, L., Frieder, S., Xu, Z., et al. (2021). "Associative memories via predictive coding," in *Advances in Neural Information Processing Systems 34*, NeurIPS 2021.

Samuel, D., Atzmon, Y., and Chechik, G. (2021). "From generalized zero-shot learning to long-tail with class descriptors," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, Waikoloa, HI, 286–295. doi: 10.1109/WACV48630.2021.00033

Serra, J., Suris, D., Miron, M., and Karatzoglou, A. (2018). "Overcoming catastrophic forgetting with hard attention to the task," in *International Conference on Machine Learning* (PMLR), Stockholm, 4548–4557.

Snell, J., Swersky, K., and Zemel, R. (2017). "Prototypical networks for few-shot learning," in *Advances in Neural Information Processing Systems 30*, NeurIPS 2017, Long Beach, CA.

Sokar, G., Mocanu, D. C., and Pechenizkiy, M. (2021). Addressing the stability-plasticity dilemma via knowledge-aware continual learning. *arXiv preprint arXiv:2110.05329*. doi: 10.48550/arXiv.2110.05329

Susman, L., Brenner, N., and Barak, O. (2019). Stable memory with unstable synapses. *Nat. Commun.* 10, 1–9. doi: 10.1038/s41467-019-12306-2

Suzuki, Y., Ikeda, H., Miyamoto, T., Miyakawa, H., Seki, Y., Aonishi, T., et al. (2015). Noise-robust recognition of wide-field motion direction and the underlying neural mechanisms in *Drosophila melanogaster*. *Sci. Rep.* 5, 1–12. doi: 10.1038/srep10253

Takesian, A. E., and Hensch, T. K. (2013). Balancing plasticity/stability across brain development. *Prog. Brain Res.* 207, 3–34. doi: 10.1016/B978-0-444-63327-9.00001-1

Wardle, S. G., Taubert, J., Teichmann, L., and Baker, C. I. (2020). Rapid and dynamic processing of face pareidolia in the human brain. *Nat. Commun.* 11, 1–14. doi: 10.1038/s41467-020-18325-8

Wen, H., Han, K., Shi, J., Zhang, Y., Culurciello, E., and Liu, Z. (2018). "Deep predictive coding network for object recognition," in *International Conference on Machine Learning* (PMLR), 5266–5275.

Whittington, J. C., and Bogacz, R. (2017). An approximation of the error backpropagation algorithm in a predictive coding network with local hebbian synaptic plasticity. *Neural Comput.* 29, 1229–1262. doi: 10.1162/NECO_a_00949

Woo, S., Park, J., Hong, J., and Jeon, D. (2021). "Activation sharing with asymmetric paths solves weight transport problem without bidirectional connection," in *Advances in Neural Information Processing Systems 34*, NeurIPS 2021.

Wu, Y., Zhao, R., Zhu, J., Chen, F., Xu, M., Li, G., et al. (2022). Brain-inspired global-local learning incorporated with neuromorphic computing. *Nat. Commun.* 13, 1–14. doi: 10.1038/s41467-021-27653-2

Xiao, H., Rasul, K., and Vollgraf, R. (2017). Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*. doi: 10.48550/arXiv.1708.07747

Xu, Y., and Vaziri-Pashkam, M. (2021). Limits to visual representational correspondence between convolutional neural networks and the human brain. *Nat. Commun.* 12, 1–16. doi: 10.1038/s41467-021-22244-7

Yang, G., Pan, F., and Gan, W.-B. (2009). Stably maintained dendritic spines are associated with lifelong memories. *Nature* 462, 920–924. doi: 10.1038/nature08577

Yang, S., Gao, T., Wang, J., Deng, B., Azghadi, M. R., Lei, T., et al. (2022a). SAM: a unified self-adaptive multicompartmental spiking neuron model for learning with working memory. *Front. Neurosci.* 16, 850945. doi: 10.3389/fnins.2022.850945

Yang, S., Linares-Barranco, B., and Chen, B. (2022b). Heterogeneous ensemble-based spike-driven few-shot online learning. *Front. Neurosci.* 16, 850932. doi: 10.3389/fnins.2022.850932

Yang, S., Tan, J., and Chen, B. (2022c). Robust spike-based continual meta-learning improved by restricted minimum error entropy criterion. *Entropy* 24, 455. doi: 10.3390/e24040455

Yger, P., Stimberg, M., and Brette, R. (2015). Fast learning with weak synaptic plasticity. *J. Neurosci.* 35, 13351–13362. doi: 10.1523/JNEUROSCI.0607-15.2015

Zhang, H., Cisse, M., Dauphin, Y. N., and Lopez-Paz, D. (2017). mixup: beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*. doi: 10.48550/arXiv.1710.09412

Zhou, Z., and Firestone, C. (2019). Humans can decipher adversarial images. *Nat. Commun.* 10, 1–9. doi: 10.1038/s41467-019-08931-6

frontiers | Frontiers in Computational Neuroscience

TYPE Original Research
PUBLISHED 16 November 2022
DOI 10.3389/fncom.2022.1037976

OPEN ACCESS

EDITED BY
Yuqi Han,
Tsinghua University, China

REVIEWED BY
Rachmad Vidya Wicaksana Putra,
Vienna University of Technology, Austria
Chenglong Zou,
Peking University, China
Oliver Rhodes,
The University of Manchester, United Kingdom
Federico Corradi,
Eindhoven University of Technology, Netherlands

*CORRESPONDENCE
Nicolas Skatchkovsky
nicolas.skatchkovsky@kcl.ac.uk

RECEIVED 06 September 2022
ACCEPTED 26 October 2022
PUBLISHED 16 November 2022

CITATION
Skatchkovsky N, Jang H and Simeone O (2022) Bayesian continual learning *via* spiking neural networks. *Front. Comput. Neurosci.* 16:1037976. doi: 10.3389/fncom.2022.1037976

Bayesian continual learning *via* spiking neural networks

Nicolas Skatchkovsky[1]*, Hyeryung Jang[2] and Osvaldo Simeone[1]

[1]King's Communication, Learning and Information Processing (KCLIP) Lab, Department of Engineering, King's College London, London, United Kingdom, [2]Department of Artificial Intelligence, Dongguk University, Seoul, South Korea

Among the main features of biological intelligence are energy efficiency, capacity for continual adaptation, and risk management *via* uncertainty quantification. Neuromorphic engineering has been thus far mostly driven by the goal of implementing energy-efficient machines that take inspiration from the time-based computing paradigm of biological brains. In this paper, we take steps toward the design of neuromorphic systems that are capable of adaptation to changing learning tasks, while producing well-calibrated uncertainty quantification estimates. To this end, we derive online learning rules for spiking neural networks (SNNs) within a Bayesian continual learning framework. In it, each synaptic weight is represented by parameters that quantify the current epistemic uncertainty resulting from prior knowledge and observed data. The proposed online rules update the distribution parameters in a streaming fashion as data are observed. We instantiate the proposed approach for both real-valued and binary synaptic weights. Experimental results using Intel's Lava platform show the merits of Bayesian over frequentist learning in terms of capacity for adaptation and uncertainty quantification.

KEYWORDS

spiking neural networks, Bayesian learning, neuromorphic learning, neuromorphic hardware, artificial intelligence

1. Introduction

Recent advances in machine learning and artificial intelligence systems have been largely driven by a pursuit of accuracy *via* resource-intensive pattern recognition algorithms run in a *train-and-then-deploy* fashion. In stark contrast, neuroscience paints a picture of intelligence that revolves around *continual adaptation*, *uncertainty quantification*, and resource budgeting (allostasis) for the parsimonious processing of *event-driven* information (Doya et al., 2007; Friston, 2010; Feldman Barrett, 2021; Hawkins, 2021). Taking inspiration from neuroscience, over the last decade, neuromorphic engineering has pursued the goal of implementing energy-efficient machines that process information *with* time *via* sparse inter-neuron binary signals—or *spikes* (Davies et al., 2021). The main aim of this paper is to introduce algorithmic solutions to endow neuromorphic models, namely spiking neural networks (SNNs), with the capacity for adaptation to changing learning tasks, while ensuring the reliable quantification of uncertainty of the model's decisions.

1.1. Managing uncertainty *via* Bayesian learning

Training algorithms for SNNs have been overwhelmingly derived by following the *frequentist* approach which consists in minimizing the training loss with respect to the model parameter vector (Shrestha and Orchard, 2018; Zenke and Ganguli, 2018; Bellec et al., 2020; Kaiser et al., 2020). This is partly motivated by the dominance of frequentist learning, and associated software tools, in the literature on deep learning for conventional artificial neural networks (ANNs). Frequentist learning is well justified when enough data are available to make the training loss a good empirical approximation of the underlying population loss (Clayton, 2021). When this condition is not satisfied, while the model's average accuracy may be satisfactory on test data, the decisions made by the trained model can be badly calibrated, often resulting in overconfident predictions (Nguyen et al., 2015; Guo et al., 2017). The problem is particularly significant for decisions made on test data that differ significantly from the data observed during training—a common occurrence for applications such as self-driving vehicles. Furthermore, the inability of frequentist learning to account for uncertainty limits its capacity to adapt to new tasks while retaining the capacity to operate on previous tasks (Ebrahimi et al., 2020).

The main cause of the poor calibration of frequentist learning is the selection of a single parameter vector, which disregards any uncertainty on the best model to use for a certain task due to the availability of limited data. A more principled approach that has the potential to properly account for such *epistemic uncertainty*, i.e., for uncertainty related to the availability of limited data, is given by *Bayesian learning* (Jaynes, 2003) and by its generalized form known as *information risk minimization* (see, e.g., Zhang, 2006; Guedj, 2019; Knoblauch et al., 2019; Jose and Simeone, 2021; Simeone, 2022). Bayesian learning maintains a *distribution* over the model parameter vector that represents the partial information available to the learner. This way, Bayesian models can provide well-calibrated decisions, which quantify accurately the associated degree of uncertainty and can be used to detect out-of-distribution inputs (Daxberger and Hernández-Lobato, 2019). In the self-driving example provided earlier, the vehicle may hand back control to the driver when the certainty of its decision is below a certain threshold.

Bayesian reasoning is at the core of the *Bayesian brain* hypothesis in neuroscience, according to which biological brains constantly update an internal model of the world in an attempt to minimize their information-theoretic surprise. This hypothesis is formalized by the free energy principle, which measures surprise in terms of a variational free energy (Friston, 2012). In this context, synaptic plasticity has been hypothesized to be well-modeled as Bayesian learning, which keeps track of the distributions of synaptic weights over time (Aitchison et al., 2021).

In the present paper, we propose (generalized) Bayesian learning rules for SNNs with binary and real-valued synaptic weights that can adapt over time to changing learning tasks.

1.2. Related work

Bayesian learning, and its application to deep ANNs, typically labeled as *Bayesian deep learning*, is receiving increasing attention in the literature. We refer to the following work for a recent overview (Wang and Yeung, 2020). Natural gradient descent rule known as the *Bayesian learning rule* was introduced in Khan and Lin (2017), then applied in Meng et al. (2020) to train binary ANNs, and to a variety of other scenarios in Khan and Rue (2021). Khan and Rue (2021) demonstrates that the Bayesian learning rule recovers many state-of-the-art machine learning algorithms in a principled fashion. We also point to the Kreutzer et al. (2020) that explores the use of natural gradient descent for frequentist learning in spiking neurons.

As mentioned, the choice of a Bayesian learning framework is in line with the importance of the Bayesian brain hypothesis in computational neurosciences (Friston, 2012). The recent Aitchison et al. (2021) explores a Bayesian paradigm to model biological synapses as an explanation of the capacity of the brain to perform learning in the presence of noisy observations. A Bayesian approach to neural plasticity was previously proposed for synaptic sampling, by modeling synaptic plasticity as sampling from a posterior distribution (Kappel et al., 2015). Apart from the conference version (Jang et al., 2021) of the present work, this paper is the first to explore the definition of Bayesian learning and Bayesian continual learning rules for general SNNs adopting the standard spike response model [SRM, see, e.g., (Gerstner and Kistler, 2002)].

Continual learning is a key area of machine learning research, which is partly motivated by the goal of understanding how biological brains maintain previously acquired skills while adding new capabilities. Unlike traditional machine learning, whereby one performs training based on a single data source, in continual learning, several datasets, corresponding to different tasks, are sequentially presented to the learner. A challenge in continual learning is the ability of the learning algorithm to perform competitively on previous tasks after training on the subsequently observed datasets. In this context, *catastrophic forgetting* indicates the situation in which performance drops sharply on previously encountered tasks after learning new ones. Many continual learning techniques follow the principle of preserving synaptic connections that are deemed important to perform well on previously learned tasks *via* a regularization of the learning objective (Kirkpatrick et al., 2017; Zenke et al., 2017). Bayesian approaches have also been proposed for this purpose, whereby priors are selected as the posterior evaluated on the previous task to prevent the new posterior distribution from deviating too much from

learned states. Biological mechanisms are explicitly leveraged in works such as Laborieux et al. (2021) and Soures et al. (2021), which combine a variety of neural mechanisms to obtain state-of-the-art performance for SNNs on standard continual learning benchmarks. Putra and Shafique (2022) also proposes a continual learning algorithm for SNNs in an unsupervised scenario by assuming limited precision for the weights. In the present paper, we demonstrate how Bayesian learning allows obtaining similar biologically inspired features by following a principled objective grounded in information risk minimization.

Traditionally, training of SNNs has relied on biologically realistic Hebbian rules, among which spike-timing dependent plasticity (STDP) is the most popular. STDP modulates the synaptic weight between two neurons based on the firing times of both neurons. A long-term potentiation (i.e., an increase in the weight) of the synapse occurs when the pre-synaptic neuron spikes right before the post-synaptic neuron, while long-term depression (i.e., a decrease in the weight) of a synapse happens when the pre-synaptic neuron spikes after the post-synaptic neuron. STDP implements a form of unsupervised learning, and can be leveraged to perform tasks such as clustering, while also supporting continual learning (Vaila et al., 2019).

Supervised learning based on the minimization of the training loss is challenging in SNNs due to the activation function of spiking neurons, the derivative of which is always zero, except at the spike time, where it is not differentiable. Modern training algorithms (Zenke and Ganguli, 2018; Bellec et al., 2020; Kaiser et al., 2020) overcome this difficulty through the use of *surrogate gradients*, i.e., by replacing the true derivative with that of a well-defined differentiable function (Neftci et al., 2019). An alternative approach, reviewed in Jang et al. (2019), is to view the SNN as a probabilistic model whose likelihood can be directly differentiated. Further extensions of the probabilistic modeling approach and associated training rules are presented in Jang and Simeone (2022) and Jang et al. (2020b).

An application of Bayesian principles to SNNs has first been proposed in the conference version of this paper (Jang et al., 2021). Jang et al. (2021) focuses on SNNs with binary synaptic weights and offline learning, presenting limited experimental results. In contrast, the current paper provides all the necessary background, including frequentist learning; it covers frequentist and Bayesian continual learning; and it provides extensive experimental results on a variety of tasks.

1.3. Main contributions

In this work, we derive online learning rules for SNNs within a Bayesian continual learning framework. In it, each synaptic weight is represented by parameters that quantify the current epistemic uncertainty associated with prior knowledge and data observed thus far. Bayesian methods are key to handling uncertainty over time, providing the model knowledge of what is to be retained, and what can be forgotten (Ebrahimi et al., 2020). The main contributions are as follows.

i) We introduce general frameworks for the definition of single-task and continual Bayesian learning problems for SNNs that are based on information risk minimization and variational inference. Following the desiderata formulated in Farquhar and Gal (2019a), we focus on the standard formulation of continual learning in which there exist clear demarcations between subsequent tasks, but the learner is unaware of the identity of the current task. For example, in the typical example of an autonomous vehicle navigating in several environments, the vehicle may be aware that it is encountering a new terrain, while being a priori unaware of the type of new terrain. Furthermore, the model is not modified between tasks, and tasks may be encountered more than once;

ii) We instantiate the general Bayesian learning frameworks for SNNs with real-valued synapses. To this end, we adopt a Gaussian variational distribution for the synaptic weights, and demonstrate learning rules that can adapt the parameters of the weight distributions online. This choice of variational posterior has been previously explored for ANNs, and can yield state-of-the-art performance on real-life datasets (Osawa et al., 2019);

iii) We then introduce Bayesian single-task and continual learning rules for SNNs with binary weights, with the main goal of supporting more efficient hardware implementations (Courbariaux et al., 2016; Rastegari et al., 2016), including platforms based on beyond-CMOS memristors (Mehonic et al., 2020);

iv) Through experiments on both synthetic and real neuromorphic datasets, we demonstrate the advantage of the Bayesian learning paradigm in terms of accuracy and calibration for both single-task and continual learning. As neuromorphic algorithms are designed to be run on dedicated hardware, we run the experiments using Intel's Lava software emulator platform (Intel Corporation, 2021), accounting for the limited precision of synaptic weights in hardware.

2. Methods

We first introduce the adopted SNN model, namely the standard spike response model (SRM), before giving a short overview of frequentist, Bayesian, continual, and biologically inspired learning. We then detail learning rules for offline and continual frequentist learning, and derive associated online Bayesian learning rules.

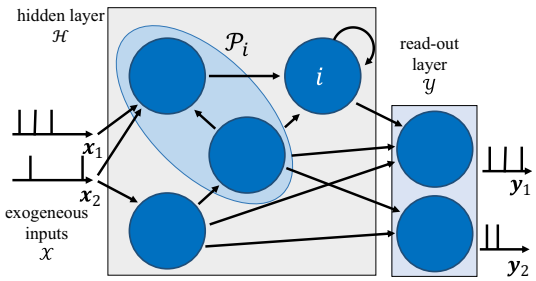


FIGURE 1
Illustration of the internal architecture of an SNN. The behavior of neurons in the read-out layer is guided by the training data, while that of neurons in the hidden layer is adjusted to fit the data. The blue shaded area represents the set of pre-synaptic neurons $\mathcal{P}_i$ to neuron i.

2.1. SNN model

2.1.1. Spike response model

The architecture of an SNN is defined by a network of spiking neurons connected over an arbitrary graph, which possibly includes (directed) cycles. As illustrated in Figure 1, the directed graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ is described by a set $\mathcal{N}$ of nodes, representing the neurons, and by a set $\mathcal{E}$ of directed edges $i \to j$ with $i \neq j \in \mathcal{N}$, representing synaptic connections.

Focusing on a discrete-time implementation, each spiking neuron $i \in \mathcal{N}$ produces a binary value $s_{i,t} \in \{0, 1\}$ at discrete time $t = 1, 2, \ldots$, with "1" denoting the firing of a spike. We collect in an $|\mathcal{N}| \times 1$ vector $\boldsymbol{s}_t = (s_{i,t} : i \in \mathcal{N})$ the spikes emitted by all neurons $\mathcal{N}$ at time t, and denote by $\boldsymbol{s}^t = (\boldsymbol{s}_1, \ldots, \boldsymbol{s}_t)$ the spike sequences of all neurons up to time t. Without loss of generality, we consider time-sequences of length T, and write $\boldsymbol{s} := \boldsymbol{s}^T$. Each neuron i receives input spike signals $\{s_{j,t}\}_{j\in\mathcal{P}_i} = \boldsymbol{s}_{\mathcal{P}_i,t}$ at time t from the set $\mathcal{P}_i = \{j \in \mathcal{N} : (j \to i) \in \mathcal{E}\}$ of parent, or pre-synaptic, neurons, which are connected to neuron i *via* directed links in the graph $\mathcal{G}$. With some abuse of notations, this set is taken to include also exogeneous input signals.

Each neuron i maintains a scalar analog state variable $u_{i,t}$, known as the *membrane potential*. Mathematically, neuron i outputs a binary signal $s_{i,t}$, or spike, at time t when the membrane potential $u_{i,t}$ is above a threshold ϑ, i.e.,

$$s_{i,t} = \Theta(u_{i,t} - \vartheta), \tag{1}$$

with $\Theta(\cdot)$ being the Heaviside step function and ϑ being the fixed firing threshold. Following the standard discrete-time SRM (Gerstner and Kistler, 2002), the membrane potential $u_{i,t}$ is obtained by summing filtered contributions from pre-synaptic neurons in set $\mathcal{P}_i$ and from the neuron's own output. In particular, the membrane potential evolves as

$$u_{i,t} = \sum_{j\in\mathcal{P}_i} w_{ij}(\alpha_t * s_{j,t}) - \beta_t * s_{i,t}, \tag{2}$$

where w_{ij} is a learnable synaptic weight from pre-synaptic neuron $j \in \mathcal{P}_i$ to post-synaptic neuron i; and we collect in vector $\boldsymbol{w} = \{\boldsymbol{w}_i\}_{i\in\mathcal{N}}$ the model parameters, with $\boldsymbol{w}_i := \{w_{ij}\}_{j\in\mathcal{P}_i}$ being the synaptic weights for each neuron i. We have denoted as α_t and β_t the spike responses of synapses and somas, respectively; while $*$ denotes the convolution operator $f_t * g_t = \sum_{\delta>0} f_\delta g_{t-\delta}$. When implemented with autoregressive filters, the SRM is equivalent to leaky integrate-and-fire (LIF) neuron model (Gerstner and Kistler, 2002; Kaiser et al., 2020). The techniques developed in this work can be directly generalized to other, more complex, neuron models, such as resonate-and-fire (Izhikevich, 2001), but we leave an investigation of this point to future work.

2.1.2. Real-valued and binary-valued synapses

In this paper, we will consider two implementations of the SRM introduced in the previous subsection. In the first, the synaptic weights in vector $\boldsymbol{w}$ are real-valued, i.e., $w_{ij} \in \mathbb{R}$, with possibly limited resolution, as dictated by deployment on neuromorphic hardware (see Section 3). In contrast, in the second implementation, the weights are binary, i.e., $w_{ij} \in \{+1, -1\}$. The advantages of models with binary-valued synapses, which we call binary SNNs, include a reduced complexity for the computation of the membrane potential $u_{i,t}$ in Equation (2). Furthermore, binary SNNs are particularly well suited for implementations on chips with nanoscale components that provide discrete conductance levels for the synapses (Mehonic et al., 2020). In this regard, we note that the methods described in this paper can be generalized to models with weights having any discrete number of values.

2.2. Frequentist vs. Bayesian learning

With traditional frequentist learning, the vector of synaptic weights $\boldsymbol{w}$ is optimized by minimizing a training loss. The training loss is adopted as a proxy for the population loss, i.e., for the loss averaged over the true, unknown, distribution of the data. Therefore, frequentist learning disregards the inherent uncertainty caused by the availability of limited training data, which causes the training loss to be a potentially inaccurate estimate of the population loss. As a result, frequentist learning is known to potentially yield poorly calibrated, and overconfident decisions for ANNs (Nguyen et al., 2015).

In contrast, as seen in Figure 2A, Bayesian learning optimizes over a distribution $q(\boldsymbol{w})$ in the space of the synaptic weight vector $\boldsymbol{w}$. The distribution $q(\boldsymbol{w})$ captures the *epistemic* uncertainty induced by the lack of knowledge of the true

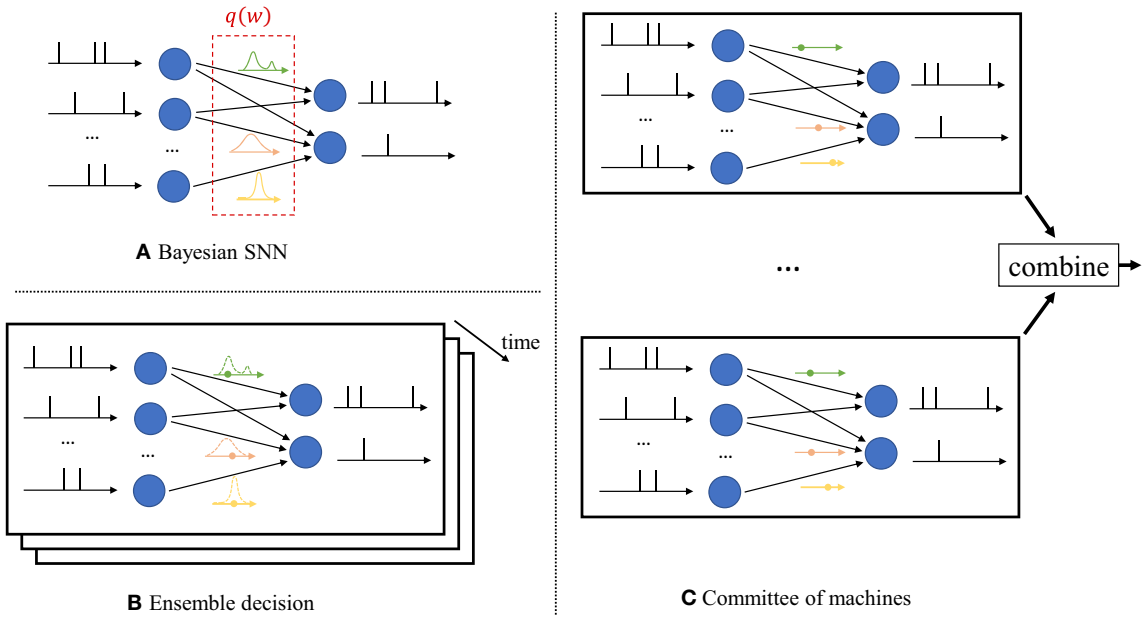


FIGURE 2
Illustration of Bayesian learning in an SNN. **(A)** In a Bayesian SNN, the synaptic weights $\boldsymbol{w}$ are assigned a joint distribution $q(\boldsymbol{w})$, often simplified as a product distribution across weights. **(B)** An ensemble decision is obtained by sampling several times from the distribution $q(\boldsymbol{w})$, and by averaging the predictions of the sampled models. Sampling is done independently for each new input. **(C)** In a committee machine, the weights for several models are drawn only once from $q(\boldsymbol{w})$, and the same models are run in parallel for any new input.

distribution of the data. This is done by assigning similar values of $q(\boldsymbol{w})$ to model parameters that fit equally well the data, while also being consistent with prior knowledge. As a consequence, Bayesian learning is known to produce better calibrated decisions, i.e., decisions whose associated confidence better reflects the actual accuracy of the decision (Guo et al., 2017). Furthermore, models trained *via* Bayesian learning can better detect out-of-distribution data, i.e., data that is not covered by the distribution of the training set (Daxberger and Hernández-Lobato, 2019; Kristiadi et al., 2020).

Once distribution $q(\boldsymbol{w})$ is optimized *via* Bayesian learning, at inference time a decision on any new test input is made by averaging the decisions of multiple models, with each being drawn from the distribution $q(\boldsymbol{w})$. The average over multiple models can be realized in one of two ways.

i) Ensemble predictor: Given a test input, as seen in Figure 2B, one draws a new synaptic weight vector several times from the distribution $q(\boldsymbol{w})$, and an *ensemble* decision is obtained by averaging the decisions produced by running the SNN with each sampled weight vector;

ii) Committee machine: Alternatively, one can sample a number of realizations from the distribution $q(\boldsymbol{w})$ that are kept fixed and reused for all test inputs. This solution foregoes the sampling step at inference time as illustrated in Figure 2C. However, the approach generally requires a larger memory to store all samples $\boldsymbol{w}$ to be used for inference, while the ensemble predictor can make decisions using different weight vectors $\boldsymbol{w} \sim q(\boldsymbol{w})$ sequentially over time.

2.3. Offline vs. continual learning

Offline learning denotes the typical situation where the system is presented with a single training dataset $\mathcal{D}$, which is used to measure a training loss. In offline learning, optimization of the training loss is carried out once and for all, resulting in a synaptic weight vector $\boldsymbol{w}$ or in a distribution $q(\boldsymbol{w})$ for frequentist or Bayesian learning, respectively. Offline learning is hence, by construction, unable to adapt to changing conditions, and it is deemed to be a poor representation of how intelligence works in biological organisms (Kudithipudi and Aguilar-Simon, 2022).

In continual learning, the system is sequentially presented datasets $\mathcal{D}^{(1)}, \mathcal{D}^{(2)}, \ldots$ corresponding to distinct, but related, learning tasks, where each task is selected, possibly with replacement, from a pool of tasks, and its identity is unknown to the system. For each task k, the system is given a training set $\mathcal{D}^{(k)}$, and its goal is to learn to make predictions that generalize well on the new task, while causing minimal loss of accuracy on previous tasks $1, \ldots, k-1$. In frequentist continual learning, the model parameter vector $\boldsymbol{w}$ is updated as data from successive tasks is collected. Conversely, in Bayesian continual learning, the distribution $q(\boldsymbol{w})$ is updated over time as illustrated in Figure 3. The updates should be sufficient to address the needs of the

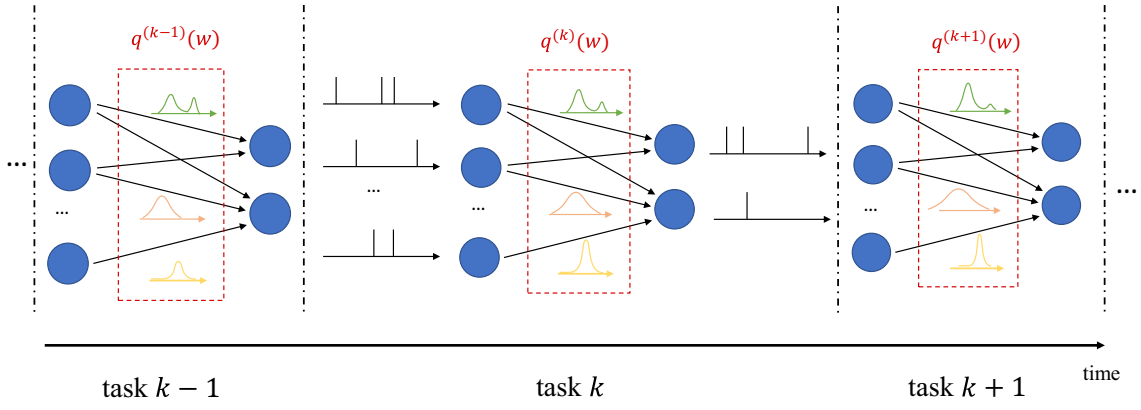


FIGURE 3
Illustration of Bayesian continual learning: the system is successively presented with similar, but different, tasks. Bayesian learning allows the model to retain information about previously learned information.

new task, while not disrupting performance on previous tasks, operating on a *stability-plasticity trade-off*.

2.4. Biological principles of learning

Many existing works on continual learning draw their inspiration from the mechanisms underlying the capability of biological brains to carry out life-long learning (Soures et al., 2021; Kudithipudi and Aguilar-Simon, 2022). Learning is believed to be achieved in biological systems by modulating the strength of synaptic links. In this process, a variety of mechanisms are at work to establish short-to intermediate-term and long-term memory for the acquisition of new information over time (Kandel et al., 2014). These mechanisms operate at different time and spatial scales.

One of the best understood mechanisms, *long-term potentiation*, contributes to the management of long-term memory through the consolidation of synaptic connections (Morris, 2003; Malenka and Bear, 2004). Once established, these are rendered resistant to disruption by changing their capacity to change *via metaplasticity* (Abraham and Bear, 1996; Finnie and Nader, 2012). As a related mechanism, return to a base state is ensured after exposition to small, noisy changes by *heterosynaptic plasticity*, which plays a key role in ensuring the stability of neural systems (Chistiakova et al., 2014). *Neuromodulation* operates at the scale of neural populations to respond to particular events registered by the brain (Marder, 2012). Finally, *episodic replay* plays a key role in the maintenance of long-term memory, by allowing biological brains to re-activate signals seen during previous active periods when inactive (i.e., sleeping) (Kudithipudi and Aguilar-Simon, 2022).

2.5. Frequentist offline learning

We now review frequentist offline training algorithms for SNNs, under the SRM model described in Section 2.1.1. This will provide the necessary background for Bayesian learning and its continual version, described in Sections 2.6 and 2.8, respectively.

2.5.1. Empirical risk minimization

To start, as illustrated in Figure 1, we divide the set $\mathcal{N}$ of neurons of the SNN into two subsets $\mathcal{Y}$ and $\mathcal{H}$ with $\mathcal{N} = \mathcal{Y} \cup \mathcal{H}$: a set of read-out, or output, neurons $\mathcal{Y}$ and a set of hidden neurons $\mathcal{H}$. The set of exogeneous inputs is defined as $\mathcal{X}$. We focus on supervised learning, in which a dataset $\mathcal{D}$ is given by $|\mathcal{D}|$ pairs $(\boldsymbol{x}, \boldsymbol{y})$ of signals generated from an unknown distribution $p(\boldsymbol{x}, \boldsymbol{y})$, with $\boldsymbol{x}$ being exogeneous input signals, one for each element of the set $\mathcal{X}$, and $\boldsymbol{y}$ the corresponding desired output signals. Both $\boldsymbol{x}$ and $\boldsymbol{y}$ are vector sequences of length T, with $\boldsymbol{x}$ comprising $|\mathcal{X}|$ signals, and $\boldsymbol{y}$ including $|\mathcal{Y}|$ signals. Each output samples $y_{m,t}$ in $\boldsymbol{y}$ dictates the desired behavior of the mth neuron in the read-out set $\mathcal{Y}$. The sequences in $\boldsymbol{x}$ and $\boldsymbol{y}$ can generally take arbitrary real values (see Section 3 for specific examples).

In frequentist learning, the goal is to minimize the training loss over the parameter vector $\boldsymbol{w}$ using the training dataset $\mathcal{D} = \{(\boldsymbol{x}, \boldsymbol{y})\}$. To elaborate, we define the loss $\mathcal{L}_{\boldsymbol{x},\boldsymbol{y}}(\boldsymbol{w})$ measured with respect to a data $(\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{D}$ as the error between the reference signals $\boldsymbol{y}$ and the output spiking signals produced by the SNN with parameters $\boldsymbol{w}$, given the input $\boldsymbol{x}$. Accordingly, the loss is written as a sum over time instants $t = 1, \dots, T$ and over the $|\mathcal{Y}|$ read-out neurons as

$$\mathcal{L}_{\boldsymbol{x},\boldsymbol{y}}(\boldsymbol{w}) = \sum_{t=1}^{T} \mathcal{L}_{\boldsymbol{x}^t,\boldsymbol{y}_t}(\boldsymbol{w}) = \sum_{t=1}^{T} \sum_{m \in \mathcal{Y}} L\big(y_{m,t}, f_m(\boldsymbol{w}, \boldsymbol{x}^t)\big), \quad (3)$$

where function $L\big(y_{m,t}, f_m(\boldsymbol{w}, \boldsymbol{x}^t)\big)$ is a local loss measure comparing the target output $y_{m,t}$ of neuron m at time t and the actual output $f_m(\boldsymbol{w}, \boldsymbol{x}^t)$ of the same neuron. The notations $f_m(\boldsymbol{w}, \boldsymbol{x}^t)$ and $\mathcal{L}_{\boldsymbol{x}^t, \boldsymbol{y}_t}(\boldsymbol{w})$ are used as a reminder that the output of the SNN and the corresponding loss at time t generally depend on the input $\boldsymbol{x}^t$ up to time t, and on the target output $\boldsymbol{y}_t$ at time t. Specifically, the notation $f_m(\boldsymbol{w}, \boldsymbol{x}^t)$ makes it clear that the output of neuron $m \in \mathcal{Y}$ is produced with the model parameters $\boldsymbol{w}$ from exogeneous input $\boldsymbol{x}^t$, consisting of all input samples up to time t, using the SRM (Equations 1, 2).

The training loss $\mathcal{L}_{\mathcal{D}}(\boldsymbol{w})$ is an empirical estimate of the population loss based on the data samples in the training dataset $\mathcal{D}$, and is given as

$$\mathcal{L}_{\mathcal{D}}(\boldsymbol{w}) = \frac{1}{|\mathcal{D}|} \sum_{(\boldsymbol{x},\boldsymbol{y}) \in \mathcal{D}} \mathcal{L}_{\boldsymbol{x},\boldsymbol{y}}(\boldsymbol{w}). \tag{4}$$

Frequentist learning addresses the empirical risk minimization (ERM) problem

$$\min_{\boldsymbol{w}} \mathcal{L}_{\mathcal{D}}(\boldsymbol{w}). \tag{5}$$

Problem (Equation 5) cannot be directly solved using standard gradient-based methods since: *(i)* the spiking mechanism (Equation 1) is not differentiable in $\boldsymbol{w}$ due to the presence of the threshold function $\Theta(\cdot)$; and *(ii)* in the case of binary SNNs, the domain of the weight vector $\boldsymbol{w}$ is the discrete set of binary values.

To tackle the former problem, as detailed in Section 2.5.2, surrogate gradients (SG) methods replace the derivative of the threshold function $\Theta(\cdot)$ in Equation (1) with a suitable differentiable approximation (Neftci et al., 2019). In a similar manner, for the latter issue, optimization over binary weights is conventionally done *via* the straight-through estimator (STE) (Bengio et al., 2013; Jang et al., 2021), which is covered in Section 2.5.3.

2.5.2. Surrogate gradient

As discussed in the previous subsections, the gradient $\nabla_{\boldsymbol{w}} \mathcal{L}_{\boldsymbol{x},\boldsymbol{y}}(\boldsymbol{w})$ is typically evaluated *via* SG methods. SG techniques approximate the Heaviside function $\Theta(\cdot)$ in Equation (1) when computing the gradient $\nabla_{\boldsymbol{w}} \mathcal{L}_{\boldsymbol{x},\boldsymbol{y}}(\boldsymbol{w})$. Specifically, the derivative $\Theta'(\cdot)$ is replaced with the derivative of a differentiable surrogate function, such as rectifier or sigmoid. For example, with a sigmoid surrogate, given by function $\sigma(x) = (1 + e^{-x})^{-1}$, we have $\partial s_{i,t} / \partial u_{i,t} \approx \sigma'(u_{i,t} - \vartheta)$, with derivative $\sigma'(x) = \sigma(x)(1 - \sigma(x))$. Using the loss decomposition in Equation (3), the partial derivative of the training loss $\mathcal{L}_{\boldsymbol{x}^t, \boldsymbol{y}_t}(\boldsymbol{w})$ at each time instant t with respect to a synaptic weight w_{ij} can be accordingly approximated as

$$\frac{\partial \mathcal{L}_{\boldsymbol{x}^t, \boldsymbol{y}_t}(\boldsymbol{w})}{\partial w_{ij}} \approx \underbrace{\sum_{m \in \mathcal{Y}} \frac{\partial L(y_{m,t}, f_{m,t})}{\partial s_{i,t}}}_{e_{i,t}} \cdot \underbrace{\frac{\partial s_{i,t}}{u_{i,t}}}_{\sigma'(u_{i,t} - \vartheta)} \cdot \underbrace{\frac{\partial u_{i,t}}{\partial w_{ij}}}_{\alpha_t * s_{j,t}}, \tag{6}$$

where the first term $e_{i,t}$ is the derivative of the loss at time t with respect to the output $s_{i,t}$ of post-synaptic neuron i at time t; and the third term can be directly computed from Equation (2) as the filtered pre-synaptic trace of neuron j. For simplicity of notation, we have defined $f_{m,t} := f_m(\boldsymbol{w}, \boldsymbol{x}^t)$ and omitted the explicit dependence of $s_{i,t}$ and $u_{i,t}$ on exogeneous inputs $\boldsymbol{x}^t$ and synaptic weights $\boldsymbol{w}$. The second term is the source of the approximation, as the derivative of the threshold function $\Theta'(\cdot)$ from Equation (1), which is zero almost everywhere, is replaced using the derivative of the sigmoid function.

At every time instant $t = 1, \dots, T$, using Equation (6), the online update is obtained *via* stochastic gradient descent (SGD) as

$$w_{ij,t+1} \leftarrow w_{ij,t} - \eta \cdot \frac{1}{|\mathcal{B}|} \sum_{(\boldsymbol{x},\boldsymbol{y}) \in \mathcal{B}} \frac{\partial \mathcal{L}_{\boldsymbol{x}^t, \boldsymbol{y}_t}(\boldsymbol{w}_t)}{\partial w_{ij}}, \tag{7}$$

where $\eta > 0$ is a learning rate, and $\mathcal{B} \subseteq \mathcal{D}$ is a mini-batch of examples $(\boldsymbol{x}, \boldsymbol{y})$ from the training dataset. Note that the sequential implementation of the update (Equation 7) over time t requires running a number of copies of the SNN model equal to the size of the mini-batch $\mathcal{B}$. In fact, each input $\boldsymbol{x}$, with $(\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{B}$, generally causes the spiking neurons to follow distinct trajectories in the space of the membrane potentials. Henceforth, when referring to online learning rules, we will implicitly assume that parallel executions of the SNN are possible when the mini-batch size is larger than 1.

The weight update in the direction of the negative gradients in Equation (7) implements a standard *three-factor* rule. Three-factor rules generalize two-factor Hebbian updates such as STDP (Gerstner et al., 2018), and can be implemented on hardware with similar complexity (Zenke and Ganguli, 2018; Kaiser et al., 2020; Stewart et al., 2020). In fact, the partial derivative (Equation 6) can be written as

$$\frac{\partial \mathcal{L}_{\boldsymbol{x}^t, \boldsymbol{y}_t}(\boldsymbol{w})}{\partial w_{ij}} = \underbrace{e_{i,t}}_{\text{error signal}} \cdot \underbrace{\sigma'(u_{i,t} - \vartheta)}_{\text{post}_{i,t}} \cdot \underbrace{\big(\alpha_t * s_{j,t}\big)}_{\text{pre}_{j,t}}, \tag{8}$$

where we distinguish three terms. The first is the per-neuron error signal $e_{i,t}$, which can be in principle computed *via* backpropagation through time (Huh and Sejnowski, 2018). In practice, this term is approximated, e.g. *via* local signals (Bellec et al., 2020), or *via* random projections (Kaiser et al., 2020). The latter technique has previously been likened to the biological mechanisms behind short-term memory (Zou et al., 2022). We will discuss a specific implementation in Section 3.2. The second contribution is given by the local post-synaptic term $\sigma'(u_{i,t} - \vartheta)$, which measures the

current sensitivity to changes in the membrane potential of the neuron i. Finally, the last term is the local pre-synaptic trace $\alpha_t * s_{j,t}$ that depends on the activity of the neuron j.

2.5.3. Straight-through estimator

As mentioned in Section 2.5.1, optimization over binary weights can be carried out using STE (Bengio et al., 2013; Jang et al., 2021), which maintains latent, real-valued weights to compute gradients during training. Binary weights, obtained *via* quantization of the real-valued latent weights, are used as the next iterate. To elaborate, in addition to the binary weight vector $\boldsymbol{w} \in \{+1, -1\}^{|\boldsymbol{w}|}$, we define the real-valued weight vector $\boldsymbol{w}^{\mathrm{r}} \in \mathbb{R}^{|\boldsymbol{w}|\times 1}$. We use $|\boldsymbol{w}|$ to denote the size of vector $\boldsymbol{w}$. With STE, gradients are estimated by differentiating over the real-valued latent weights $\boldsymbol{w}^{\mathrm{r}}$, instead of discrete binary weights $\boldsymbol{w}$, to compute the gradient $\nabla_{\boldsymbol{w}^{\mathrm{r}}} \mathcal{L}_{\boldsymbol{x}^t, \boldsymbol{y}_t}(\boldsymbol{w}^{\mathrm{r}})|_{\boldsymbol{w}^{\mathrm{r}}=\boldsymbol{w}}$. The technique can be naturally combined with the SG method, detailed in Section 2.5.2, to obtain the gradients with respect to the real-valued latent weights.

The training algorithm proceeds iteratively by selecting a mini-batch $\mathcal{B}$ of examples $(\boldsymbol{x}, \boldsymbol{y})$ from the training dataset $\mathcal{D}$ at each iteration as in Equation (7). Accordingly, the real-valued latent weight vector $\boldsymbol{w}^{\mathrm{r}}$ is updated *via* online SGD as

$$w^{\mathrm{r}}_{ij,t+1} \leftarrow w^{\mathrm{r}}_{ij,t} - \eta \cdot \frac{1}{|\mathcal{B}|} \sum_{(\boldsymbol{x},\boldsymbol{y})\in\mathcal{B}} \frac{\partial \mathcal{L}_{\boldsymbol{x}^t, \boldsymbol{y}_t}(\boldsymbol{w}^{\mathrm{r}}_t)}{\partial w^{\mathrm{r}}_{ij,t}}\bigg|_{w^{\mathrm{r}}_{ij,t}=w_{ij,t}}, \tag{9}$$

and the next iterate for the binary weights $\boldsymbol{w}$ is obtained by quantization as

$$w_{ij,t+1} = \mathrm{sign}\big(w^{\mathrm{r}}_{ij,t+1}\big), \tag{10}$$

where the sign function is defined as $\mathrm{sign}(x) = +1$ for $x \geq 0$ and $\mathrm{sign}(x) = -1$ for $x < 0$.

2.6. Bayesian offline learning

In this section, we describe the formulation of Bayesian offline learning, and then develop two Bayesian training algorithms for SNNs with real-valued and binary synaptic weights.

2.6.1. Information risk minimization

Bayesian learning formulates the training problem as the optimization of a probability distribution $q(\boldsymbol{w})$ in the space of synaptic weights, which is referred to as the *variational posterior*. To this end, the ERM problem (Equation 5) is replaced by the information risk minimization (IRM) problem

$$\min_{q(\boldsymbol{w})} \Big\{ \mathcal{F}\big(q(\boldsymbol{w})\big) = \mathbb{E}_{q(\boldsymbol{w})}\Big[\mathcal{L}_{\mathcal{D}}(\boldsymbol{w})\Big] + \rho \cdot \mathrm{KL}\big(q(\boldsymbol{w})||p(\boldsymbol{w})\big)\Big\}, \tag{11}$$

where $\rho > 0$ is a "temperature" constant, $p(\boldsymbol{w})$ is an arbitrary prior distribution over synaptic weights, and $\mathrm{KL}(\cdot||\cdot)$ is the Kullback-Leibler divergence

$$\mathrm{KL}(q(\boldsymbol{w})||p(\boldsymbol{w})) = \mathbb{E}_{q(\boldsymbol{w})}\left[\log \frac{q(\boldsymbol{w})}{p(\boldsymbol{w})}\right]. \tag{12}$$

The objective function in IRM problem (Equation 11) is known as (variational) free energy (Jose and Simeone, 2021).

The problem of minimizing the free energy in Equation (11) must strike a balance between fitting the data—i.e., minimizing the first term—and not deviating too much from the reference behavior defined by prior $p(\boldsymbol{w})$—i.e., keeping the second term small. Note that with $\rho = 0$, the IRM problem (Equation 11) reduces to the ERM problem (Equation 5) in the sense that the optimal solution of the IRM problem with $\rho = 0$ is a distribution concentrated at the solution of the ERM problem (assuming that the latter is unique). The KL divergence term in Equation (11) is hence essential to Bayesian learning, and it is formally justified as a regularizing penalty that accounts for epistemic uncertainty due to the presence of limited data in the context of PAC Bayes theory (Zhang, 2006). It can also be used as a model of bounded rationality accounting for the complexity of information processing (Jose and Simeone, 2021).

If no constraints are imposed on the variational posterior $q(\boldsymbol{w})$, the optimal solution of Equation (11) is given by the *Gibbs posterior*

$$q^{\star}(\boldsymbol{w}) = \frac{p(\boldsymbol{w}) \exp\big(-\mathcal{L}_{\mathcal{D}}(\boldsymbol{w})/\rho\big)}{\mathbb{E}_{p(\boldsymbol{w})}\Big[\exp\big(-\mathcal{L}_{\mathcal{D}}(\boldsymbol{w})/\rho\big)\Big]}. \tag{13}$$

Due to the intractability of the normalizing constant in Equation (13), we adopt a mean-field variational inference (VI) approximation that limits the optimization domain for problem (Equation 11) to a class of factorized distributions (see, e.g., Angelino et al., 2016; Simeone, 2022). More specifically, we focus on Gaussian and Bernoulli variational approximations, targeting SNN models with real-valued and binary synaptic weights, respectively, which are detailed in the rest of this section.

2.6.2. Gaussian mean-field variational inference

In this subsection, we derive a Gaussian mean-field VI algorithm that approximately solves the IRM problem (Equation 11) by assuming variational posteriors of the form $q(\boldsymbol{w}) =$

Input: dataset $\mathcal{D}$, learning rate η, temperature parameter ρ, prior $(\boldsymbol{m}_0, \boldsymbol{p}_0)$
Output: learned parameters pair $(\boldsymbol{m}, \boldsymbol{p})$
initialize parameters $(\boldsymbol{m}_1, \boldsymbol{p}_1)$
repeat
  select mini-batch $\mathcal{B} \subseteq \mathcal{D}$
  for each time-step $t = 1, \dots, T$ **do**
    sample weights $\boldsymbol{w}$ as $\boldsymbol{w} \sim \mathcal{N}(\boldsymbol{w}|\boldsymbol{m}_t, \boldsymbol{P}_t^{-1})$.
    for each $(\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{B}$ **do**
      compute the gradient $\nabla_{\boldsymbol{w}} \mathcal{L}_{\boldsymbol{x}^t, \boldsymbol{y}_t}(\boldsymbol{w})$ locally at each synapse using SG (see Section 2.5.2).
    end for
    update the mean and precision parameters $(m_{ij,t}, p_{ij,t})$ for all synapses $(i,j) \in \mathcal{E}$ as

$$p_{ij,t+1} \leftarrow (1 - \eta\rho) \cdot p_{ij,t} + \eta \cdot \left[\frac{1}{|\mathcal{B}|} \sum_{(\boldsymbol{x},\boldsymbol{y}) \in \mathcal{B}} \left(\frac{\partial \mathcal{L}_{\boldsymbol{x}^t, \boldsymbol{y}_t}(\boldsymbol{w})}{\partial w_{ij}} \right)^2 + \rho \cdot p_{ij,0} \right]$$

$$m_{ij,t+1} \leftarrow m_{ij,t} - \eta \cdot p_{ij,t+1}^{-1} \cdot \left[\frac{1}{|\mathcal{B}|} \sum_{(\boldsymbol{x},\boldsymbol{y}) \in \mathcal{B}} \frac{\partial \mathcal{L}_{\boldsymbol{x}^t, \boldsymbol{y}_t}(\boldsymbol{w})}{\partial w_{ij}} - \rho \cdot p_{ij,0} \cdot \left(m_{ij,0} - m_{ij,t} \right) \right].$$

  end for
  set $(\boldsymbol{m}_1, \boldsymbol{p}_1) = (\boldsymbol{m}_T, \boldsymbol{p}_T)$
until convergence

Algorithm 1. Bayesian offline learning with real-valued synapses.

$\mathcal{N}(\boldsymbol{w}|\boldsymbol{m}, \boldsymbol{P}^{-1})$, where $\boldsymbol{m}$ is a mean vector and $\boldsymbol{P}$ is a precision diagonal matrix with positive vector $\boldsymbol{p}$ on the main diagonal. For the $|\boldsymbol{w}| \times 1$ weight vector $\boldsymbol{w}$, the distribution of the parameters $\boldsymbol{w}$ is defined by the $|\boldsymbol{w}| \times 1$ mean vector $\boldsymbol{m}$ and $|\boldsymbol{w}| \times 1$ precision vector $\boldsymbol{p} = \{p_{ij}\}_{(i,j)\in\mathcal{E}}$ with $p_{ij} > 0$ for all $(i,j) \in \mathcal{E}$. This variational model is well suited for real-valued synapses, which can be practically realized to the fixed precision allowed by the hardware implementation (Davies et al., 2018). We choose the prior $p(\boldsymbol{w})$ as the Gaussian distribution $p(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w}|\boldsymbol{m}_0, \boldsymbol{P}_0^{-1})$ with mean $\boldsymbol{m}_0$ and precision matrix $\boldsymbol{P}_0$ with positive diagonal vector $\boldsymbol{p}_0$.

We tackle the IRM problem (Equation 11) with respect to the so-called *variational parameters* $(\boldsymbol{m}, \boldsymbol{p})$ of the Gaussian variational posterior $q(\boldsymbol{w})$ *via* the Bayesian learning rule (Khan and Rue, 2021). The Bayesian learning rule is derived by applying *natural gradient descent* to the variational free energy $\mathcal{F}(q(\boldsymbol{w}))$ in Equation (11). The derivation leverages the fact that the distribution $q(\boldsymbol{w})$ is an exponential-family distribution with natural parameters $\boldsymbol{\lambda} = (\boldsymbol{P}\boldsymbol{m}, -1/2\boldsymbol{P})$, sufficient statistics $\mathbf{T} = (\boldsymbol{w}, \boldsymbol{w}\boldsymbol{w}^T)$ and mean parameters $\boldsymbol{\mu} = (\boldsymbol{m}, \boldsymbol{P}^{-1} + \boldsymbol{m}\boldsymbol{m}^T)$. Updates to the mean $\boldsymbol{m}_t$ and precision $\boldsymbol{p}_t$ parameters at iteration t can be obtained as Osawa et al. (2019) and Khan and Rue (2021).

$$p_{ij,t+1} \leftarrow (1 - \eta\rho) \cdot p_{ij,t} + \eta \cdot \mathbb{E}_{q_t(\boldsymbol{w})} \left[\frac{1}{|\mathcal{B}|} \sum_{(\boldsymbol{x},\boldsymbol{y}) \in \mathcal{B}} \left(\frac{\partial \mathcal{L}_{\boldsymbol{x}^t, \boldsymbol{y}_t}(\boldsymbol{w})}{\partial w_{ij}} \right)^2 + \rho \cdot p_{ij,0} \right] \tag{14}$$

$$m_{ij,t+1} \leftarrow m_{ij,t} - \eta \cdot p_{ij,t+1}^{-1} \cdot \mathbb{E}_{q_t(\boldsymbol{w})} \left[\frac{1}{|\mathcal{B}|} \sum_{(\boldsymbol{x},\boldsymbol{y}) \in \mathcal{B}} \frac{\partial \mathcal{L}_{\boldsymbol{x}^t, \boldsymbol{y}_t}(\boldsymbol{w})}{\partial w_{ij}} - \rho \cdot p_{ij,0} \cdot \left(m_{ij,0} - m_{ij,t} \right) \right] \tag{15}$$

where $\eta > 0$ is a learning rate; $\mathcal{B} \subseteq \mathcal{D}$ is a mini-batch of examples $(\boldsymbol{x}, \boldsymbol{y})$ from the training dataset; and $q_t(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w}|\boldsymbol{m}_t, \boldsymbol{P}_t^{-1})$ is the variational posterior at iteration t with $\boldsymbol{m}_t$ and $\boldsymbol{p}_t$.

In practice, the updates (Equations 14, 15) are estimated by evaluating the expectation over distribution $q_t(\boldsymbol{w})$ *via* one or more randomly drawn samples $\boldsymbol{w} \sim q_t(\boldsymbol{w})$. Furthermore, the gradients $\nabla_{\boldsymbol{w}} \mathcal{L}_{\boldsymbol{x}^t, \boldsymbol{y}_t}(\boldsymbol{w})$ can be approximated using the online SG method described in Section 2.5.2. The overall training algorithm proceeds iteratively by selecting a mini-batch $\mathcal{B} \subseteq \mathcal{D}$ of examples $(\boldsymbol{x}, \boldsymbol{y})$ from the training dataset at each iteration, and is summarized in Algorithm 1. Note that, as mentioned in Section 2.5.2, the implementation of a rule operating with mini-batches requires running $|\mathcal{B}|$ SNN models in parallel, where $|\mathcal{B}|$ is the cardinality of the mini-batch. When this is not possible, the rule can be applied with mini-batches of size $|\mathcal{B}| = 1$.

2.6.3. Bernoulli mean-field variational inference

In this subsection, we turn to the case of binary synaptic weights $w_{ij} \in \{+1, -1\}$. For this setting, we adopt the variational posterior $q(\boldsymbol{w}) = \text{Bern}(\boldsymbol{w}|\boldsymbol{p})$, with

$$q(\boldsymbol{w}) = \prod_{i \in \mathcal{N}} \prod_{j \in \mathcal{P}_i} p_{ij}^{\frac{1+w_{ij}}{2}} (1 - p_{ij})^{\frac{1-w_{ij}}{2}}, \tag{16}$$

where the $|\boldsymbol{w}| \times 1$ vector $\boldsymbol{p} = \{\{p_{ij}\}_{j\in\mathcal{P}_i}\}_{i\in\mathcal{N}}$ defines the variational posterior, with p_{ij} being the probability that synaptic weights w_{ij} equals $+1$.

The variational posterior (Equation 16) can be reparameterized in terms of the mean parameters $\boldsymbol{\mu} = \{\{\mu_{ij}\}_{j\in\mathcal{P}_i}\}_{i\in\mathcal{N}}$ as

$$q(\boldsymbol{w}) = \text{Bern}\left(\boldsymbol{w} \middle| \frac{\boldsymbol{\mu} + \mathbf{1}}{2}\right) \tag{17}$$

by setting $p_{ij} = (\mu_{ij} + 1)/2$, where $\mathbf{1}$ is the all-ones vector. It can also be expressed in terms of the logits, or natural parameters, $\boldsymbol{w}^{\text{r}} = \{\{w_{ij}^{\text{r}}\}_{j\in\mathcal{P}_i}\}_{i\in\mathcal{N}}$ as $q(\boldsymbol{w}) = \text{Bern}(\boldsymbol{w}|\sigma(2\boldsymbol{w}^{\text{r}}))$ by setting

$$w_{ij}^{\text{r}} = \frac{1}{2} \log \left(\frac{p_{ij}}{1 - p_{ij}} \right) = \frac{1}{2} \log \left(\frac{1 + \mu_{ij}}{1 - \mu_{ij}} \right), \tag{18}$$

Input: dataset $\mathcal{D}$, learning rate η, temperature parameter ρ, GS trick parameter τ, logits $\boldsymbol{w}_0^{\mathrm{r}}$ of prior distribution
Output: learned real-valued weights $\boldsymbol{w}^{\mathrm{r}}$
initialize real-valued weights $\boldsymbol{w}_1^{\mathrm{r}}$
repeat
  select mini-batch $\mathcal{B} \subseteq \mathcal{D}$
  for each time-step $t = 1, \dots, T$ **do**
    sample relaxed binary weights as

$$w_{ij} = \tanh\left(\frac{w_{ij,t}^{\mathrm{r}} + \delta_{ij}}{\tau}\right),$$

    with $\delta_{ij} = \frac{1}{2}\log\frac{\epsilon_{ij}}{1-\epsilon_{ij}}$ and $\epsilon_{ij} \overset{\text{i.i.d.}}{\sim} \mathcal{U}(0,1)$ for all $(i,j) \in \mathcal{E}$.
    for each $(\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{B}$ **do**
      compute the gradient $\nabla_{\boldsymbol{w}} \mathcal{L}_{\boldsymbol{x}^t, \boldsymbol{y}_t}(\boldsymbol{w})$ locally at each synapse using SG (see Section 2.5.2).
    end for
    update the real-valued weights $w_{ij,t}^{\mathrm{r}}$ for all synapses $(i,j) \in \mathcal{E}$ as

$$w_{ij,t+1}^{\mathrm{r}} \leftarrow (1-\eta\rho)\cdot w_{ij,t}^{\mathrm{r}} - \eta \cdot \Big[\frac{1-w_{ij}^2}{\tau\left(1-\tanh^2(w_{ij,t}^{\mathrm{r}})\right)} \cdot \frac{1}{|\mathcal{B}|}\sum_{(\boldsymbol{x},\boldsymbol{y})\in\mathcal{B}} \frac{\partial \mathcal{L}_{\boldsymbol{x}^t,\boldsymbol{y}_t}(\boldsymbol{w})}{\partial w_{ij}} - \rho \cdot w_{ij,0}^{\mathrm{r}}\Big].$$

  end for
  set $\boldsymbol{w}_1^{\mathrm{r}} = \boldsymbol{w}_T^{\mathrm{r}}$
until convergence

Algorithm 2. Bayesian offline learning with binary-valued synapses.

for all $(i,j) \in \mathcal{E}$. The notation $\boldsymbol{w}^{\mathrm{r}}$ has been introduced to suggest a relationship with the STE method described in Section 2.5.3, as defined below. We assume that the prior distribution $p(\boldsymbol{w})$ also follows a mean-field Bernoulli distribution of the form $p(\boldsymbol{w}) = \mathrm{Bern}(\boldsymbol{w}|\sigma(2\boldsymbol{w}_0^{\mathrm{r}}))$, for some vector of $\boldsymbol{w}_0^{\mathrm{r}}$ logits. For example, setting $\boldsymbol{w}_0^{\mathrm{r}} = \boldsymbol{0}$ indicates that the binary weights are equally likely to be either $+1$ or -1 a priori.

In a manner similar to the case of Gaussian VI developed in the previous subsection, we apply natural gradient descent to minimize the variational free energy in Equation (11) with respect to the variational parameters $\boldsymbol{w}^{\mathrm{r}}$ defining the variational posterior $q(\boldsymbol{w})$. Following Meng et al. (2020), and applying the online SGD rule detailed in Section 2.5.2, this yields the update

$$w_{ij,t+1}^{\mathrm{r}} \leftarrow (1-\eta\rho)\cdot w_{ij,t}^{\mathrm{r}} - \eta \cdot \left[\frac{\partial}{\partial \mu_{ij,t}} \mathbb{E}_{q_t(\boldsymbol{w})}\left[\frac{1}{|\mathcal{B}|}\sum_{(\boldsymbol{x},\boldsymbol{y})\in\mathcal{B}} \mathcal{L}_{\boldsymbol{x}^t,\boldsymbol{y}_t}(\boldsymbol{w})\right] - \rho \cdot w_{ij,0}^{\mathrm{r}}\right], \quad (19)$$

where $\eta > 0$ is a learning rate and $q_t(\boldsymbol{w})$ the variational posterior with $\boldsymbol{w}_t^{\mathrm{r}}$ and $\boldsymbol{\mu}_t$ related through (Equation 18). Note that the gradient in Equation 19 is with respect to the mean parameters $\boldsymbol{\mu}_t$.

In order to estimate the gradient in Equation 19, we leverage the *reparameterization* trick *via* the *Gumbel-Softmax* (GS) distribution (Jang et al., 2016; Meng et al., 2020). Accordingly, we first obtain a sample $\boldsymbol{w}$ that is approximately distributed according to $q_t(\boldsymbol{w}) = \mathrm{Bern}\big(\boldsymbol{w}|\sigma(2\boldsymbol{w}_t^{\mathrm{r}})\big)$. This is done by drawing a vector $\boldsymbol{\delta} = \{\{\delta_{ij}\}_{j\in\mathcal{P}_i}\}_{i\in\mathcal{N}}$ of i.i.d. Gumbel variables, and computing

$$w_{ij} = \tanh\left(\frac{w_{ij,t}^{\mathrm{r}} + \delta_{ij}}{\tau}\right), \quad (20)$$

where $\tau > 0$ is a parameter. When τ in Equation (20) tends to zero, the $\tanh(\cdot)$ function tends to the $\mathrm{sign}(\cdot)$ function, and the vector $\boldsymbol{w}$ follows distribution $q_t(\boldsymbol{w})$ (Meng et al., 2020). To generate $\boldsymbol{\delta}$, one can set $\delta_{ij} = \frac{1}{2}\log\left(\frac{\epsilon_{ij}}{1-\epsilon_{ij}}\right)$, with $\epsilon_{ij} \sim \mathcal{U}(0,1)$ being i.i.d. samples.

With this sample, for each example $(\boldsymbol{x}, \boldsymbol{y})$, we then obtain an approximately unbiased estimate of the gradient in Equation (19) by using the following approximation

$$\begin{aligned}
&\frac{\partial}{\partial \mu_{ij,t}} \mathbb{E}_{q_t(\boldsymbol{w})}\left[\mathcal{L}_{\boldsymbol{x}^t,\boldsymbol{y}_t}(\boldsymbol{w})\right] \\
&\overset{(a)}{\approx} \mathbb{E}_{p(\boldsymbol{\delta})}\left[\frac{\partial \mathcal{L}_{\boldsymbol{x}^t,\boldsymbol{y}_t}(\boldsymbol{w})}{\partial \mu_{ij,t}}\bigg|_{\boldsymbol{w}=\tanh\left(\frac{\boldsymbol{w}_t^{\mathrm{r}}+\boldsymbol{\delta}}{\tau}\right)}\right] \\
&\overset{(b)}{=} \mathbb{E}_{p(\boldsymbol{\delta})}\left[\frac{\partial \mathcal{L}_{\boldsymbol{x}^t,\boldsymbol{y}_t}(\boldsymbol{w})}{\partial w_{ij}} \cdot \frac{\partial}{\partial \mu_{ij,t}} \tanh\left(\frac{w_{ij,t}^{\mathrm{r}} + \delta_{ij}}{\tau}\right)\right] \\
&= \mathbb{E}_{p(\boldsymbol{\delta})}\left[\frac{\partial \mathcal{L}_{\boldsymbol{x}^t,\boldsymbol{y}_t}(\boldsymbol{w})}{\partial w_{ij}} \cdot \frac{1-w_{ij}^2}{\tau\left(1-\tanh^2(w_{ij,t}^{\mathrm{r}})\right)}\right],
\end{aligned} \quad (21)$$

where the approximate equality (a) is exact when $\tau \to 0$ and the equality (b) follows the chain rule. We note that the gradient $\nabla_{\boldsymbol{w}} \mathcal{L}_{\boldsymbol{x}^t,\boldsymbol{y}_t}(\boldsymbol{w})$ can be computed as detailed in Section 2.5.2.

As summarized in Algorithm 2, the resulting rule proceeds iteratively by selecting a mini-batch $\mathcal{B}$ of examples $(\boldsymbol{x}, \boldsymbol{y})$ from the training dataset $\mathcal{D}$ at each iteration. Using the samples w_{ij} from Equation (20), we obtain at every time-step t the estimate of the gradient (Equation 19) as

$$\frac{\partial}{\partial \mu_{ij,t}} \mathbb{E}_{q_t(\boldsymbol{w})}\left[\mathcal{L}_{\boldsymbol{x}^t,\boldsymbol{y}_t}(\boldsymbol{w})\right] \approx \frac{1-w_{ij}^2}{\tau\left(1-\tanh^2(w_{ij,t}^{\mathrm{r}})\right)} \cdot \frac{1}{|\mathcal{B}|}\sum_{(\boldsymbol{x},\boldsymbol{y})\in\mathcal{B}} \frac{\partial \mathcal{L}_{\boldsymbol{x}^t,\boldsymbol{y}_t}(\boldsymbol{w})}{\partial w_{ij}} - \rho \cdot w_{ij,0}^{\mathrm{r}}. \quad (22)$$

This is unbiased when the limit $\tau \to 0$ holds.

2.7. Frequentist continual learning

We now consider a continual learning setting, in which the system is sequentially presented datasets $\mathcal{D}^{(1)}, \mathcal{D}^{(2)}, \ldots$ corresponding to distinct, but related, learning tasks. Applying a frequentist approach, at every subsequent task k, the system minimizes a new objective based on dataset $\mathcal{D}^{(k)}$ in order to update the model parameter vector $\boldsymbol{w}$, where we have used superscript (k) to denote the quantities corresponding to the kth task. We first describe an algorithm based on coresets and regularization (Farquhar and Gal, 2019b). Then, we briefly review a recently proposed biologically inspired rule.

2.7.1. Regularization-based continual learning

In a similar manner to Equation (4), let us first define as

$$\mathcal{L}_{\mathcal{D}^{(k)}}(\boldsymbol{w}) = \frac{1}{|\mathcal{D}^{(k)}|} \sum_{(\boldsymbol{x},\boldsymbol{y})\in\mathcal{D}^{(k)}} \mathcal{L}_{\boldsymbol{x},\boldsymbol{y}}(\boldsymbol{w}) \tag{23}$$

the training loss evaluated on dataset $\mathcal{D}^{(k)}$ for the kth task. A general formulation of the continual learning problem in a frequentist framework is then obtained as the minimum of the objective

$$\mathcal{L}_{\mathcal{D}^{(k)}}(\boldsymbol{w}) + \sum_{k'=1}^{k-1} \mathcal{L}_{\mathcal{C}^{(k')}}(\boldsymbol{w}) + \alpha \cdot R\big(\boldsymbol{w}, \{\boldsymbol{w}^{(k')}\}_{k'=1}^{k-1}\big), \tag{24}$$

where $\mathcal{L}_{\mathcal{C}^{(k')}}(\boldsymbol{w})$ is the training loss evaluated on a *coreset*, that is, a subset $\mathcal{C}^{(k')} \subset \mathcal{D}^{(k')}$ of examples randomly selected from a previous task $k' < k$ and maintained for use when future tasks are encountered; $\alpha \geq 0$ determines the strength of the regularization; and $R(\boldsymbol{w}, \{\boldsymbol{w}^{(k')}\}_{k'=1}^{k-1})$ is a regularization function aimed at preventing the current weights from differing too much from previously learned weights $\{\boldsymbol{w}^{(k')}\}_{k'=1}^{k-1}$, hence mitigating the problem of catastrophic forgetting (Parisi et al., 2019).

A popular choice for the regularization function, yielding the Elastic Weight Consolidation (EWC) method, proposes to estimate the relative importance of synapses for previous tasks *via* the Fisher information matrices (FIM) computed on datasets $k' < k$ (Kirkpatrick et al., 2017). This corresponds to the choice of the regularizer

$$R\big(\boldsymbol{w}, \{\boldsymbol{w}^{(k')}\}_{k'=1}^{k-1}\big) = \sum_{k'=1}^{k-1} (\boldsymbol{w} - \boldsymbol{w}^{(k')})^T F^{(k')}(\boldsymbol{w}^{(k')})(\boldsymbol{w} - \boldsymbol{w}^{(k')}), \tag{25}$$

where $F^{(k)}(\boldsymbol{w}) = \mathrm{diag}\big(\sum_{(\boldsymbol{x},\boldsymbol{y})\in\mathcal{D}^{(k)}} (\nabla_{\boldsymbol{w}} \mathcal{L}_{\boldsymbol{x},\boldsymbol{y}}(\boldsymbol{w}))^2\big)$ is an approximation of the FIM estimated on dataset $\mathcal{D}^{(k)}$. The square operation in vector $(\nabla_{\boldsymbol{w}} \mathcal{L}_{\boldsymbol{x},\boldsymbol{y}}(\boldsymbol{w}))^2$ is evaluated element-wise. Intuitively, a larger value of an entry in the diagonal of the matrix $F^{(k)}(\boldsymbol{w})$ indicates that the corresponding entry of the vector $\boldsymbol{w}$ plays a significant role for the kth task.

2.7.2. Biologically inspired continual learning

The authors of Soures et al. (2021) introduce a biologically inspired, frequentist, continual learning rule for SNNs, which we briefly review here. The approach operates online in discrete time t, and implements the mechanisms described in Section 2.4. It considers a leaky integrate-and-fire (LIF) neuron model. The LIF is a special case of the SRM (Equations 1, 2) in which the synaptic response α implemented as the *alpha-function* spike response $\alpha_t = \exp(-t/\tau_{\mathrm{mem}}) - \exp(-t/\tau_{\mathrm{syn}})$ and the exponentially decaying feedback filter $\beta_t = -\exp(-t/\tau_{\mathrm{ref}})$ for $t \geq 1$ with some positive constants $\tau_{\mathrm{mem}}, \tau_{\mathrm{syn}}$, and τ_{ref}. This choice enables an autoregressive update of the membrane potential (Jang et al., 2020a; Kaiser et al., 2020).

A metaplasticity parameter ν_{ij} is introduced for each synapse $(i,j) \in \mathcal{E}$ that determines the degree to which the synapse is prone to change. This quantity is increased by a fixed step $\Delta\nu$ as

$$\nu_{ij,t+1} \leftarrow \nu_{ij,t} + \Delta\nu \tag{26}$$

when the pre- and post-synaptic neurons spiking rates, i.e., the spiking rate of neuron i and j, respectively, pass a predetermined threshold. Furthermore, each synapse $(i,j) \in \mathcal{E}$ maintains a reference weight w_{ij}^{ref} to mimic heterosynaptic plasticity by adjusting the weight updates to drive synaptic weights toward this resting state. It is updated over time as

$$w_{ij,t+1}^{\mathrm{ref}} \leftarrow w_{ij,t}^{\mathrm{ref}} + \kappa \cdot \Big(w_{ij,t} - w_{ij,t}^{\mathrm{ref}}\Big), \tag{27}$$

where $\kappa > 0$, and serves as a reference to implement heterosynaptic plasticity.

With these definitions, the update of each synaptic weight $\boldsymbol{w}$ is computed according to the online learning rule

$$w_{ij,t+1} \leftarrow w_{ij,t} - \exp\big(-|\nu_{ij} \cdot w_{ij,t}|\big) \Big(\eta \cdot e_{i,t} \cdot s_{j,t} \cdot \sigma'(u_{i,t} - \vartheta) + \gamma \cdot (w_{ij,t} - w_{ij,t}^{\mathrm{ref}}) \cdot s_{i,t}\Big), \tag{28}$$

where η and γ are respectively learning and decay rates, and $e_{i,t}$ is an error signal from neuron i (see Soures et al., 2021 for details). The rule (Equation 28) takes a form similar to that of three-factor rules (Equation 8), with the term $e_{i,t} \cdot s_{j,t} \cdot \sigma'(u_{i,t} - \vartheta)$ evaluating the product of error, post-synaptic, and pre-synaptic signals. The update (Equation 28) implements metaplasticity *via* the term $\exp\big(-|\nu_{ij} \cdot w_{ij,t}|\big)$ that decreases the magnitude of the updates during the training procedure for active synapses. It also accounts for heterosynaptic plasticity thanks to the term $(w_{ij,t} - w_{ij,t}^{\mathrm{ref}})$, which drives the updates toward learned "resting" weight $w_{ij,t}^{\mathrm{ref}}$ when the pre-synaptic neuron is active.

2.8. Bayesian continual learning

In this section, we generalize the Bayesian formulation seen in Section 2.6 from the offline setting to continual learning.

2.8.1. Bayesian continual learning

To allow the adaptation to task k without catastrophic forgetting, we consider the problem (Farquhar and Gal, 2019a).

$$\min_{q^{(k)}(\boldsymbol{w})} \mathcal{F}^{(k)}\left(q^{(k)}(\boldsymbol{w})\right) \tag{29}$$

of minimizing the free energy metric

$$\mathcal{F}^{(k)}\left(q^{(k)}(\boldsymbol{w})\right) = \mathbb{E}_{q^{(k)}(\boldsymbol{w})}\left[\mathcal{L}_{\mathcal{D}^{(k)}}(\boldsymbol{w}) + \sum_{k'=1}^{k-1} \mathcal{L}_{\mathcal{C}^{(k')}}(\boldsymbol{w})\right] + \rho \cdot \mathrm{KL}\left(q^{(k)}(\boldsymbol{w}) || q^{(k-1)}(\boldsymbol{w})\right), \tag{30}$$

which combines the IRM formulation (Equation 11) with the use of coresets. Minimizing the free energy objective (Equation 30) must strike a balance between fitting the new training data $\mathcal{D}^{(k)}$, as well as the coresets $\{\mathcal{C}^{(k')}\}_{k'=1}^{k-1}$ from the previous tasks, while not deviating too much from previously learned distribution $q^{(k-1)}(\boldsymbol{w})$. Comparing (Equation 30) with the free energy (Equation 11), we observe that the distribution $q^{(k-1)}(\boldsymbol{w})$ plays the role of prior for the current task k.

2.8.2. Continual gaussian mean-field variational inference

Similarly to the approach for offline learning described in Section 2.6, we first assume a Gaussian variational posterior $q(\boldsymbol{w})$, and address the problem (Equation 30) *via* natural gradient descent. To this end, we adopt the variational posterior $q^{(k)}(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w}|\boldsymbol{m}^{(k)}, (\boldsymbol{P}^{(k)})^{-1})$, with mean vector $\boldsymbol{m}^{(k)}$ and diagonal precision matrix $\boldsymbol{P}^{(k)}$ with positive diagonal vector $\boldsymbol{p}^{(k)}$ of size $|\boldsymbol{w}| \times 1$ for every task k. We choose the prior $p(\boldsymbol{w})$ for dataset $\mathcal{D}_1$ as the Gaussian distribution $p(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w}|\boldsymbol{m}_0, \boldsymbol{P}_0^{-1})$ with positive diagonal vector $\boldsymbol{p}_0$ of size $|\boldsymbol{w}| \times 1$. Applying the Bayesian learning rule (Khan and Rue, 2021) as in Section 2.6.2, updates to the mean and precision parameters can be obtained *via* online SGD as

$$p_{ij,t+1}^{(k)} \leftarrow (1-\eta\rho)\cdot p_{ij,t}^{(k)} + \eta \cdot \mathbb{E}_{q_t^{(k)}(\boldsymbol{w})} \left[\frac{1}{|\mathcal{B}|}\sum_{(\boldsymbol{x},\boldsymbol{y})\in\mathcal{B}} \left(\frac{\partial \mathcal{L}_{\boldsymbol{x}^t,\boldsymbol{y}_t}(\boldsymbol{w})}{\partial w_{ij}}\right)^2 + \rho \cdot p_{ij}^{(k-1)}\right] \tag{31}$$

$$m_{ij,t+1}^{(k)} \leftarrow m_{ij,t}^{(k)} - \eta \cdot (p_{ij,t+1}^{(k)})^{-1} \cdot \mathbb{E}_{q_t^{(k)}(\boldsymbol{w})} \left[\frac{1}{|\mathcal{B}|}\sum_{(\boldsymbol{x},\boldsymbol{y})\in\mathcal{B}} \frac{\partial \mathcal{L}_{\boldsymbol{x}^t,\boldsymbol{y}_t}(\boldsymbol{w})}{\partial w_{ij}} - \rho \cdot p_{ij}^{(k-1)} \cdot \left(m_{ij}^{(k-1)} - m_{ij,t}^{(k)}\right)\right], \tag{32}$$

where mini-batch $\mathcal{B}$ is now selected at random from both dataset $\mathcal{D}^{(k)}$ and coresets from previous tasks, i.e., $\mathcal{B} \subseteq \mathcal{D}^{(k)} \cup_{k'=1}^{k} \mathcal{C}^{(k')}$. The rule can be directly derived by following the steps detailed in Section 2.6.2, and using for prior at every task k the mean $\boldsymbol{m}^{(k-1)}$ and precision $\boldsymbol{P}^{(k-1)}$ obtained at the end of training on the previous task.

2.8.3. On the biological plausibility of the Bayesian learning rule

The continual learning rule (Equations 31, 32) exhibits some of the mechanisms thought to enable memory retention in biological brains as described in Section 2.4. In particular, synaptic consolidation and metaplasticity for each synapse $(i,j) \in \mathcal{E}$ are modeled by the precision p_{ij}. In fact, a larger precision $p_{ij,t+1}$ effectively reduces the step size $1/p_{ij,t+1}$ of the synaptic weight update (Equation 32). This is a similar mechanism to the metaplasticity parameter $v_{ij,t}$ introduced in the rule (Equation 28). Furthermore, by Equation 31, the precision p_{ij} is increased to a degree that depends on the relevance of the synapse $(i,j) \in \mathcal{E}$ as measured by the estimated FIM $(\partial \mathcal{L}_{\boldsymbol{x}^t,\boldsymbol{y}_t}(\boldsymbol{w})/\partial w_{ij})^2$ for the current mini-batch $\mathcal{B}$ of examples.

Heterosynaptic plasticity, which drives the updates toward previously learned and resting states to prevent catastrophic forgetting, is obtained from first principles *via* the IRM formulation with a KL regularization term, rather than from the addition of the reference weight $\boldsymbol{w}_{\mathrm{ref}}$ in the previous work (Soures et al., 2021). This mechanism drives the updates of the precision $p_{ij,t+1}^{(k)}$ and mean parameter $m_{ij,t+1}^{(k)}$ toward the corresponding parameters of the variational posterior obtained at the previous task, namely $p_{ij}^{(k-1)}$ and $m_{ij}^{(k-1)}$.

Finally, the use of coresets implements a form of replay, or reactivation, in biological brains (Buhry et al., 2011).

2.8.4. Continual bernoulli mean-field variational inference

We now consider continual learning with a Bernoulli mean-field variational posterior, and force the synaptic weight w_{ij} to be binary, i.e., $w_{ij} \in \{+1, -1\}$. Following Equation (16), the posterior is of the form $q^{(k)}(\boldsymbol{w}) = \mathrm{Bern}\left(\boldsymbol{w}|\boldsymbol{p}^{(k)}\right)$.

We leverage the Gumbel-softmax trick, and use the reparametrization in terms of the natural parameters at task k

$$w_{ij}^{\mathrm{r},(k)} = \frac{1}{2}\log\left(\frac{1+\mu_{ij}^{(k)}}{1-\mu_{ij}^{(k)}}\right). \tag{33}$$

We then apply the Bayesian learning rule, and, following the results obtained in the offline learning case of Section 2.6.3, we obtain the learning rule at task k as

$$w_{ij,t+1}^{\mathrm{r},(k)} \leftarrow (1-\eta\rho)\cdot w_{ij,t}^{\mathrm{r},(k)} - \eta \cdot \left[\mathbb{E}_{p(\boldsymbol{\delta})}\left[\frac{1}{|\mathcal{B}|}\sum_{(\boldsymbol{x},\boldsymbol{y})\in\mathcal{B}} \frac{\partial \mathcal{L}_{\boldsymbol{x}^t,\boldsymbol{y}_t}(\boldsymbol{w})}{\partial w_{ij}} \cdot \frac{1-w_{ij}^2}{\tau\left(1-\tanh^2\left(w_{ij,t}^{\mathrm{r},(k)}\right)\right)}\right] - \rho \cdot w_{ij}^{\mathrm{r},(k-1)}\right], \tag{34}$$

where we denote as $\boldsymbol{w}^{\mathrm{r},(k-1)}$ the logits obtained at the end of the previous task $k-1$, and mini-batch $\mathcal{B}$ is selected at random as $\mathcal{B} \subseteq \mathcal{D}^{(k)} \cup_{k'=1}^{k} \mathcal{C}^{(k')}$.

3. Experiments

In this section, we compare the performance of frequentist and Bayesian learning schemes in a variety of experiments, using both synthetic and real neuromorphic datasets. All experiments consist of classification tasks with C classes. In each such task, we are given a dataset $\mathcal{D}'$ consisting of spiking inputs $\boldsymbol{x}$ and label $c_{\boldsymbol{x}} \in \{0, 1, \dots, C-1\}$. Each pair $(\boldsymbol{x}, c_{\boldsymbol{x}})$ is converted into a pair of spiking signals $(\boldsymbol{x}, \boldsymbol{y})$ to obtain the dataset $\mathcal{D}$. To do this, the target signals $\boldsymbol{y}$ are such that each sample $\boldsymbol{y}_t$ is the $C \times 1$ one-hot encoding vector of label $c_{\boldsymbol{x}}$ for all time-steps $t = 1, \dots, T$.

3.1. Datasets

3.1.1. Two-moons dataset

We first consider an offline 2D binary classification task on the two-moons dataset (Scikit-Learn library, 2020). Training is done on 200 examples per class with added noise with standard deviation $\sigma = 0.1$ as proposed in Meng et al. (2020) for 100 epochs. The inputs $\boldsymbol{x}$ are obtained *via* population encoding following (Jang et al., 2020a) over $T = 100$ time-steps and *via* 10 neurons.

3.1.2. DVS-gestures

Next, we consider a real-world neuromorphic dataset for offline classification, namely the DVS-Gestures dataset (Amir et al., 2017). The dataset comprises 11 classes of hand movements, captured with a DVS camera. Movements are recorded from 30 different persons under 5 lighting conditions. To evaluate the calibration of Bayesian learning algorithms, we obtain in- and out-of-distribution dataset $\mathcal{D}_{\text{id}}$ and $\mathcal{D}_{\text{ood}}$ by partitioning the dataset by users and lighting conditions. We selected the first 15 users for the training set, while the remaining 15 users are used for testing. The first 4 lighting conditions are used for in-distribution testing; and the one left out from the training set is used for out-of-distribution testing. Images are of size $128 \times 128 \times 2$, and preprocessed following (Amir et al., 2017) to yield inputs of size $32 \times 32 \times 2$, with sequences of length 500 ms for training and 1,500 ms for testing, with a sampling rate of 10 ms.

3.1.3. Split-MNIST and MNIST-DVS

For continual learning, we first conduct experiments on the 5-ways split-MNIST dataset (Farquhar and Gal, 2019a; Soures et al., 2021). Examples from the MNIST dataset, of size 28×28 pixels, are hence rate-encoded over $T = 50$ time-steps (Jang et al., 2020a), and training examples drawn from subsets of two classes are successively presented to the system for training. The order of the pairs is selected as $\{0, 1\}$, then $\{2, 3\}$, and so on. We restrict here our study to rate encoding, although the proposed methods are applicable to any spike encoding scheme. In a similar way, we also consider a neuromorphic continual learning setting based on the neuromorphic counterpart to the MNIST dataset, namely, the MNIST-DVS dataset (Serrano-Gotarredona and Linares-Barranco, 2015). Following the preprocessing adopted in Skatchkovsky et al. (2020a,b, 2021), we cropped images spatially to 26×26 pixels, capturing the active part of the image, and temporally to a duration of 2 s. For each pixel, positive and negative events are encoded as (unsigned) spikes over two different input channels, and the input $\boldsymbol{x}$ is of size 1,352 spiking signals. Uniform downsampling over time is then carried out to restrict the length to $T = 80$ time-samples. The training dataset is composed of 900 examples per class, and the test dataset contains 100 examples per class. For continual learning, classes are presented to the network in pairs by following the lexicographical order, i.e., the classes $\{0, 1\}$ are presented first, then $\{2, 3\}$, and so on.

3.2. Implementation

All schemes are implemented using the SG technique DECOLLE (Kaiser et al., 2020) to compute the gradients. In DECOLLE, the SNN is organized into L layers, with the first $L-1$ layers encompassing the hidden neurons in set $\mathcal{H}$, and the Lth layer containing the read-out neurons in set $\mathcal{Y}$. To evaluate the partial derivative (Equation 8), we need to specify how to compute error signals $e_{i,t}$ for each neuron $i \in \mathcal{N}$. To this end, at each time t, the spiking outputs $\boldsymbol{s}_t^{(l)}$ of each layer $l \in \{1, \dots, L\}$ are used to compute local per-layer errors

$$L(y_{m,t}, \boldsymbol{s}_t^{(l)}) = -y_{m,t} \cdot \log\Big(\text{Softmax}_m\big(\boldsymbol{B}^{(l)} \boldsymbol{s}_t^{(l)}\big)\Big), \tag{35}$$

where $\boldsymbol{B}^{(l)} \in \mathbb{R}^{C \times |l|}$ are random, fixed weights, $|l|$ is the cardinality of layer l, and $\text{Softmax}_m(\boldsymbol{a}) = \exp(a_m) / \sum_{1 \le n \le C} \exp(a_n)$ is the ith element of the softmax of vector $\boldsymbol{a}$ with length C. The local losses (Equation 35) at every time-step t are then used to compute the error signals $e_{i,t}$ in Equation (8) for every neuron $i \in l$ as

$$e_{i,t} = \sum_{m \in \mathcal{Y}} \frac{\partial L(y_{m,t}, \boldsymbol{s}_t^{(l)})}{\partial s_{i,t}}. \tag{36}$$

While the algorithms introduced in this work are valid for any SNN architecture as highlighted in Figure 1, DECOLLE is limited to feedforward layered architectures, which we hence adopt for our experiments (Kaiser et al., 2020). Furthermore, we consider autoregressive filters for the spike responses of synapses α_t and somas β_t in the membrane potential (Equation 2), as discussed in Section 2.1.1.

Results have been obtained by using Intel's Lava software framework (Intel Corporation, 2021), under Loihi-compatible

fixed-point precision (Davies et al., 2018)[1]. We use as benchmark the frequentist algorithms detailed in Sections 2.5, 2.7, for which gradients are as described in the previous paragraph. For Bayesian learning with real-valued (fixed-precision) synapses, we set the threshold of each neuron as $\vartheta = 64$; while for binary synapses the threshold ϑ is selected as the square-root of the fan-in of the corresponding layer.

Implementation of the proposed methods on hardware is left for future work. While Loihi supports the injection of Gaussian noise to the membrane potential of the neurons (Davies et al., 2018), it does not provide mechanisms for the sampling of the model parameters. In contrast, recent work (Dalgaty et al., 2021) has proposed leveraging the inherent noise of nanoscale devices in order to implement Bayesian inference.

3.3. Performance measures

Apart from the test accuracy, performance metrics include calibration measures, namely reliability diagrams and expected calibration error (ECE), which are described next. We note that, as the hardware implementation of Bayesian SNNs is currently an open problem (see Section 3.2), we are unable to provide measurements in terms of energy expenditure and computation time. As a general remark, as discussed in Section 2.2, Bayesian learning requires a larger memory to store all samples for the weights distribution to be used for inference using a committee machine implementation, while an ensemble predictor implementation increases inference latency.

3.3.1. Confidence levels

For frequentist learning, predictive probabilities are obtained from a single pass through the network with parameter vector $\boldsymbol{w}$ as

$$p(c_{\boldsymbol{x}} = k \mid \boldsymbol{x}, \boldsymbol{w}) = \frac{1}{T}\sum_{t=1}^{T} \text{Softmax}_k\big(\boldsymbol{B}^{(L)} f(\boldsymbol{w}, \boldsymbol{x}^t)\big), \tag{37}$$

where $f(\boldsymbol{w}, \boldsymbol{x}^t)$ is the output of read-out layer L for weights $\boldsymbol{w}$, as detailed in the previous subsection.

In contrast, for Bayesian learning, decisions and confidence levels are obtained by drawing N_S samples $\{\boldsymbol{w}_s\}_{s=1}^{N_S}$ from the distribution $q(\boldsymbol{w})$, and by averaging the read-out outputs of the model to obtain the probability assigned to each class as

$$p\big(c_{\boldsymbol{x}} = k \mid \boldsymbol{x}, \{\boldsymbol{w}_s\}_{s=1}^{N_S}\big) = \frac{1}{N_S}\frac{1}{T}\sum_{s=1}^{N_S}\sum_{t=1}^{T} \text{Softmax}_k\big(\boldsymbol{B}^{(L)} f(\boldsymbol{w}_s, \boldsymbol{x}^t)\big). \tag{38}$$

1 Our implementation can be found at: https://github.com/kclip/bayesian-snn.

Unless mentioned otherwise, the predictions (Equation 38) are obtained by using the committee machine approach, and hence the weights $\{\boldsymbol{w}_s\}_{s=1}^{N_S}$ are kept fixed for all test inputs $\boldsymbol{x}$ (see Section 2.2). All results presented are averaged over three repetitions of the experiments and 10 draws from the posterior distribution $q(\boldsymbol{w})$, i.e., we set $N_S = 10$ in all experiments.

For Bayesian learning, the hard prediction of the model is hence obtained as

$$c_{\boldsymbol{x}}^* = \underset{1\le k\le C}{\text{argmax}}\, p\big(c_{\boldsymbol{x}} = k \mid \boldsymbol{x}, \{\boldsymbol{w}_s\}_{s=1}^{N_S}\big), \tag{39}$$

corresponding to the predictive probability

$$p\big(c_{\boldsymbol{x}}^* \mid \boldsymbol{x}, \{\boldsymbol{w}_s\}_{s=1}^{N_S}\big) = \max_{1\le k\le C} p\big(c_{\boldsymbol{x}} = k \mid \boldsymbol{x}, \{\boldsymbol{w}_s\}_{s=1}^{N_S}\big). \tag{40}$$

The probability (Equation 40) can be interpreted as the confidence of the model in making decisions (Equation 39).

A model is considered to be well calibrated when there is no mismatch between confidence level $p\big(c_{\boldsymbol{x}}^* \mid \boldsymbol{x}, \{\boldsymbol{w}_s\}_{s=1}^{N_S}\big)$ and the actual probability for the model to correctly classify input $\boldsymbol{x}$ (Guo et al., 2017). Definitions (Equations 39, 40) can be straightforwardly adapted to the frequentist case by replacing the average over draws $\{\boldsymbol{w}_s\}_{s=1}^{N_S}$ with a single parameter vector $\boldsymbol{w}$.

3.3.2. Reliability diagrams

Reliability diagrams plot the actual probability of correct detection as a function of the confidence level (Equation 40). This is done by first dividing the probability interval $[0, 1]$ into M intervals of equal length, and then evaluating the average accuracy and confidence for all inputs $\boldsymbol{x}$ in each mth interval $(\frac{m-1}{M}, \frac{m}{M}]$, also referred to as mth *bin*. We denote as $\mathcal{B}_m$ the subset of examples whose associated confidence level $p\big(c_{\boldsymbol{x}}^* \mid \boldsymbol{x}, \{\boldsymbol{w}_s\}_{s=1}^{N_S}\big)$ lies within bin m, that is, Guo et al. (2017)

$$\mathcal{B}_m = \left\{\boldsymbol{x} \in \mathcal{D} \;\middle|\; p\big(c_{\boldsymbol{x}}^* \mid \boldsymbol{x}, \{\boldsymbol{w}_s\}_{s=1}^{N_S}\big) \in \left(\frac{m-1}{M}, \frac{m}{M}\right]\right\}. \tag{41}$$

The average empirical accuracy of the predictor within bin m is defined as

$$\text{acc}(\mathcal{B}_m) = \frac{1}{|\mathcal{B}_m|}\sum_{\boldsymbol{x}\in\mathcal{B}_m} \mathbf{1}(c_{\boldsymbol{x}}^* = c_{\boldsymbol{x}}), \tag{42}$$

with $\mathbf{1}(\cdot)$ being the indicator function; while the average empirical confidence of the predictor for bin m is defined as

$$\text{conf}(\mathcal{B}_m) = \frac{1}{|\mathcal{B}_m|}\sum_{\boldsymbol{x}\in\mathcal{B}_m} p\big(c_{\boldsymbol{x}}^* \mid \boldsymbol{x}, \{\boldsymbol{w}_s\}_{s=1}^{N_S}\big). \tag{43}$$

Reliability diagrams plot the per-bin accuracy $\text{acc}(\mathcal{B}_m)$ vs. confidence level $\text{conf}(\mathcal{B}_m)$ across all bins m. A model is said to be perfectly calibrated when, for all bins m, the

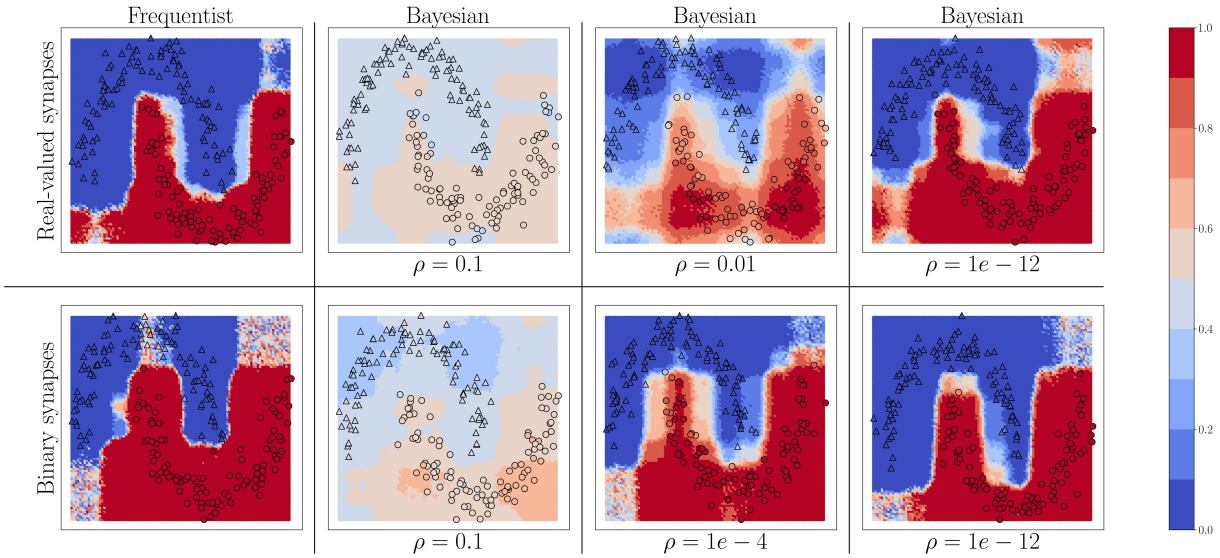


FIGURE 4
Predictive probabilities (Equation 40) evaluated on the two-moons dataset after training with different values of the temperature ρ in Equation (11) for Bayesian learning. **Top row:** Real-valued synapses; **Bottom row:** Binary synapses.

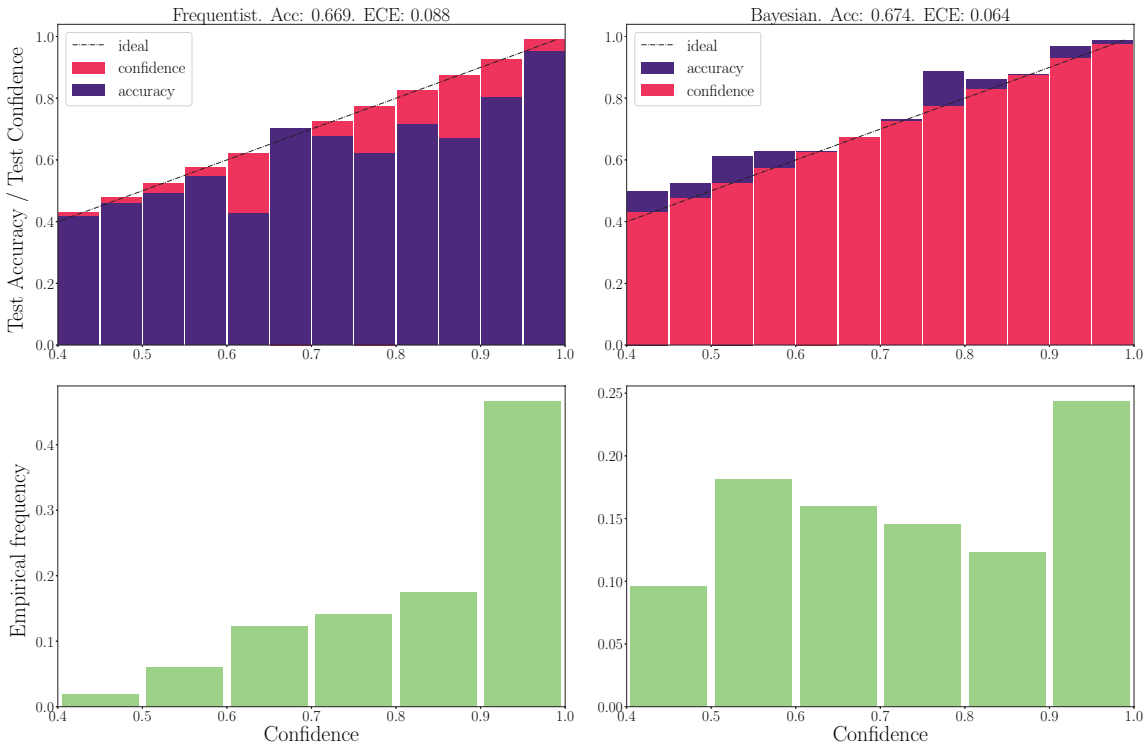


FIGURE 5
Top: Reliability diagrams (for in-distribution data) with real-valued synapses for the DVS-Gestures dataset. **Bottom**: Corresponding empirical confidence histograms for in-distribution data.

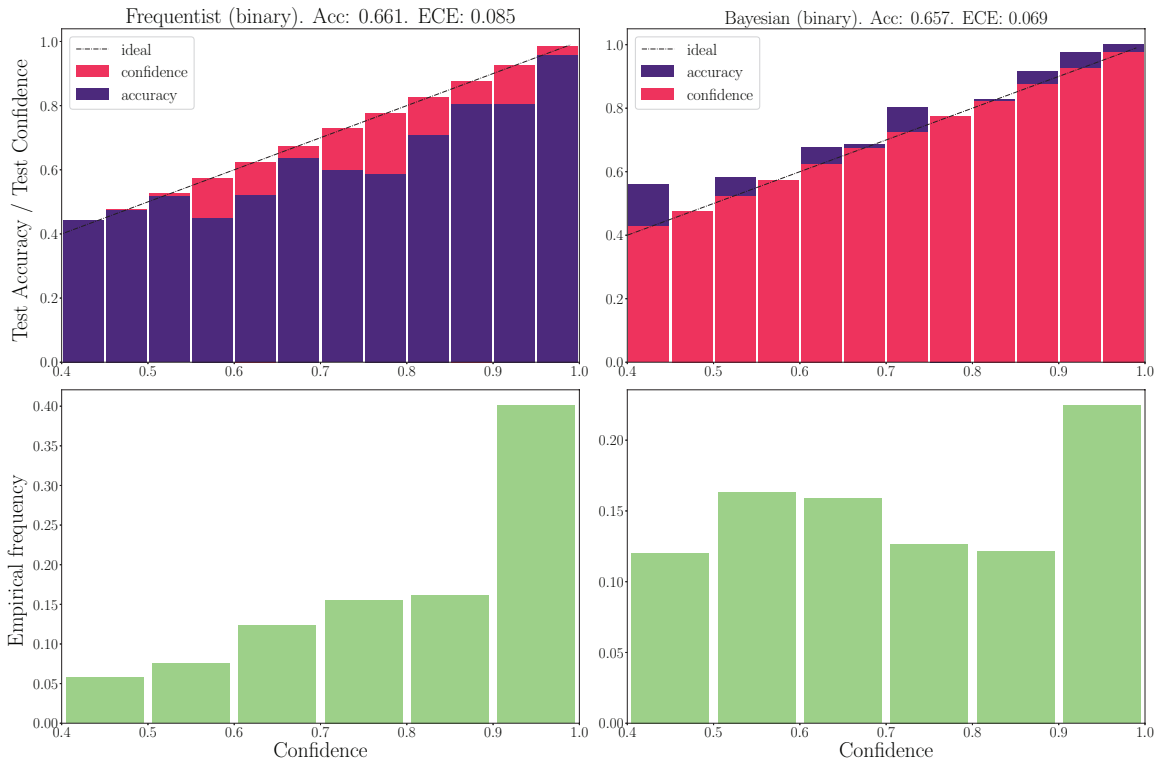


FIGURE 6
Top: Reliability diagrams (for in-distribution data) with binary synapses for the DVS-Gestures dataset. **Bottom**: Corresponding empirical confidence histograms for in-distribution data.

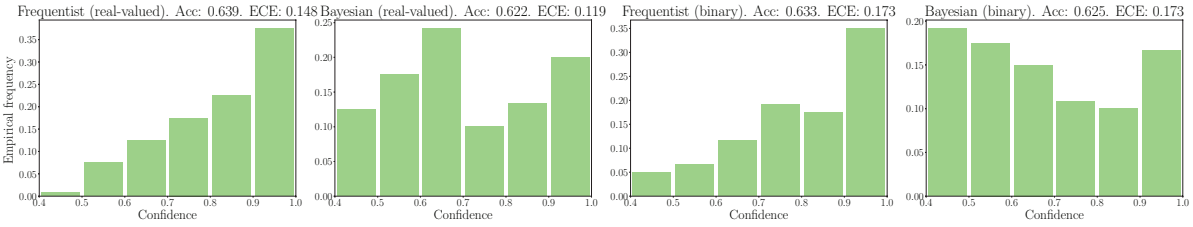


FIGURE 7
Out-of-distribution empirical confidence histograms for SNNs with real-valued and binary synapses on the DVS-Gestures dataset.

equality $\text{acc}(\mathcal{B}_m) = \text{conf}(\mathcal{B}_m)$ holds. If in the mth bin, the empirical accuracy and empirical confidence are different, the predictor is considered to be over-confident when the inequality $\text{acc}(\mathcal{B}_m) < \text{conf}(\mathcal{B}_m)$ holds, and under-confident when the reverse inequality $\text{acc}(\mathcal{B}_m) > \text{conf}(\mathcal{B}_m)$ holds.

3.3.3. Expected calibration error

While reliability diagrams offer a fine-grained description of calibration, the ECE provides a scalar measure of the global miscalibration of the model. This is done by computing the average

difference between per-bin confidence and accuracy as Guo et al. (2017).

$$\mathrm{ECE} = \frac{1}{|\mathcal{D}|}\sum_{m=1}^{M} |\mathcal{B}_m| \left|\mathrm{conf}(\mathcal{B}_m) - \mathrm{acc}(\mathcal{B}_m)\right|. \tag{44}$$

Models with a lower ECE are considered to be better calibrated.

3.3.4. Out-of-distribution empirical confidence

Reliability diagrams and ECE assume that the test data follows the same distribution as the training data. A well-calibrated model is also expected to assign lower probabilities to out-of-distribution data, i.e., data that does not follow the training distribution (DeGroot and Fienberg, 1983). To gauge the capacity of a model to recognize out-of-distribution data, a common approach is to plot the histogram of the predictive probabilities $\left\{p\left(c^*_{\boldsymbol{x}}|\boldsymbol{x}, \{\boldsymbol{w}_s\}_{s=1}^{N_S}\right)\right\}_{\boldsymbol{x}\in\mathcal{D}_{\mathrm{ood}}}$ evaluated on a dataset $\mathcal{D}_{\mathrm{ood}}$ of out-of-distribution examples (DeGroot and Fienberg, 1983; Daxberger and Hernández-Lobato, 2019). Such examples may correspond, as discussed, to examples recorded in different lighting conditions with a neuromorphic camera.

3.4. Offline learning

3.4.1. Two-moons dataset

We start by considering the two-moons dataset. For this experiment, the SNN comprises two fully connected layers with 256 neurons each. Bayesian learning is implemented with different values of the temperature parameter ρ in the free energy (Equation 11). In Figure 4, triangles indicate training points for a class "0," while circles indicate training points for a class "1." The color intensity represents the predictive probabilities (Equation 37) for frequentist learning and Equation (38) for Bayesian learning: the more intense the color, the higher the prediction confidence determined by the model. Bayesian learning is observed to provide better calibrated predictions, that are more uncertain outside the input regions covered by training data points.

For both real-valued and binary synapses, the temperature parameter ρ has an important role to play in preventing overfitting and underfitting of the training data, while also enabling uncertainty quantification. When the parameter ρ is too large, the model cannot fit the data correctly, resulting in inaccurate predictions; while when ρ is too small, the training data is fit more tightly, leading to a poor representation of the prediction uncertainty outside the training set. A well-chosen value of ρ strikes the best trade-off between faithfully fitting the training data and allowing for uncertainty quantification. Frequentist algorithms, obtained in the limit when $\rho \to 0$, yield the most over-confident estimates.

3.4.2. DVS-gestures

We now turn to the DVS-Gestures dataset, for which we plot the performance for real-valued and binary-valued SNNs, in terms of accuracy, reliability diagrams (DeGroot and Fienberg, 1983), and ECE (Guo et al., 2017) in Figures 5, 6. In all cases, the SNNs have two fully connected layers comprising 4,096 neurons each, and they are trained for 200 epochs. The architecture was chosen to highlight the benefits of Bayesian learning over frequentist learning in regimes characterized by epistemic uncertainty, and it was not optimized for maximal accuracy. The figures confirm that Bayesian SNNs generally produce better calibrated outcomes. In fact, reliability diagrams (top rows) demonstrate that frequentist learning algorithms produce overconfident decisions, while Bayesian learning outputs decisions whose confidence levels match well the test accuracies. This improvement is reflected, for models with real-valued synapses (with fixed precision), in a lower ECE of 0.064, as compared to 0.088 for frequentist SNNs; while, for binary SNNs, the reduction in ECE is from 0.085 for frequentist learning, to 0.069 for Bayesian learning. This benefit may come at the cost of a slight decrease in terms of accuracy, which is only observed here for binary synapses. The bottom parts of Figures 5, 6 also show that frequentist learning tends to output high-confidence decisions with a larger probability.

We now turn to evaluate the performance in terms of robustness to out-of-distribution data. As explained in Section 3.3, to this end, we evaluate the histogram of the confidence levels produced by frequentist and Bayesian learning, as shown in Figure 7. From the figure, it is remarked that Bayesian learning correctly provides low confidence levels on out-of-distribution data, while frequentist learning outputs decisions with confidence levels similar to the case of in-distribution data, which are shown in Figures 5, 6.

This point is further illustrated in Figure 8 by showing the three largest probabilities assigned by the different models on selected examples, considering real-valued synapses in the top row and binary synapses in the bottom row. In the left column, we observe that, when both models predict the wrong class, Bayesian SNNs tend to do so with a lower level of certainty, and typically rank the correct class higher than their frequentist counterparts. Specifically, in the examples shown, Bayesian models with both real-valued and binary synapses rank the correct class second, while the frequentist models rank it third. Furthermore, as seen in the middle column, in a number of cases, the Bayesian models manage to predict the correct class, while the frequentist models predict a wrong class with high certainty. Finally, in the right column, we show that even when frequentist models predict the correct class and Bayesian models fail to do so, they still assign lower probabilities (i.e., $< 50\%$) to the predicted class.

A key advantage of SNNs is the possibility to obtain intermediate decisions during the observation of the T

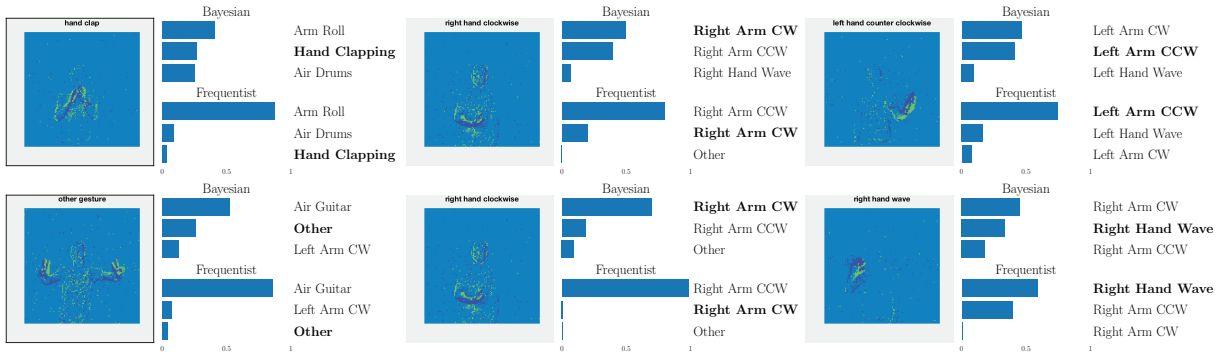


FIGURE 8
Top three classes predicted by both Bayesian and frequentist models on selected examples. **Top:** real-valued synapses. **Bottom:** binary synapses. The correct class is indicated in bold font.

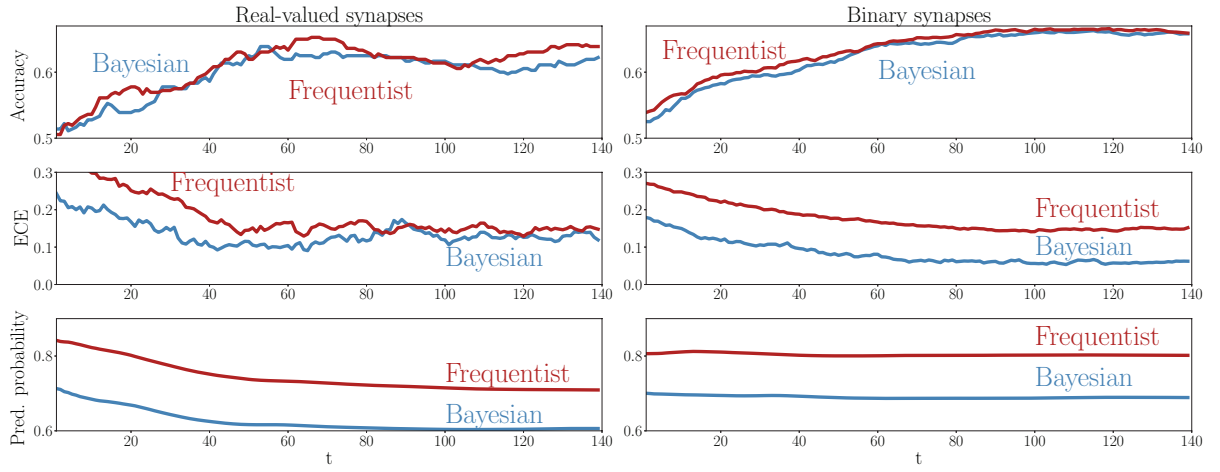


FIGURE 9
Evolution of the accuracy **(top)**, ECE **(middle)**, and predictive probabilities **(bottom)** during the presentation of out-of-distribution test examples for the DVS-Gestures dataset. The horizontal axis represents the time instants t within the presentation of each test example. **Left**: Real-valued synapses. **Right**: Binary synapses.

samples of a test example. To elaborate on this aspect, Figure 9 reports the evolution of the mean test accuracy, ECE, and predictive probabilities (Equations 38, 37) for all examples in the out-of-distribution dataset as a function of the discrete time-steps $t = 1, 2, \ldots, T$. Although both Bayesian and frequentist methods show similar improvements in accuracy over time, frequentist algorithms remain poorly calibrated, even after the observation of many time samples. The bottom plots show that frequentist learning tends to be more confident in its decisions, especially when a few samples t have been observed. On the contrary, Bayesian algorithms offer better calibration and confidence estimates, even when only part of the input signal $\boldsymbol{x}$ has been observed.

TABLE 1 **Final average test accuracy and ECE on the split-MNIST dataset (real-valued synapses).**

Model	Accuracy	ECE
TACOS (Soures et al., 2021) (Full Precision)	83.45 ± 0.55%	N/A
Frequentist (Kirkpatrick et al., 2017)	77.19 ± 0.65%	0.39 ± 0.01
Bayesian committee machine	**85.44 ± 0.16%**	**0.36 ± 0.01**
Bayesian ensemble decision	85.03 ± 0.54%	**0.36 ± 0.01**

3.5. Continual learning

We now turn to continual learning benchmarks. Starting with the rate encoded MNIST dataset, we use coresets

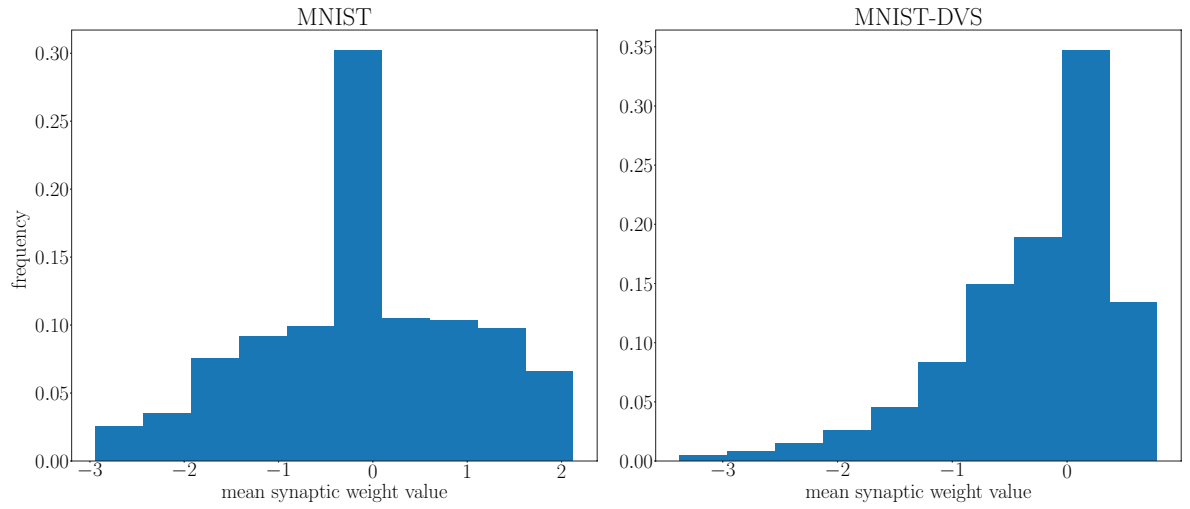


FIGURE 10
Distribution of the mean parameter $\boldsymbol{m}$ at the end of training on the MNIST and MNIST-DVS datasets.

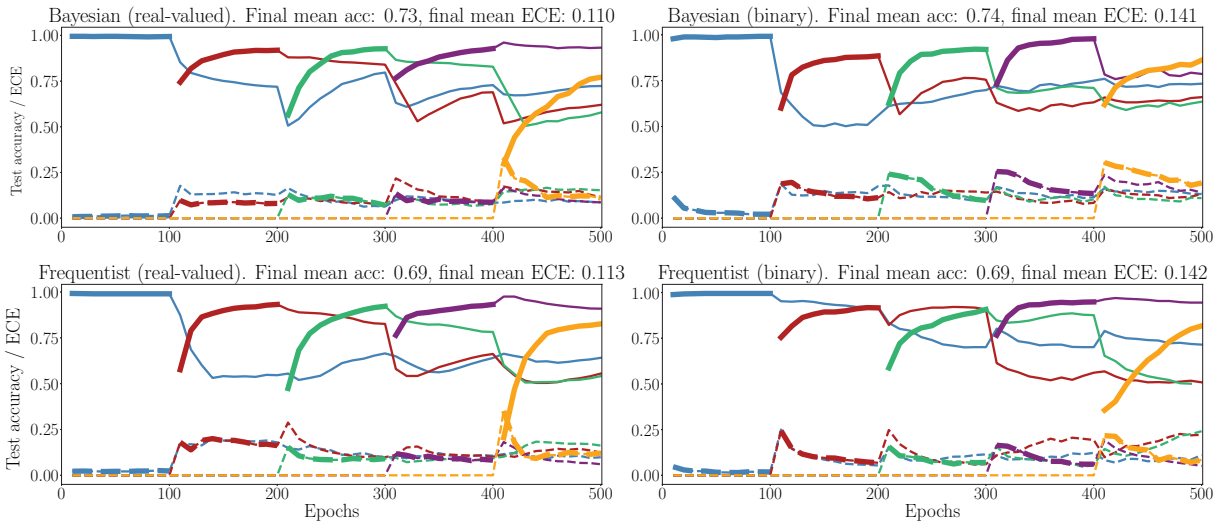


FIGURE 11
Evolution of the average test accuracies and ECE on all tasks of the split-MNIST-DVS across training epochs, with Gaussian and Bernoulli variational posteriors, and frequentist schemes for both real-valued and binary synapses. Continuous lines: test accuracy, dotted lines: ECE, bold: current task. Blue: {0, 1}; Red: {2, 3}; Green: {4, 5}; Purple:{6, 7}; Yellow: {8, 9}.

representing 7.5% of randomly selected training examples for each class. To establish a fair comparison with the protocol adopted in Soures et al. (2021), we train SNNs comprising a single layer with 400 neurons for one epoch on each subtask. This choice was found to be advantageous for Bayesian techniques—a result that may be related to the known asymptotic behavior of Bayesian neural networks as non-parametric models (Neal, 1996). In Table 1, we show the average accuracy over all tasks at the end of training on the last task, as well as the average ECE at that point for real-valued synapses, enabling a comparison with Soures et al. (2021). Bayesian continual learning is seen to achieve the best accuracy and calibration across all the methods studied here, including the solution introduced in Soures et al. (2021). The latter incurs a 2.5× memory overhead as compared to standard frequentist methods. Considering that we performed training

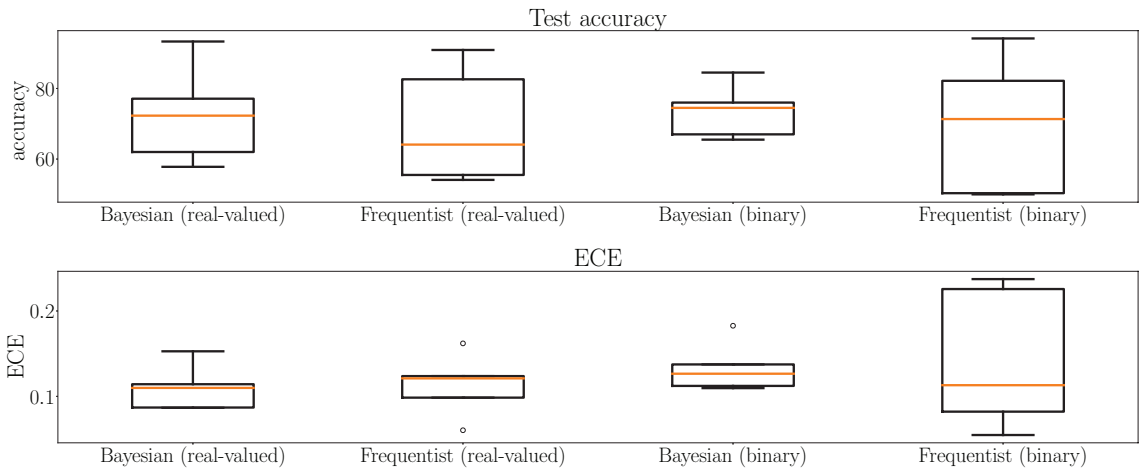


FIGURE 12
Box plots for final test accuracy **(top)** and ECE **(bottom)** across all tasks for Bayesian and frequentist learning with real-valued and binary synapses after the 500th epoch of training on the MNIST-DVS dataset (see Figure 11). The horizontal bar represents the median value across the tasks, while the box extends from the first to the third quartile. The whiskers extend from the box by 1.5 times the inter-quartile range. Circles represent outliers.

using the 8-bit precision imposed by the neuromorphic chip Loihi, our solution outperforms the state-of-the-art with a 5× memory consumption improvement. This saving can be leveraged, e.g., to store several samples of the weights for a committee machine implementation.

Next, for the MNIST-DVS dataset (Serrano-Gotarredona and Linares-Barranco, 2015), we use coresets representing 10% of randomly selected training examples for each class, and implement multilayer SNNs with $2,048 - 4,096 - 4,096 - 2,048 - 1024$ neurons per layer, that we train on each subtask for 100 epochs. This task requires a larger architecture and longer training time to allow for the processing of the richer spatio-temporal information recorded by neuromorphic cameras, as compared to the spatial information from static image datasets, such as MNIST, encoded into spikes *via* rate encoding (Jang et al., 2020a).

We highlight the requirement for a larger architecture on the MNIST-DVS dataset in Figure 10 by comparing the distribution of the mean parameter $\boldsymbol{m}$ at the end of training on the MNIST and MNIST-DVS datasets. For the larger network trained on the MNIST-DVS dataset, 83.5% of the mean parameters are non-zero, a larger proportion than that of the network trained on the MNIST dataset, for which only 80.1% of the mean weight parameters are non-zero. This demonstrates that the larger number of weights used for this task is important for the network to perform well.

In Figure 11, we show the evolution of the test accuracy and ECE on all tasks, represented with lines of different colors, during training. The performance on the current task is shown as a thicker line. We consider frequentist and Bayesian learning, with both real-valued and binary synapses. With Bayesian learning, the test accuracy on previous tasks does not decrease excessively when learning a new task, which shows the capacity of the technique to tackle catastrophic forgetting. Also, the ECE across all tasks is seen to remain more stable for Bayesian learning as compared to the frequentist benchmarks. For both real-valued and binary synapses, the final average accuracy and ECE across all tasks show the superiority of Bayesian over frequentist learning.

This point is further elaborated in Figure 12, which shows test accuracy and ECE on all tasks at the final epoch—the 500th—in Figure 12. Bayesian learning can be seen to offer a better test accuracy and ECE on average across tasks, as well as a lower dispersion among tasks.

4. Conclusion

In this work, we have introduced a Bayesian learning framework for SNNs with both real-valued and binary-valued synapses. Bayesian learning is particularly well suited for applications characterized by limited data—a situation that is likely to be encountered in use cases of neuromorphic computing such as edge intelligence. We have demonstrated the benefits of Bayesian learning in terms of calibration metrics that gauge the effectiveness of uncertainty quantification over a variety of offline and continual learning. We have also argued that the proposed rules exhibit mechanisms resembling those that enable lifelong learning in biological brains from a theoretically motivated information risk minimization framework. While this work focused on variational inference Bayesian learning methods, future research may explore Monte-Carlo based solutions. Finally, we recall the importance of investigating solutions for hardware design, adopting either

ensemble predictors or committees of machines. As an example, consider ensemble predictions based on binary synapses. An implementation based on digital hardware would need to store the real-valued parameters of the parameter vector distribution, and to sample from the distribution using auxiliary circuitry, which incurs energy and memory overheads. Alternatively, one could leverage the inherent stochasticity of analog hardware for sampling (Dalgaty et al., 2021), a line of research that we reserve for future work.

Data availability statement

The original contributions presented in the study are included in the article/Supplementary material, further inquiries can be directed to the corresponding author.

Author contributions

OS first proposed to train SNNs *via* Bayesian learning. HJ derived the rule for binary synapses. NS extended it to real-valued synapses and designed and implemented the experiments. NS, HJ, and OS wrote the text. All authors contributed to the article and approved the submitted version.

Funding

This study received funding from Intel Labs through the Intel Neuromorphic Research Community (INRC). The work of HJ was supported in part by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2021R1F1A10663288). The work of OS has also been supported by the European Research Council (ERC) under the European Union's Horizon 2020 Research and Innovation Programme (Grant Agreement No. 725731) and by an Open Fellowship of the EPSRC. The funders were not involved in the study design, collection, analysis, interpretation of data, the writing of this article or the decision to submit it for publication.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

Supplementary material

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fncom.2022.1037976/full#supplementary-material

References

Abraham, W. C., and Bear, M. F. (1996). Metaplasticity: the plasticity of synaptic plasticity. *Trends Neurosci*. 19, 126–130. doi: 10.1016/S0166-2236(96)80018-X

Aitchison, L., Jegminat, J., Menendez, J. A., Pfister, J.-P., Pouget, A., and Latham, P. E. (2021). Synaptic plasticity as Bayesian inference. *Nat. Neurosci.* 24, 565–571. doi: 10.1038/s41593-021-00809-5

Amir, A., Taba, B., Berg, D., Melano, T., McKinstry, J., Di Nolfo, C., et al. (2017). "A low power, fully event-based gesture recognition system," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (Honolulu, HI: IEEE), 7243–7252.

Angelino, E., Johnson, M., and Adams, R. (2016). Patterns of scalable Bayesian inference. *Foundat. Trends Mach. Learn.* 9, 119–247. doi: 10.1561/9781680832198

Bellec, G., Scherr, F., Subramoney, A., Hajek, E., Salaj, Darjan Legenstein, R., and Maass, W. (2020). A solution to the learning dilemma for recurrent networks of spiking neurons. *Nat. Commun.* 11, 3625. doi: 10.1038/s41467-020-17236-y

Bengio, Y., Léonard, N., and Courville, A. (2013). Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*. doi: 10.48550/arXiv.1308.3432

Buhry, L., Azizi, A. H., and Cheng, S. (2011). Reactivation, replay, and preplay: how it might all fit together. *Neural Plast.* 2011, 1–11. doi: 10.1155/2011/203462

Chistiakova, M., Bannon, N. M., Bazhenov, M., and Volgushev, M. (2014). Heterosynaptic plasticity: multiple mechanisms and multiple roles. *Neuroscientist* 20, 483–498. doi: 10.1177/1073858414529829

Clayton, A. (2021). *Bernoulli's Fallacy*. New York, NY: Columbia University Press.

Courbariaux, M., Hubara, I., Soudry, D., El-Yaniv, R., and Bengio, Y. (2016). Binarized neural networks: training deep neural networks with weights and activations constrained to +1 or -1. *arXiv preprint arXiv:1602.02830*. doi: 10.48550/arXiv.1602.02830

Dalgaty, T., Vianello, E., and Querlioz, D. (2021). "Harnessing intrinsic memristor randomness with bayesian neural networks," in *2021 International Conference on IC Design and Technology* (Dresden: ICICDT), 1–4.

Davies, M., Srinivasa, N., Lin, T. H., Chinya, G., Cao, Y., Harsha, S., et al. (2018). Loihi: a neuromorphic manycore processor with on-chip learning. *IEEE Micro* 38, 82–99. doi: 10.1109/MM.2018.112130359

Davies, M., Wild, A., Orchard, G., Sandamirskaya, Y., Guerra, G. A. F., Joshi, P., et al. (2021). Advancing neuromorphic computing with loihi: a survey of results and outlook. *Proc. IEEE* 109, 911–934. doi: 10.1109/JPROC.2021.3067593

Daxberger, E. A., and Hernández-Lobato, J. M. (2019). Bayesian variational autoencoders for unsupervised out-of-distribution detection. *arXiv preprint 1912.05651*, abs/1912.05651. doi: 10.48550/arXiv.1912.05651

DeGroot, M. H., and Fienberg, S. E. (1983). The comparison and evaluation of forecasters. *J. R. Stat Soc.*32, 12–22. doi: 10.2307/2987588

Doya, K., Ishii, S., Pouget, A., and Rao, R. P. N. (Eds.). (2007). *Bayesian Brain. Computational Neuroscience Series*. London: MIT Press.

Ebrahimi, S., Elhoseiny, M., Darrell, T., and Rohrbach, M. (2020). Uncertainty-guided continual learning with Bayesian neural networks. *arXiv preprint 1906.02425*. doi: 10.48550/arXiv.1906.02425

Farquhar, S., and Gal, Y. (2019a). Towards robust evaluations of continual learning. *arXiv preprint 1805.09733*. doi: 10.48550/arXiv.1805.09733

Farquhar, S., and Gal, Y. (2019b). A unifying bayesian view of continual learning. *arXiv preprint 1902.06494*. doi: 10.48550/arXiv.1902.06494

Feldman Barrett, L. (Ed.). (2021). *Seven and a Half Lessons About the Brain*. London: Picador.

Finnie, P. S., and Nader, K. (2012). The role of metaplasticity mechanisms in regulating memory destabilization and reconsolidation. *Neurosci. Biobeh. Rev*. 36, 1667–1707. doi: 10.1016/j.neubiorev.2012.03.008

Friston, K. (2010). The free-energy principle: a unified brain theory? *Nat. Rev. Neurosci*. 11, 127–138. doi: 10.1038/nrn2787

Friston, K. (2012). The history of the future of the bayesian brain. *Neuroimage* 62, 1230–1233. doi: 10.1016/j.neuroimage.2011.10.004

Gerstner, W., and Kistler, W. M. (2002). *Spiking Neuron Models: Single Neurons, Populations, Plasticity*. Cambridge: Cambridge University Press.

Gerstner, W., Lehmann, M., Liakoni, V., Corneil, D., and Brea, J. (2018). Eligibility traces and plasticity on behavioral time scales: experimental support of neohebbian three-factor learning rules. *Front. Neural Circ*. 12, 53. doi: 10.3389/fncir.2018.00053

Guedj, B. (2019). A primer on PAC-bayesian learning. doi: 10.48550/arXiv.1901.05353

Guo, S., Yu, Z., Deng, F., Hu, X., and Chen, F. (2017). Hierarchical Bayesian inference and learning in spiking neural networks. *IEEE Trans. Cybern*. 49, 133–145. doi: 10.1109/TCYB.2017.2768554

Hawkins, J. (2021). *A Thousand Brains: A New Theory of Intelligence* (Hachette).

Huh, D., and Sejnowski, T. J. (2018). "Gradient descent for spiking neural networks," in *Advances in Neural Information Processing Systems, Vol. 31* (Montreal, QC).

Intel Corporation (2021). *Lava Software Framework*. Available online at: https://lava-nc.org/

Izhikevich, E. M. (2001). Resonate-and-fire neurons. *Neural Netw*. 14, 883–894. doi: 10.1016/S0893-6080(01)00078-8

Jang, E., Gu, S., and Poole, B. (2016). Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*. doi: 10.48550/arXiv.1611.01144

Jang, H., and Simeone, O. (2022). Multisample online learning for probabilistic spiking neural networks. *IEEE Trans. Neural Net. Learn. Syst*. 33, 2034–2044. doi: 10.1109/TNNLS.2022.3144296

Jang, H., Simeone, O., Gardner, B., and Grüning, A. (2019). An introduction to probabilistic spiking neural networks: probabilistic models, learning rules, and applications. *IEEE Sig. Proc. Mag*. 36, 64–77. doi: 10.1109/MSP.2019.2935234

Jang, H., Skatchkovsky, N., and Simeone, O. (2020a). Spiking neural networks-parts I, II, and III. doi: 10.48550/arXiv.2010.14217

Jang, H., Skatchkovsky, N., and Simeone, O. (2020b). VOWEL: a local online learning rule for recurrent networks of probabilistic spiking winner-take-all circuits. *arXiv preprint arXiv:2004.09416*. doi: 10.48550/arXiv.2004.09416

Jang, H., Skatchkovsky, N., and Simeone, O. (2021). "BiSNN: training spiking neural networks with binary weights via bayesian learning," in *2021 IEEE Data Science and Learning Workshop (DSLW)* (Toronto, ON: IEEE), 1–6.

Jaynes, E. T. (2003). *Probability Theory*. Cambridge: Cambridge University Press.

Jose, S. T., and Simeone, O. (2021). Free energy minimization: a unified framework for modeling, inference, learning, and optimization [Lecture Notes]. *IEEE Signal Proc. Mag*. 38, 120–125. doi: 10.1109/MSP.2020.3041414

Kaiser, J., Mostafa, H., and Neftci, E. (2020). Synaptic plasticity dynamics for deep continuous local learning (DECOLLE). *Front. Neurosci*. 14, 424. doi: 10.3389/fnins.2020.00424

Kandel, E. R., Dudai, Y., and Mayford, M. R. (2014). The molecular and systems biology of memory. *Cell* 157, 163–186. doi: 10.1016/j.cell.2014.03.001

Kappel, D., Habenschuss, S., Legenstein, R., and Maass, W. (2015). Network plasticity as bayesian inference. *PLoS Comput. Biol*. 11, e1004485. doi: 10.1371/journal.pcbi.1004485

Khan, M., and Lin, W. (2017). "Conjugate-computation variational inference converting variational inference in non-conjugate models to inferences in conjugate models," in *2017 International Conference on Artificial Intelligence and Statistics* (Ft. Lauderdale, FL), 878–887.

Khan, M. E., and Rue, H. (2021). The bayesian learning rule. *arXiv preprint 2107.04562*. doi: 10.48550/arXiv.2107.04562

Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., et al. (2017). Overcoming catastrophic forgetting in neural networks. *Proc. Natl. Acad. Sci. U.S.A*. 114, 3521–3526. doi: 10.1073/pnas.1611835114

Knoblauch, J., Jewson, J., and Damoulas, T. (2019). Generalized variational inference. *arXiv preprint arXiv:1904.02063*. doi: 10.48550/arXiv.1904.02063

Kreutzer, E., Petrovici, M. A., and Senn, W. (2020). "Natural gradient learning for spiking neurons," in *Proceedings of the Neuro-inspired Computational Elements Workshop*, 1–3.

Kristiadi, A., Hein, M., and Hennig, P. (2020). "Being bayesian, even just a bit, fixes overconfidence in ReLU networks," in *2020 International Conferences on Machine Learning*, 5436–5446.

Kudithipudi, D., and Aguilar-Simon, M. B. J. E. A. (2022). Biological underpinnings for lifelong learning machines. *Nat. Mach. Intell*. 4, 196–210. doi: 10.1038/s42256-022-00452-0

Laborieux, A., Ernoult, M., Hirtzlin, T., and Querlioz, D. (2021). Synaptic metaplasticity in binarized neural networks. *Nat. Commun*. 12, 2549. doi: 10.1038/s41467-021-22768-y

Malenka, R. C., and Bear, M. F. (2004). LTP and LTD: an embarrassment of riches. *Neuron* 44, 5–21. doi: 10.1016/j.neuron.2004.09.012

Marder, E. (2012). Neuromodulation of neuronal circuits: back to the future. *Neuron* 76, 1–11. doi: 10.1016/j.neuron.2012.09.010

Mehonic, A., Sebastian, A., Rajendran, B., Simeone, O., Vasilaki, E., and Kenyon, A. J. (2020). Memristors–from in-memory computing, deep learning acceleration, and spiking neural networks to the future of neuromorphic and bio-inspired computing. *Adv. Intell. Syst*. 2, 2000085. doi: 10.1002/aisy.202000085

Meng, X., Bachmann, R., and Khan, M. E. (2020). Training binary neural networks using the bayesian learning rule. *arXiv preprint arXiv:2002.10778*. doi: 10.48550/arXiv.2002.10778

Morris, R. G. M. (2003). Long-term potentiation and memory. *Physiol. Rev*. 358, 643–647. doi: 10.1098/rstb.2002.1230

Neal, R. M. (1996). *Bayesian Learning for Neural Networks*. New York, NY: Springer.

Neftci, E., Mostafa, H., and Zenke, F. (2019). Surrogate gradient learning in spiking neural networks: bringing the power of gradient-based optimization to spiking neural networks. *IEEE Sig. Proc. Mag*. 36, 51–63. doi: 10.1109/MSP.2019.2931595

Nguyen, A., Yosinski, J., and Clune, J. (2015). "Deep neural networks are easily fooled: high confidence predictions for unrecognizable images," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (Boston, MA: IEEE), 427–436.

Osawa, K., Swaroop, S., Jain, A., Eschenhagen, R., Turner, R. E., Yokota, R., et al. (2019). Practical deep learning with bayesian principles. *arXiv preprint 1906.02506*. doi: 10.48550/arXiv.1906.02506

Parisi, G. I., Kemker, R., Part, J. L., Kanan, C., and Wermter, S. (2019). Continual lifelong learning with neural networks: a review. *Neural Netw*. 113, 54–71. doi: 10.1016/j.neunet.2019.01.012

Putra, R. V. W., and Shafique, M. (2022). lpSpikeCon: enabling low-precision spiking neural network processing for efficient unsupervised continual learning on autonomous agents. *arXiv preprint 2205.12295*. doi: 10.48550/arXiv.2205.12295

Rastegari, M., Ordonez, V., Redmon, J., and Farhadi, A. (2016). "XNOR-Net: imagenet classification using binary convolutional neural networks," in *Proceedings of European Conference on Computer Vision* (Amsterdam: Springer), 525–542.

Scikit-Learn library (2020). *Two Moons Dataset*. Available online at: https://scikit-learn.org/stable/modules/generated/sklearn.datasets.make_moons.html

Serrano-Gotarredona, T., and Linares-Barranco, B. (2015). Poker-DVS and MNIST-DVS. Their history, how they were made, and other details. *Front. Neurosci*. 9, 481. doi: 10.3389/fnins.2015.00481

Shrestha, S. B., and Orchard, G. (2018). "SLAYER: spike layer error reassignment in time," in *Advances in Neural Information Processing Systems, Vol. 31* (Montreal, QC).

Simeone, O. (2022). *Machine Learning for Engineers*. Cambridge: Cambridge University Press.

Skatchkovsky, N., Jang, H., and Simeone, O. (2020a). "End-to-end learning of neuromorphic wireless systems for low-power edge artificial intelligence," in *Asilomar Conference on Signals, Systems, and Computers* (Pacific Grove, CA).

Skatchkovsky, N., Jang, H., and Simeone, O. (2020b). "Federated neuromorphic learning of spiking neural networks for low-power edge intelligence," in *ICASSP*

2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (Barcelona: IEEE), 8524–8528.

Skatchkovsky, N., Simeone, O., and Jang, H. (2021). "Learning to time-decode in spiking neural networks through the information bottleneck," in *Advances in Neural Information Processing Systems*, 17049–17059.

Soures, N., Helfer, P., Daram, A., Pandit, T., and Kudithipudi, D. (2021). "Tacos: task agnostic continual learning in spiking neural networks," in *ICML Workshop*.

Stewart, K., Orchard, G., Shrestha, S. B., and Neftci, E. (2020). "Live demonstration: on-chip few-shot learning with surrogate gradient descent on a neuromorphic processor," in *2020 2nd IEEE Int. Conf. on Artificial Intelligence Circuits and Systems (AICAS)* (Genova: IEEE), 128–128.

Vaila, R., Chiasson, J. N., and Saxena, V. (2019). Deep convolutional spiking neural networks for image classification. *arXiv preprint 1903.12272*. doi: 10.48550/arXiv.1903.12272

Wang, H., and Yeung, D.-Y. (2020). A survey on bayesian deep learning. *ACM Comput. Surv.* 53, 1–37. doi: 10.1145/3409383

Zenke, F., and Ganguli, S. (2018). SuperSpike: supervised learning in multilayer spiking neural networks. *Neural Comput.* 30, 1514–1541. doi: 10.1162/neco_a_01086

Zenke, F., Poole, B., and Ganguli, S. (2017). Continual learning through synaptic intelligence. *arXiv preprint 1703.04200*. doi: 10.48550/arXiv.1703.04200

Zhang, T. (2006). Information-theoretic upper and lower bounds for statistical estimation. *IEEE Trans. Inf. Theory* 52, 1307–1321. doi: 10.1109/TIT.2005.864439

Zou, Z., Alimohamadi, H., Zakeri, A., Imani, F., Kim, Y., Najafi, M. H., et al. (2022). Memory-inspired spiking hyperdimensional network for robust online learning. *Sci. Rep.* 12, 7641. doi: 10.1038/s41598-022-11073-3

frontiers | Frontiers in Computational Neuroscience

TYPE Original Research
PUBLISHED 18 November 2022
DOI 10.3389/fncom.2022.1012559

OPEN ACCESS

EDITED BY
Yuqi Han,
Tsinghua University, China

REVIEWED BY
Tianshan Liu,
Hong Kong Polytechnic University, Hong Kong SAR, China
Zbigniew Kowalewski,
AGH University of Science and Technology, Poland

*CORRESPONDENCE
V. Srinivasa Chakravarthy
schakra@ee.iitm.ac.in

RECEIVED 05 August 2022
ACCEPTED 26 October 2022
PUBLISHED 18 November 2022

CITATION
Kumari S, Shobha Amala VY, Nivethithan M and Chakravarthy VS (2022) BIAS-3D: Brain inspired attentional search model fashioned after what and where/how pathways for target search in 3D environment. *Front. Comput. Neurosci.* 16:1012559. doi: 10.3389/fncom.2022.1012559

BIAS-3D: Brain inspired attentional search model fashioned after what and where/how pathways for target search in 3D environment

Sweta Kumari[1], V. Y. Shobha Amala[2], M. Nivethithan[1] and V. Srinivasa Chakravarthy[1]*

[1]Computational Neuroscience (CNS) Lab, Department of Biotechnology, IIT Madras, Chennai, India, [2]IIT BHU, Varanasi, India

We propose a brain inspired attentional search model for target search in a 3D environment, which has two separate channels—one for the object classification, analogous to the "what" pathway in the human visual system, and the other for prediction of the next location of the camera, analogous to the "where" pathway. To evaluate the proposed model, we generated 3D Cluttered Cube datasets that consist of an image on one vertical face, and clutter or background images on the other faces. The camera goes around each cube on a circular orbit and determines the identity of the image pasted on the face. The images pasted on the cube faces were drawn from: MNIST handwriting digit, QuickDraw, and RGB MNIST handwriting digit datasets. The attentional input of three concentric cropped windows resembling the high-resolution central fovea and low-resolution periphery of the retina, flows through a Classifier Network and a Camera Motion Network. The Classifier Network classifies the current view into one of the target classes or the clutter. The Camera Motion Network predicts the camera's next position on the orbit (varying the azimuthal angle or "θ"). Here the camera performs one of three actions: move right, move left, or do not move. The Camera-Position Network adds the camera's current position (θ) into the higher features level of the Classifier Network and the Camera Motion Network. The Camera Motion Network is trained using Q-learning where the reward is 1 if the classifier network gives the correct classification, otherwise 0. Total loss is computed by adding the mean square loss of temporal difference and cross entropy loss. Then the model is trained end-to-end by backpropagating the total loss using Adam optimizer. Results on two grayscale image datasets and one RGB image dataset show that the proposed model is successfully able to discover the desired search pattern to find the target face on the cube, and also classify the target face accurately.

KEYWORDS

attention, memory, human visual system, what and where pathway, convolutional neural network, search in 3D, flip-flop neurons

1. Introduction

Human visual system (HVS) processes a restricted field of view of about 150° in the horizontal line and 210° in the vertical line (Knapp, 1938). However, the eye orientates itself in such a manner that the image of the region of interest falls inside the central part of the retina or fovea to obtain precise information from that part of the visual field. Information from the fovea in high resolution and periphery in low resolution is passed through the visual hierarchy, and the features related to the form, color, and motion are analyzed by respective visual cortical areas. Due to this anatomical constraint, the eye does not process the entire scene at once: the eye makes darting movements called saccades and attends the salient parts of the scene sequentially and integrates the pieces of the image to get a more comprehensive understanding of the scene.

Visual attention is a popular topic in both computer vision and visual neuroscience. Many computational models of visual attention, proposed in the past couple of decades, may be divided into two categories: bottom-up approaches (Le Meur et al., 2006; Gao et al., 2008), and top-down approaches (Gao et al., 2009; Kanan et al., 2009; Borji et al., 2012). The models are basically developed to predict the saliency map, where a brighter pixel has a higher probability of receiving human attention and vice versa. Bottom-up attention is considered to be stimulus driven whereas top-down attention is considered to be task driven, which receives human attention based on the explicit understanding of the image content. Prior attempts in the field of top-down attention mechanisms (Gao et al., 2009; Kanan et al., 2009; Borji et al., 2012) have mainly used non-deep approaches such as the Bayesian approach (Borji et al., 2011), based on a limited understanding of visual attention. In a recent model of visual attention, Mnih et al. (2014) have developed a recurrent attention model (RAM) which takes a glimpse of the attention window as input and uses the internal state of the network to find the next location to focus on in a non-static environment. Their proposed network processes multiple glimpses of windows to attend to a part of the image at different levels of resolutions. Training of their model is done by using the reinforcement learning approach for classification of MNIST dataset for modeling task-driven visual attention. Design of their network is based on fully connected layers, which leads to a rapid increase in computational cost with image size, and therefore the network is perhaps not feasible for more complex real world tasks such as search in a 3D environment.

There is an extensive number of research studies that demonstrate the application of attentional search methods to solve real world problems in 2D space such as image cropping (Xu et al., 2019), object recognition (Gao and Vasconcelos, 2004), object segmentation (Shen et al., 2014; Wang et al., 2015a,b), video understanding (Zhang et al., 2015, 2017; Yang et al., 2016a,b,c), and egocentric activity recognition (Liu et al., 2021, 2022). These models are based on covert attention, where the mental shift of attention occurs at the output activity map without explicit eye movement. But the use of overt visual attention in a 3D environment is still relatively under-explored. Earliest work in 3D target search is the Shape Nets (Wu et al., 2015) where the objective was to voxelize the target and use deep belief networks for training and prediction. Minut and Mahadevan (2001) used Q learning to identify the next movement of the camera (action) out of the eight possible actions in order to focus on the object of interest. At a lower level, this approach uses histogram back projection color maps and symmetry map to identify the objects. Unlike reinforcement learning based approaches, the model proposed by Kanezaki et al. (2018) named RotationNet, focuses on convolutional neural networks (CNNs) based approaches where multiple views of the object are taken into consideration for learning. The model predicts the class and the pose (orientation) of the object of consideration. This was an improvement over the previous CNN based networks, that recognized the object but failed to predict the pose. The model yielded an accuracy of 94% on Modelnet40 dataset (Wu et al., 2015) consisting of 40 categories including chair, airplane, etc. Multiview CNN (Su et al., 2015) was one of the earliest attempts in 3D object recognition that acts as a precursor of the RotationNet.

In the model known as the SaccadeNet developed by Lan et al. (2020), a model closest in approach to ours, four module classifiers are used to recognize objects. These modules are—center attentive module, the corner attentive module, the attention transitive module, and the aggregation attentive module. Each module works on identifying the main key points of the object of interest, perhaps the center, corners, attend object centers, and bounding boxes. This technique works similar to the proposed saccade approach inspired by human visual search. The drawback is that it works mainly on 2D inputs. While performing a target search in a 3D environment, the model needs to predict the next location of the camera and identify the object that the camera is looking for. To perform such search tasks in 3D space, time is one of the constraints which depends on the network design and input.

We propose a Brain Inspired Attentional Search model in 3D space (BIAS-3D) that takes the attentional glimpse instead of the entire image. The design of the model contains convolutional layers instead of fully connected layers to extract features and contains Elman and Jordan recurrence layers as well as JK-flip-flop recurrence layer (Sweta et al., 2021) instead of Long Short Term Memory (LSTM) to integrate the temporal attention history in the network. To generate the attentional glimpse, a set of concentric attention windows is used by taking the inspiration from Ba et al. (2014), Mnih et al. (2014), and Kahou et al. (2017).

The proposed model has the following brain-inspired features: (1) it has separate channels for image classification and camera movement, analogous to the "What pathway" and

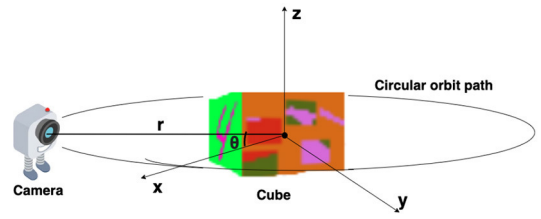


FIGURE 1
Simulated environment: "*r*" is the radius of the circular orbit around the cube or the line connecting the camera position and origin of the spherical coordinate system. Polar angle "ϕ" is assumed to be fixed at 0, and azimuth angle "θ" is varying.

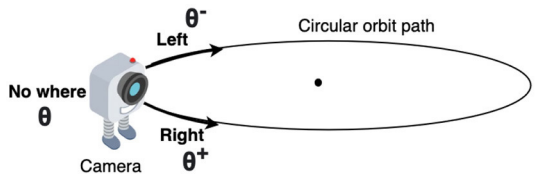


FIGURE 2
Direction of all three movements of the camera on the orbital path, supposed to be predicted by the model.

"Where pathway" in HVS; (2) it incorporates three types of recurrence connections: (a) Local recurrence connection of Elman type (Elman, 1991), (b) Global recurrence connection of Jordan type (Jordan, 1986), (c) Flip-flop neurons (Holla and Chakravarthy, 2016) that are capable of storing information for a long time. In this study, we show that the BIAS-3D is effectively able to learn task-specific strategies and identify the targets. Our simulation results successfully show that an attention-based network can be an efficient approach in dealing with target search tasks in a 3D environment, which is demonstrated by using 3D Cluttered MNIST Cube dataset, 3D Cluttered QuickDraw Cube dataset, and 3D Cluttered RGB MNIST Cube dataset.

2. The proposed approach

2.1. Environment overview

The virtual environment used in this study is created using OpenGL (Segal and Akeley, 2010) (Figure 1). The environment contains a cube placed at the origin of a spherical coordinate system and a camera placed on a circular orbit around the cube. On this orbit of radius "r," the camera revolves around the cube, always looking inwards toward the center of the cube (Figure 2). As the camera moves on the orbit, it processes the views of the cube it captures and searches for the face that has a target pattern displayed on it (Figure 3B). The possible movements of the camera on the orbit are: "move right" (θ^+), "move left" (θ^-), or "do not move" (θ; Figure 2).

2.2. Architecture overview

The architecture design of the proposed brain inspired attentional search model in 3D space (BIAS-3D) is depicted in Figure 3D. The model takes two inputs: (i) the *attentional glimpse* which consists of the contents at different resolutions and sizes of the attended region, where multiple concentric attention windows are applied to the center location of the camera view, and (ii) the *camera-position* in the form of a point on the unit circle at an angle θ or the azimuth angle of the camera position on its circular orbit. The model predicts two outputs at each timestep: (i) the next location of the camera on the orbit, and (ii) the class of the object seen in the camera view. The model consists of three parallel pipelines (Figure 3D): (i) the upper pipeline processes the class information of the object seen in the view, called the Classifier Network, (ii) the middle pipeline processes the location of the target object over the cube and predicts the next position of the camera, called the Camera Motion Network, and (iii) the lower pipeline, which incorporates the camera position into the high level features of the Classifier Network and the Camera Motion Network, is called the Camera-Position Network. Outputs of all the three pipelines are concatenated in one flattened layer which connects with a fully connected layer, and the output of the fully connected layer passes through one linear output layer and one softmax output layer in parallel. Linear output layer computes the *Q*-values corresponding to the three actions that can be taken by the camera, and softmax output layer computes the classification probabilities of the object present inside the attentional glimpse. A Deep Q-learning algorithm is applied to train the model and learn the optimal policy for camera control (Fan et al., 2020). As the model takes the sequential input, the network requires memory to store the past information of the following details: (i) the extracted features of the attentional glimpse, (ii) its corresponding location on the cube, and (iii) the camera position. For storing this input history, the model uses three recurrent neural features: the flip-flop neuron layer (Holla and Chakravarthy, 2016), Elman and Jordan recurrence layers.

2.3. The BIAS-3D

The proposed attention model is a deep neural network, which has three pipelines: Classifier Network, Camera Motion Network, and Camera-Position Network (Figure 3). The classifier network consists of three convolutional layers (Convs), three maxpool layers, and one fully connected (FC) Elman Jordan recurrence layer (FCEJ). The camera

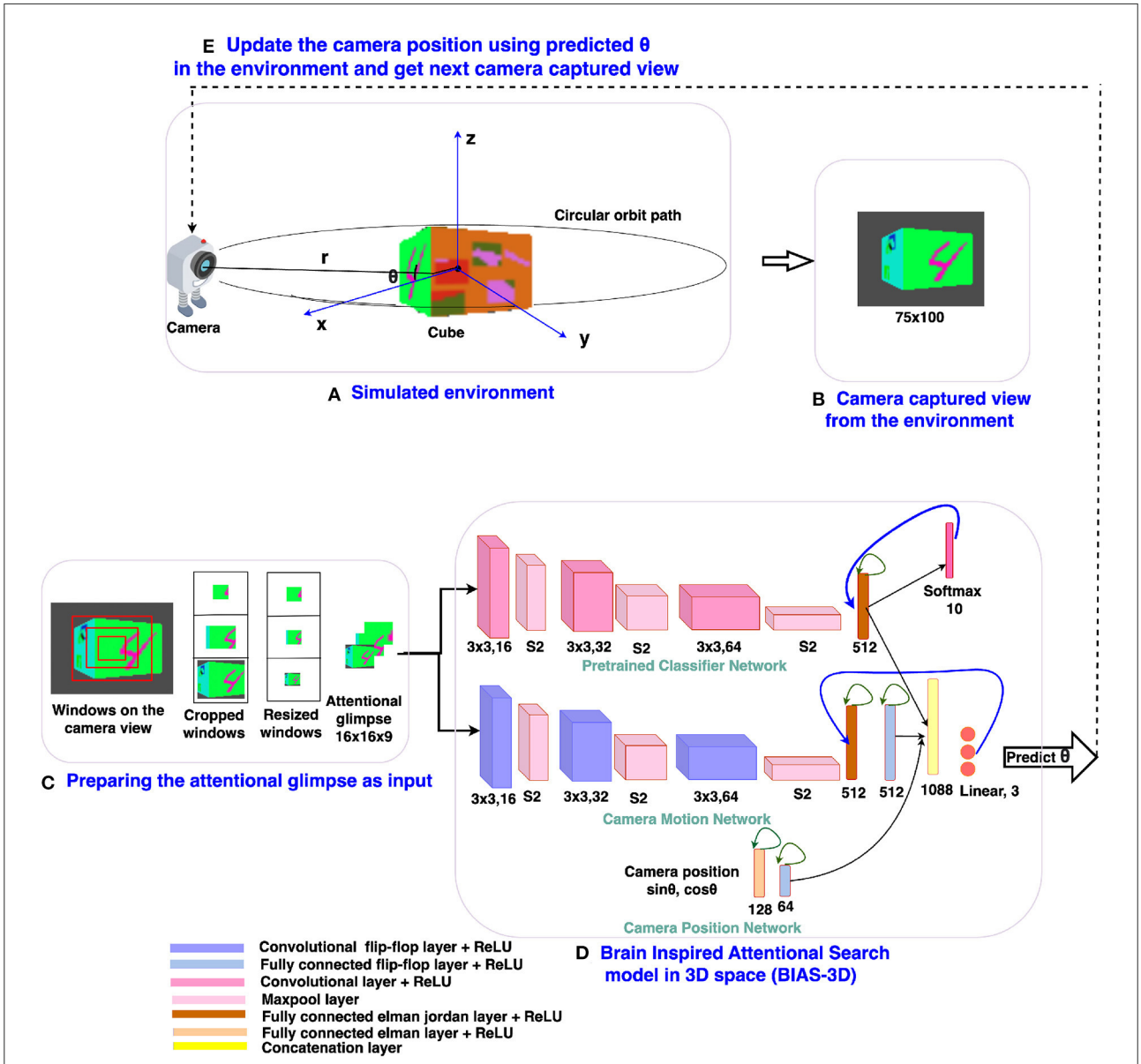


FIGURE 3
The design of the BIAS-3D: **(A)** Simulated environment with the 3D Cluttered RGB MNIST cube and the camera, **(B)** camera captures the image from the environment, **(C)** the attentional glimpse generated from the camera captured view, **(D)** the BIAS-3D predicts the class of the target and position of the camera or "θ," and **(E)** Update the camera position using the predicted "θ."

motion network consists of three convolutional flip-flop layers (ConvJKFF), three maxpool layers, one FCEJ layer, and one FC flip-flop layer (FCJKFF). The camera-position network consists of one FCEJ layer, and one FCJKFF layer; this network encodes the revolving direction of the camera. The aforementioned layers are discussed in greater detail in the following paragraphs.

Convolutional layers (Convs) are used to extract features by sharing the weights across different spatial locations. Input and output to the Conv layer are 3D tensors, called feature maps. The output feature map is calculated by convolving the input feature map with 3D linear filters. Then a bias term is added up into the convolved output. In this paper, the bold notations in all the equations stands for the matrix or the matrices. If $\mathbf{X}^{l-1}$ is the input feature map of *lth* Conv layer and $\mathbf{W}^l$ and $\mathbf{b}^l$ are filter weights and bias terms, respectively, then the output feature map $\mathbf{X}^l$ of *lth* layer is calculated *via* Equation-1:

$$\mathbf{X}^l = \mathbf{X}^{l-1}\mathbf{W}^l + \mathbf{b}^l, \quad (1)$$

$$l = 1, ..., L,$$

$$\mathbf{X}^0 = \mathbf{I},$$

In the above equation, L is the total number of layers, $\mathbf{X}^0$ is the input image **I** to the first Conv layer. The output feature maps from each Conv layer are passed through a non-linear ReLU activation function (Nair and Hinton, 2010) (equation-2).

$$\mathbf{f}(\mathbf{X}) = max(0, \mathbf{X}) \tag{2}$$

The output feature maps from the activation function, are normalized using local response normalization (LRN) (Krizhevsky et al., 2012). LRN normalizes the feature maps within the channels and is a form of lateral inhibition (Equation 3).

$$\mathbf{N}^f_{x,y} = \mathbf{X}^f_{x,y} / \left(k + \alpha \sum_{j=max(0,f-c/2)}^{min(C-1,f+c/2)} (\mathbf{X}^j_{x,y})^2 \right)^{\beta} \tag{3}$$

where $\mathbf{X}(x, y)$ and $\mathbf{N}(x, y)$ are the pixel values at (x, y) position before and after normalization, respectively, f denotes the filter. C stands for the total number of channels. The constants k, α, β, and c are hyperparameters. k is used to avoid "division by zero," α is a normalization constant, while β is used as a contrasting constant. The constant c is used to define the length of the neighborhood, that is, the number of consecutive pixel values need to be considered while calculating the normalization. $(k, \alpha, \beta, c) = (0, 1, 1, C)$ case is considered as the standard normalization. Normalized features from the Conv layer are passed through the maxpool layer (Scherer et al., 2010). Several convolutional layers and pooling layers are assembled alternately across depth in the first three Conv or ConvJKFF layers in both classifier and camera motion networks (Figure 3D).

To implement the Elman recurrence layer (Elman, 1991), the output vector of the FC layer at time "$t-1$" is stored in a context layer and the content of the context layer is fed back to the same FC layer at time "t," named as FC Elman recurrence layer which is a short range storage connection. The Elman recurrence layer is implemented only in the first FC layer of all three networks. Similarly, to implement the Jordan recurrence layer (Jordan, 1986), the output vector of the last FC layer at time step "$t-1$" is stored in a context layer and this context layer is fed back to the first FC layer at time step "t" in their corresponding pipeline, named as FC Jordan recurrence layer which is a long-range storage connection. In this way, the first FC layer in Classifier and Camera Motion Networks has both Elman and Jordan recurrences; so we call this layer a FCEJ layer. The computation of FCEJ is shown in the following (Equation 4)

$$\mathbf{X}^l_t = f\left(\mathbf{X}^{l-1}_t \mathbf{W}^{l-1,l} + \mathbf{X}^l_{t-1} \mathbf{W}^{l,l} + \mathbf{X}^L_{t-1} \mathbf{W}^{L,l} + \mathbf{b}_l \right) \tag{4}$$

In Equation (4), $\mathbf{X}^{l-1}_t$ is the output of the $l-1th$ layer at time "t" and going as input to the *lth* layer at time "t" (FC layer). $\mathbf{X}^l_{t-1}$ is the output of the *lth* layer at time "$t-1$" and going as input to the same *lth* layer at time "t" (Elman recurrence layer). $\mathbf{X}^L_{t-1}$ is the output of the *Lth* layer at time "$t-1$" and going as input to the *lth* layer at time "t" (Jordan recurrence layer). $\mathbf{W}'s$ and b are the corresponding weights and bias, respectively. f is the ReLU activation function.

Memory of the past information in the layers of the proposed network is stored using a third mechanism—the flip-flop neurons (Holla and Chakravarthy, 2016). A flip-flop is a digital electronic circuit to store state information. There are four types of digital implementations of flip-flops: D flip-flops, Toggle flip-flops, SR flip-flops, and JK flip-flops (Roth et al., 2020). In the proposed network, JK flip-flop neurons are used in place of LSTM neurons because of the performance advantage shown in Holla and Chakravarthy (2016) and Sweta et al. (2021). In both of these papers, the experiments conducted on the sequential data shows that flip-flop neurons outperform the LSTM neurons, using only half the number of training parameters in comparison to LSTM. Likewise, to get the advantage of fewer parameters and better performance, in the current study we used the JK flip-flop neuron. The JK flip-flop neuron uses two gating variables with "**J** and **K**" nodes, whereas LSTM uses four gating variables. In this paper, the term flip-flop will be used to refer to JK-flip-flop. Furthermore, the flip-flop neurons are considered similar to the UP/DOWN neurons found in the prefrontal cortex (PFC), responsible for working memory (Gruber et al., 2006).

In the proposed model, the flip-flop layer is designed in two ways: flip-flop neurons in convolutional layer (named as "convolutional flip-flop layer" or ConvJKFF), and flip-flop neurons in the FC layer (named as "fully connected flip-flop layer" or FCJKFF). Training rules of these flip-flop neurons in the network were also developed. The two gate outputs "**J**" and "**K**," the hidden state of the JK flip-flops, and the final flip-flops output are computed by using Equations (5–7, respectively) below.

$$\mathbf{J} = \sigma\left(In_t \mathbf{W}_j\right), \mathbf{K} = \sigma\left(In_t \mathbf{W}_k\right) \tag{5}$$

$$\mathbf{H}_t = \mathbf{J}.\left(\mathbf{1} - \mathbf{H}_{t-1}\right) + (\mathbf{1} - \mathbf{K}).\mathbf{H}_{t-1} \tag{6}$$

$$\mathbf{O}_t = \tanh\left(\mathbf{H}_t \mathbf{W}_{out}\right) \tag{7}$$

In Equation (6), "." stands for the pointwise multiplication. $In_t = (\mathbf{X}_t; \mathbf{H}_t)$ is the input to the flip-flop layer, where $\mathbf{X}_t$ is the output from the previous layer and $\mathbf{H}_t$ is the hidden state at time "t," which initialize with ones at time 0. **1** is a matrix of ones. **J** and **K** are the gate variables, which has weight parameters $\mathbf{W}_j$ and, $\mathbf{W}_k$, respectively. $\mathbf{O}_t$ is the output of the flip-flop layer at time 't'. To train the flip-flop neurons, the partial derivatives w.r.t **J** and **K** were used to backpropagate the corresponding **J** and **K** nodes (Equation 8).

$$\frac{\partial \mathbf{H}_t}{\partial \mathbf{J}} = 1 - \mathbf{H}_{t-1}; \frac{\partial \mathbf{H}_t}{\partial \mathbf{K}} = -\mathbf{H}_{t-1} \tag{8}$$

2.4. Implementation detail

2.4.1. Camera motion network

The camera motion network takes the attentional glimpse of size $h \times w \times a$ as input, where "h" is the height, "w" is the width, and "a" is the number of the cropped attention windows. Here, the number of attention windows is chosen to be 3 (i.e., $a = 3$). The size of one attention window is twice the previous attention window's size. Similar multiscale concentric attention windows were used in other models (Mnih et al., 2014; Haque et al., 2016; Shaikh et al., 2019). All the attention windows, except the smallest one, get resized to the size of the smallest attention window. For example, to generate the attentional glimpse where $h = 16$, $w = 16$, and $a = 3$ from location $y = 35$ and $x = 50$ in the given image of size 75×100, the first, second, and third attention windows are cropped out of size 16×16, 32×32, and 50×50 from pixel location $(y, x) = (27$ to 43, 42 to 58), $(y, x) = (19$ to $51, 34$ to 66), and $(y, x) = (10$ to $60, 25$ to 75), respectively. The second and third cropped attention windows are resized to the size of the first cropped attention window, which is 16×16. After resizing, all the three attention windows are stacked together, which finally becomes an attentional glimpse of size $16 \times 16 \times 3$. This type of attentional glimpse having a size of $h \times w \times a$ shown in Figure 3C is passed to the first ConvJKFF layer of 16 kernels, each of size 3×3, of the classifier network (shown in the top pipeline of the BIAS-3D in Figure 3D). The spatial dimension of the features generated from the first ConvJKFF layer is $h \times w \times 16$, which are normalized using LRN, and passed into ReLU activation function. Output from ReLU activation function is passed to the maxpool layer with a window of size 2×2 and stride by 2, which translates the feature's spatial dimensions into $h/2 \times w/2 \times 16$. The translated feature maps are passed as input to the second ConvJKFF layer of 32 kernels, each of size 3×3, to extract the higher level features of size $h/2 \times w/2 \times 32$. Then, similar to the previous layer, features generated from the second ConvJKFF layer are passed through the LRN layer, ReLU activation function, and maxpool layer with a window of size 2×2 and stride 2. After passing into the maxpool layer, feature maps of size $h/4 \times w/4 \times 32$ are generated, which are further converted into a flattened layer to reshape the 3D features into 1D vectors. The flattened vectors are passed through one FCEJ layer of 512 neurons, which is followed by one FCJKFF layer of 512 neurons. Output from the FCJKFF layer of the camera motion network is concatenated with the output vectors of the last layer of the other two channels.

2.4.2. Classifier network

The classifier network gets the same attentional glimpse as input which has been passed to the camera motion network. This network predicts the class of the object present in the attentional glimpse. The object present in the attentional glimpse may belong to one of the "$n + 1$" classes, where "n" classes are the object or target class and one is the nontarget or clutter class. The network consists of 3 Conv layers followed by one FCEJ layer. The first Conv layer of 16 kernels of size 3×3 generates the feature maps of spatial dimension $h \times w \times 16$. Generated features are passed through the LRN layer and ReLU activation function. After this, the maxpool layer with a window of size 2×2 and stride by 2 has been applied to the output of ReLU activation function, which gives the feature maps of spatial dimension $h/2 \times w/2 \times 16$. Then, the feature maps are passed through a second Conv layer of 32 kernels, each of size 3×3, LRN layer, ReLU activation function, and maxpool layer with a window of size 2×2 and stride by 2. Feature maps of spatial dimension $h/4 \times w/4 \times 32$ are passed through a third Conv layer of 64 kernels each of size 3×3 with ReLU activation function, which further generates the feature maps of size $h/4 \times w/4 \times 64$. Then the flattened layer reshapes the 3D tensor of feature maps into vectors, and these vectors are input to the FCEJ layer of 512 neurons. The output of the FCEJ layer gets concatenated with the output vectors of the last layer of the camera motion network and the camera position network.

2.4.3. Camera position network

The camera's position in the environment is inferred from the spherical coordinates, where the camera is assumed, as described before, on a circular object centered on the origin, and the center of the cube is located at the origin. The camera's position is defined by three variables: ("r," "θ," "ϕ"), where "$r = R$" is the radius of the circular orbit of the camera or line connecting the camera point and the origin of the spherical coordinate system, "θ" is the azimuth angle and "ϕ" is the polar angle of the spherical coordinate system. In the current simulated environment, the camera moves only in one degree of freedom, that is "θ." Therefore, "$r = R$" and "$\phi = 0$" are considered to be constant. Only "θ" varies as the camera moves on a circular orbital path around the origin of the spherical coordinate system or the cube. In the camera position network, sinusoidal functions of "θ" are passed as input to the first FC Elman (FCE) layer having 128 neurons, followed by one FCJKFF of 64 neurons. Output from the FCJKFF layer is concatenated to the output vectors of the last layer of the classifier network and the camera motion network.

Outputs from three pipelines are concatenated in one common flattened layer, which further connects with two output layers in parallel. One output layer with linear activation function is responsible to predict one direction out of the three considered directions in which the camera will move on the orbit to look and locate the target face present in the given cube. The other output layer with softmax activation function is responsible to predict the class of the object seen on the view of the camera.

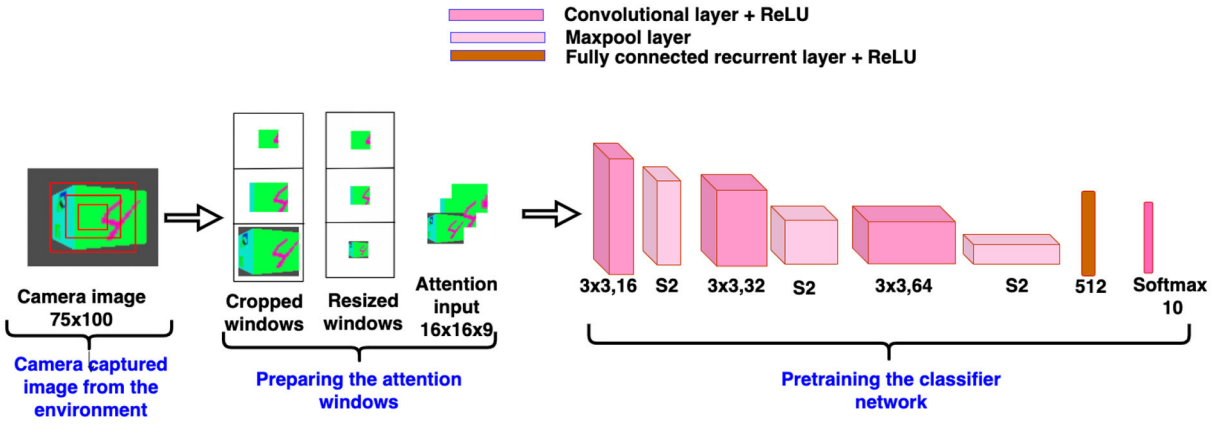


FIGURE 4
Pretrained classifier network.

2.4.4. Training and testing

Tensorflow framework is used to implement the proposed attention model. Xavier's initialization (Glorot and Bengio, 2010; Equation 9) with random normal distribution is used to initialize the weights for each layer of the three networks. The Xavier initialization is able to avoid the exploding or vanishing gradients (Bengio et al., 2001) problem by fixing the variance of the activations across each layer as the same.

$$\mathbf{W}^l = \mathbf{N}\left(0, \frac{2}{m^{l-1} + m^l}\right) \tag{9}$$

where, $\mathbf{N}$ stands for the normal distribution. m^{l-1} and m^l is the number of neurons in the previous layer and current layer, respectively. $\mathbf{W}^l$ denotes the weights at *lth* layer with Xavier initialization.

Before training the model, the classifier network is pretrained on the camera captured views. To pretrain the classifier network, we collect views of the simulated environment by explicitly revolving the camera from 0 to 360°, where 0° is assumed to be exactly at the front of the face containing the target object. Advancing in steps of 9 degrees over the range of 0–360°, a total of 40 views is collected for each cube in the dataset. Views between −45 to +45 range are labeled as one of the "n classes" and views between +46 to +180 and −46 to −180 range are labeled as "background class." Therefore, the total number of classes present in the dataset is $n + 1$. To make the views data uniform, the same number of views of the background class are chosen randomly as the number of views of the other class. The classifier network is pretrained on such views of targets and background or nontarget class. We assume that the camera's focus is always fixed in the center of the view. Therefore, we create a glimpse of three concentric windows from the center location of the camera view. Detailed architecture of the pretraining classifier network is shown in Figure 4. The classifier network without recurrent layers in the BIAS-3D is pretrained on the glimpse of the camera views. Total loss of the model is calculated in two parts: one is classification loss, calculated using the cross-entropy loss function (Goodfellow et al., 2016) and the other one is prediction loss, calculated using mean square error of temporal difference (Sutton and Barto, 1998). Equations of the both loss functions are shown in Equations (10) and (13).

$$\mathbf{L}_{ce} = -\sum_{i=1}^{n+1} \mathbf{d}_i \log\left(\mathbf{p}_i\right) \tag{10}$$

In Equation (10), $\mathbf{d}_i$ denotes the desired classification probability and $\mathbf{p}_i$ denotes the predicted classification probability of *ith* class. "$n + 1$" is the total number of classes that are present in the dataset including background class. Here, the camera is assumed as an agent and the agent learns a defined policy of the reward function (Equation 11) (Armstrong and Murlis, 2007). When the agent is in the current state, Q-values of all three actions are predicted by passing the information of the current state (like the attention input and the θ value of the camera) into the deep neural network. Based on the predicted Q-value of all the actions in the current state, the agent makes an action decision using a softmax action selection policy (Abed-alguni, 2018). In this policy, the predicted Q-values are passed through a softmax activation function to produce the action probabilities. The action with the highest probability is selected and performed by the agent in the current state of the environment. After performing the action, the current state is updated to the next state and then the agent receives a reward, either "1" or "0" depending on the reward policy shown in Equation (11).

$$\mathbf{r} = \begin{cases} 1 \;\; \arg\max_{i \in n}(\mathbf{p_i}) == \arg\max_{i \in n}(\mathbf{d_i}) \\ \quad max(\mathbf{p}) \geq \lambda \\ \\ 0 \;\; otherwise \end{cases} \tag{11}$$

Apart from the softmax action selection policy, we used a race model (Rowe et al., 2010) which ensures that the selected action is correct. Race models have been applied in many behavioral, perceptual, and oculomotor decisions and such decisions are based on trial-to-trial modifications in a race among all the responses (Carpenter and McDonald, 2007). Race model works based on two neurophysiological evidences to show the relatedness. Firstly, if monkeys are trained to make their decision on coherently moving direction of dots, accumulating neuronal activity is formed that mirrors the decision even when there is no coherent motion. Here, both choices are equally rewarded (Churchland et al., 2008). Secondly, the decision threshold is considered constant for a selected action, regardless of its being a specifically cued action (Roitman and Shadlen, 2002). We have taken the motivation to apply the race model based on the second evidence. The action predicted by the network is the action which crosses the threshold, λ, first and if the action predicted is correct, the agent gets reward "1"; it otherwise gets reward "0."

The Q-values of the actions in the next step are estimated by passing the next state information into the target network, where the target network is the separate copy of the networks of the model. Target Q-value is calculated by adding the current state reward and maximum of the next state Q-values multiplied with a discount factor γ. Discount factor defines how much the current state Q-value depends on the future reward. Now, the temporal difference (TD) is calculated by calculating the difference between the target Q-value and the predicted Q-value (Equation 12).

$$\mathbf{TD} = \left(\mathbf{r} + \gamma * \mathbf{Q_{max}}\left(S_{t+1}\right)\right) - \mathbf{Q}\left(S_t\right) \tag{12}$$

$$\mathbf{L}_{mse} = \frac{1}{n}\sum_{i=0}^{n}\mathbf{TD}^2 \tag{13}$$

$$\mathbf{l}_{total} = \mathbf{L}_{ce} + \mathbf{L}_{mse} \tag{14}$$

where, $\mathbf{r}$ is the reward which the agent gets while going from the state $\mathbf{S}_t$ to the state $\mathbf{S}_{t+1}$. $\mathbf{Q}\left(S_{t+1}\right)$ and $\mathbf{Q}\left(S_t\right)$ is the Q-value of the state $\mathbf{S}_{t+1}$ and, $\mathbf{S}_t$, respectively. γ is the discount factor. Then these two losses, the cross-entropy loss of the classifier network (Equation 10), $\mathbf{L}_{ce}$ and the mean square error of temporal difference of the camera motion network (Equation 13), $\mathbf{L}_{mse}$, are added up to get the total loss (Equation 14). The total loss is back propagated into the network (Voleti, 2021). The network parameters are updated by using the mini-batch Adam optimizer (Kingma and Ba, 2014). L2 regularization (Kratsios and Hyndman, 2020) is used to avoid the overfitting problem of the network. During inference, the camera starts from a random location and moves toward the target face of the cube. Once it finds the target face, the camera continues to fixate around that face. The number of trainable parameters of the model are 2,668,362. The model achieves a processing speed of 0.0001 s per input image on a workstation with an NVIDIA GeForce GTX 1,080Ti 11 GB GPU, i7-8700 CPU @ 3.20 GHz 3.19 GHz, 64-bit operating system, and 32.0 GB RAM.

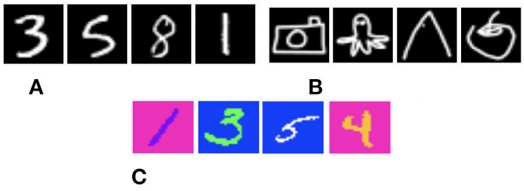


FIGURE 5
A sample of image datasets are shown here. **(A)** 28 × 28 MNIST digits. **(B)** 28 × 28 QuickDraw. **(C)** 28 × 28 RGB MNIST digits.

3. Simulation results

We evaluate our model on "painted cube" data, where each cube has a target object on one vertical face and nontarget objects on the other three vertical faces. The model is supposed to move the camera around the cube on a circular orbit and search the target object image present on one of the four vertical faces of the cube. For target object image, we used image datasets. Totally three 3D Cluttered Cube datasets were considered in the experiment. The cube datasets were generated using their related image data. Grayscale MNIST digit image dataset, QuickDraw image dataset, and RGB MNIST digit image dataset (Samples are shown in Figures 5A–C, respectively) were used to generate cube datasets like 3D Cluttered Grayscale MNIST Cube dataset, 3D Cluttered QuickDraw Cube dataset, and 3D Cluttered RGB MNIST Cube dataset respectively. The first two of these are cube datasets with grayscale images, and the last one is a cube dataset with RGB images. Based on the grayscale and RGB cube datasets, we designed the experiments in two parts: one part of the experiment shows the target search capability of the proposed model on the cubes which has all four vertical faces of grayscale images (called grayscale cubes) and the other part of the experiment shows the target search capability of the model on the cubes which has all vertical faces of RGB images (called RGB cubes).

3.1. Searching on grayscale cubes

In the first part of the experiment, we evaluated our model on two datasets of Grayscale cubes. For that, we used two different datasets of grayscale images: MNIST handwritten digits (LeCun et al., 1998) and QuickDraw (Jongejan et al., 2016). Both datasets with 10 different classes contain 48,000 examples in the training set, 12,000 examples in the validation set, and 10,000

TABLE 1 Accuracy on testing set of all three datasets.

Dataset	Testing accuracy (%)
3D cluttered grayscale MNIST cube dataset	95.6
3D cluttered grayscale QuickDraw cube dataset	83
3D cluttered RGB MNIST cube dataset	91.5

examples in the testing set. We have generated a 3D Cluttered MNIST Cube dataset using MNIST dataset. To generate such a cube dataset, each of the cubes were created with a 28×28 MNIST digit image (target) on one vertical face and 28×28 a random clutter image (nontarget) on the other three vertical faces. In this experiment, the bottom and top faces of the cube are not considered for searching. Similarly, a 3D Cluttered QuickDraw Cube dataset was generated using QuickDraw image dataset.

Once the cube datasets are generated, we place the cube in the environment in such a way that the center of the cube is at the origin of the spherical coordinate system. Then the camera is placed at a random value of azimuth angle "θ" at initial time ($t = 0$). The polar angle "ϕ," and radius "r" are set to 0, and 2.5, respectively. The camera placed at (r, θ, ϕ) captures the view of size 75×100. Then a glimpse is extracted from the center location of the captured view. To extract the glimpse, three concentric windows of size 16×16, 32×32, and 50×50 are cropped out from the center of the view. After cropping out, windows of size 32×32 and 50×50 are resized into the size of 16×16. Then resized windows with the smallest window of size 16×16 are arranged together across depth to generate an attentional glimpse of dimension $16 \times 16 \times 3$. Since the image size in the QuickDraw image dataset is same as the image size in the MNIST dataset, the same dimensions of the camera view and attentional glimpse were considered in case of the 3D Cluttered QuickDraw Cube dataset.

The proposed model takes the attentional glimpse of size $16 \times 16 \times 3$ from the center location of the image view of the camera of size 75×100. Achieved accuracy on both grayscale datasets are listed in Table 1. The results of the camera's movement predicted by our model in the testing set are shown in Figures 6–9. In this figure, images of the camera view of dimension 75×100 are shown in one row and their corresponding plots for predicted classification probabilities for that view (dotted dashed-blue curve) and ground truth target classification probabilities (green curve) are shown in the row just below. At the bottom of the plots, timestep and ground truth target class labels are denoted by using variables "t" and "c," respectively. In the row of images of the camera view, three concentric red windows depict the glimpse.

The model has the ability to move the camera to the position where the target face of the cube is visible from the camera. For example, in Figure 6, the class of digit 2 in the fourth image of the first row has the view of nontarget or clutter face at timestep $t = 0$ and its corresponding predicted classification probabilities shown in the plot just below that image is low for all classes. But at timestep $t = 1$ (θ is decided by the model), the camera has moved toward the right and has seen some part of the target face that has the digit 2. At the same time, the highest of the predicted classification probabilities is for digit 2. The camera again moved to the right at timestep $t = 2$, where an adequate part of the digit 2 on the cube face is visible (first image in fourth row of Figure 6) and therefore, the maximum value of predicted classification probabilities is close to 1 for digit 2, which crosses a testing threshold of value 0.95. Similarly, for the other digits, the camera starts moving appropriately, searching for the target. The camera stops moving when the maximum value of the predicted classification probabilities crosses the testing threshold. The testing threshold is set based on the feature complexity of the image datasets.

In the case of the 3D Cluttered QuickDraw Cube dataset, we can observe the same search behavior of the camera. For example, in Figure 8, class 5 (bicycle) in the third image of the tenth row has the camera view showing non-target objects on the cube face at timestep $t = 0$ and its corresponding predicted classification probabilities shown in the plot just below the that image is low for all classes. At the next timestep ($t = 1$), the camera has moved to the left and the camera continues to move in the left direction 3 more times even though the target is not visible. At timestep $t = 4$, a very small part of the bicycle is visible (second image in the thirteenth row of Figure 9) and at this time the classification probability for class 5 or bicycle becomes the highest. The camera stops moving once the maximum value of the predicted classification probabilities crosses a testing threshold of value 0.85.

3.2. Searching on RGB cubes

In the second part of the experiment, we evaluated our model on RGB cubes to investigate that the model is able to search for the target object on the cube face even in the case of color images. To this end, we generated a cube dataset using RGB MNIST image dataset. Here, we first create the RGB MNIST digit image dataset by assigning different colors to the digits and the background of the images available in Grayscale MNIST digit image dataset (LeCun et al., 1998). Our created RGB MNIST handwriting digit dataset is available in this link. The dataset with 10 different classes contains 48,000 examples in the training set, 12,000 examples in the validation set, and 10,000 examples in the testing set. Once the image dataset is ready, we generate a 3D Cluttered RGB MNIST Cube dataset using RGB MNIST image dataset. To generate a 3D Cluttered RGB MNIST Cube dataset, each of the cubes is created with a $28 \times 28 \times 3$ RGB MNIST image (target) on one vertical face and

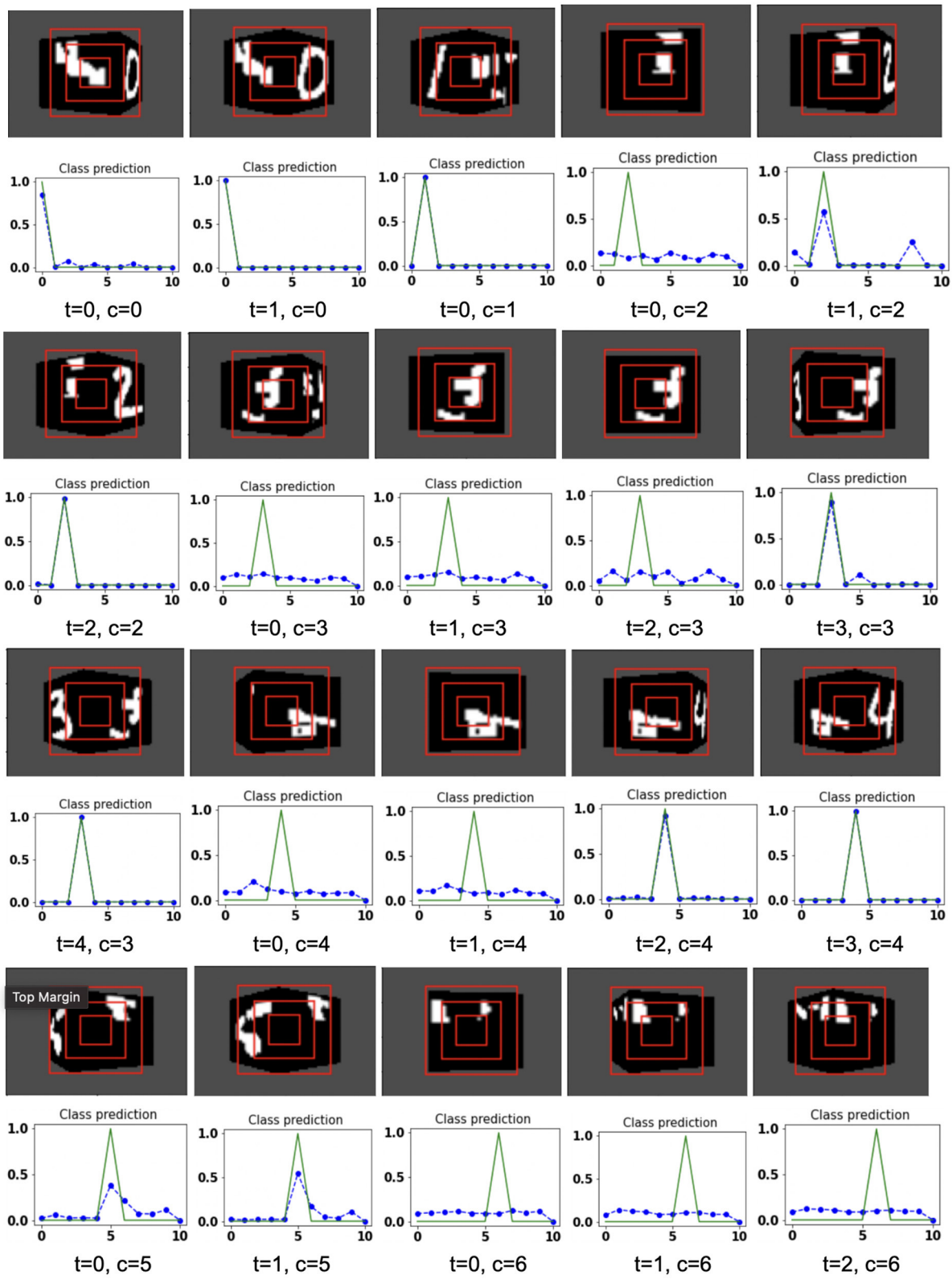


FIGURE 6
Illustrates the camera movements around the cube to search the target face in the view of size 75 × 100, predicted by our model in 3D Cluttered Grayscale MNIST Cube dataset. For each class at time *t*, there is a movement (shown in the row of camera view images) and corresponding classification probabilities (shown in the row of plots). In the row of camera view images, the three concentric red windows depict the glimpse at the center of the view image. In the plot corresponding with the above view image, the green curve is the desired classification probabilities and the dotted dashed-blue curve is the predicted classification probabilities at time *t*.

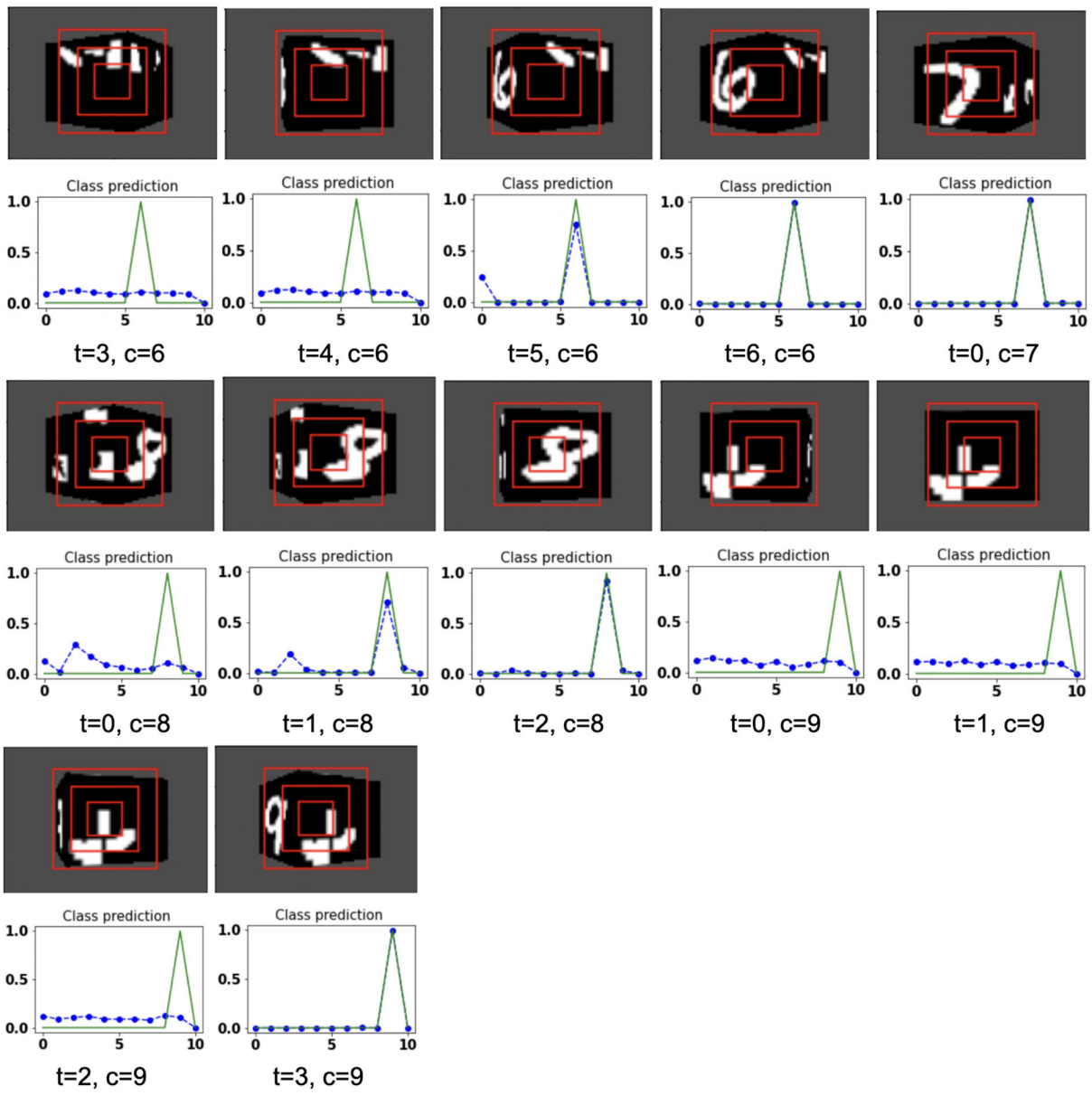


FIGURE 7
Illustrates the camera movements around the cube to search the target face in the view of size 75 × 100, predicted by our model in 3D Cluttered Grayscale MNIST Cube dataset. For each class at time *t*, there is a movement (shown in the row of camera view images) and corresponding classification probabilities (shown in the row of plots). In the row of camera view images, the three concentric red windows depict the glimpse at the center of the view image. In the plot corresponding with the above view image, the green curve is the desired classification probabilities and the dotted dashed-blue curve is the predicted classification probabilities at time *t*.

$28 \times 28 \times 3$ random clutter image (non-target) on the other three vertical faces.

The model is evaluated by placing the RGB cube in the environment in the same way of grayscale cube datasets. The camera captures the view of size 75 × 100 of the 3D Cluttered RGB MNIST Cube. The camera extracts the glimpse from the center of the captured view. To extract the glimpse, three concentric windows of size 16 × 16, 32 × 32, and 50 × 50 are cropped out from the center of the view to generate an attentional glimpse of size 16 × 16 × 9. The proposed model takes the attentional glimpse of size 16 × 16 × 9 from the center location of the image view of the camera of size 75 × 100 in case of 3D Cluttered RGB MNIST Cube dataset. The achieved accuracy on the RGB cube dataset is listed in Table 1. The results of the camera's movement predicted by our attention model in the testing set are shown

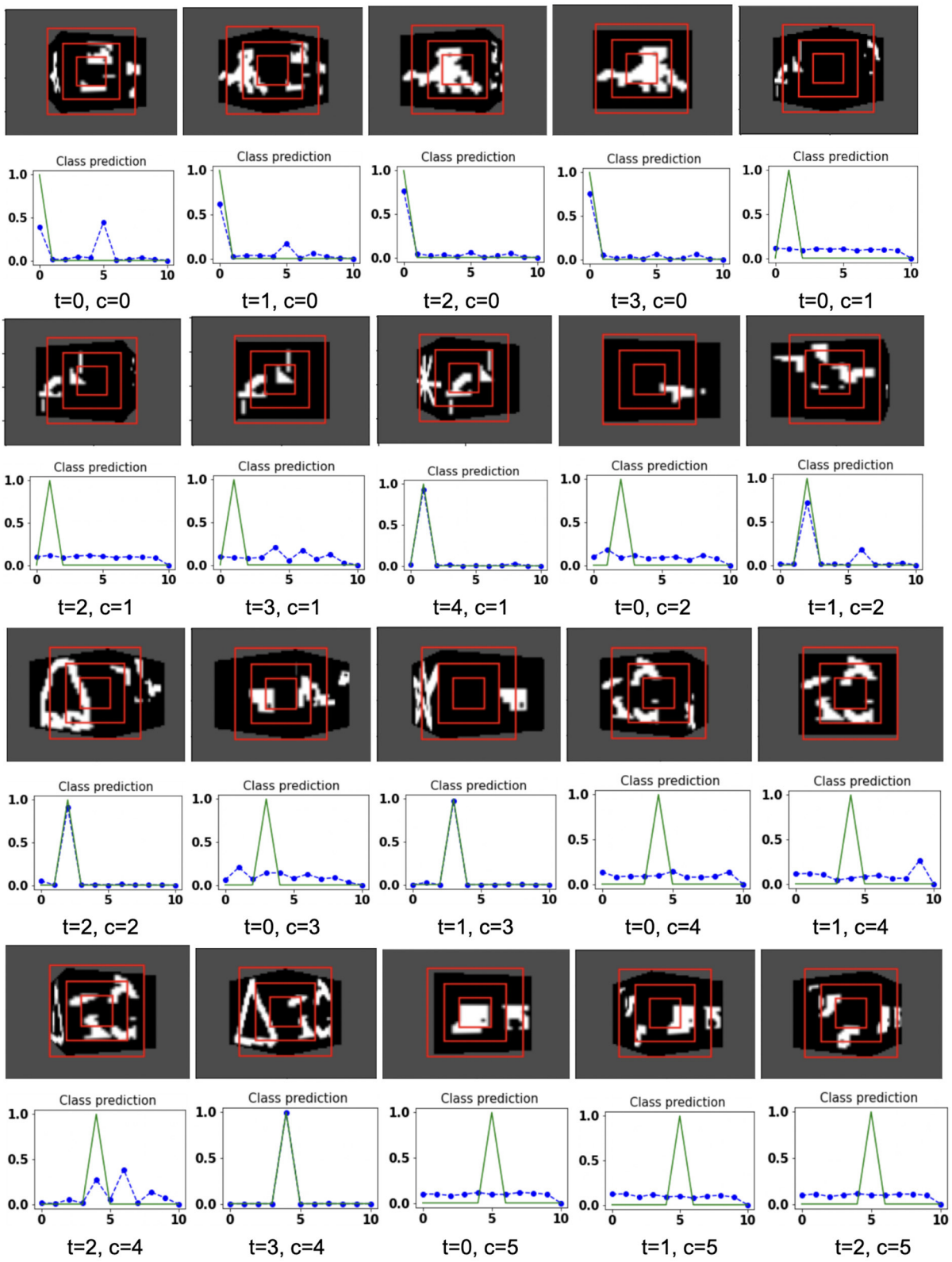


FIGURE 8
Illustrates the camera movements around the cube to search the target face in the view of size 75 × 100, predicted by our model in 3D Cluttered Grayscale QuickDraw Cube dataset. For each class at time *t*, there is a movement (shown in the row of camera view images) and corresponding classification probabilities (shown in the row of plots). In the row of camera view images, the three concentric red windows depict the glimpse at the center of the view image. In the plot corresponding with the above view image, the green curve is the desired classification probabilities and the dotted dashed-blue curve is the predicted classification probabilities at time *t*.

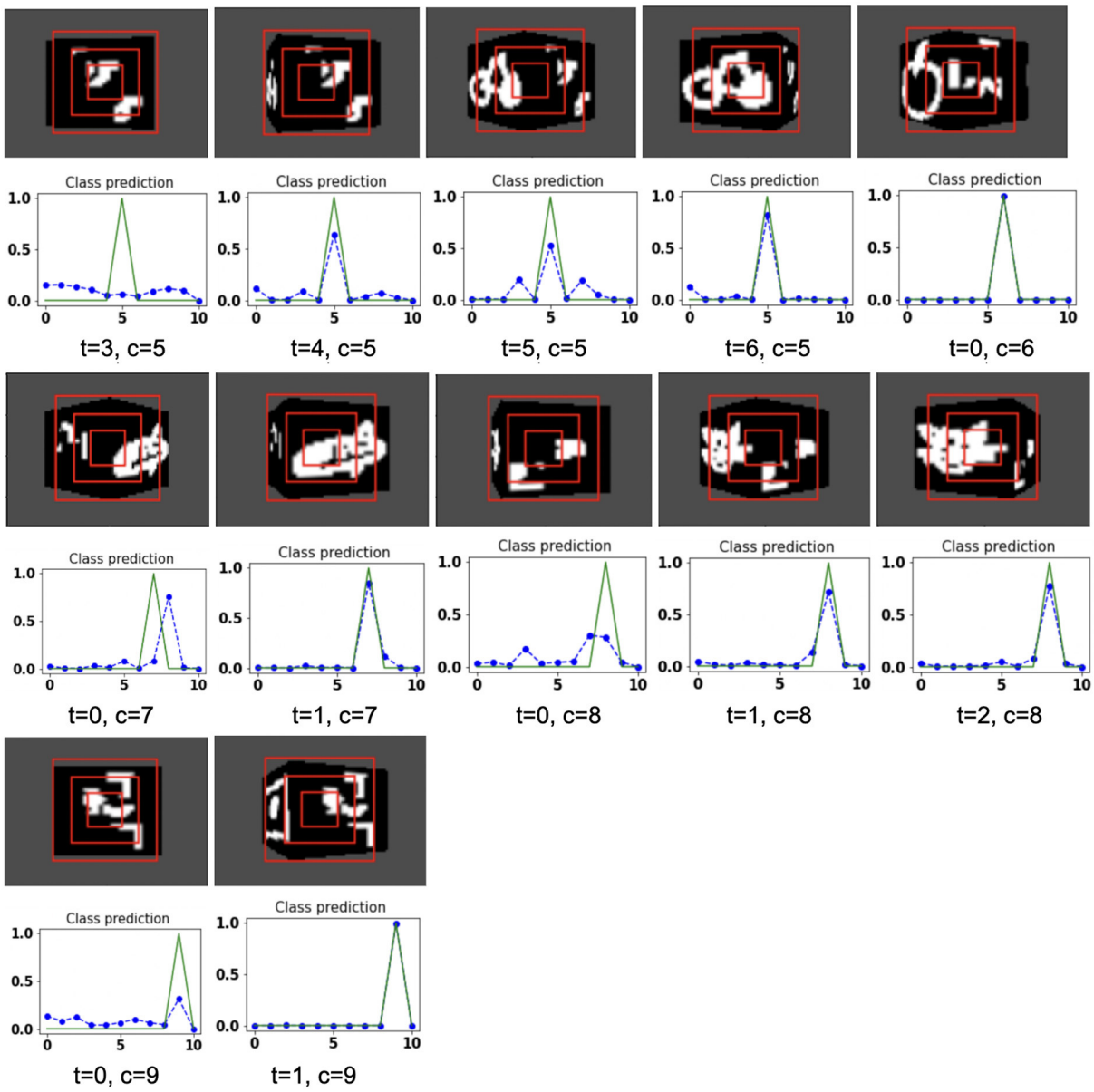


FIGURE 9
Illustrates the camera movements around the cube to search the target face in the view of size 75 × 100, predicted by our model in 3D Cluttered Grayscale QuickDraw Cube dataset. For each class at time *t*, there is a movement (shown in the row of camera view images) and corresponding classification probabilities (shown in the row of plots). In the row of camera view images, the three concentric red windows depict the glimpse at the center of the view image. In the plot corresponding with the above view image, the green curve is the desired classification probabilities and the dotted dashed-blue curve is the predicted classification probabilities at time *t*.

in Figures 10–12. Plots of accuracy, and reward vs. epoch, are shown in Figure 13.

The hyperparameters of the model are tuned and chosen as follows: 0.0001 learning rate, 0.43 discount factor, 0.85 threshold (λ), and 0.1 regularization factor (β) with the best performance in case of 3D Cluttered Grayscale MNIST Cube dataset. The model explores the actions with ϵ equal to 0.99 and the exploration gets reduced by a decay factor of 0.999 while training. The minimum value of ϵ is set with 0.1. The model is trained for 25 epochs and 50 timesteps per cube, in case of 3D Cluttered Grayscale MNIST Cube dataset. In the case of the 3D Cluttered QuickDraw Cube dataset, the model is trained for 20 epochs and 50 timesteps. During the inference, time-steps are varied depending upon the classification probabilities. Prediction is considered to be done as soon as the maximum value of the

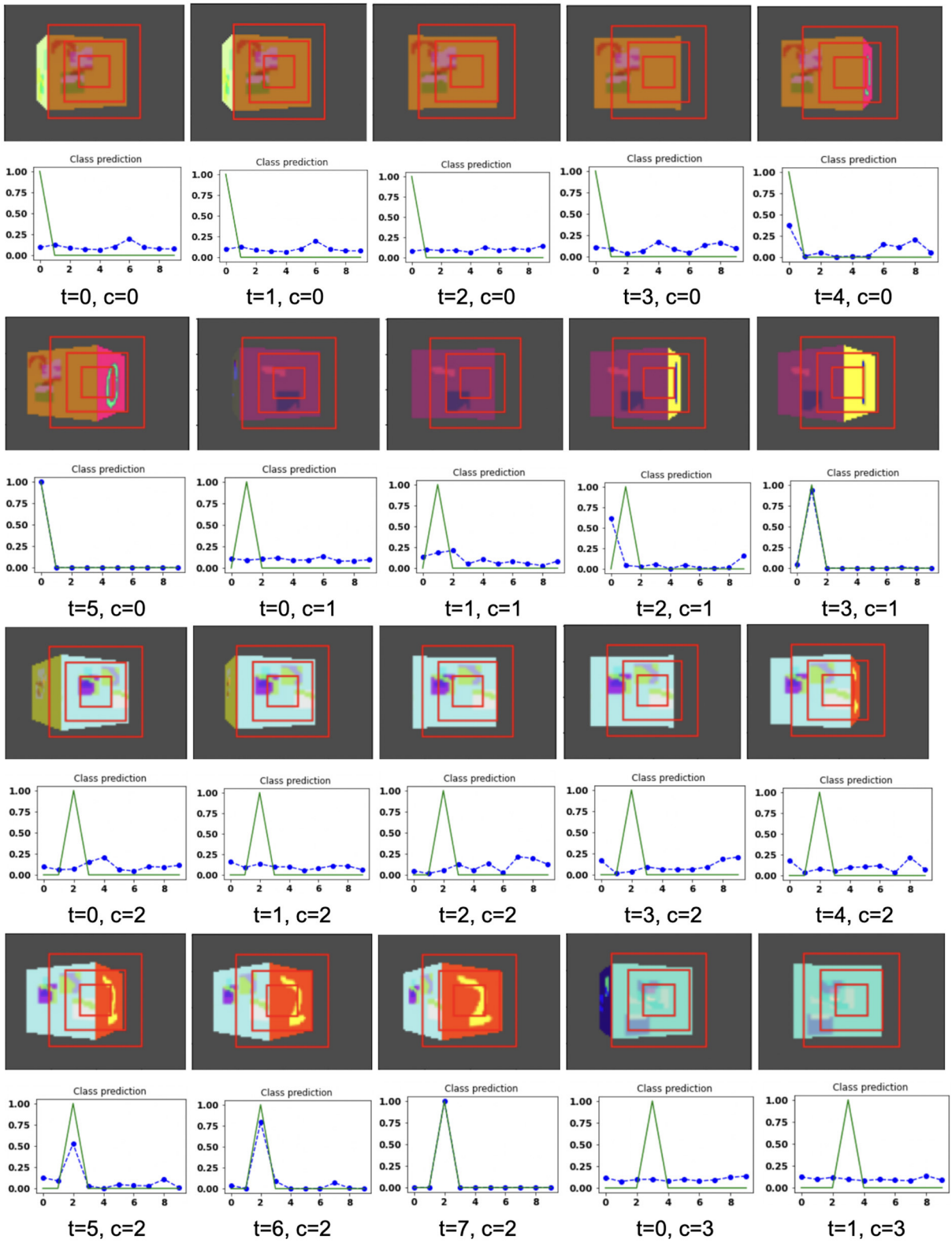


FIGURE 10
Illustrates the camera movements around the cube to search the target face in the view of size 75 × 100, predicted by our model in 3D Cluttered RGB MNIST Cube dataset. For each class at time *t*, there is a movement (shown in the row of camera view images) and corresponding classification probabilities (shown in the row of plots). In the row of camera view images, the three concentric red windows depict the glimpse at the center of the view image. In the plot corresponding with the above view image, the green curve is the desired classification probabilities and the dotted dashed-blue curve is the predicted classification probabilities at time *t*.

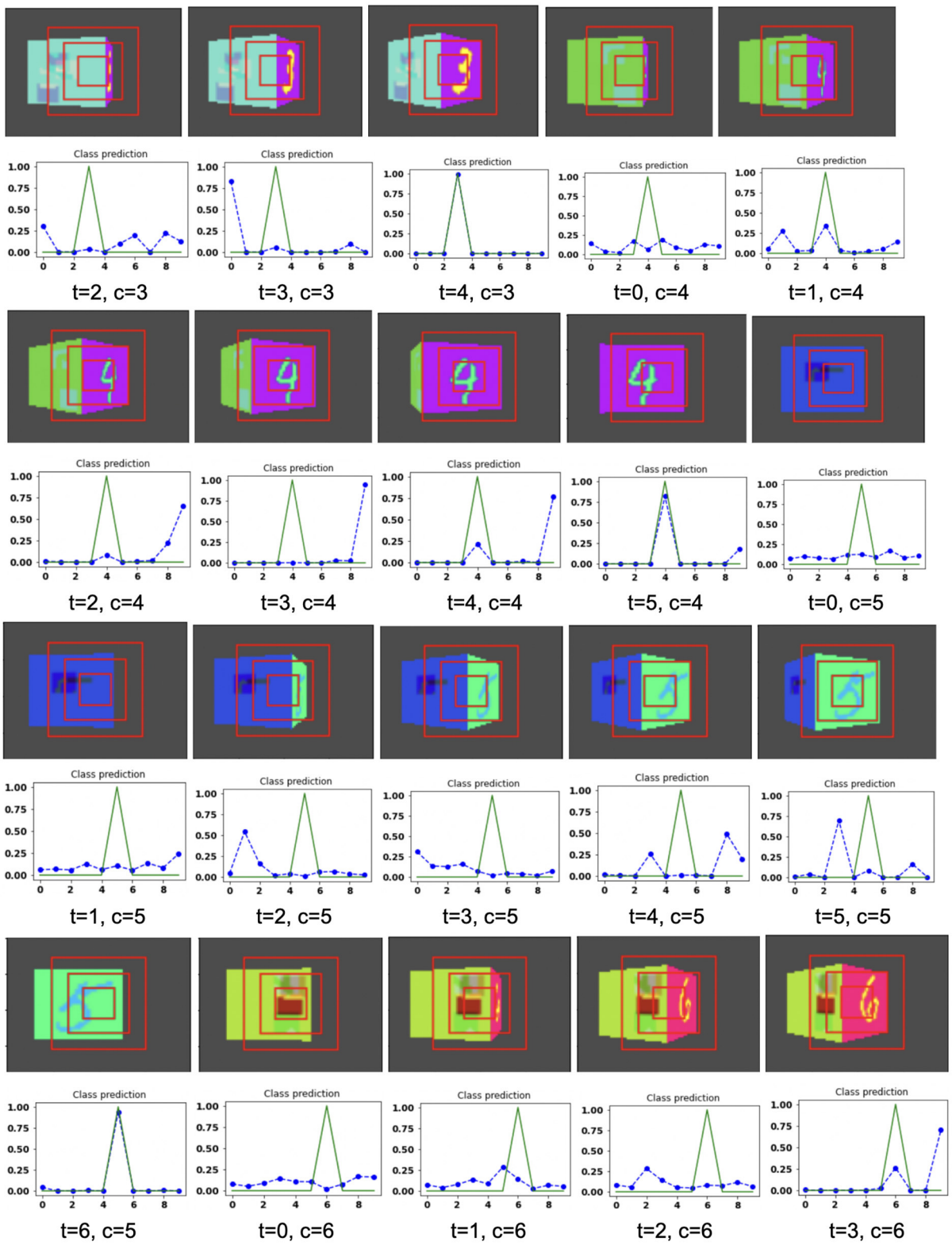


FIGURE 11
Illustrates the camera movements around the cube to search the target face in the view of size 75 × 100, predicted by our model in 3D Cluttered RGB MNIST Cube dataset. For each class at time *t*, there is a movement (shown in the row of camera view images) and corresponding classification probabilities (shown in the row of plots). In the row of camera view images, the three concentric red windows depict the glimpse at the center of the view image. In the plot corresponding with the above view image, the green curve is the desired classification probabilities and the dotted dashed-blue curve is the predicted classification probabilities at time *t*.

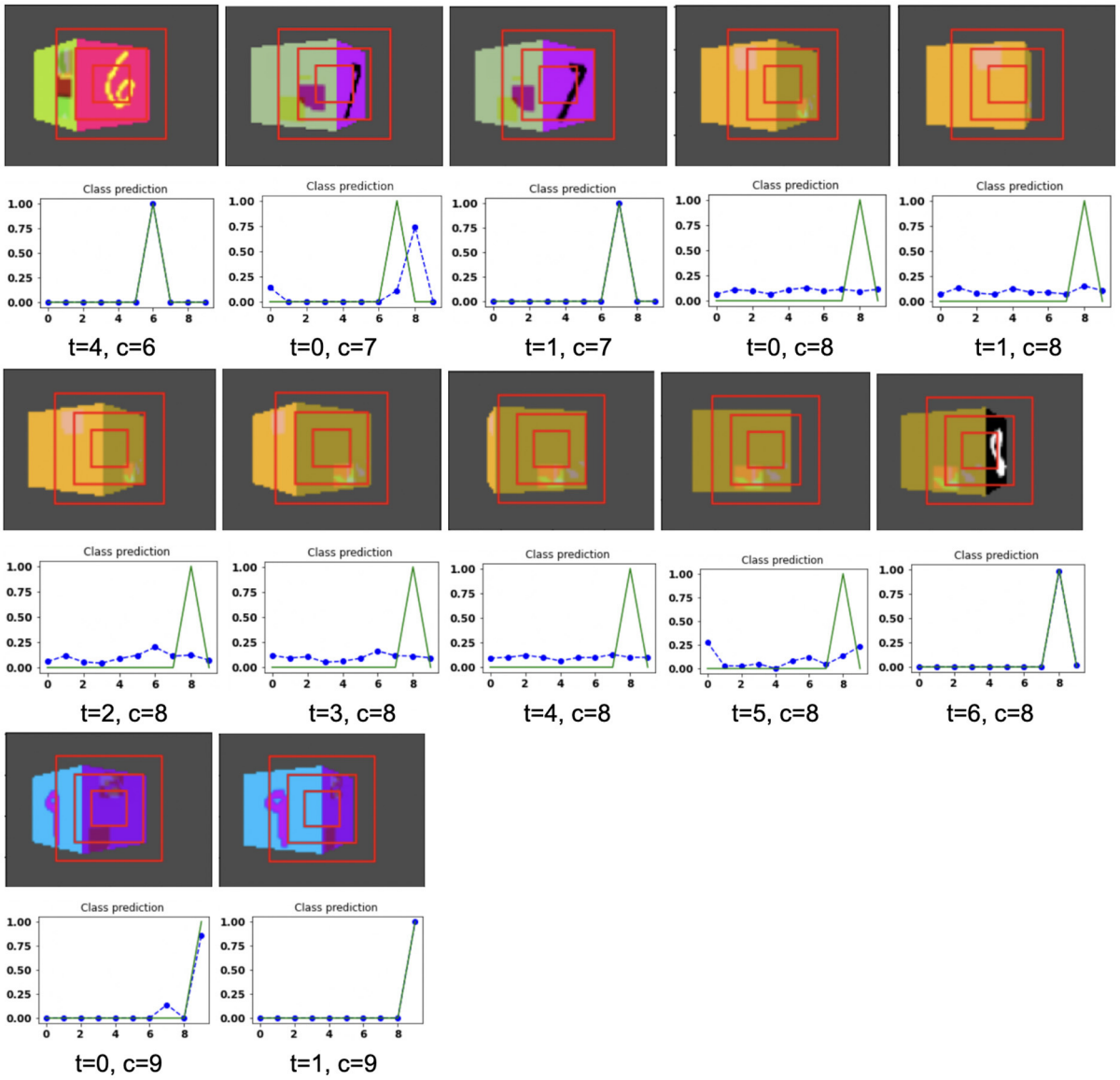


FIGURE 12
Illustrates the camera movements around the cube to search the target face in the view of size 75 × 100, predicted by our model in 3D Cluttered RGB MNIST Cube dataset. For each class at time *t*, there is a movement (shown in the row of camera view images) and corresponding classification probabilities (shown in the row of plots). In the row of camera view images, the three concentric red windows depict the glimpse at the center of the view image. In the plot corresponding with the above view image, the green curve is the desired classification probabilities and the dotted dashed-blue curve is the predicted classification probabilities at time *t*.

classification probabilities crosses a certain testing threshold (= 0.95). A slight variation in values of the hyperparameters is used for the 3D Cluttered RGB MNIST Cube dataset after tuning.

Jump length is the displacement from one location to the next location. The jump length of the camera from one location to the next location on the orbit is considered as a predefined parameter. The jump length of the camera is 12 in case of Grayscale 3D Cluttered MNIST Cube dataset, and 20 in case of 3D Cluttered QuickDraw Cube dataset and 3D Cluttered RGB MNIST Cube dataset.

4. Discussion

To search for the entrance of a building, where there is neither a boundary wall, nor a clear path leading to the entrance, we usually move on the circular path around the building in

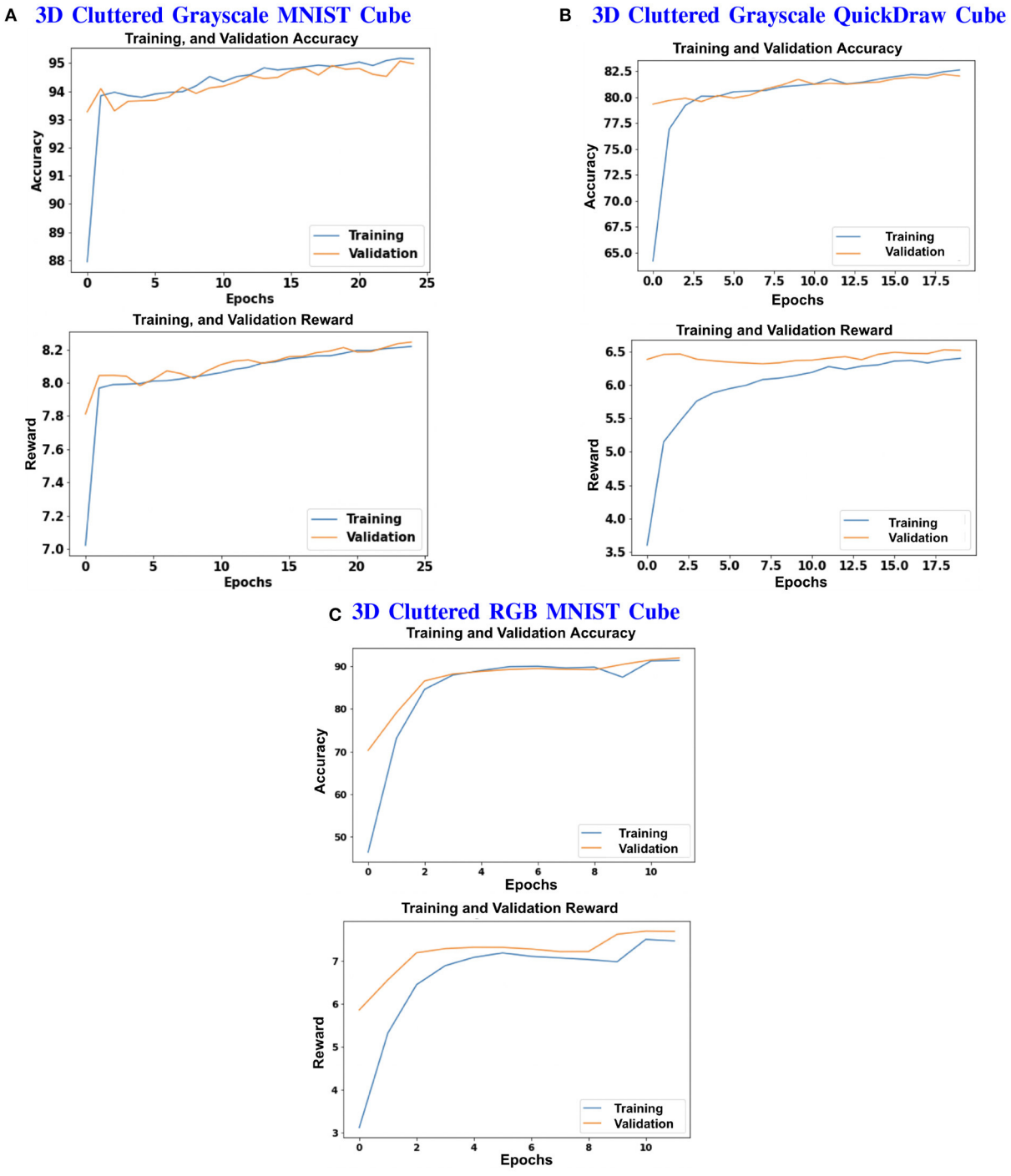


FIGURE 13
(A–C) Shows the plots of Accuracy (1*st* and 3*rd* row) and Reward (2*nd* and 4*th* row) vs. Epochs of 3D Cluttered Grayscale MNIST Cube dataset, 3D Cluttered Grayscale QuickDraw Cube dataset, and 3D Cluttered RGB MNIST Cube dataset, respectively.

either clockwise or anticlockwise direction until we find the entrance. While performing such a task, we also take care that the movement should not involve rapid alternation between the two directions, and must progress continuously in one direction.

The best application of the current model can be in space. For example, geostationary satellites and spy satellites revolving around the earth in a circular orbit require a searching capability of one specific large area of the earth to get a bird's eye view or

to obtain information about various weather, natural calamities, deforestation, and similar activities. From the results of camera movement shown in Figures 6–12, the proposed model is able to avoid alternative movements and is always able to follow the continuous movements to search the target face of the cube. There are three major components to consider the proposed model biologically inspired. First, the model takes the input of multiple concentric windows of different scales, which resembles the differential spatial resolution of the central fovea and the peripheral regions of the retinal. Second, the model processes the view and its corresponding functions of the camera's location, θ, which is analogous to determining the position using path integration and using it to navigate the world. The classifier and camera motion networks are analogous to the processing of visual information along the "what and where/how" pathways (Schenk and McIntosh, 2010), respectively. Third, the model uses Elman, Jordan, JK-flip-flop recurrence layers as memory to store the history of the view and corresponding location, which resemble the feedback loops present among the visual cortical areas, for example from V_1 to thalamus or from V_2 to V_1, (Angelucci and Sainsbury, 2006). The output layers of the classifier and the camera motion network are used to attribute a specialized role to both of the networks for classification and searching tasks, by feeding the outputs back into the first fully connected Elman and Jordan layers in their corresponding channels. The output vector of the camera motion network (Q-values) which has information about the action to be taken by the camera is fed back into the fully connected Elman and Jordan layer and the output vectors of this layer passed through fully connected flip-flop layer and gets concatenated with the output of the last layer of the camera position network; this wide loop is responsible for storing the history of location and view.

5. Conclusions

In the proposed model, we have shown how the "classifier" and "camera motion" networks coordinate with each other to perform the 3D visual search task. The BIAS-3D successfully performed the classification task on a 3D environment on three datasets (Table 1). As shown in the results, movements generated by the model to search a target in the given cube always aim at the target face and take meaningful movements so that the camera looks at the target and classifies it correctly. Based on the results described herewith, we want to extend the model to more complicated full 3D searches in a 3D environment like, for example, searching for defects on the surface of a 3D structure. The model can then be applied to full scale object detection and recognition in 3D space.

Data availability statement

The datasets generated during the current study are available in the RGB MNIST-dataset repository: https://github.com/sweta111/RGB_MNIST-dataset.

Author contributions

The experiments were conducted by SK. SK wrote the entire text. VA and MN created the setup of the environment. VC contributed in providing the key ideas, editing the manuscript drafts, and providing insight into structure. All authors contributed to the article and approved the submitted version.

Acknowledgments

We acknowledge the support from Pavan Holla and Vigneswaran for the implementation of flip-flop neurons. We also acknowledge Sowmya Manojna to generate RGB MNIST dataset. SK acknowledges the financial support of Ministry of Human Resource Development for graduate assistantship.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

Supplementary material

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fncom.2022.1012559/full#supplementary-material

References

Abed-alguni, B. H. (2018). Action-selection method for reinforcement learning based on cuckoo search algorithm. *Arab. J. Sci. Eng.* 43, 6771–6785. doi: 10.1007/s13369-017-2873-8

Angelucci, A., and Sainsbury, K. (2006). Contribution of feedforward thalamic afferents and corticogeniculate feedback to the spatial summation area of macaque V1 AND LGN. *J. Compar. Neurol.* 498, 330–351. doi: 10.1002/cne.21060

Armstrong, M., and Murlis, H. (2007). *Reward Management: A Handbook of Remuneration Strategy and Practice*. London: Kogan Page Publishers.

Ba, J., Mnih, V., and Kavukcuoglu, K. (2014). Multiple object recognition with visual attention. *arXiv preprint arXiv:1412.7755*. doi: 10.48550/arXiv.1412.7755

Bengio, Y., Frasconi, P., urgen Schmidhuber, J., and Elvezia, C. (2001). *Gradient Flow in Recurrent Nets: The Difficulty of Learning Long-Term Dependencies*. Fakultat fur Informatik.

Borji, A., Sihite, D. N., and Itti, L. (2011). "Computational modeling of top-down visual attention in interactive environments," in *BMVC, Vol. 85* (Los Angeles, CA), 1–12. doi: 10.5244/C.25.85

Borji, A., Sihite, D. N., and Itti, L. (2012). "Probabilistic learning of task-specific visual attention," in *2012 IEEE Conference on Computer Vision and Pattern Recognition* (Los Angeles, CA), 470–477. doi: 10.1109/CVPR.2012.6247710

Carpenter, R., and McDonald, S. A. (2007). Later predicts saccade latency distributions in reading. *Exp. Brain Res.* 177, 176–183. doi: 10.1007/s00221-006-0666-5

Churchland, A. K., Kiani, R., and Shadlen, M. N. (2008). Correction: Corrigendum: decision-making with multiple alternatives. *Nat. Neurosci.* 11, 851–851. doi: 10.1038/nn0708-851c

Elman, J. L. (1991). Distributed representations, simple recurrent networks, and grammatical structure. *Mach. Learn.* 7, 195–225. doi: 10.1007/BF00114844

Fan, J., Wang, Z., Xie, Y., and Yang, Z. (2020). "A theoretical analysis of deep q-learning," in *Learning for Dynamics and Control* (Sardinia), 486–489.

Gao, D., Han, S., and Vasconcelos, N. (2009). Discriminant saliency, the detection of suspicious coincidences, and applications to visual recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* 31, 989–1005. doi: 10.1109/TPAMI.2009.27

Gao, D., Mahadevan, V., and Vasconcelos, N. (2008). On the plausibility of the discriminant center-surround hypothesis for visual saliency. *J. Vis.* 8, 13–13. doi: 10.1167/8.7.13

Gao, D., and Vasconcelos, N. (2004). Discriminant saliency for visual recognition from cluttered scenes. *Adv. Neural Inform. Process. Syst.* 17, 481–488.

Glorot, X., and Bengio, Y. (2010). "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics* (California), 249–256.

Goodfellow, I., Bengio, Y., Courville, A., and Bengio, Y. (2016). *Deep Learning, Vol. 1*. Cambridge: MIT Press.

Gruber, A. J., Dayan, P., Gutkin, B. S., and Solla, S. A. (2006). Dopamine modulation in the basal ganglia locks the gate to working memory. *J. Comput. Neurosci.* 20:153. doi: 10.1007/s10827-005-5705-x

Haque, A., Alahi, A., and Fei-Fei, L. (2016). "Recurrent attention models for depth-based person identification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Stanford), 1229–1238. doi: 10.1109/CVPR.2016.138

Holla, P., and Chakravarthy, S. (2016). "Decision making with long delays using networks of flip-flop neurons," in *2016 International Joint Conference on Neural Networks (IJCNN)* (Vancouver, BC), 2767–2773. doi: 10.1109/IJCNN.2016.7727548

Jongejan, J., Rowley, H., Kawashima, T., Kim, J., and Fox-Gieg, N. (2016). *The Quick, Draw!-AI Experiment*. Available online at: http://quickdraw.withgoogle.com

Jordan, M. (1986). *Serial Order: A Parallel Distributed Processing Approach*. Technical Report, California University, Institute for Cognitive Science, San Diego, CA.

Kahou, S. E., Michalski, V., Memisevic, R., Pal, C., and Vincent, P. (2017). "RATM: recurrent attentive tracking model," in *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)* (Honolulu, HI), 1613–1622. IEEE. doi: 10.1109/CVPRW.2017.206

Kanan, C., Tong, M. H., Zhang, L., and Cottrell, G. W. (2009). Sun: top-down saliency using natural statistics. *Vis. Cogn.* 17, 979–1003. doi: 10.1080/13506280902771138

Kanezaki, A., Matsushita, Y., and Nishida, Y. (2018). "Rotationnet: joint object categorization and pose estimation using multiviews from unsupervised viewpoints," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Tokyo), 5010–5019. doi: 10.1109/CVPR.2018.00526

Kingma, D. P., and Ba, J. (2014). Adam: a method for stochastic optimization. *arXiv preprint arXiv:1412.6980*. doi: 10.48550/arXiv.1412.6980

Knapp, A. (1938). An introduction to clinical perimetry. *Arch. Ophthalmol.* 20, 1116–1117. doi: 10.1001/archopht.1938.00850240232021

Kratsios, A., and Hyndman, C. (2020). Deep arbitrage-free learning in a generalized HJM framework *via* arbitrage-regularization. *Risks* 8:40. doi: 10.3390/risks8020040

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Adv. Neural Inform. Process. Syst.* 25, 1097–1105. doi: 10.1145/3065386

Lan, S., Ren, Z., Wu, Y., Davis, L. S., and Hua, G. (2020). "SaccadeNet: a fast and accurate object detector," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (Seattle, WA), 10397–10406. doi: 10.1109/CVPR42600.2020.01041

Le Meur, O., Le Callet, P., Barba, D., and Thoreau, D. (2006). A coherent computational approach to model bottom-up visual attention. *IEEE Trans. Pattern Anal. Mach. Intell.* 28, 802–817. doi: 10.1109/TPAMI.2006.86

LeCun, Y., Cortes, C., and Burges, C. J. C. (1998). *The MNIST Database of Handwritten Digits* (New York, NY). Available online at: http://yann.lecun.com/exdb/mnist/

Liu, T., Lam, K.-M., Zhao, R., and Kong, J. (2021). Enhanced attention tracking with multi-branch network for egocentric activity recognition. *IEEE Trans. Circuits Syst. Video Technol.* doi: 10.1109/TCSVT.2021.3104651

Liu, T., Zhao, R., Jia, W., Lam, K.-M., and Kong, J. (2022). Holistic-guided disentangled learning with cross-video semantics mining for concurrent first-person and third-person activity recognition. *IEEE Trans. Neural Netw. Learn. Syst.* 32, 1–15. doi: 10.1109/TNNLS.2022.3202835

Minut, S., and Mahadevan, S. (2001). "A reinforcement learning model of selective visual attention," in *Proceedings of the fifth international conference on Autonomous agents* (New York, NY), 457–464. doi: 10.1145/375735.376414

Mnih, V., Heess, N., Graves, A., and Kavukcuoglu, K. (2014). Recurrent models of visual attention. *arXiv preprint arXiv:1406.6247*. doi: 10.48550/arXiv.1406.6247

Nair, V., and Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. *Icml.* 10, 807–814.

Roitman, J. D., and Shadlen, M. N. (2002). Response of neurons in the lateral intraparietal area during a combined visual discrimination reaction time task. *J. Neurosci.* 22, 9475–9489. doi: 10.1523/JNEUROSCI.22-21-09475.2002

Roth, C. H. Jr., Kinney, L. L., and John, E. B. (2020). *Fundamentals of Logic Design*. Boston, MA: Cengage Learning.

Rowe, J. B., Hughes, L., and Nimmo-Smith, I. (2010). Action selection: a race model for selected and non-selected actions distinguishes the contribution of premotor and prefrontal areas. *Neuroimage* 51, 888–896. doi: 10.1016/j.neuroimage.2010.02.045

Schenk, T., and McIntosh, R. D. (2010). Do we have independent visual streams for perception and action? *Cogn Neurosci.* 1, 52–62. doi: 10.1080/17588920903388950

Scherer, D., Müller, A., and Behnke, S. (2010). "Evaluation of pooling operations in convolutional architectures for object recognition," in *International Conference on Artificial Neural Networks* (Heidelberg: Springer), 92–101. doi: 10.1007/978-3-642-15825-4_10

Segal, M., and Akeley, K. (2010). *The openGLr Graphics System: A Specification (Version 4.0 (Core Profile))*. Beaverton, OR: The Khronos Group Inc.

Shaikh, M., Kollerathu, V. A., and Krishnamurthi, G. (2019). "Recurrent attention mechanism networks for enhanced classification of biomedical images," in *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)* (Venice), 1260–1264. IEEE. doi: 10.1109/ISBI.2019.8759214

Shen, J., Du, Y., Wang, W., and Li, X. (2014). Lazy random walks for superpixel segmentation. *IEEE Trans. Image Process.* 23, 1451–1462. doi: 10.1109/TIP.2014.2302892

Su, H., Maji, S., Kalogerakis, E., and Learned-Miller, E. (2015). "Multi-view convolutional neural networks for 3D shape recognition," in *Proceedings of the IEEE International Conference on Computer Vision* (Massachusetts), 945–953. doi: 10.1109/ICCV.2015.114

Sutton, R. S., and Barto, A. G. (1998). *Introduction to Reinforcement Learning, Vol. 135*. Cambridge: MIT Press.

Sweta, K., Vigneswaran, C., and Chakravarthy, V. S. (2021). The flip-flop neuron - a memory efficient alternative for solving challenging sequence processing and decision-making problems. *BioRxiv*. doi: 10.1101/2021.11.16.468605

Voleti, R. (2021). Unfolding the evolution of machine learning and its expediency. *IJCSMC*. 10, 1–7. doi: 10.47760/ijcsmc.2021.v10i01.001

Wang, W., Shen, J., and Porikli, F. (2015a). "Saliency-aware geodesic video object segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Boston, MA), 3395–3402.

Wang, W., Shen, J., and Shao, L. (2015b). Consistent video saliency using local gradient flow optimization and global refinement. *IEEE Trans. Image Process*. 24, 4185–4196. doi: 10.1109/TIP.2015.2460013

Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., et al. (2015). "3D shapenets: a deep representation for volumetric shapes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Boston, MA), 1912–1920.

Xu, B., Liu, J., Hou, X., Liu, B., Garibaldi, J., Ellis, I. O., et al. (2019). Attention by selection: A deep selective attention approach to breast cancer classification. *IEEE Trans. Med. Imaging* 39, 1930–1941. doi: 10.1109/TMI.2019.2962013

Yang, Y., Deng, C., Gao, S., Liu, W., Tao, D., and Gao, X. (2016a). Discriminative multi-instance multitask learning for 3d action recognition. *IEEE Trans. Multim*. 19, 519–529. doi: 10.1109/TMM.2016.2626959

Yang, Y., Deng, C., Tao, D., Zhang, S., Liu, W., and Gao, X. (2016b). Latent max-margin multitask learning with skelets for 3-D action recognition. *IEEE Trans. Cybern*. 47, 439–448. doi: 10.1109/TCYB.2016.2519448

Yang, Y., Liu, R., Deng, C., and Gao, X. (2016c). Multi-task human action recognition *via* exploring super-category. *Signal Process*. 124, 36–44. doi: 10.1016/j.sigpro.2015.10.035

Zhang, D., Han, J., Jiang, L., Ye, S., and Chang, X. (2017). Revealing event saliency in unconstrained video collection. *IEEE Trans. Image Process*. 26, 1746–1758. doi: 10.1109/TIP.2017.2658957

Zhang, L., Zhang, Q., and Xiao, C. (2015). Shadow remover: image shadow removal based on illumination recovering optimization. *IEEE Trans. Image Process*. 24, 4623–4636. doi: 10.1109/TIP.2015.2465159

frontiers | Frontiers in Computational Neuroscience

TYPE Original Research
PUBLISHED 24 November 2022
DOI 10.3389/fncom.2022.1077439

OPEN ACCESS

EDITED BY
Chenwei Deng,
Beijing Institute of Technology, China

REVIEWED BY
Ping Zhou,
Ministry of Natural Resources of the People's Republic of China, China
Jianhua Wan,
China University of Petroleum Qingdao, China

*CORRESPONDENCE
Guo Zhang
guozhang@whu.edu.cn

RECEIVED 23 October 2022
ACCEPTED 10 November 2022
PUBLISHED 24 November 2022

CITATION
Tong J, Zhang G, Kong P, Rao Y, Wei Z, Cui H and Guan Q (2022) An interpretable approach for automatic aesthetic assessment of remote sensing images.
Front. Comput. Neurosci. 16:1077439.
doi: 10.3389/fncom.2022.1077439

An interpretable approach for automatic aesthetic assessment of remote sensing images

Jingru Tong[1], Guo Zhang[2]*, Peijie Kong[1], Yu Rao[1], Zhengkai Wei[1], Hao Cui[2] and Qing Guan[2]

[1]School of Remote Sensing and Information Engineering, Wuhan University, Wuhan, China, [2]State Key Laboratory of Information Engineering in Surveying, Mapping, and Remote Sensing, Wuhan University, Wuhan, China

The increase of remote sensing images in recent decades has resulted in their use in non-scientific fields such as environmental protection, education, and art. In this situation, we need to focus on the aesthetic assessment of remote sensing, which has received little attention in research. While according to studies on human brain's attention mechanism, certain areas of an image can trigger visual stimuli during aesthetic evaluation. Inspired by this, we used convolutional neural network (CNN), a deep learning model resembling the human neural system, for image aesthetic assessment. So we propose an interpretable approach for automatic aesthetic assessment of remote sensing images. Firstly, we created the Remote Sensing Aesthetics Dataset (RSAD). We collected remote sensing images from Google Earth, designed the four evaluation criteria of remote sensing image aesthetic quality—color harmony, light and shadow, prominent theme, and visual balance—and then labeled the samples based on expert photographers' judgment on the four evaluation criteria. Secondly, we feed RSAD into the ResNet-18 architecture for training. Experimental results show that the proposed method can accurately identify visually pleasing remote sensing images. Finally, we provided a visual explanation of aesthetic assessment by adopting Gradient-weighted Class Activation Mapping (Grad-CAM) to highlight the important image area that influenced model's decision. Overall, this paper is the first to propose and realize automatic aesthetic assessment of remote sensing images, contributing to the non-scientific applications of remote sensing and demonstrating the interpretability of deep-learning based image aesthetic evaluation.

KEYWORDS

remote sensing images, aesthetic assessment, aesthetic quality, interpretability, attention mechanism, deep learning

Introduction

In recent decades, remote sensing has advanced rapidly, becoming increasingly important in geological mapping, environmental monitoring, urban development, etc. These studies mainly focus on the scientific uses of remote sensing. However, with the increase of remote sensing images, they have emerged in various non-scientific applications and been used non-scientific users. These individuals only regard remote sensing images as images, not as a source of scientific information. In such case, we need to pay attention to the aesthetic assessment of remote sensing images. Visually appealing remote sensing images, which offer a distinctive perspective from above, can be meaningful to fields such as environmental protection, education, and art. When policymakers are exposed to natural splendors, they may be motivated to adopt more environmentally friendly measures (Wang et al., 2016). When creating artworks, artists such as photographers and painters can be inspired by the beauty of the Earth (Grayson, 2016). Beautiful remote sensing images can also be used by educators to trigger students' passion in nature. According to studies on human brain's attention mechanism, certain areas of an image can trigger visual stimuli, influencing aesthetic evaluation. Inspired by this, we used convolutional neural network (CNN), a deep learning model resembling the human neural system, to perform automatic aesthetic assessment of remote sensing images. By comparing the key image area that affected the model's decision with human aesthetic standards, we discussed the interpretability of deep-learning based image aesthetic evaluation.

Aesthetic assessment is the process of classifying images into high or low aesthetic quality (Wong and Low, 2009; Luo et al., 2011), or predict their aesthetic scores (Datta and Wang, 2010; Li et al., 2010). Aesthetic quality can be understood as the pleasure people obtain from appreciating images (Kalivoda et al., 2014). Recent advances in cognitive neuroscience have suggested correspondence between the physical properties of stimuli and the sensations they cause (Skov and Nadal, 2020). Therefore, images of high aesthetic quality can be deemed as "visually pleasing." Though people's aesthetic preference or criteria may differ (Kim et al., 2018), such subjectivity does not preclude objective research into aesthetic quality. Just as many people may feel more comfortable and delightful with certain rhythms in music (Li and Chen, 2009), many may have similar feelings towards certain images. The same goes for remote sensing images. And if we can identify the factors that affect people's judgment on the aesthetic quality of remote sensing images, we may establish the evaluation standards behind aesthetic evaluation. Using data-driven methods, we can then measure the aesthetic quality of remote sensing images in a scientific way.

In past decades, researchers have designed handcrafted features to quantify image aesthetic quality. These features range from low-level image statistics, such as edge distributions and color histograms, to high-level photographic rules, such as the rule of thirds and the golden ratio. For example, Datta et al. (2006) designed a set of visual features, including color metrics, rule of thirds, depth of field, etc. Using professional photography techniques Luo and Tang (2008) first extracted the subject region from a photo and then formulated many high-level semantic features based on this subject and background division. Recently, researchers began to apply deep learning in image aesthetic evaluation. They typically cast it as a classification or regression problem (Deng et al., 2017). A model is trained by assigning a single label (i.e., a class or score) to an image to indicate its level of aesthetic quality. Compared with hand-crafted features designed primarily based on domain-specific knowledge, automatically learned deep features can better capture the underlying aesthetic characteristics from massive training images (Tian et al., 2015). Among the deep learning methods, CNN proved to be effective in analyzing image aesthetics. It is the most similar to human visual processing systems, has a structure well-suited to processing 2D and 3D images, and can effectively learn and extract 2D feature abstractions. The max-pooling layers of CNN can effectively detect shape changes. And it is good at extracting mid-to-high level abstract features from raw images by interleaving convolutional and pooling layers (i.e., by spatially shrinking feature maps layer by layer).

Here, we tackle the aesthetic assessment problem by binary classification, discriminating a remote sensing image into "high aesthetic quality" or "low aesthetic quality." And CNNs have excellent performance in image aesthetic classification. In Lu et al. (2014), proposed the Rating Pictorial Aesthetics using Deep Learning (RAPID) model, it was the first attempt to apply CNNs in image aesthetic evaluation. The network structure was close to AlexNet and aimed at the binary aesthetic classification. CNN's robustness in image aesthetic classification is also demonstrated in image style classification (Karayev et al., 2013) and image popularity estimation (Khosla et al., 2014). In image classification, network depth is crucial, but stacking more conventional layers to increase depth can easily lead to the problem of gradient explosion (Liu et al., 2019). Existing CNN networks, such as AlexNet and VGG, are usually built to directly learn the mapping between input and output, which can hardly alleviate gradient explosion. To address this problem, He et al. (2016) proposed ResNet in 2016, which used residual blocks to create a shortcut between the target and the input. The ResNet residual module can solve the problem of vanishing gradients and accelerate training (Wu et al., 2020).

Despite the good performance of deep neural networks in image aesthetic assessment, they are hard to interpret because they cannot be decomposed into intuitive and understandable components (Lipton, 2018). Evidence from human perception process (Mnih et al., 2014) demonstrates the importance of attention mechanism, which uses top information to guide

bottom-up feed-forward process. In the cognitive process of visual aesthetics, the region of the object's prominent visual properties, such as color, shape, and composition, receives initial attention (Cela-Conde et al., 2011). These prominent regions would trigger stimulus within the ventral visual stream. The feed-forward process would then enhance the visual experience of the object, leading to aesthetic assessment. In other words, rather than processing the whole scene in its entirety, humans selectively focus on specific parts of the image (Wang et al., 2018). So inspired by such attention mechanism involved in image aesthetic evaluation, we adopt the Gradient-weighted Class Activation Mapping (Grad-CAM) proposed by Selvaraju et al. (2017). Grad-CAM can use the gradient information learned by convolutional neurons to highlight the important image area that influenced the model's decision. The highlighted area generated by Grad-CAM is comparable to the prominent area that draws attention and triggers visual stimulus during the cognitive process of aesthetic assessment.

The increase of remote sensing images in recent decades has resulted in their use by non-scientific users who only see them as images rather than a source of scientific information. In this situation, we need to focus on the aesthetic assessment of remote sensing, which has received little attention in research. Though convolutional neural network (CNN) performs well in image aesthetic evaluation, it lacks interpretability. While according to studies on human brain's attention mechanism, certain areas of an image can trigger visual stimuli during aesthetic evaluation. Therefore, inspired by the brain's cognitive process and the use of CNN in image aesthetic assessment, we propose an interpretable approach for automatic aesthetic assessment of remote sensing images. Firstly, we created the Remote Sensing Aesthetics Dataset (RSAD). We collected remote sensing images from Google Earth, designed the four evaluation criteria of remote sensing image aesthetic quality—color harmony, light and shadow, prominent theme, and visual balance—and then labeled the samples based on expert photographers' judgment on the four evaluation criteria. Secondly, we feed RSAD into the ResNet-18 architecture for training. Experimental results show that the proposed method can accurately identify visually pleasing remote sensing images. Finally, we provided a visual explanation of aesthetic assessment by adopting Grad-CAM to highlight the important image area that influenced model's decision. Overall, this paper is the first to propose and realize automatic aesthetic assessment of remote sensing images, contributing to the non-scientific applications of remote sensing and demonstrating the interpretability of image aesthetics. Our work has the potential to promote the use of remote sensing in non-scientific fields such as environmental protection, education, and art.

Materials and methods

Our method consists of three steps, as shown in **Figure 1**. We first created the Remote Sensing Aesthetics Dataset. We

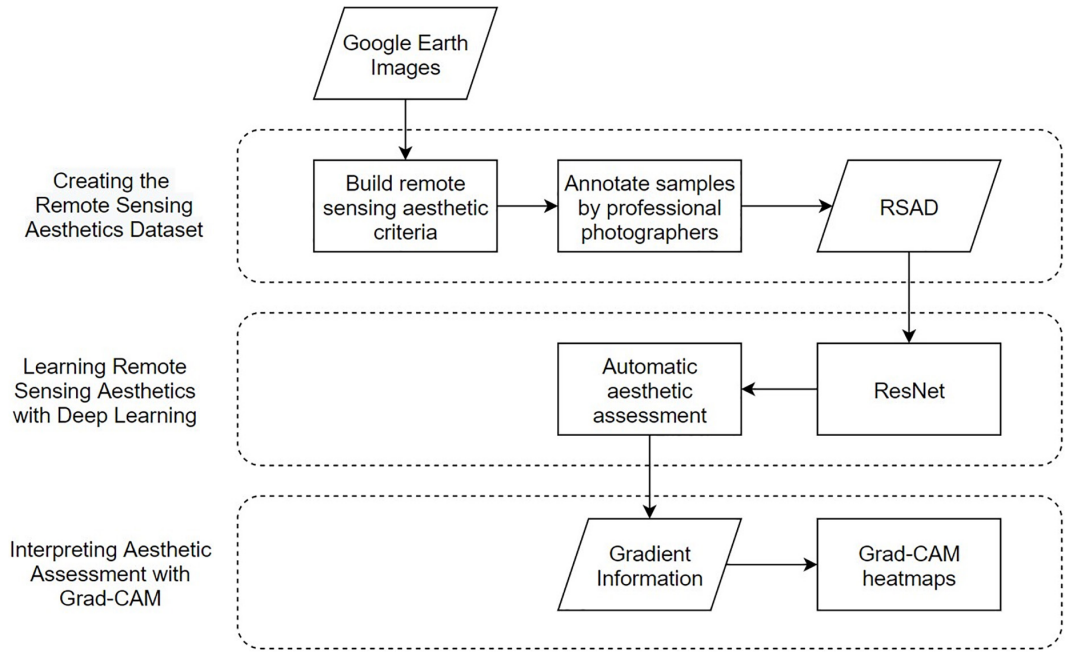


FIGURE 1
Overall technical route of automatic remote sensing aesthetic assessment.

collected remote sensing images from Google Earth, established four evaluation criteria of remote sensing aesthetics, and labeled the images based on professional photographers' judgment of the four criteria. Secondly, we fed the dataset into a deep learning model to classify remote sensing images in high or low aesthetic quality. Finally, we tried to interpret model's aesthetic assessment with Grad-CAM.

The remote sensing aesthetics dataset

Data source

To enable aesthetic evaluation, the remote sensing images we gather should adhere to certain technical requirements. First, all images should be in true color. They should be combination of the three channels that are sensitive to the red, green, and blue visible light, producing what our naked eyes see in the natural world. As we will explain in the following subsection, color plays a significant role in aesthetic evaluation, and dealing colors we are familiar with is a good place to start when exploring remote sensing aesthetics. **Figure 2** compares remote sensing image in true color (**Figure 2B**) with false color (**Figure 2A**). Second, samples ought to have a high resolution. In this way, people can identify features on the image and determine whether the image have a prominent theme or visual weight. Finally, images should not contain any artifacts. Artifacts can appear during image mosaicking as a result of color differences or geometric misalignments between adjacent images (Yin et al., 2022), as shown in **Figure 2C**.

To meet the following technical requirements, we collected images from Google Earth, an open-source platform that includes data integration of satellite and aerial images. Both image types can be regarded as remote sensing images because they are passively collected remotely sensed data. Google Earth includes a wide range of true-color visible spectrum imagery (380–760 nm wavelength) derived from a combination of freely available public domain Landsat imagery, government orthophotos, and high resolution commercial data sets from DigitalGlobe, GeoEye, and SPOT (Fisher et al., 2012). Whatever imaging modalities are used for different data sources, these images all truly reflect the earth's surface. Also, Google image has a resolution of below 100 m, usually 30 m, and a viewing angle of about 15 km above sea level. As a result, Google Earth images can be used as a data source for assessing remote sensing aesthetic quality.

In order for an effective and thorough investigation of remote sensing aesthetics, we should ensure that the dataset had enough variety. Therefore, we gathered remote sensing images covering eight content categories: river, mountain, farmland, beach, desert, forest, glacier, and plain. These categories are based on typical landscape types and remote sensing features, and they are selected for two reasons. First, these are natural features. These images are simpler and clearer than those with artificial features such as airport, industrial, and residential regions, making it relatively easier for aesthetics quality evaluation. Second, these features are common on the Earth's surface. They contain a variety of spatial patterns that are representative in terms of texture and color, and most of them vary sufficiently between different regions. For instance, Mount Himalayan, Sahara Desert volcanoes, and frost-covered Arctic mountains are located at different latitudes, and they look completely different.

We collected all images from a viewing height of 1,500 m, and we avoided images with artifacts. In addition, to increase diversity, remote sensing images are carefully selected from continents worldwide, covering as many latitudes and regions as possible. And these images are selected from different years and seasons. **Figure 3** is a schematic diagram of some images and their selected locations.

Evaluation criteria of remote sensing aesthetic quality

Researchers found that image aesthetic quality can be affected by numerous factors, including lighting (Freeman, 2007), contrast (Itten, 1975), color scheme (Shamoi et al., 2020), and image composition (London et al., 2011), etc. While judging the aesthetic quality of remote sensing images, viewers also have certain criteria or pay attention to certain features in mind. Therefore, we first design a questionnaire to study the factors that may influence how humans evaluate the aesthetic quality of remote sensing images.

We recruited a total of 30 college students between the ages of 18 and 25 as volunteers to fill in the questionnaires. To ensure variety, these students come from a variety of backgrounds and major in fields including journalism, law, economics, computer science, psychology, and electrical engineering, etc. There is a nearly equal distribution of genders. In the questionnaire, we presented volunteers with several remote sensing images and asked them to list more than two factors that they felt crucial for assessing the aesthetic quality of these images. They were also encouraged to further explain how the factor affected the aesthetic evaluation. The top four frequently mentioned factors are "Composition," "Color," "Content," and "Light/Brightness." Other factors mentioned include "Texture," "Balance," "Imagination," "Perspective," "Mood," etc.

In response to the survey results, we summarized four evaluation criteria: color harmony, light and shadow, prominent theme, and visual balance, which addressed both the image's content and composition. As was previously stated, the bottom-up attention mechanism involved in aesthetic evaluation is stimulus-driven. Thus, these four criteria together work as visual stimuli that draw viewers' attention. In our work, we assume that remote sensing images of high aesthetic quality are used for non-scientific users. These individuals regard remote sensing images solely as images, or in a broader sense, artworks of nature. Therefore, when concluding the aforementioned criteria, we considered the general guidelines for both art and photography.

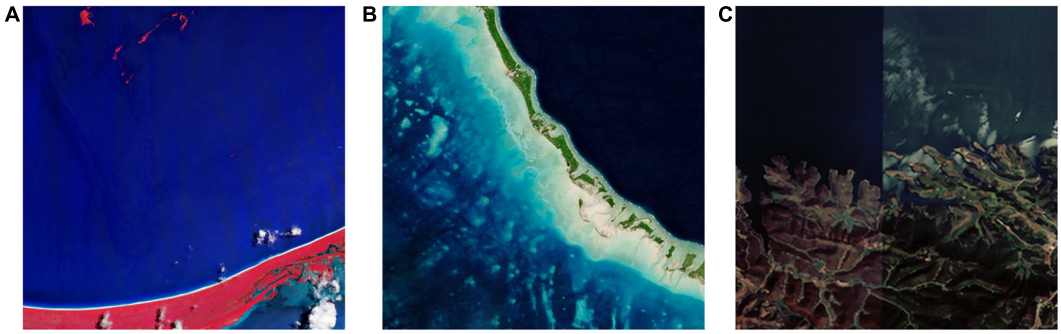


FIGURE 2
Remote sensing image in false color **(A)**, true color **(B)**, and with artifacts **(C)**.

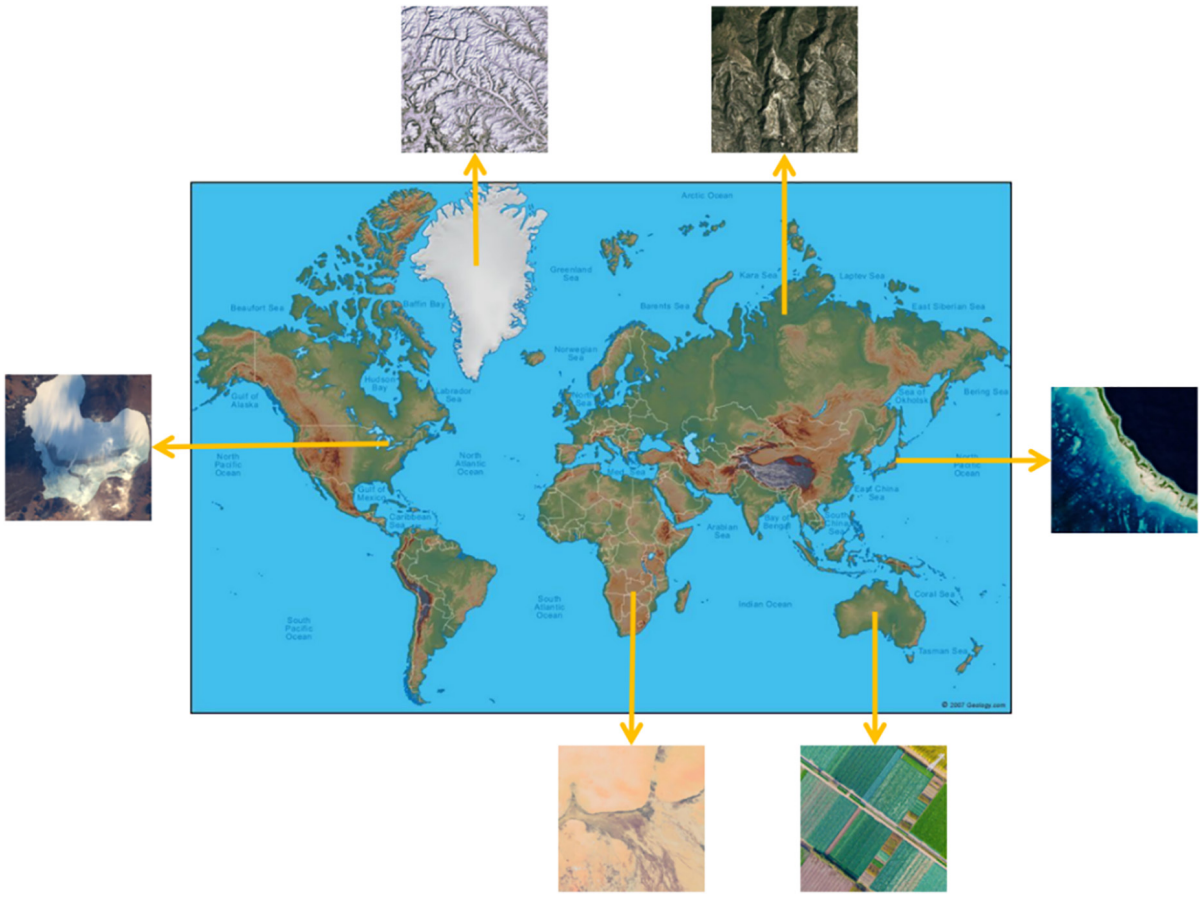


FIGURE 3
Schematic diagram of remote sensing aesthetics dataset (RSAD) images and their selected locations.

We also considered the properties of remote sensing images. The four criteria are elaborated as follows.

Color harmony

Color is what we notice first when we appreciate an image. When two or more colors are brought together to produce a satisfying affective response, they are said to be harmonized (Burchett, 2002). Color harmony is therefore related to the relationship between colors, including cool-warm colors, complementary colors, and the arrangement relations of colors, as shown in **Figure 4**. A remote sensing image can cover a wide range of features, and the various colors of these features

can result in color harmony, leading to high aesthetic quality. The illustration and examples of color harmony in different contexts are provided below.

In modern color theories, an imaginary dividing line running through the color wheel separates the colors into warm and cool, as is displayed in **Figure 4A**. Cool-warm colors are linked to the feelings they evoke and the emotions with which we identify when looking at them. Red, orange, and yellow are warm colors, while blue, green, and purple are cool (Moreland, 2009). **Figure 5A** cleverly combines cool and warm colors. Cool colors like green and blue predominate in the farmland on the left portion of the image, while warm red predominates in the right portion. They form an overall structure of cool-warm contrast. Meanwhile, the left part is interspersed by warm red color patches, creating a local contrast between cool and warm. Complementary colors are a pair of color stimuli (dependent on appropriate wavelength pairs and luminance ratios) whose mixture color matches a given neutral (Brill, 2007). These color pairs can create a striking visual impact when they appear in the same picture. According to the RGB additive color mode, red and cyan, green and magenta, blue, and yellow are typical complementary color pairs, as is shown by **Figure 4B**. Remote sensing images that capture complementary color pairs in nature can have a strong visual impact on the audience, resulting in a high aesthetic quality. **Figure 5B** is an excellent example of red-green complementation, with scattered red islands dotting the green salty lake, bringing liveliness to the whole scene. When colors are arranged in certain relations, they engage the viewer and create an inner sense of order, a balance in the visual experience (Brady and Phillips, 2003). One typical of color arrangement relations is that colors of similar hues undergo progressive changes in brightness or saturation. The gradual change in color will serve as a one-way visual guide, leading humans to appreciate the scene in a specific direction. The progressive red color transition can be seen in the meandering river in **Figure 5C**.

Light and shadow

Optical remote sensing images, in most cases, use sunlight as a source of illumination (Yamazaki et al., 2009). When sunlight reaches the ground features, it will cast a shadow. A right proportion of light and shade can impart depth perception to the scene, creating a stereoscopic effect (Todd et al., 1997). The amount of shadow produced by the light is determined by its direction. In remote sensing images, the direction of light depends on the solar zenith angle, which is related to the latitude of the direct solar point, the local latitude, and the local time (Zhang et al., 2021). In the morning or afternoon, due to the low solar zenith angle, half of the feature is in sunlight and the other half is in shadow. At this time, the contrast between the bright and dark portions of the image is sharp, and the stereoscopic effect at its peak. However, the ground features' large shadow area lowers the aesthetic quality at the same time. At noon, the solar zenith angle is close to 90 degrees, so the ground features are evenly exposed to light and can be clearly identified. But the shadow is also the shortest, and the stereoscopic effect is weak. Remote sensing images of high aesthetic quality should have a light-shadow balance. **Figure 6B** shows an ideal light-shadow distribution that results in high aesthetic quality. The right amount of shadow is produced with enough light and the right light direction: just enough to create the stereoscopic effect without shading over other features. While **Figure 6A** suffers from the lack of sunlight which results in a dim image, the light direction in **Figure 6C** creates too large shadow area that obscures the ice in the image, lowering the overall aesthetic quality.

Prominent theme

Since remote sensing images are taken from high altitudes, they are often occupied by dense ground features, which can easily make the viewer feel monotonous because of the lack of focus. Therefore, remote sensing image of high aesthetic quality should highlight the theme, drawing the viewer's attention to the key area of the picture. And the theme is often emphasized by image composition (Dhar et al., 2011), including rule of thirds, framing and repetition. When composing an image, professional photographers often divide the image using the imagery horizontal and vertical thirds lines and place important objects along these lines or at their intersections. This particular visual element placement is known as the rule of thirds (Krages, 2005). In **Figure 7A**, for example, the heart-shaped cloud is located at the intersection of two dividing lines. The cloud becomes a standout theme, with the green terrain serving as the backdrop. Just as the frame of a painting naturally draws people's attention to the painting, the frame of an object within an image does the same. A frame can be regular, complete, and closed, or it can be irregular, incomplete, and open. In **Figure 7B**, dark green woodlands, winding roads and houses form a frame to surround and highlight the colorful terraces. Apart from traditional image composition techniques, repetition can also be used to create a prominent theme. Repetition means using repeating shapes or a repetitive pattern inside the frame as part of the composition. While the overall repetition can easily draw attention and deepen the viewer's memory of the repeated objects, the repetitions that are slightly different from each other can produce a unique sense of rhythm in the picture (Shinkle, 2004). In remote sensing images, repetitive objects can be seen everywhere. From the bird's-eye view, these repetitive objects appear as different regular geometric figures, highlighting the distinct theme. **Figure 7C** serves as a good example. The repetitive circles in different shades of green, which are dotted with rectangle fields of corn and wheat, emphasize the image theme of farmlands.

Visual balance

Visual balance, a sense of weighted clarity created in a composition (Arnheim, 1956), influences how we perceive

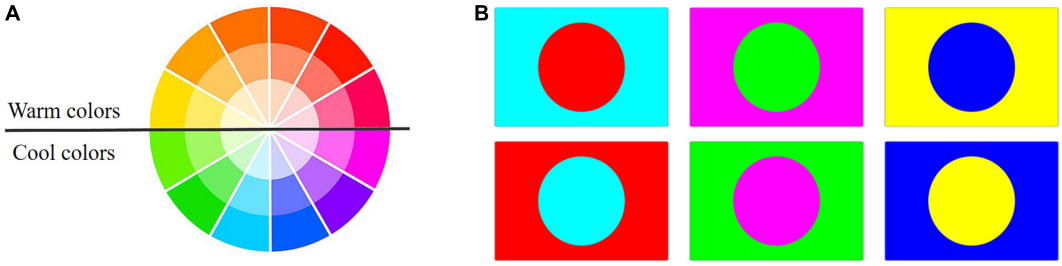


FIGURE 4
Cool-warm colors **(A)** and complementary colors **(B)**.

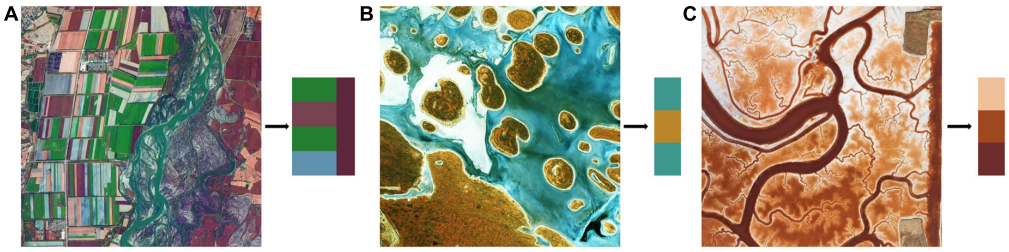


FIGURE 5
Remote sensing images with cool-warm contrast **(A)**, red-green color complementation **(B)**, and progressive color arrangement **(C)**.

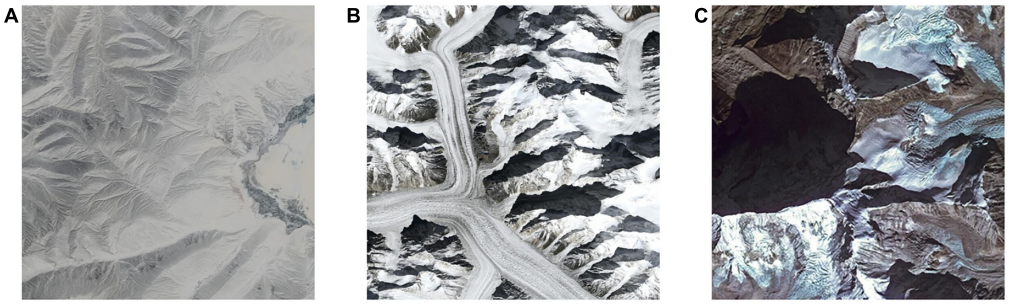


FIGURE 6
Remote sensing images with a lack of sunlight **(A)**, ideal light-shadow distribution **(B)**, and too large shadow area **(C)**.

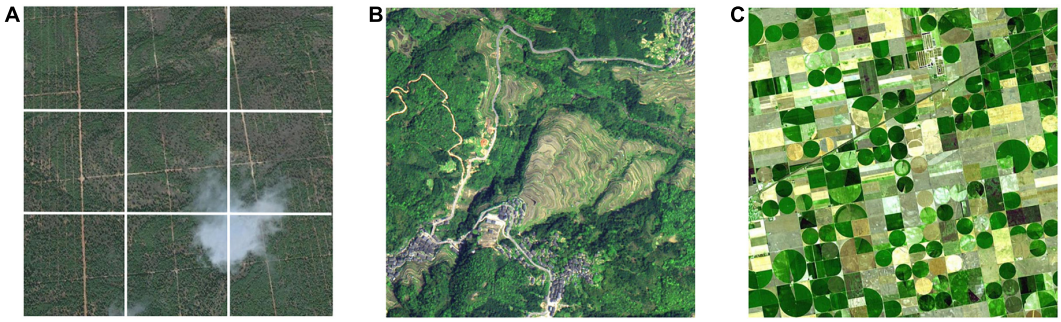


FIGURE 7
Remote sensing images that emphasize the theme using the rule of thirds **(A)**, framing **(B)**, and repetition **(C)**.

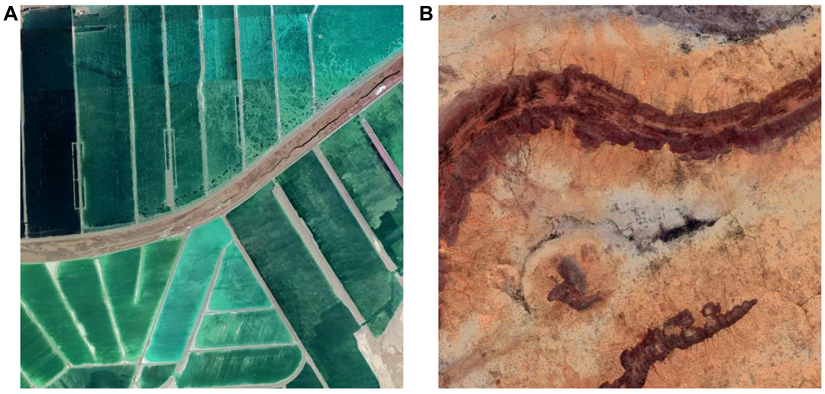


FIGURE 8
Object area **(A)** and distance from the image center **(B)** impact visual balance.

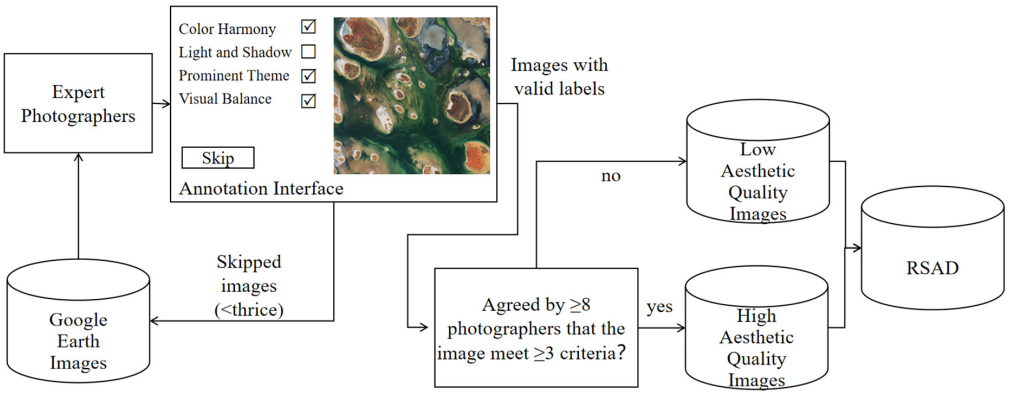


FIGURE 9
The overall labeling procedure of remote sensing aesthetics dataset (RSAD).

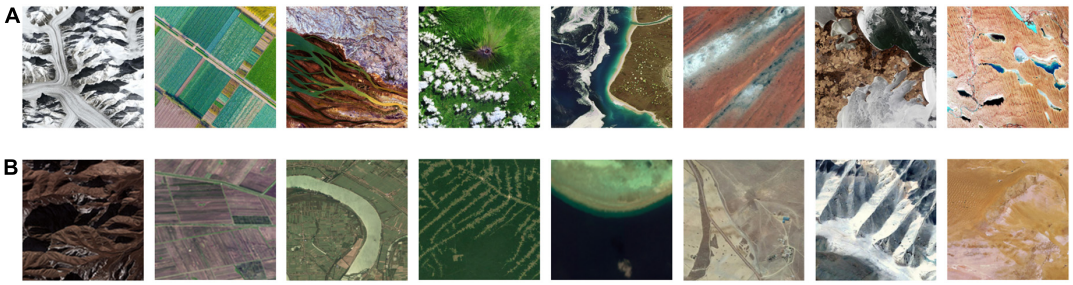


FIGURE 10
Remote sensing aesthetics dataset (RSAD) samples of high **(A)** and low **(B)** aesthetic quality.

aesthetic quality (Palmer et al., 2013). Visual balance builds upon the notion of visual weight, a perceptual analog to physical weight (Lok et al., 2004). An object is visually heavy if it takes up large area. The larger the area occupied by an object, the greater its visual weight is. Also, objects far from the image center frequently appear visually heavier than objects close to the image center. This is the visual Principle of Lever: Since the feature in the image represents a heavy object and the image center represents the lever's fulcrum, the distance between them functions as a lever (Xia, 2020). **Figure 8A** shows how object area impacts visual balance. The top and bottom portions of the image divided by a tilted line are roughly the

same size, whereas the two parts at the bottom are almost equally sized and are divided by a second, nearly diagonal line. Figure 8B shows how distance from the image center impacts visual balance. The long ridge on the upper part of the remote sensing image is of high visual weight. However, such visual weight is balanced by a smaller ridge farther from the center.

Dataset creation

After that, images are manually annotated. We invited professional photographers to evaluate the aesthetic quality of remote sensing images because they master photographic skills and understands the aesthetic preference of the public. They can decide whether the image satisfies the four evaluation criteria: color harmony, light and shadow, prominent theme and visual balance. If a photographer thinks an image satisfies at least three standards, the image will be considered beautiful. 15 photographers participated in the labeling procedure. If 8 or more photographers agree on the aesthetic quality of an image, then we will assign it the label of "high aesthetic quality". And the remaining images will be of "low aesthetic quality". In addition, we have added a "skip" option. To put it another way, if the photographer is unable to determine whether a remote sensing image satisfies the four standards, he can skip it. After three skips, an image's aesthetic quality is suggested to be blurred, so it will be removed from the dataset. The overall annotation process is depicted in Figure 9.

The expert photographers evaluated 1,500 samples, 117 of which were skipped, leaving 1,383 samples with valid labels. The RSAD dataset was finished with 875 positive samples and 508 negative samples. Figure 10 depicts samples of high (A) and low (B) aesthetic quality; images in the same column are of the same content type.

Learning remote sensing aesthetics with deep learning

In this study, we used binary classification to discriminate a remote sensing image into "high aesthetic quality" or "low aesthetic quality." And ResNet-18 served as the backbone network. The ResNet residual module can solve the problem of vanishing gradients and is calculated as follows. Define a residual block in the form of $y_l = h(X_l) + F(X_l, W_L)$, where x and y are the input and output vectors of the residual block, respectively, $h(X_l)$ is the feature mapping function, and $F(X_l, W_L)$ is the residual mapping function to be learned, $f(y_l)$ is the activation function.

$$y_l = h(X_l) + F(X_l, W_L) \quad (1)$$

$$X_{l+1} = f(y_l) \quad (2)$$

Figure 11 depicts the ResNet-18 network structure and parameters, including the input, output, and convolutional and pooling layers in the middle. Input images of 512 x 512 and get the output of 1 x 2 after training. The first parameter represents the probability of being unaesthetic, and the second digit is the probability of being aesthetic. If the probability of being aesthetic is greater than the probability of not being aesthetic, the image is considered visually appealing, and vice versa.

The input section consists of a large convolution kernel (7 x 7, stride 2) and a max-pooling (3 x 3, stride 2). This step converts the 512 x 512 input image to a 128 x 128 feature map. The convolution layer then extracts feature information using two 3 x 3 convolutions and adds it directly to the original data in a residual block; the output part converts the feature map to 1 x 1 using global adaptive average pooling and passes it through the fully connected layer. Table 1 displays the model's input and output for each layer.

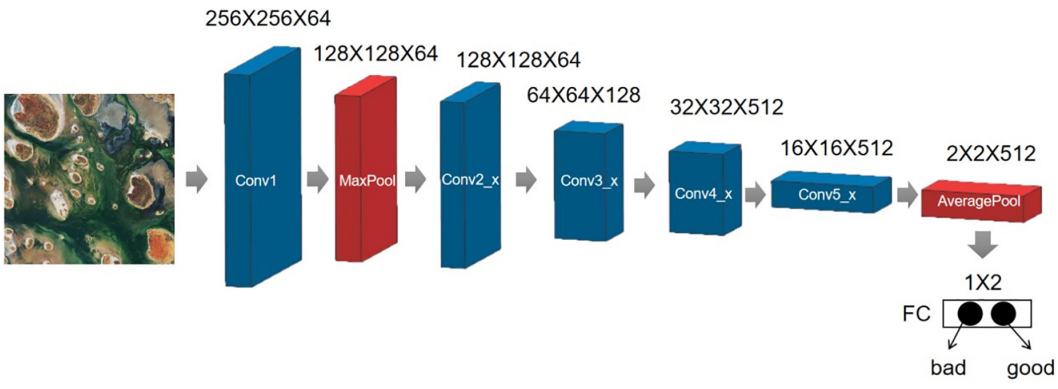


FIGURE 11
The ResNet-18 network structure and parameters.

TABLE 1 ResNet-18's input and output for each layer.

Layer name	Operation	Input	Output
Conv1	7 × 7, 64, stride2	512 × 512 × 1	256 × 265 × 64
Max pool	3 × 3, max_pooling, stride2	256 × 265 × 64	128 × 128 × 64
Conv2_x	$\begin{bmatrix} 3 \times 3 & 64 \\ 3 \times 3 & 64 \end{bmatrix} \times 2$	128 × 128 × 64	128 × 128 × 64
Conv3_x	$\begin{bmatrix} 3 \times 3 & 128 \\ 3 \times 3 & 128 \end{bmatrix} \times 2$	128 × 128 × 64	64 × 64 × 128
Conv4_x	$\begin{bmatrix} 3 \times 3 & 256 \\ 3 \times 3 & 256 \end{bmatrix} \times 2$	64 × 64 × 128	32 × 32 × 256
Conv5_x	$\begin{bmatrix} 3 \times 3 & 512 \\ 3 \times 3 & 512 \end{bmatrix} \times 2$	32 × 32 × 256	16 × 16 × 512
Average pool	avg_pooling	16 × 16 × 512	2 × 2 × 512
FC	1,000–d fc + softmax	2 × 2 × 512	1 × 2

Interpreting aesthetic assessment with gradient-weighted class activation mapping

While deep learning enables good performance in the aesthetic classification of remote sensing images, it lacks interpretability. As the process of aesthetic evaluation involves visual stimulation (Cheung et al., 2019), visualizing the prominent image area that influenced model's decision can be a solution. Therefore, in an effort to interpret the deep-learning based aesthetic assessment and compare it with the cognitive process of human brain, we adopted the class activation map Grad-CAM proposed by Selvaraju et al. (2017). By referring to the gradient information learned by convolutional neurons, we can generate visual explanations from any CNN-based network without architectural changes or retraining.

Gradient-weighted class activation mapping (Grad-CAM) uses the gradient information flowing into the last convolutional layer to draw a heat map, as shown in **Figure 12**. The network first performs forward propagation to obtain the output of feature layer A (the last convolutional layer of ResNet in this case) and the predicted value y. Assuming that the predicted value of a remote sensing image by the network is y^c, then back-propagating y^c can obtain the gradient information $\overline{A}$ that is back-transmitted to the feature layer. The importance of each channel of the feature layer A is obtained by calculation and then weighted and summed. After passing through the residual module ReLU, we can obtained the final result of Grad-CAM.

Equations 3 and 4 show the Grad-CAM calculation formula. Among them, A_{ij}^k represents the point (i, j) of the kth channel of feature map A, y^c represents the output of class c, and $\frac{\partial y^c}{\partial A_{ij}^k}$ represents the partial derivative of y^c for all feature maps A_{ij}^k of the last layer of CNN. The ReLU function produces a heat map whose values are positively correlated with class c. The negative part indicates that it does not belong to class c, which can be viewed as posing an inhibitory effect and thus can be filtered out with ReLU.

$$\alpha_k^c = \frac{1}{Z}\sum_i\sum_j \frac{\partial y^c}{\partial A_{ij}^k} \tag{3}$$

$$L_{Grad-CAM}^c = ReLU\left(\sum_k \alpha_k^c A^k\right) \tag{4}$$

The Grad-CAM heat map can show which area contributes the most to an image's aesthetic quality prediction. The redder parts of the heat map have a greater impact on the prediction than the bluer parts. As a result, using Grad-CAM, we can verify the four evaluation criteria we have concluded of remote sensing image aesthetic quality: color harmony, light and shadow, prominent theme and visual balance.

Experimental results and analysis

Experimental design

In this paper, we conducted experiments on the Remote Sensing Aesthetics Dataset. 80% of the samples are for training, and the remaining 20% are for testing. To facilitate network training, we resized the images to 512 x 512 and fed them into the ResNet-18 architecture. After that, we used quantitative indicators to assess model performance.

Regarding training parameters, we trained 100 epochs with ResNet-18, batch size = 16, without any pre-trained weights. Stochastic gradient descent is the optimizer used in back-propagation, with the hyperparameter learning rate set to 1×10^{-4}. The learning rate controls the update of the weights, and a lower learning rate allows the model to converge better. Cross-entropy is the loss function, and it is defined as follows: y_i represents the aesthetic label of sample i, the positive class is 1, and the negative class is 0; p_i represents the probability that sample i is predicted to be a positive class.

$$L = \frac{1}{N}\sum_i L_i = \frac{1}{N}\sum_i -\left[y_i \cdot \log\left(p_i\right) + \left(1 - y_i\right) \cdot \log\left(1 - p_i\right)\right] \tag{5}$$

Evaluation metrics

In this paper, finding visually attractive remote sensing images is a binary classification task in which samples are classified as either high or low aesthetic quality. The confusion matrix is thus used to calculate the four parameters TP, FP, TN, and FN to evaluate model performance. Each parameter in the confusion matrix is explained as follows.

- TP (True Positive): High-aesthetic-quality image predicted of high aesthetic quality.
- TN (True Negative): Low-aesthetic-quality image predicted of low aesthetic quality.
- FP (False Positive): Low-aesthetic-quality image predicted of high aesthetic quality.
- FN (False Negative): High-aesthetic-quality image predicted of low aesthetic quality.

We can calculate accuracy, recall, precision, and F1-score of our method based on these four parameters, shown in Equations 6–9.

- Accuracy: the proportion of accurately predicted images in all images.

$$Accuracy = \frac{(TP+TN)}{(TP+FN+TN+FP)} \times 100 \tag{6}$$

- Recall: the proportion of accurately predicted aesthetic images in all correct predictions.

$$Recall = \frac{TP}{TP+FN} \tag{7}$$

- Precision: the proportion of images predicted as high aesthetic quality of all aesthetic images.

$$Precision = \frac{TP}{TP+FP} \tag{8}$$

- F1-score: the harmonic mean of precision and recall, reflecting the robustness of our model.

$$F1 = \frac{2TP}{2TP+FN+FP} = \frac{2 \cdot \mathrm{Pr}\,ecision \cdot Recall}{\mathrm{Pr}\,ecision + Recall} \tag{9}$$

Results and analysis

Automatic remote sensing aesthetic assessment

The test set contains 277 samples, and it has an accuracy of 91.34%. **Figure 13A** depicts the confusion matrix for the test set, and the classification results for each cell of the confusion matrix are visualized in **Figure 13B**.

Judging from the True-Positive cell where images of high aesthetic quality are correctly predicted, we can conclude that our model can distinguish the images that meet the four evaluation standards. In the lower-right image of farmland, there is feature repetition and a prominent theme. Light and shadow contrast can be found in the upper-left image of glacier. And color harmony exists in the upper-right image of coral reef. Similar conclusion can be reached when we examine all images in the True-Negative cell. Looking at the farmland image in the upper-left corner with a meandering purple outline and the image of meandering rivers in the lower-right corner, we can see that the model may find the winding shape visually unappealing.

Based on the confusion matrix, we calculated accuracy, recall, precision and F1-score. The accuracy is 91.34%, demonstrating the overall good performance. The precision is 0.90, which indicates the effectiveness of the model in identifying images of low aesthetic quality. Meanwhile, the model is good at identifying high-aesthetic-quality images, as the recall reaches 0.67. While F1-score of 0.77 proves the robustness of the model as well.

From the analysis above, we can conclude that the ResNet model we trained can accurately distinguish between remote sensing images of high and low aesthetic quality.

Attention mechanism in automatic aesthetic assessment

In an effort to interpret the deep-learning based aesthetic assessment, we adopted Grad-CAM to highlight the prominent image area that influenced model's decision, as shown in **Figure 14**. By examining how those areas matches human attention on the four aesthetic standards, we can compare how ResNet performs aesthetic evaluation with the actual cognitive process of aesthetics in the human brain.

Color harmony

Color harmony is related to the relationship between colors, including cool-warm colors, complementary colors and the arrangement relations of colors. Cool-warm colors are linked to the feelings they evoke and the emotions with which we identify when looking at them. Complementary colors are a pair of color stimuli whose mixture color matches a given neutral. And color arrangement relations are the progressive color changes in hue, brightness or saturation. Grad-CAM highlighted the warm red blocks in the lower right corner and the cool-toned mountains in **Figure 14A**, indicating the cool-warm color contrast.

Light and shadow

When ground features are exposed to sunlight, shadows will occur. A right proportion of light and shade can impart depth perception to the scene, creating a stereoscopic effect. However, a large shadow area will reduce the aesthetic quality. So remote sensing image of high aesthetic quality should have light-shadow balance, as shown in **Figure 14B**. The lower-right corner of the image has more shade areas whereas the upper-left corner has more exposure to light, both are highlighted on the heat map.

Prominent theme

Remote sensing image of high aesthetic quality should highlight the theme, drawing the viewer's attention to the key area of the picture. And prominent theme is realized by repetition, rule of thirds and framing. **Figure 14C** serves as a good example of repetition. Grad-CAM captures the repetitive

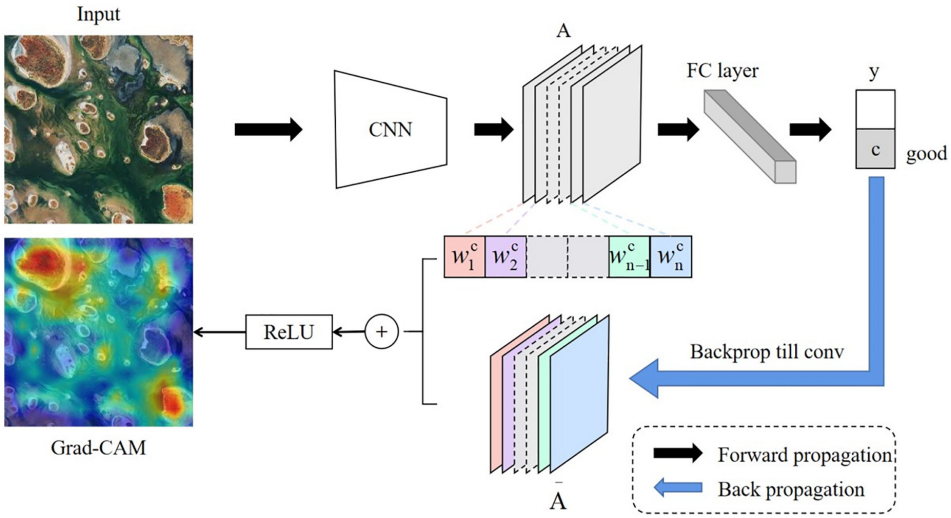


FIGURE 12
The work flow of gradient-weighted class activation mapping (Grad-CAM).

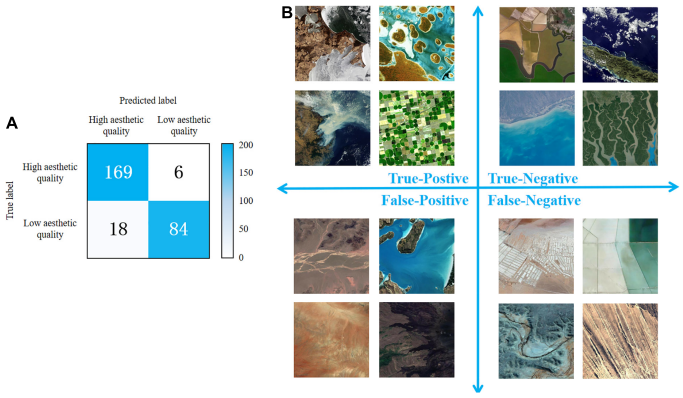


FIGURE 13
Confusion matrix for the test set **(A)** and the classification results for each cell of the confusion matrix **(B)**.

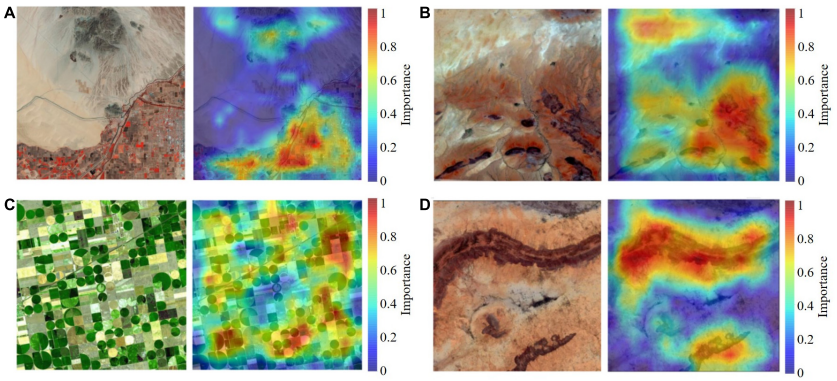


FIGURE 14
Gradient-weighted class activation mapping (Grad-CAM) captures the image's cool-warm contrast **(A)**, the light and shadow area **(B)**, the repetitive circles and the rectangle fields of corn and wheat **(C)**, and the two balancing portions of visual weight **(D)**.

circles in various shades of green, as well as the rectangle fields of corn and wheat, all of which emphasize the image theme of farmlands.

Visual balance

Visual balance, a sense of weighted clarity created in a composition, is influenced by the feature's area and its distance from the image center. In **Figure 14D**, the long ridge on the upper part of the remote sensing image is of high visual weight. A smaller ridge farther from the center, however, balances such visual weight. And both ridges are highlighted on the heat map.

Judging from the heat maps' highlighted regions, we can conclude that ResNet's aesthetic evaluation is involved with something similar to the attention mechanism of the brain's visual aesthetic process. It proves the interpretability of automatic remote sensing aesthetic assessment as well.

Conclusion and future work

To enable non-scientific application of remote sensing images, while inspired by the brain's cognitive process and the use of CNN in image aesthetic assessment, we propose an interpretable approach for automatic aesthetic assessment of remote sensing images. Firstly, we created the Remote Sensing Aesthetics Dataset. We collected remote sensing images from Google Earth, designed the four evaluation criteria of remote sensing image aesthetic quality—color harmony, light and shadow, prominent theme, and visual balance—and then labeled the samples based on expert photographers' judgment on the four evaluation criteria. Secondly, we feed RSAD into the ResNet-18 architecture for training. Experimental results show that the proposed method can accurately identify visually pleasing remote sensing images. Finally, we provided a visual explanation of aesthetic assessment by adopting Grad-CAM to highlight the important image area that influenced model's decision. Overall, this paper is the first to propose and realize automatic aesthetic assessment of remote sensing images, contributing to the non-scientific applications of remote sensing and demonstrating the interpretability of deep-learning based image aesthetic evaluation.

But some limitations still exist, so we need to further our research. First, we treat aesthetic assessment as a binary classification problem in this paper. This is because assigning an aesthetic quality score requires more voters and samples. Therefore, estimating an aesthetic quality score for each remote sensing image using regression methods will be part of the future work. Second, we only used ResNet, a scene-based CNN, as the backbone of evaluation, which is not a novel method. To ensure that the model is more dedicated to remote sensing aesthetic quality, we should fine-tune the backbone network by adjusting its blocks and layers. Third, objectivity and subjectivity coexist in aesthetic assessment. So we are unable to verify the aesthetic classification results due to the possible subjectivity of aesthetics. Thus, we will continue to work on bridging the objective and subjective aspects of remote sensing aesthetics through well-designed psychology surveys. To sum up, more research and practice in the fields of neural science, remote sensing, deep learning, aesthetics, and psychology will be needed in the future for the automatic aesthetic evaluation of remote sensing images.

Data availability statement

The original contributions presented in this study are included in the article/supplementary material, further inquiries can be directed to the corresponding author.

Author contributions

JT: conceptualization, methodology, formal analysis, writing—original draft, and writing—review and editing. GZ: conceptualization and supervision. PK: validation and writing—review and editing. YR: validation and visualization. ZW: investigation and data curation. HC: resources and writing—review and editing. QG: investigation and writing—review and editing. All authors contributed to the article and approved the submitted version.

Funding

This work was supported by the Postdoctoral Innovation Talent Support Program (BX2021222), in part by the National Key Research and Development Program of China (No: 2018YFC0825803).

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

References

Arnheim, R. (1956). *Art and visual perception: A psychology of the creative eye*. California, CA: Univ of California Press.

Brady, L., and Phillips, C. (2003). Aesthetics and usability: A look at color and balance. *Usability News* 5, 2–5.

Brill, M. H. (2007). Camera color gamut: Spray-painting the invisible definition. *Color Res. Appl.* 32, 236–237. doi: 10.1002/col.20317

Burchett, K. E. (2002). Color harmony. *Color Res. Appl.* 27, 28–31. doi: 10.1002/col.10004

Cela-Conde, C. J., Agnati, L., Huston, J. P., Mora, F., and Nadal, M. (2011). The neural foundations of aesthetic appreciation. *Progress Neurobiol.* 94, 39–48. doi: 10.1016/j.pneurobio.2011.03.003

Cheung, M.-C., Law, D., Yip, J., and Wong, C. W. (2019). Emotional responses to visual art and commercial stimuli: Implications for creativity and aesthetics. *Front. Psychol.* 10:14. doi: 10.3389/fpsyg.2019.00014

Datta, R., and Wang, J. Z. (2010). "Acquine: Aesthetic quality inference engine-real-time automatic rating of photo aesthetics," in *Proceedings of the international conference on Multimedia information retrieval*, New York, NY, 421–424. doi: 10.1145/1743384.1743457

Datta, R., Joshi, D., Li, J., and Wang, J. Z. (2006). "Studying aesthetics in photographic images using a computational approach," in *Proceedings of the European conference on computer vision*. (Berlin: Springer), 288–301. doi: 10.1007/11744078_23

Deng, Y., Loy, C. C., and Tang, X. (2017). Image aesthetic assessment: An experimental survey. *IEEE Signal Process. Mag.* 34, 80–106. doi: 10.1109/MSP.2017.2696576

Dhar, S., Ordonez, V., and Berg, T. L. (2011). "High level describable attributes for predicting aesthetics and interestingness," in *2011 IEEE conference on computer vision and pattern recognition (CVPR) Cvpr 2011*. (Piscataway, NJ: IEEE), 1657–1664. doi: 10.1109/CVPR.2011.5995467

Fisher, G. B., Amos, C. B., Bookhagen, B., Burbank, D. W., and Godard, V. (2012). Channel widths, landslides, faults, and beyond: The new world order of high-spatial resolution Google Earth imagery in the study of earth surface processes. *Geol. Soc. Am. Spec. Pap.* 492, 1–22. doi: 10.1130/2012.2492(01)

Freeman, M. (2007). *The complete guide to light & lighting in digital photography*. New York, NY: Sterling Publishing Company, Inc.

Grayson, C. (2016). *An artist's surreal view of Australia – created from satellite data captured 700km above Earth* [Online]. *theconversation*. Available online at: https://theconversation.com/an-artists-surreal-view-of-australia-created-from-satellite-data-captured-700km-above-earth-96718 (accessed August 16, 2022).

He, K., Zhang, X., Ren, S., and Sun, J. (2016). "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, Silver Spring, MD, 770–778. doi: 10.1109/CVPR.2016.90

Itten, J. (1975). *Design and form: The basic course at the Bauhaus and later*. Hoboken, NJ: John Wiley & Sons.

Kalivoda, O., Vojar, J., Skøivanová, Z., and ZahradníK, D. (2014). Consensus in landscape preference judgments: The effects of landscape visual aesthetic quality and respondents' characteristics. *J. Environ. Manag.* 137, 36–44. doi: 10.1016/j.jenvman.2014.02.009

Karayev, S., Trentacoste, M., Han, H., Agarwala, A., Darrell, T., Hertzmann, A., et al. (2013). Recognizing image style. *arXiv* [Preprint]. arXiv 1311.3715. doi: 10.5244/C.28.122

Khosla, A., Das Sarma, A., and Hamid, R. (2014). "What makes an image popular?," in *Proceedings of the 23rd international conference on World wide web*, New York, NY, 867–876. doi: 10.1145/2566486.2567996

Kim, W.-H., Choi, J.-H., and Lee, J.-S. (2018). Objectivity and subjectivity in aesthetic quality assessment of digital photographs. *IEEE Trans. Affect. Comput.* 11, 493–506. doi: 10.1109/TAFFC.2018.2809752

Krages, B. P. (2005). *The art of composition*. New York, NY: Allworth Communications.

Li, C., and Chen, T. (2009). Aesthetic visual quality assessment of paintings. *IEEE J. Sel. Top. Signal Process.* 3, 236–252. doi: 10.1109/JSTSP.2009.2015077

Li, C., Gallagher, A., Loui, A. C., and Chen, T. (2010). "Aesthetic quality assessment of consumer photos with faces," in *Proceedings of the 2010 IEEE international conference on image processing*. (Piscataway, NJ: IEEE), 3221–3224. doi: 10.1109/ICIP.2010.5651833

Lipton, Z. C. (2018). The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery. *Queue* 16, 31–57. doi: 10.1145/3236386.3241340

Liu, S., Tian, G., and Xu, Y. (2019). A novel scene classification model combining ResNet based transfer learning and data augmentation with a filter. *Neurocomputing* 338, 191–206. doi: 10.1016/j.neucom.2019.01.090

Lok, S., Feiner, S., and Ngai, G. (2004). "Evaluation of visual balance for automated layout," in *Proceedings of the 9th international conference on intelligent user interfaces*, NewYork, NY, 101–108. doi: 10.1145/964442.964462

London, B., Upton, J., Stone, J., Kobre, K., and Brill, B. (2011). *Photography*. London: Pearson.

Lu, X., Lin, Z., Jin, H., Yang, J., and Wang, J. Z. (2014). "Rapid: Rating pictorial aesthetics using deep learning," in *Proceedings of the 22nd ACM international conference on multimedia*, New York, NY, 457–466. doi: 10.1145/2647868.2654927

Luo, W., Wang, X., and Tang, X. (2011). "Content-based photo quality assessment," in *Proceedings of the 2011 international conference on computer vision*. (Piscataway, NJ: IEEE), 2206–2213. doi: 10.1109/ICCV.2011.6126498

Luo, Y., and Tang, X. (2008). "Photo and video quality evaluation: Focusing on the subject," in *Proceedings of the European conference on computer vision*. (Berlin: Springer), 386–399. doi: 10.1007/978-3-540-88690-7_29

Mnih, V., Heess, N., and Graves, A. (2014). Recurrent models of visual attention. *Adv. Neural Inf. Process. Syst.* 27, 13–21.

Moreland, K. (2009). "Diverging color maps for scientific visualization," in *Proceedings of the international symposium on visual computing*. (Berlin: Springer), 92–103. doi: 10.1007/978-3-642-10520-3_9

Palmer, S. E., Schloss, K. B., and Sammartino, J. (2013). Visual aesthetics and human preference. *Annu. Rev. Psychol.* 64, 77–107. doi: 10.1146/annurev-psych-120710-100504

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D. (2017). "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *Proceedings of the IEEE international conference on computer vision*, Piscataway, NJ, 618–626. doi: 10.1109/ICCV.2017.74

Shamoi, P., Inoue, A., and Kawanaka, H. (2020). Modeling aesthetic preferences: Color coordination and fuzzy sets. *Fuzzy Sets Syst.* 395, 217–234. doi: 10.1016/j.fss.2019.02.014

Shinkle, E. (2004). Boredom, repetition, inertia: Contemporary photography and the aesthetics of the banal. *Mosaic Interdiscip. Crit. J.* 37, 165–184.

Skov, M., and Nadal, M. (2020). A farewell to art: Aesthetics as a topic in psychology and neuroscience. *Perspect. Psychol. Sci.* 15, 630–642. doi: 10.1177/1745691619897963

Tian, X., Dong, Z., Yang, K., and Mei, T. (2015). Query-dependent aesthetic model with deep learning for photo quality assessment. *IEEE Trans. Multimed.* 17, 2035–2048. doi: 10.1109/TMM.2015.2479916

Todd, J. T., Norman, J. F., Koenderink, J. J., and Kappers, A. M. (1997). Effects of texture, illumination, and surface reflectance on stereoscopic shape perception. *Perception* 26, 807–822. doi: 10.1068/p260807

Wang, R., Zhao, J., and Liu, Z. (2016). Consensus in visual preferences: The effects of aesthetic quality and landscape types. *Urban For. Urban Green.* 20, 210–217. doi: 10.1016/j.ufug.2016.09.005

Wang, W., Shen, J., and Ling, H. (2018). A deep network solution for attention and aesthetics aware photo cropping. *IEEE Trans. Pattern Anal. Mach. Intell.* 41, 1531–1544. doi: 10.1109/TPAMI.2018.2840724

Wong, L.-K., and Low, K.-L. (2009). "Saliency-enhanced image aesthetics class prediction," in *Proceedings of the 2009 16th IEEE international conference on image processing (Icip)*. (Piscataway, NJ: IEEE), 997–1000.

Wu, J., Xing, B., Si, H., Dou, J., Wang, J., Zhu, Y., et al. (2020). Product design award prediction modeling: Design visual aesthetic quality assessment via Dcnns. *IEEE Access* 8, 211028–211047. doi: 10.1109/ACCESS.2020.3039715

Xia, Y. (2020). Visual psychological analysis of photographic composition balance. *Hubei Inst. Fine Arts J.* 4, 20–23.

Yamazaki, F., Liu, W., and Takasaki, M. (2009). "Characteristics of shadow and removal of its effects for remote sensing imagery," in *Proceedings of the 2009 IEEE international geoscience and remote sensing symposium*. (Piscataway, NJ: IEEE), Iv–426–Iv–429. doi: 10.1109/IGARSS.2009.5417404

Yin, H., Li, Y., Shi, J., Jiang, J., Li, L., and Yao, J. (2022). Optimizing local alignment along the seamline for parallax-tolerant orthoimage mosaicking. *Remote Sens.* 14:3271. doi: 10.3390/rs14143271

Zhang, T., Stackhouse, P. W. Jr., Macpherson, B., and Mikovitz, J. C. (2021). A solar azimuth formula that renders circumstantial treatment unnecessary without compromising mathematical rigor: Mathematical setup, application and extension of a formula based on the subsolar point and atan2 function. *Renew. Energy* 172, 1333–1340. doi: 10.1016/j.renene.2021.03.047

frontiers | Frontiers in Neuroscience

TYPE Original Research
PUBLISHED 30 November 2022
DOI 10.3389/fnins.2022.1074706

OPEN ACCESS

EDITED BY
Yuqi Han,
Tsinghua University, China

REVIEWED BY
Ying Zhu,
Wuhan Institute of Technology, China
Junying Zeng,
Wuyi University, China
Zhaocheng Wang,
Hebei University of Technology, China

*CORRESPONDENCE
Yupei Wang
wangyupei2019@outlook.com

SPECIALTY SECTION
This article was submitted to Perception Science, a section of the journal Frontiers in Neuroscience

RECEIVED 19 October 2022
ACCEPTED 08 November 2022
PUBLISHED 30 November 2022

CITATION
Shi H, He C, Li J, Chen L and Wang Y (2022) An improved anchor-free SAR ship detection algorithm based on brain-inspired attention mechanism. *Front. Neurosci.* 16:1074706. doi: 10.3389/fnins.2022.1074706

An improved anchor-free SAR ship detection algorithm based on brain-inspired attention mechanism

Hao Shi[1,2,3], Cheng He[1,3], Jianhao Li[1,3], Liang Chen[1,2,3] and Yupei Wang[1,2,3]*

[1]Radar Research Lab, School of Information and Electronics, Beijing Institute of Technology, Beijing, China, [2]Chongqing Innovation Center, Beijing Institute of Technology, Chongqing, China, [3]Beijing Key Laboratory of Embedded Real-Time Information Processing Technology, Beijing Institute of Technology, Beijing, China

As a computing platform that can deal with problems independently and adapt to different environments, the brain-inspired function is similar to the human brain, which can effectively make use of visual targets and their surrounding background information to make more efficient and accurate decision results. Currently synthetic aperture radar (SAR) ship target detection has an important role in military and civilian fields, but there are still great challenges in SAR ship target detection due to the problems of large span of ship scales and obvious feature differences. Therefore, this paper proposes an improved anchor-free SAR ship detection algorithm based on brain-inspired attention mechanism, which efficiently focuses on target information ignoring the interference of complex background. First of all, most target detection algorithms are based on the anchor method, which requires a large number of anchors to be defined in advance and has poor generalization capability and performance to be improved in multi-scale ship detection, so this paper adopts an anchor-free detection network to directly enumerate potential target locations to enhance algorithm robustness and improve detection performance. Secondly, in order to improve the SAR ship target feature extraction capability, a dense connection module is proposed for the deep part of the network to promote more adequate deep feature fusion. A visual attention module is proposed for the shallow part of the network to focus on the salient features of the ship target in the local area for the input SAR images and suppress the interference of the surrounding background with similar scattering characteristics. In addition, because the SAR image coherent speckle noise is similar to the edge of the ship target, this paper proposes a novel width height prediction constraint to suppress the noise scattering power effect and improve the SAR ship localization accuracy. Moreover, to prove the effectiveness of this algorithm, experiments are conducted on the SAR

ship detection dataset (SSDD) and high resolution SAR images dataset (HRSID). The experimental results show that the proposed algorithm achieves the best detection performance with metrics AP of 68.2% and 62.2% on SSDD and HRSID, respectively.

KEYWORDS

anchor-free, synthetic aperture radar, ship detection, brain-inspired, attention mechanism

1. Introduction

The brain-inspired concept originates from the human brain, which can focus on the target information while selectively ignoring the interference of redundant information when facing a large amount of information, and this attention mechanism in the human brain can enhance the target cognition and understanding. By imitating the processing mode of information in the human brain, the brain-inspired can improve the information acquisition ability of the target in practical applications, and finally complete the cognitive and understanding of the target.

In SAR ship detection, the target information usually contains a large number of redundant interference components, and being able to obtain the target information accurately plays an important role in the detection results. Because the brain-inspired ability to effectively pay attention to key regions in the target scene, we take SAR ship detection as an example to explore an algorithm that can effectively extract SAR ship information and improve SAR ship detection accuracy.

Synthetic aperture radar (SAR) is an active microwave imaging sensor that can effectively collect large area data under any weather conditions, such as day, night, and foggy days, and eventually generate high-resolution SAR images. Because of its all-day and all-weather high-resolution imaging capability, SAR plays an important role in marine ship target detection (Li et al., 2016), such as marine rescue, marine law enforcement and other civilian fields, as well as precise detection, ship target detection, and other military fields. However, it is difficult to detect ship targets in SAR images due to the large scale span of ship targets and obvious feature differences. Therefore, an efficient target detector is needed to detect SAR ship targets.

Traditional SAR target detection methods can be broadly classified into three categories: threshold (Wang et al., 2016), statistical (Song and Yang, 2015), and transform methods (He et al., 2019). The main steps include the pre-processing stage of processing the input image into a more recognizable image, the candidate region extraction stage of extracting possible target pixels as candidate targets, and the recognition stage of identifying targets within the potential region. Among the existing conventional SAR target detection algorithms, the constant false alarm rate (CFAR) method (Wang et al., 2017) is one of the most commonly used techniques, which is based on the main idea of establishing a sea clutter distribution model based on local sea clutter data and plotting the probability density curve of the sea clutter distribution model, then calculating the adaptive threshold based on the typical false alarm probability, and finally using the adaptive threshold to detect the target in the SAR image. Although the CFAR method has been widely used for SAR ship target detection, it relies on the modeling of sea clutter data and adapts to simple scenarios, and does not adapt to multi-scale ship detection in complex backgrounds.

With the rapid theoretical development of deep learning, various deep learning models have emerged, which are widely used in the field of image processing due to their advantages such as powerful feature characterization ability and automatic learning. For feature misalignment and variation of target appearance in SAR multi-scale target detection, Tang et al. (2022) proposed scale-aware feature pyramid network with scale-adaptive feature extraction module and learnable anchor point assignment strategy. For redundancy-oriented computation and background interference in the remote sensing domain, Deng et al. (2022) proposed fast anchor point refinement network with rotational alignment module and balanced regression loss function. To improve the SAR multi-scale ship detection performance, Cui et al. (2019) proposed dense attention pyramid network by fusing the convolutional attention module with the features of each layer to highlight the salient features of each layer. Since SAR ship targets are difficult to distinguish from the surrounding background, Yang et al. (2022) proposed robust detection network by introducing coordinate attention approach to obtain more representative semantic features. To obtain better detection performance in practical industrial applications, Gao et al. (2022) proposed efficient SAR ship detection network with targeted skill fusion strategy based on Yolov4.

The above detection algorithms are all based on anchor detectors, and although these methods achieve better performance in target detection, there are still some shortcomings. Firstly, the algorithms need to manually set some hyperparameters according to the data, which are sensitive to ship targets with large scale span. Secondly, the algorithms usually generate a large number of anchor boxes

on the image, while SAR ship targets account for a small percentage of the image, and a large number of irrelevant anchor boxes waste computational resources. Moreover, when the targets are densely arranged, the overlapping area of candidate anchor boxes is large, and some targets are missed under non-maximum suppression. Therefore, it is necessary to propose an efficient anchor-free detector in SAR ship target detection.

The general anchor-free detector is designed based on natural scene images, while SAR images are very different from natural scene images, and the detection results are not good if the anchor-free detector is directly applied to SAR ship target detection. First of all, the SAR image coherent speckle noise is relatively large, the ship target is relatively similar to the clutter, and the island, port and building backgrounds have high grayscale characteristics easily confused with the ship target, so the SAR ship target features are difficult to extract, and problems such as missed detection and false detection are easy to occur in the detection results. In order to detect objects more effectively in existing detection algorithms, Deng et al. (2021) introduced dynamic weights to encourage the filters to focus on more reliable regions during the training phase. Han et al. (2019) added global context patches in the training phase of the model to better distinguish the target from the background. Zhao et al. (2017) adopted high confidence update strategies and study mechanisms to avoid model corruption and handle occlusion. Han et al. (2017) utilized a co-training paradigm to formulate multi-feature templates with inherently complementary information into a correlation filter model to extract valid feature targets. Wang et al. (2022) introduced deep residual networks into dictionary learning to extract rich image information. Lin et al. (2017a) developed top-down architectures with lateral connections for building high-level feature maps at various scales. Although existing feature extraction networks can effectively extract target features, they often lack the targeting of different feature layers in the network. The deep part of the network has a relatively large perceptual field and rich semantic features, and we propose a dense connection module for the deep part of the network to promote more adequate deep feature fusion. The shallow part of the network has a relatively small perceptual field and rich fine-grained details, and we propose a visual attention module for the shallow part of the network to focus on the salient features of the ship target in the local area for the input SAR images and suppress the interference of the surrounding background with similar scattering characteristics. In addition, because the scattered power distribution of the surrounding background in the near-shore scene of SAR images is similar to the edge of the ship target, it is easy to lead to the offset between some ship predicted positions and real positions, and the ship target is not localized correctly. For this reason, we propose a novel width height prediction constraint, which considers the overlapping area of the predicted box and the real box, the real difference between the width and length of the edge and the loss gradient reweighting to improve the ship target localization accuracy.

In conclusion, drawing on the idea that the brain-inspired can effectively use visual targets and their surrounding background information, we propose an improved anchor-free SAR ship detection algorithm based on brain-inspired attention mechanism. The main contributions are summarized as follows.

1. We propose an improved anchor-free SAR ship detection algorithm, which directly enumerates potential target locations and classifies them with better generalization capability compared to the anchor method, and makes targeted improvements to different feature layers of the network to improve SAR ship detection accuracy.
2. We design a dense connection module and a visual attention module for feature extraction. The deep part of the network is richer in semantic features, and the dense connection module promotes more adequate deep feature fusion. The shallow part of the network is richer in fine-grained details, and the visual attention module focuses on the salient features of the target in the local area and suppresses the surrounding background interference, which can eventually detect the SAR ship target more effectively.
3. We design a novel width height prediction constraint, which considers the overlapping area of the prediction box and the real box, the real difference between the length and width of the edge and the loss gradient reweighting, which suppresses the influence of the near-shore background on SAR ship target localization and improves the SAR ship target localization accuracy.

2. Related work

Since the concept of deep learning (Hinton and Salakhutdinov, 2006) was proposed, deep learning has gradually shown great advantages over traditional methods for various classification and regression tasks, and target detection using deep learning has now become mainstream. Existing target detection methods are mainly divided into two categories: anchor-based detectors and anchor-free detectors.

In the anchor-based detectors, first a series of sliding windows are predefined on the feature map, then they are divided into positive and negative samples according to the IOU values, and finally the detection results are obtained by classification regression on the divided positive and negative samples. The anchor-based detectors can be classified into two-stage and one-stage detectors according to the number of classification regression. Typical representatives of two-stage detectors are Faster R-CNN (Ren et al., 2017), Cascade R-CNN (Cai and Vasconcelos, 2018), etc., while typical representatives of one-stage detectors are RetinaNet (Lin et al., 2017b), SSD (Liu et al., 2016), etc. Generally speaking, two-stage detectors can obtain higher accuracy, but the processing speed is slower.

One-stage detectors have faster processing speed, but obtain poorer accuracy.

Anchor-based detectors require a series of predefined sliding windows before detecting targets, while brain-inspired detects important area targets directly without predefined operations. In addition, the predefined sliding windows are not suitable for the targets with large scale span in remote sensing image processing, so the anchor-free detectors have been developed and researched.

In the anchor-free detectors, they can be mainly divided into key point detectors and pixel point detectors. In this paper, we focus on key point detectors, which detect the key points of the same instance object after prediction by identifying the location of bounding box characteristics as key points. The typical representatives of key point detectors are CornerNet (Law and Deng, 2018), ExtremeNet (Zhou et al., 2019b), and CenterNet (Zhou et al., 2019a), etc. CornerNet predicts the top-left and bottom-right points of the target and determines the connection between the two points through the localization vector to complete the target detection, but when the target is irregular, the extracted information of the two points is weak. ExtremeNet predicts the center point of the target and the extreme points of the four edges of the target to complete the target detection, but the network outputs a large number of key points and requires a large number of extreme points to be matched, resulting in a slow operation. Based on the above methods, Centernet determines the target location directly by predicting the center of the target without subsequent grouping and post-processing, and the network will be described in detail later. Although anchor-based detectors dominate in target detection, the anchor-free detectors processing idea is more scientific and have great potential for development.

3. Methods

The overall architecture of our proposed algorithm is shown in Figure 1, using an anchor-free network with an encoder-decoder structure, which performs targeted feature extraction for the deep and shallow parts of the network with target width and height prediction constraint to finally obtain detection results. In the deep part of the network, a dense connection module is made from the encoder layer En3 to the decoder layer De3 to promote a more adequate deep feature fusion. In the shallow part of the network, the encoder layer En2 is processed with a visual attention module to focus on the salient features of the local area ship targets for the input SAR images and suppress the interference of the surrounding background with similar scattering characteristics. In the prediction head part of the network, the decoder layer De2 outputs heatmap, target center offset, and constrained target width and height to obtain the final detection results.

In this section, we first introduce the anchor-free network with an encoder-decoder structure used as the algorithm baseline. Next, the designed dense connection module and visual attention module are described in detail. Then we present a novel width height prediction constraint designed in the prediction head.

3.1. Anchor-free network

The proposed algorithm builds on the key point anchor-free detector, which determines the target center by key point estimation and regresses at the target center to obtain other target attributes, such as target center offset and target width and height.

The feature extraction part uses an encoder-decoder structure. In the encoder, Resnet101 is used for feature extraction, and the extracted features are En1, En2, En3, En4, En5, with scales corresponding to 1/2, 1/4, 1/8, 1/16, 1/32 of the original image, reflecting the information of SAR image from shallow to deep. The shallow features are richer in fine-grained details and highlight the boundary of the target, while the deep features are richer in semantic features and highlight the location of the target. In the decoder, the features extracted by the encoder are up-sampled three times to gradually recover the feature map resolution, and the up-sampled features are De4, De3, De2, with scales corresponding to 1/16, 1/8, 1/4 of the original image. The final network output features are not only rich in feature extraction, but also have higher resolution, which is convenient for target detection.

In the prediction head part of the network, the output heatmap, target center offset and target width and height are shown in Figure 1. Heatmap is used to locate the key points to be determined in the input image, and the peak in the heatmap is determined as the center of the target by sigmoid function processing. Since the spatial resolution of the output heatmap is 1/4 of the original image, the target center offset is used to compensate for the pixel error caused by mapping the points on the heatmap to the original image. The output target width and height is used to predict the size of the target. Compared with the anchor detector, the key point anchor-free detector directly predicts the target center to determine the target, which is more in line with the idea of brain-inspired attention mechanism.

3.2. Dense connection module

We design a dense connection module for feature extraction to promote more adequate deep feature fusion. Traditional feature extraction methods usually utilize lateral connection to combine high-level semantic feature mappings from the decoder with corresponding low-level detailed feature mappings from the encoder, which can extract effective target features but lack

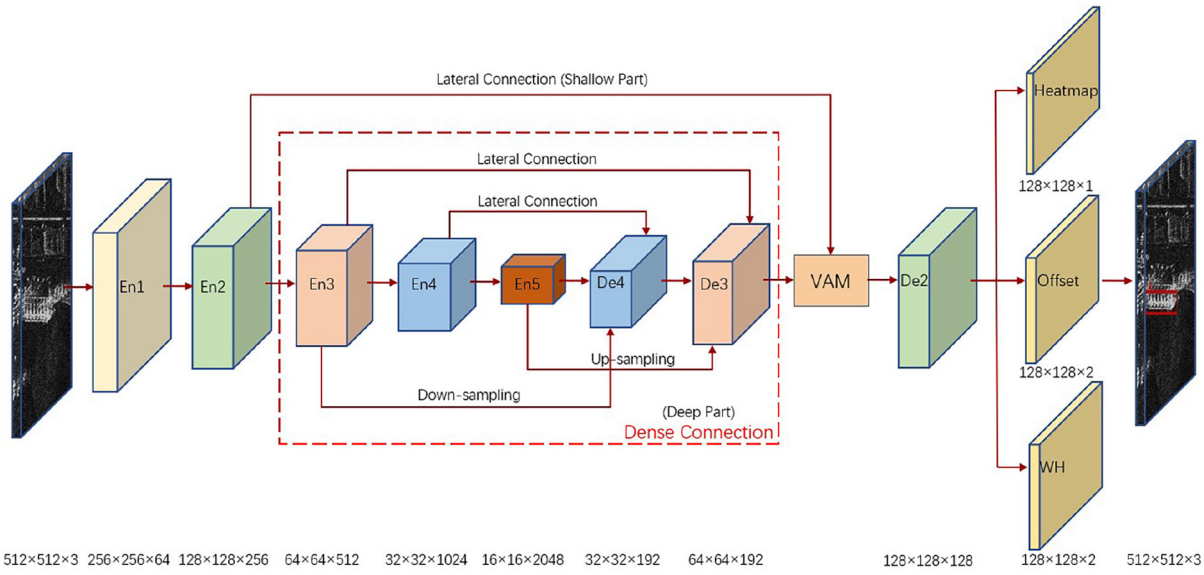


FIGURE 1
Overall architecture of the proposed algorithm. The deep part of the network is densely connected, the shallow part of the network is processed by the visual attention module (VAM), and the prediction head outputs heatmap, target center offset, and constrained target width and height.

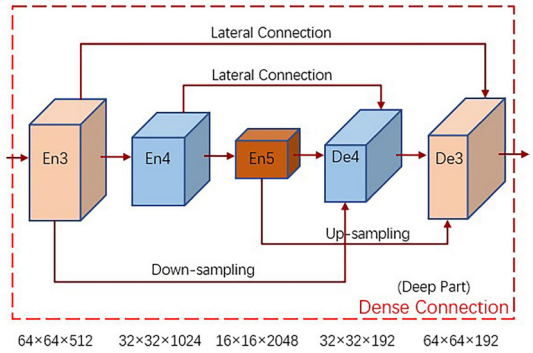


FIGURE 2
The dense connection module is used for feature extraction, which can promote the deep feature fusion more fully.

correlation between adjacent layers and feature extraction is not sufficient. For this reason, we design a dense connection module, as shown in Figure 2, with decoder feature layers from the encoder small-scale and same-scale feature mappings, and large-scale feature mappings from the decoder or encoder layer En5, to promote adequate feature fusion.

The encoder layer En1 is usually not considered in the following feature extraction, while the shallow part of the network En2 is not sufficiently extracted with still more background interference, so we only process the deep part of the network, i.e., the dense connection from the encoder layer En3 to the decoder layer De3, to promote the deep feature fusion more fully. Take how to build the decoder layer De4 as an example, its input sources are, the encoder layer En3 after down-sampling operation, the encoder layer En4 after lateral connection and the encoder layer En5 after up-sampling operation, whose feature maps have the same resolution for channel concatenation, and the number of channels of each input feature layer is 64 in order to unify the number of channels. To fuse the concatenated feature maps more fully, a fusion process is applied to them, i.e., a convolution of size 3 × 3 with 192 channels, batch normalization and ReLU activation function. The formula for constructing the decoder layer De4 is as follows:

$$De4 = FP(CONCAT(D(En3), L(En4), U(En5))) \quad (1)$$

where D(·) denotes the down-sampling operation, L(·) denotes the lateral connection, U(·) denotes the up-sampling operation, CONCAT(·) performs channel concatenation on the three processed feature maps, and FP(·) applies fusion processing on the concatenated feature maps by convolution, batch normalization with RELU activation function.

3.3. Visual attention module

We design a visual attention module to focus on local area SAR ship target salient features, suppress surrounding background interference, and finally detect the target effectively. As shown in Figure 3, encoder feature e and decoder feature d are input to the network for attention processing to obtain

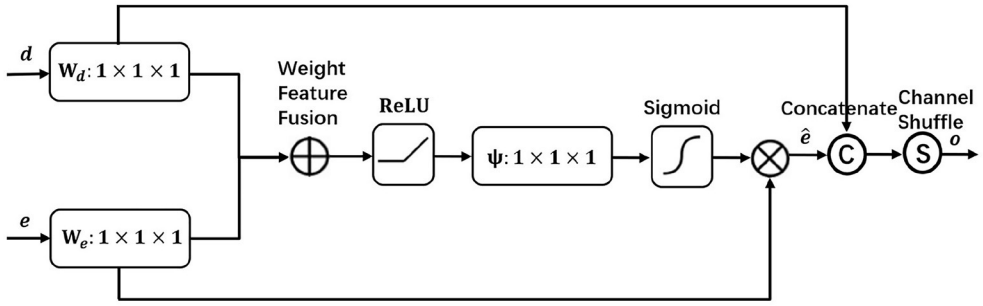


FIGURE 3
The visual attention module focuses on the salient features of the target in the local area, suppresses the surrounding background interference, and detects the object effectively.

encoder feature $\widehat{e}$ that highlights important information, and the processed encoder feature $\widehat{e}$ is then channel concatenated and shuffled with decoder feature d, thus promoting sufficient information mixing among different channels and finally obtaining feature o for target detection.

In the shallow part of the network, the encoder layer En2 is rich in fine-grained details, but it is usually ignored in the feature extraction due to insufficient feature extraction and still more background interference. To extract richer features in SAR target detection, we apply the visual attention module to the encoder layer En2 and the decoder layer De3 with the same scale after up-sampling, so as to obtain the effective feature information of the encoder layer En2 and finally achieve better detection results. In the visual attention module, the encoder layer En2 is simplified as feature e and the processed decoder layer De3 is simplified as feature d. First, they go through a 1 × 1 convolution W_e and W_d, respectively to change the channels into the same, followed by a weigh feature fusion of both, i.e., a selective element-by-element summation with differentiated fusion of different input features, and then after a Relu activation function, a 1 × 1 convolution ψ of the channel down to 1 and Sigmoid to obtain the attention coefficients. By using the attention coefficients to weight the encoder features e, the encoder features $\widehat{e}$ that highlight the effective information are obtained, and the processed encoder features $\widehat{e}$ are channel concatenated and shuffled with the decoder features d, thus promoting information mixing among different channels and finally obtaining feature o for target detection. The visual attention module is processed by the following equation:

$$WFF = Conv(\frac{\omega_1 \times I_1 + \omega_2 \times I_2}{\omega_1 + \omega_2 + \varepsilon}) \quad (2)$$

$$\widehat{e} = SIG(\Psi(RELU(WFF(e \times W_e, d \times W_d)))) \times (e \times W_e) \quad (3)$$

$$o = CS(CONCAT(\widehat{e}, d \times W_d)) \quad (4)$$

In (2), WFF represents weight feature fusion, where w is the parameter we learn to distinguish the importance of different input features I in the feature fusion process. In (3), $\widehat{e}$ represents the encoder features with salient important information, where SIG(·) denotes the sigmoid function. In (4), o represents the features processed by the visual attention module for target detection, where CS(·) denotes channel shuffle and CONCAT(·) denotes channel concatenation.

3.4. Width height prediction constraint

In predicting the width and height of the target, the scattered power distribution of the surrounding background in the near-shore scene of SAR image is relatively similar to the edge of the ship target, which is easy to have an impact on the ship target localization. So we propose a new width height prediction constraint, considering the overlapping area of the prediction box and the real box, the real difference of width and height edge and the loss gradient reweighting to improve the ship target localization accuracy. The relative position of the prediction box and the real box is shown in Figure 4. In wide and high prediction, the network only computes the positive sample loss values, so the prediction box overlaps with the true box at the center.

The overlapping area of prediction box and real box is better for ship targets with large scale differences, which can make the width and height regressions have the same contribution at different scales. The true difference of width and height edge can minimize the difference between the width and height of the prediction box and the true box to improve the detection accuracy. Loss gradient reweighting is better when focusing on high IOU targets by adaptively enhancing the weighting of the loss and gradient of high IOU objects. The width height prediction constraint loss function is as follows:

$$L_{size} = 1 - IOU^{\alpha}(A, B) + \frac{\rho^{2\alpha}(w, w_{gt})}{{w_c}^{2\alpha}} + \frac{\rho^{2\alpha}(h, h_{gt})}{{h_c}^{2\alpha}} \quad (5)$$

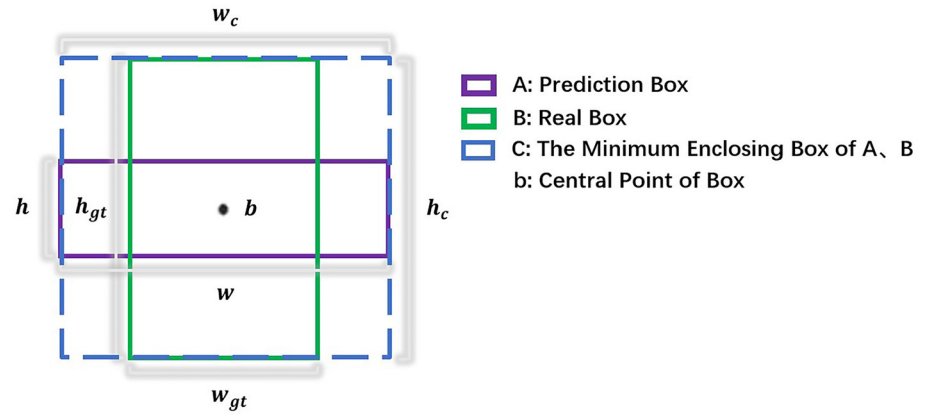


FIGURE 4
The relative position of the prediction box and the real box, both of which have the same center.

where L_{size} represents the width height prediction constraint loss value, $IOU(A, B)$ considers the overlapping area of the prediction box and the real box, $\frac{\rho^2(w,w_{gt})}{{w_c}^2} + \frac{\rho^2(h,h_{gt})}{{h_c}^2}$ considers the real difference of width and height edge, w_c and h_c denote the width and height of the minimum external box covering the prediction box and the real box, and α considers the loss gradient reweighting, by adjusting the parameter α, the detector can flexibly achieve different IOU target regression accuracy, the parameter α is taken as 3.

4. Experiments

To evaluate the performance of the proposed algorithm, we conducted experiments on the SAR ship detection dataset (SSDD) and high resolution SAR images dataset (HRSID). Firstly, the adopted dataset, experimental setup and evaluation metric are described. Then ablation experiments are performed on the algorithm to verify the effectiveness of the proposed dense connection module, visual attention module, and width height prediction constraint. Finally, it is compared with multiple target detection methods to demonstrate that the proposed algorithm can achieve better results in SAR ship target detection.

4.1. Implementations

4.1.1. Dataset

SSDD is the first publicly available dataset at home and abroad dedicated to SAR image ship target detection, which can be used for training and testing to check algorithms and is widely used. SSDD contains a total of 1,160 images, each image size is about 500 × 500, with a total of 2,456 ships, and the average number of ships per image is 2.12. The data mainly has RadarSat-2, TerraSAR-X and Sentinel-1 sensors with four polarizations of HH, HV, VV, and VH, and resolutions of 1–15 m, with ship targets in large areas of the sea and nearshore. We choose the suffix images with indexes 1 and 9 as the test set (232 images). The images with index suffix 7 are set as the validation set (116 images). The remaining images in SSDD are set as the training set (812 images). The image size is resized to 512 × 512 in our experiment.

HRSID is a high-resolution SAR ship detection dataset that includes SAR images of different resolutions, polarization, sea state, sea area, and coastal ports. The dataset is collected by Sentinel-1 and TerraSAR-X satellites and contains a total of 5,604 high-resolution SAR images and 16,951 labeled ship targets. Based on the original report in the HRSID dataset, the whole dataset is divided into training and test sets according to 13:7. The image size is 800 × 800 in our experiment.

4.1.2. Experimental setup

The proposed algorithm is implemented on pytorch 1.4.0, CUDA 10.1, and NVIDIA TITAN RTX GPU. Adam is used to optimize the target, the initial learning rate is 1.25e-4, the batch size is 16, and the feature extraction backbone is Resnet-101.

4.1.3. Evaluation metric

To evaluate the algorithm performance, we use the COCO metrics, which are AP, AP_{50}, AP_{75}, AP_s, AP_m, and AP_l. The average precision (mAP) is the area under the precision–recall curve, which reflects the average precision of multiple types of targets. mAP = AP since there is only one type of target for SAR ships. The IoU threshold is calculated every 0.05 on the interval from 0.5 to 0.95, and the final average is taken as the

final result of AP. AP_{50} is the AP at IOU = 0.5 and AP_{75} is the AP at IOU = 0.75. AP_{75} requires more stricter target localization accuracy. AP_s, AP_m and AP_l correspond to the AP of small-scale, medium-scale and large-scale targets, respectively. The precision and recall equations are as follows:

$$Precision = \frac{TP}{TP + FP} \tag{6}$$

$$Recall = \frac{TP}{TP + FN} \tag{7}$$

where TP is the number of correctly detected ships, FP is the number of false alarm ships, FN is the number of missed ships.

4.2. Ablation experiments

We performed ablation experiments on SSDD to analyze the contribution of the proposed different modules. To ensure the validity of the experimental results, all experimental settings are the same. The results are shown in Table 1, and it can be seen that the proposed different modules all significantly improve the algorithm and enhance the SAR ship target detection accuracy.

In the experimental results, by comparing the results in the second row with the fifth row, the combination of the dense connection module with the width height prediction constraint improves more in the metrics AP_s and AP_{75}, with an increase of 1.7 and 1.0%, respectively. AP_s indicates the extraction ability of small-scale ships, and AP_{75} requires high target localization accuracy, which we believe is mainly due to the consideration of the overlapping area between the prediction box and the real box in the width-height prediction, and the introduction of the loss gradient reweighting. The overlapping area makes the target width and height regressions have the same contribution at different scales, which avoids the network from focusing too much on large scale ships and ignoring the importance of small scale ships. The loss gradient reweighting improves the loss of high IOU and improves the target localization accuracy. By comparing the fifth row with the last row of results, the combination of adding the visual attention module improves more in the metric AP_m, which is 2.9% higher than before. The dense connection module acts on deep features with relatively large sensory fields, which usually correspond to the extraction of medium and large scale targets, and we believe that the visual attention module adds shallow detail information to the deep extracted features, which enriches the network features and promotes the target detection accuracy.

The detection results of the different modules proposed are shown in Figure 5. The first row (Figures 5B,C) shows the detection results without and with the dense connection module, respectively. The dense connection module can detect the missed ship target and improve the target detection accuracy. The second and third rows (Figures 5B,C) show the detection results without and with the width height prediction constraint, respectively. The results in the second row show that the width height prediction constraint can avoid the small target with false alarm and improve the small target detection accuracy. The results in the third row show that the width height prediction constraint makes the ship's tail localization more accurate and improves the target localization accuracy. The last row (Figures 5B,C) shows the detection results without and with visual attention module, respectively, and the visual attention module reduces the interference of near-shore background and improves the target detection accuracy.

4.3. Performance and analysis

In order to verify the effectiveness of this algorithm in SAR ship target detection, this algorithm is compared with multiple target detection methods. The feature extraction backbone is used Resnet101 and keeps other parameters consistent. The AP metric can reflect the overall performance of target detection, and according to Table 2, the proposed algorithm achieves 68.2% AP on SSDD, which is 3.7, 5.4, 4.2, 2.6, and 1.2% higher than Faster R-CNN, RetinaNet, FCOS, ATSS, and VFNet, respectively, which proves the effectiveness of the proposed algorithm on SAR ship target detection. In addition, the proposed algorithm has higher detection accuracy than other methods except in the metric AP_m which is lower than VFNet, and metric AP_l which is lower than Faster R-CNN. According to Table 3, the proposed algorithm achieves 62.2% AP on HRSID, and the AP, AP_{50}, AP_{75}, AP_s, AP_m, and AP_l are 5.4, 3.2, 7.9, 6.2, 1, and 0.7% higher than those evaluated on baseline, respectively, which proves the robustness of the proposed algorithm on different datasets.

Figure 6 shows the detection results of other target detection methods and the proposed algorithm. In the figure, green indicates the truth box, red indicates the Faster R-CNN detection results, yellow indicates the RetinaNet detection results, blue indicates the VFNet detection results, and purple indicates the detection results of the proposed algorithm. In Figure 6, the first row shows that the proposed algorithm has better detection results for small-scale ships, the second and third rows show that the proposed algorithm can effectively detect targets in complex near-shore scenes, and the fourth and fifth rows show that the proposed algorithm can get better detection results for densely arranged ships, while the other methods have poor detection results. The detection results we obtained show that the proposed algorithm can be better applied to small-scale targets, complex scenes, and densely arranged targets.

TABLE 1 Contribution of dense connection module, visual attention module, and width height prediction constraint to the algorithm on SSDD.

Dense connection	Visual attention	Prediction constraint	AP	AP_{50}	AP_{75}	AP_s	AP_m	AP_l
×	×	×	0.605	0.938	0.723	0.565	0.667	0.673
√	×	×	0.665	0.964	0.804	0.632	0.715	0.733
×	√	×	0.622	0.965	0.734	0.582	0.679	**0.746**
×	×	√	0.629	0.942	0.750	0.594	0.680	0.712
√	×	√	0.672	0.965	0.814	**0.649**	0.707	0.716
√	√	√	**0.682**	**0.968**	**0.817**	0.647	**0.736**	0.717

Bold values indicate that the value is the largest in the same metric.

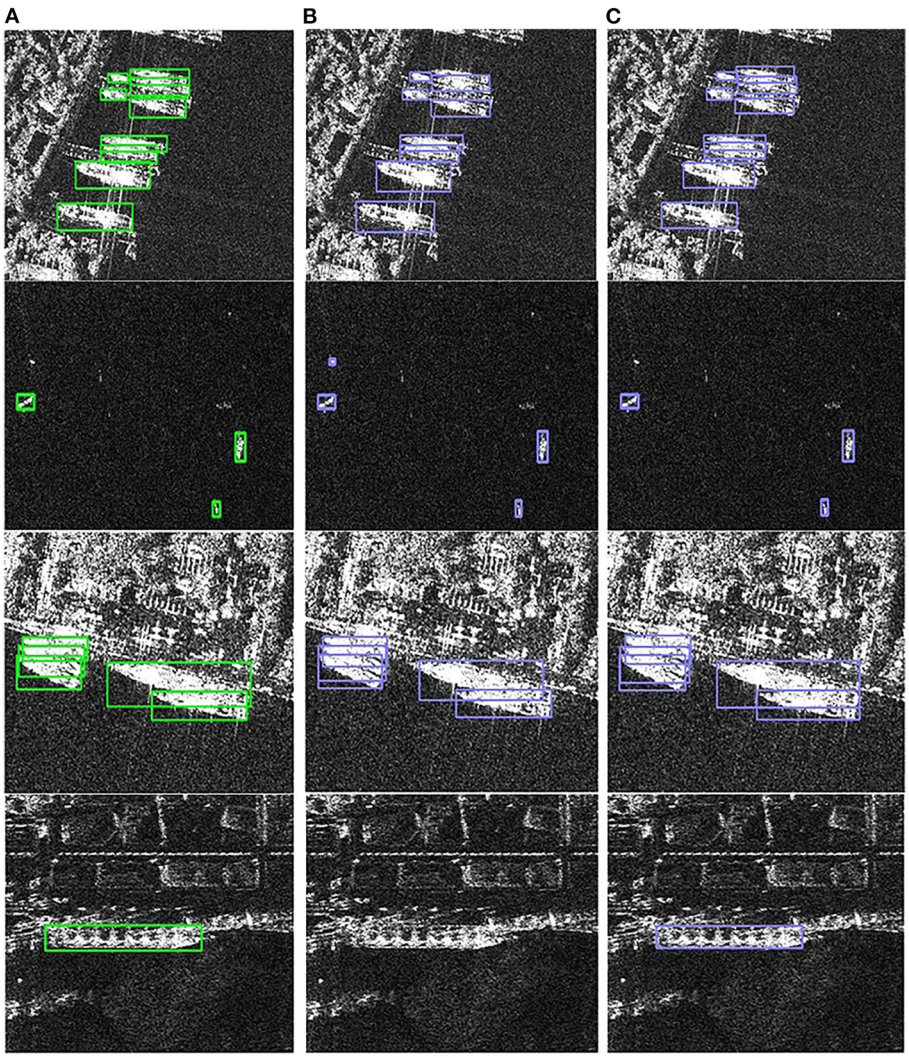


FIGURE 5
Detection results of different proposed modules. **(A)** Ground truth. **(B)** Detection results without proposed modules. **(C)** Detection results with proposed modules.

TABLE 2 Performance of other target detection methods and the proposed algorithm on SSDD.

Method	AP	AP_{50}	AP_{75}	AP_s	AP_m	AP_l
Faster R-CNN	0.645	0.925	0.774	0.592	0.724	**0.793**
RetinaNet	0.628	0.943	0.741	0.568	0.726	0.661
FCOS	0.640	0.940	0.758	0.598	0.714	0.691
ATSS	0.656	0.958	0.770	0.603	0.741	0.744
VFNet	0.670	0.965	0.802	0.622	**0.746**	0.737
Proposed	**0.682**	**0.968**	**0.817**	**0.647**	0.736	0.717

Bold values indicate that the value is the largest in the same metric.

TABLE 3 Performance of the baseline method and the proposed algorithm on HRSID.

Method	AP	AP_{50}	AP_{75}	AP_s	AP_m	AP_l
Baseline	0.568	0.866	0.619	0.567	0.679	0.347
Proposed	**0.622**	**0.898**	**0.698**	**0.629**	**0.689**	**0.354**

Bold values indicate that the value is the largest in the same metric.

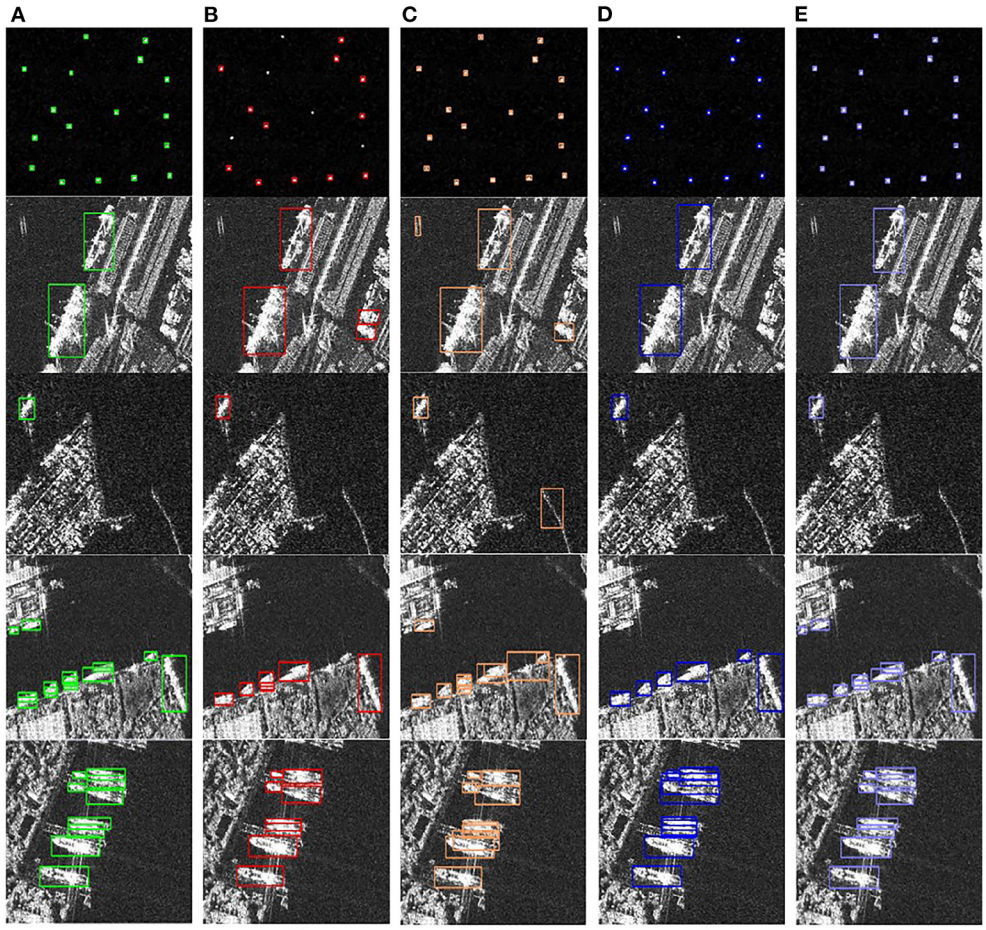


FIGURE 6
Detection results of different detection methods. **(A)** Ground truth. **(B)** Detection results of Faster R-CNN. **(C)** Detection results of RetinaNet. **(D)** Detection results of VFNet. **(E)** Detection results of the proposed algorithm.

5. Conclusion

In this article, drawing on the idea that the brain-inspired can effectively use visual targets and their surrounding background information, we propose an improved anchor-free SAR ship detection algorithm based on brain-inspired attention mechanism. The proposed algorithm improves on the anchor-free network, and in order to obtain richer target information, the deep part of the network applies a dense connection module to promote more adequate fusion of deep semantic features, and the shallow part of the network applies a visual attention module to extract features rich in fine-grained details. And in order to enable more accurate target localization in complex scenes, a novel width height prediction constraint is proposed to finally improve the target detection accuracy. After experimental validation, the proposed algorithm achieves better detection results in SAR ship target detection. In addition, there is a shortcoming during the experiment, some densely arranged ships are missed, so we will continue to improve the proposed algorithm in the future, such as considering multimodal information of ship targets, including but not limited to ship target detection under different frequency bands.

Data availability statement

The datasets analyzed for this study can be found in the online repository. SSDD data can be found here: https://github.com/TianwenZhang0825/Official-SSDD, HRSID data can be found here: https://github.com/chaozhong2010/HRSID.

Author contributions

CH and HS conceptualized the study. CH wrote the first draft of the manuscript and performed the experiments. CH, HS, and YW performed data analysis. JL and LC collected and analyzed the data. HS, YW, JL, and LC revised the manuscript. All authors contributed to the article and approved the submitted version.

Funding

This work was supported in National Natural Science Foundation of China: 62101041.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

Cai, Z., and Vasconcelos, N. (2018). "Cascade R-CNN: delving into high quality object detection," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (Salt Lake City, UT: IEEE), 6154–6162. doi: 10.1109/CVPR.2018.00644

Cui, Z., Li, Q., Cao, Z., and Liu, N. (2019). Dense attention pyramid networks for multi-scale ship detection in SAR images. *IEEE Trans. Geosci. Remote Sensing* 57, 8983–8997. doi: 10.1109/TGRS.2019.2923988

Deng, C., He, S., Han, Y., and Zhao, B. (2021). Learning dynamic spatial-temporal regularization for uav object tracking. *IEEE Signal Process. Lett.* 28, 1230–1234. doi: 10.1109/LSP.2021.3086675

Deng, C., Jing, D., Han, Y., Wang, S., and Wang, H. (2022). FAR-Net: fast anchor refining for arbitrary-oriented object detection. *IEEE Geosci. Remote Sensing Lett.* 19, 1–5. doi: 10.1109/LGRS.2022.3144513

Gao, S., Liu, J. M., Miao, Y. H., and He, Z. J. (2022). A high-effective implementation of ship detector for SAR images. *IEEE Geosci. Remote Sensing Lett.* 19, 1–5. doi: 10.1109/LGRS.2021.3115121

Han, Y., Deng, C., Zhang, Z., Li, J., and Zhao, B. (2017). "Adaptive feature representation for visual tracking," in *2017 IEEE International Conference on Image Processing (ICIP)* (Beijing: IEEE), 1867–1870. doi: 10.1109/ICIP.2017.8296605

Han, Y., Deng, C., Zhao, B., and Tao, D. (2019). State-aware anti-drift object tracking. *IEEE Trans. Image Process.* 28, 4075–4086. doi: 10.1109/TIP.2019.2905984

He, Y., He, H., and Xu, Y. (2019). "Marine multi-target detection based on improved wavelet transform," in *2019 3rd International Conference on Electronic Information Technology and Computer Engineering (EITCE)* (Xiamen: IEEE), 804–811. doi: 10.1109/EITCE47263.2019.9094990

Hinton, G. E., and Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *Science* 313, 504–507. doi: 10.1126/science.1127647

Law, H., and Deng, J. (2018). "CornerNet: detecting objects as paired keypoints," in *Computer Vision - ECCV 2018 Lecture Notes in Computer Science,* eds. V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss (Cham: Springer International Publishing), 765–781. doi: 10.1007/978-3-030-01264-9_45

Li, Z., Wu, J., Huang, Y., Sun, Z., and Yang, J. (2016). Ground-moving target imaging and velocity estimation based on mismatched compression for bistatic forward-looking SAR. *IEEE Trans. Geosci. Remote Sensing* 54, 3277–3291. doi: 10.1109/TGRS.2016.2514494

Lin, T.-Y., Dollar, P., Girshick, R., He, K., Hariharan, B., and Belongie, S. (2017a). "Feature pyramid networks for object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (Honolulu, HI: IEEE), 936–944. doi: 10.1109/CVPR.2017.106

Lin, T.-Y., Goyal, P., Girshick, R., He, K., and Dollár, P. (2017b). "Focal loss for dense object detection," in *2017 IEEE International Conference on Computer Vision (ICCV)* (Venice: IEEE), 2999–3007. doi: 10.1109/ICCV.2017.324

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., et al. (2016). "SSD: single shot multibox detector," in *Computer Vision - ECCV 2016 Lecture Notes in Computer Science*, eds. B. Leibe, J. Matas, N. Sebe, and M. Welling (Cham: Springer International Publishing), 21–37. doi: 10.1007/978-3-319-46448-0_2

Ren, S., He, K., Girshick, R., and Sun, J. (2017). Faster R-CNN: towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.* 39, 1137–1149. doi: 10.1109/TPAMI.2016.2577031

Song, S., and Yang, J. (2015). "Ship detection in polarimetric SAR images *via* tensor robust principle component analysis," in *2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)* (Milan: IEEE), 3152–3155. doi: 10.1109/IGARSS.2015.7326486

Tang, L., Tang, W., Qu, X., Han, Y., Wang, W., and Zhao, B. (2022). A scale-aware pyramid network for multi-scale object detection in SAR images. *Remote Sensing* 14:973. doi: 10.3390/rs14040973

Wang, C., Bi, F., Chen, L., and Chen, J. (2016). "A novel threshold template algorithm for ship detection in high-resolution SAR images," in *2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)* (Beijing: IEEE), 100–103. doi: 10.1109/IGARSS.2016.7729016

Wang, C., Bi, F., Zhang, W., and Chen, L. (2017). An intensity-space domain CFAR method for ship detection in HR SAR images. *IEEE Geosci. Remote Sensing Lett.* 14, 529–533. doi: 10.1109/LGRS.2017.2654450

Wang, W., Han, Y., Deng, C., and Li, Z. (2022). Hyperspectral image classification *via* deep structure dictionary learning. *Remote Sensing* 14:2266. doi: 10.3390/rs14092266

Yang, X., Zhang, X., Wang, N., and Gao, X. (2022). A robust one-stage detector for multiscale ship detection with complex background in massive SAR images. *IEEE Trans. Geosci. Remote Sensing* 60, 1–12. doi: 10.1109/TGRS.2021.3128060

Zhao, Z., Han, Y., Xu, T., Li, X., Song, H., and Luo, J. (2017). A reliable and real-time tracking method with color distribution. *Sensors* 17:2303. doi: 10.3390/s17102303

Zhou, X., Wang, D., and Krähenbühl, P. (2019a). Objects as points. *arXiv. [Preprint]*. arXiv:1904.07850. doi: 10.48550/ARXIV.1904.07850

Zhou, X., Zhuo, J., and Krähenbühl, P. (2019b). "Bottom-up object detection by grouping extreme and center points," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (Long Beach, CA: IEEE), 850–859. doi: 10.1109/CVPR.2019.00094

frontiers | Frontiers in Neuroscience

TYPE Original Research
PUBLISHED 12 December 2022
DOI 10.3389/fnins.2022.1093071

OPEN ACCESS

EDITED BY
Chenwei Deng,
Beijing Institute of Technology, China

REVIEWED BY
Shengjun Zhang,
Chongqing University, China
Shexiang Hai,
Lanzhou University of Technology, China

*CORRESPONDENCE
Ke Xiao
xiaoke@ncut.edu.cn

SPECIALTY SECTION
This article was submitted to Perception Science, a section of the journal Frontiers in Neuroscience

RECEIVED 08 November 2022
ACCEPTED 21 November 2022
PUBLISHED 12 December 2022

CITATION
Li C, Bai M, Zhang L, Xiao K, Song W and Zeng H (2022) ACLMHA and FML: A brain-inspired kinship verification framework.
Front. Neurosci. 16:1093071.
doi: 10.3389/fnins.2022.1093071

ACLMHA and FML: A brain-inspired kinship verification framework

Chen Li[1], Menghan Bai[1], Lipei Zhang[1], Ke Xiao[1]*, Wei Song[1] and Hui Zeng[2]

[1]School of Information, North China University of Technology, Beijing, China, [2]Shunde Innovation School, University of Science and Technology Beijing, Foshan, China

As an extended research direction of face recognition, kinship verification based on the face image is an interesting yet challenging task, which aims to determine whether two individuals are kin-related based on their facial images. Face image-based kinship verification benefits many applications in real life, including: missing children search, family photo classification, kinship information mining, family privacy protection, etc. Studies presented thus far provide evidence that face kinship verification still offers many challenges. Hence in this paper, we propose a novel kinship verification architecture, the main contributions of which are as follows: To boost the deep model to capture various and abundant local features from different local face regions, we propose an attention center learning guided multi-head attention mechanism to supervise the learning of attention weights and make different attention heads notice the characteristics of different regions. To combat the misclassification caused by single feature center loss, we propose a family-level multi-center loss to ensure a more proper intra/inter-class distance measurement for kinship verification. To measure the potential similarity of features among relatives better, we propose to introduce the relation comparison module to measure the similarity among features at a deeper level. Extensive experiments are conducted on the widely used kinship verification dataset—Family in the Wild (FIW) dataset. Compared with other state-of-art (SOTA) methods, encouraging results are obtained, which verify the effectiveness of our proposed method.

KEYWORDS

brain-inspired, relation comparison network, multi-head attention, facial kinship verification, deep learning

Introduction

As an extended and novel research branch of face recognition, kinship verification has received an increasing amount of attention (Hu et al., 2017; Lu et al., 2017; Wu et al., 2018; Dahan and Keller, 2020) in the recent 10 years. The purpose of kinship verification is to offer verdict whether people with different identities have kinship or not based on their facial information. Face image-based kinship verification benefits many applications in real life, including: kinship information mining (Robinson et al., 2021), missing children search (Robinson et al., 2020), family photo classification (Xia et al., 2012), family privacy protection (Kumar et al., 2020), etc. Generally, kinship can be divided

into three generations containing 11 types. The same-generation: Brother-Brother (B-B), Sister-Sister (S-S), and Brother-Sister (SIBS). The first-generation: Father-Son (F-S), Father-Daughter (F-D), Mother-Son (M-S), and Mother-Daughter (M-D). The second-generation: Grandfather-Grandson (GF-GS), Grandfather-Granddaughter (GF-GD), Grandmother-Grandson (GM-GS), and Grandmother-Granddaughter (GM-GD).

Meaningful achievement in kinship verification has been delivered. The earliest solution toward kinship verification is to construct proper handcraft features and then to calculate the similarity between features to verify the kinship of two face images. In recent years, with the development of deep learning which draws inspiration from the neurobiological mechanisms of the human brain, many data-driven kinship verification methods based on deep learning have been applied to solve the problem of face kinship verification. However, the achievements in kinship verification are relatively less inspirational compared to general face recognition or verification, due to the following challenges put forth:

1. Face datasets with family relationships are scarce. The scale of kinship verification dataset is incomparable to that of the general face recognition dataset. Therefore, data deficiency and imbalance invalidate many data-driven methods and pose great challenges for kinship verification. It is still very challenging to tackle the issue of how to boost its verification ability through limited data like the human brain.
2. Feature expressions of the latent similarity among family members are quite different compared to that of a single individual. To illustrate this issue, nine face images from Family in the Wild (FIW) dataset (Robinson et al., 2018) are shown in Figure 1, in which, faces in line A, line B, and line C belongs to three different families. And the first column are faces of fathers, the second column are faces of sons, the third column are faces of mothers. The similarities among those faces are calculated by adopting features extracted by the pretrained FaceNet. The face images in Figure 1A are those of a couple and their son. Due to gender differences, the calculated similarity between the son and his mother is lower than the similarity between him and fathers from other families, which is quite different with the human brain. Similarly, due to the differences in skin color, in group B, the faces of the son and the other father with white skin are also very similar. Therefore, feature expression for kinship verification is still very challenging.

Measurement of feature distance for kinship verification is much more complicated, compared to general face recognition. The main idea behind the face recognition problem is to reduce the intra-class distance between different samples of each individual and to expand the inter-class distance between samples of different individuals. However, deep learning-based models cannot handle the validation problem across multiple samples as well as the human brain, because there are usually gender differences and there is a large age gap lying between the relative samples, which make it very difficult to narrow down the intra-class distance with general hand-designed metric functions. Besides, a family usually contains several members with different feature representations. Simply adopting a single center for all the different family members generates an improper intra/inter-class distance for kinship verification. For instance: inter-distance between husband and wife is closer than their intra-class distance, which leads to a wrong verification of kinship.

To address these challenges in kinship verification, we propose an efficient and practical automatic kinship verification architecture inspired by the perspective of the human brain in processing visual information about relatives. The main contributions toward this article are as follows:

(1) To boost the deep model to capture various and abundant local features from different local face regions, we propose an attention center learning guided multi-head attention mechanism to supervise the learning of attention weights and make different attention heads notice the characteristics of different regions. And then, the captured local features are combined with the global feature as the final feature expression.
(2) To combat the misclassification caused by single feature center, we propose a family-level multi-center loss to ensure that the learned model can map different facial features of individuals with kinship to similar positions in the feature space.
(3) To measure the potential similarity of features among relatives better, we propose to introduce the relation comparison module to measure the relationship between features at a deeper level, instead of using a hand-designed metric function.

The rest of the article is organized as follows: In section Related work, recent influential works on kinship verification are reviewed. In section Methodology, the proposed novel methods are elaborated. In section Experimental results, extensive experiments are conducted and experimental results demonstrate that our approach achieves state-of-the-art results compared to other methods. Lastly, we summarize the main ideas and contributions of this paper.

Related work

According to the challenges discussed before, kinship verification methods based on facial features are roughly divided into local feature-based methods and metric learning-based methods.

For local feature-based methods, the key issue one needs to solve is how to abstract discriminative local features. In Zhang et al. (2015), the face image is cropped into multiple

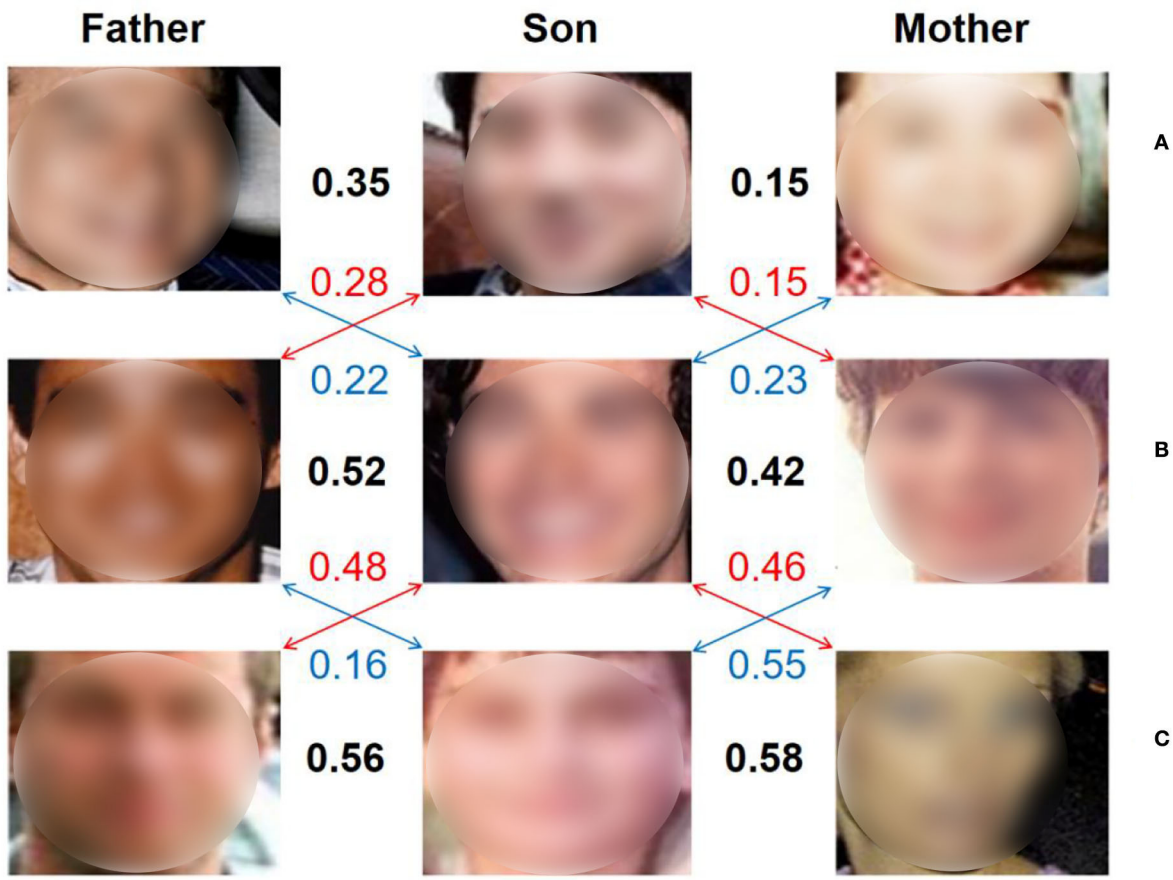


FIGURE 1
(A–C) The figure above shows the face features extracted by using the pretrained FaceNet (Schroff et al., 2015), and the similarity is calculated.

overlapping patches, and then shallow convolutional networks are used to learn the local features between relatives' face pairs for kinship verification. Dibeklioglu (2017) proposes an Age Uniform Network (AUN) to convert faces of relatives into the same age range for reducing the age features learned by the Verification Network (VFN) to be intrusive. In a similar work, a variety of artificially designed feature descriptors and deep network feature descriptors are used to extract the local and global features of the face, and the two tasks of face verification and kinship verification are combined to improve the accuracy of kinship verification (Kohli et al., 2016). In Zhang et al. (2020), adversarial loss and verification loss are added to the feature extraction process of face patches to learn the potential features of relatives' faces. Local features are used to enhance global features, which result in more effective features for kinship verification. By combining the face identification network and the face landmark prediction network, the extraction of facial appearance features and shape features is completed, and then the comparison scores of these two features are combined to finally obtain the kinship verification score (Zhang et al., 2019). In addition, in Goyal and Meenpal (2020), Dual-Tree Complex Wavelet Transform (DTCWT) is used to select a more effective patch pair for kinship verification, so as to make full use of the face patch to improve the effect of kinship verification. In Zheng et al. (2021), Residual Factorization Module is used to decompose facial features into identity and gender features, and then the adversarial training is used to reduce the negative impact of gender features on kinship verification. Indeed, local features have a positive effect on the verification of facial kinship. However, most existing local feature extraction methods rely heavily on the accuracy of facial patch crop or face landmark prediction.

To address this issue further, researchers began to introduce a non-local attention module as a complement. Visual attention is a subjective or objective mechanism of visual information selection by the brain, which concentrates on a limited amount of information and ignores other perceivable information (Cohen et al., 2012; Wang et al., 2022). It allows the human brain to be selective in processing visual input from the outside world (Yarbus, 2013), as well as enables the brain to quickly extract

different parts of interest from complex scenes and process them separately (Higgins et al., 2021). Aiming to simulate the information processing mechanism of the human brain, researchers have widely discussed the topic of attention in deep neural network. Non-local attention (Wang et al., 2018) is an attention mechanism that captures the relationship between distant pixels. After fusing the information of the global feature, an autocorrelation matrix is generated to weigh the original feature map to obtain the final attention feature. This attention model represents the importance of local regions better. Many improvements have been made on the basis of a non-local attention mechanism. DANet (Dual Attention Network) (Fu et al., 2019) increases channel attention on the basis of spatial attention in non-local attention. CCNet (Criss-Cross Network) (Huang et al., 2019) uses the cross multiplication method to reduce the computation of non-local attention. OCNet (Object Contact Network) (Yuan et al., 2018) is used to obtain the pixel-level similarity of different objects in the image to obtain the target semantic information in the image better. ABD-Net (Attentive but Diverse Network) (Chen et al., 2019) uses a non-local attention model with channel and spatial attention in person re-identification model to enhance the effectiveness of local features. It has been proven that non-local attention has a good effect in extracting the importance of the image region. NLA-FFNet (Non-local Attentional Feature Fusion Network) (Zhou et al., 2022) is proposed to enhance the robustness of feature extraction by representing the relationship between features with non-local attention through a multi-layer non-local attention mechanism. At the same time (Fu et al., 2017; Zheng et al., 2017), have shown that different channels of image features can represent specific visual patterns, and grouping them can get different regions in the image. Due to the powerful feature representation capability for Siamese models with shared weights, the Siamese networks have been used by scholars to extract global features of different images (Han et al., 2022). At present, how to boost the attention module to capture various and abundant local features from different local face regions automatically as the human brain and how to make full use of local as well as global features are worth being discussed.

Metric learning has shown a promising performance for face verification and face recognition task, which provides a positive inspiration for kinship verification. In Feng and Ma (2021), a contrastive loss function suitable for kinship comparison is proposed, and a dual-path autoencoder network is used to generate another member of the family to verify kinship pairs. In Li et al. (2017), a sampling strategy of face image triples based on family relationship information is proposed. Compared with triples based on family tags, this method is more suitable for optimization on family relationship data. At the same time, there are also studies using quadruple sampling to find more effective training pairs in kinship pairs (Zhou et al., 2019), but it is a time-consuming and labor-intensive process to construct a more suitable quadruple. In these works (Rehman et al., 2019; Nguyen et al., 2020; Yu et al., 2020), the dual-path structure network with shared or unshared weights is used to extract the face features, and various splicing methods are introduced for feature fusion, then the distance between the features is used for kinship verification. Recently, some works have started to focus on modifying the measurement method. In Wei et al. (2019), the traditional Euclidean distance or Mahalanobis distance measurement method is replaced by the adversarial learning method. In Wu et al. (2021), the framework of Mahalanobis remote metric learning is used to learn multiple distances from training data metrics. In Zhu et al. (2022), Distance and Direction based Deep Discriminant Metric Learning (D4ML) modifies and designs two loss functions to learn multiple metrics by making full use of the discriminative information contained in facial images of parents and children for minimizing the distance between relatives' faces. In conclusion, how to extract more effective shared features between kinship faces and learn the relationship between metrics is still a great challenge for metric learning-based kinship verification.

Methodology

In section Motivation, the motivation behind the proposed method is detailed. In section Proposed architecture, the overall structure of the proposed network is described. In section Attention center learning guided multi-head attention mechanism, the proposed novel attention center learning guided multi-head attention mechanism is detailed. In section Family-level multi-center loss, the introduced kinship relation compare module is illustrated, and in section Relation compare module, the novel family-level multi-center loss is elaborated.

Motivation

Multiple attention modules are introduced to extract local features under different channels in existing studies, however, the relationship between different attention modules is ignored, resulting in the inability to learn feature variability under different channels. To tackle this issue, we propose to conduct grouping at the channel level of the feature map, and then input it into a well-designed multi-headed attention module to extract local features of the face. The human brain, when processing visual information, is able to quickly focus on a few salient visual objects or multiple features, allowing for a broader range of visual information, whereas computer image processing is concerned with only a small fraction of the entire image. Therefore, to ensure the difference in various features to obtain a wider range of facial information as the human brain, we design the attention center learning module. The module is used to supervise the multi-head attention to learn diverse local features from different local regions.

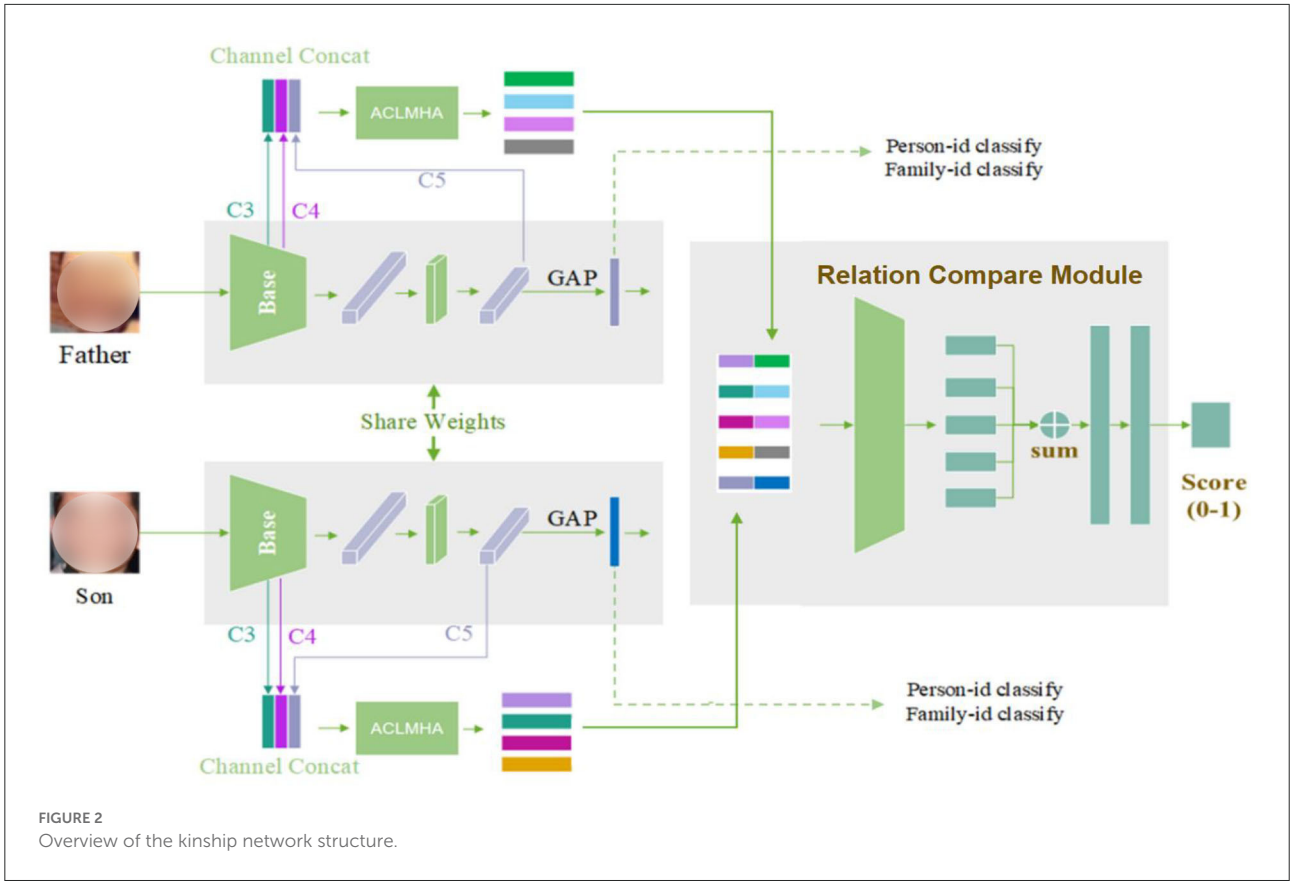


FIGURE 2
Overview of the kinship network structure.

Simply aggregating facial features of all family members to a single center generates an improper intra/inter-class distance, for example: inter-distance between husband and wife is closer than their intra-class distance, which leads to a wrong verification of kinship. The visual system of the human brain deals with the features of different individuals separately. Inspired by this reasoning, we believe that it is inappropriate to treat different members of the whole family as the same label for metric learning. Hence, we introduce a brain-inspired family-level multi-center loss so that the family feature center is not limited to one, and it is more useful to use the local area of the face to perform auxiliary metric learning.

In kinship verification, feature distance measurement is quite different and more challenging due to the existing data deficiency, gender difference, age gap, skin color, etc. Narrowing the intra-class distance with a general hand-designed metric function may not yield promising results. To handle the validation problem across multi-samples like the human brain, we propose to use a relational reasoning-based method to measure the similarity between relatives, instead of being limited to a manually set measurement function. Therefore, a non-linear relation comparable module is introduced to make the distance measurement more suitable for kinship verification.

Proposed architecture

In this section, we describe the proposed kinship verification framework. The overall structure, which consists of three main parts, is shown in Figure 2, in which the two input images are example images from FIW dataset (Robinson et al., 2018).

As shown above, the Siamese network is introduced as a feature extraction network architecture. The first part is a BaseNet module used for global feature extraction. It adopts a ResNet-50 network. To train the BaseNet better, the large-scale face dataset CASIA-WebFace (Yi et al., 2014) is introduced, and both SoftMax and center loss are applied. The second part is the attention center learning guided multi-head attention mechanism, also denoted as ACLMHA. It adopts the multi-head attention module to generate the local attention features of the face, during which, the proposed attention center learning mechanism is used to supervise the attention matrix. This helps to boost the deep model to capture various and abundant local features from different local face regions, and therefore improve the network's feature extraction ability for local areas. To capture small-scale local features better, we make full use of the feature maps (conv3_x, conv4_x, conv5_x) output by three different convolution blocks of ResNet, specifically, we perform bilinear

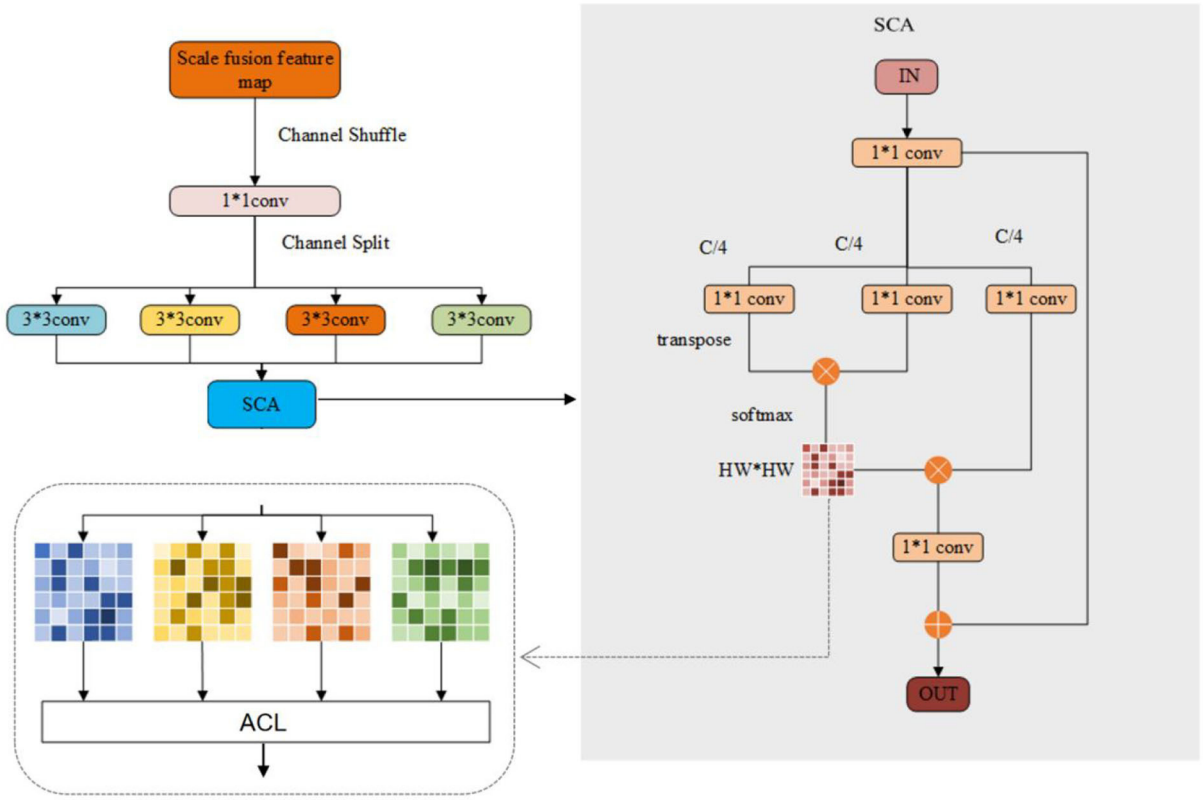


FIGURE 3
Overview of the structure of the ACLMHA (attention center learning guided multi-head attention mechanism).

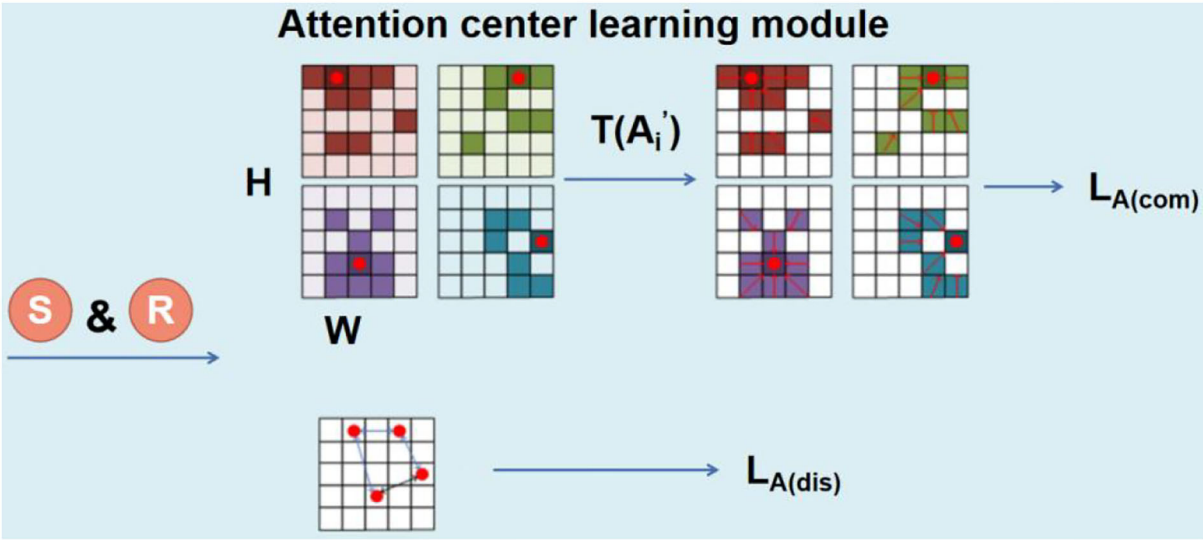


FIGURE 4
Overview of the structure of the attention center learning (ACL).

upsampling operation for C5, use 1*1 convolution for down-channel for C4, and downsample for C3. Finally, three feature maps with a size of 14*14*512 are obtained., and then we get a 14*14*1,536 feature map after splicing in the channel layer, which is input into the ACLMHA model to extract the features of the local module. Besides, the brain-inspired family-level multi-center loss is proposed to address the feature distance expression further. The third part is the relation compare module introduced to measure the complex feature distance for kinship verification. The features obtained by splicing the global and local features of different faces are fed into the module to measure their similarity, and finally the kinship between face pairs is arrived at.

Attention center learning guided multi-head attention mechanism

Attention module has been widely discussed to simulate the critical areas discovering process of the human brain. The non-local attention network expresses the importance of each pixel in the feature map through an autocorrelation matrix calculated by the correlation between the pixels of the feature map. To boost the attention module to focus on different critical regions of face images as human brain further, we propose an attention center learning mechanism to supervise the learned attention matrix. It guides the multi-head model to pay attention to different local features of the face image.

Structure of the ACLMHA

As shown in Figure 3, considering that different channels often learn different visual modes of the image, we propose to extract the features of different regions by performing channel grouping on the convolved feature maps. To be specific, the feature maps of different scales obtained by the deep convolutional network are combined, and then channel shuffling is applied to mix the channel maps of different scales, so that the information of different scales can be merged. After that, the mixed combined features are divided into k groups, and subsequently convolution operations are performed on the features through k different convolutional layers. In this paper, k is set as 4. Then, the output is fed into the spatial-channel attention (SCA) network for spatial attention learning, during which the proposed attention center learning mechanism is applied to supervise the learned attention matrix to focus on different critical regions of face images as the human brain.

The SCA network can be represented as a triple ($\boldsymbol{K}$, $\boldsymbol{Q}$, $\boldsymbol{V}$), as shown in Equation (1):

$$K^i = \theta(M^i), Q^i = \phi(M^i), V^i = \psi(M^i) \tag{1}$$

where $\boldsymbol{M} \in R^{H \times W \times C}$ is the feature map fed to the attention module, H, W, C are the width, height, and channel of the feature map separately. θ, ϕ, ψ are three different 1×1 convolution layers. We use 1×1 convolution kernel to reduce the channel number C to $\frac{C}{m}$. In this paper, $m = 4$, the feature map $\boldsymbol{K^i}$ is reshaped into $\boldsymbol{R}(\boldsymbol{K^i}) \in \mathbb{R}^{HW \times \frac{C}{m}}$ after passing through the 1*1 convolutional layer, the feature map $\boldsymbol{Q}$ is reshaped into $\boldsymbol{R}(\boldsymbol{Q^i})^{\boldsymbol{T}} \in \mathbb{R}^{\frac{C}{m} \times HW}$ after passing through the 1*1 convolutional layer and then transposed further. The autocorrelation matrix is obtained by multiplying $\boldsymbol{K^i}$ and $\boldsymbol{Q^i}$, and then the Softmax is performed row by row to get the final patch location attention matrix $\boldsymbol{A^i}$ which is represented as shown in Equation (2):

$$A^i = SoftMax(R(K^i) \bullet R(Q^i)^T) \tag{2}$$

where $\boldsymbol{R}$ is the reshape operation, $\boldsymbol{T}$ is the transpose operation, and $\bullet$ represents the matrix multiplication operation. Then $\boldsymbol{A^i}$ and $\boldsymbol{R}(\boldsymbol{V^i})$ are matrix multiplied and the residual $\boldsymbol{M'}$ is added to obtain the final local attention feature $\boldsymbol{M_a^i}$, as shown in Equation (3):

$$M_a^i = R(V^i) \bullet A^i + M' \tag{3}$$

Attention center learning module

The combination of different local features and global features of human faces has significant advantages over only global features. However, in most of the previous methods, features of different regions are extracted through facial landmark detection-based region location, which is not suitable for relatives, since the local similarities among relatives' faces are not limited to specific landmarks. Therefore, we developed an attention model that can automatically locate the salient areas between relatives, so that different attention matrices can learn different regions. We propose a feature center-based learning method to supervise the non-local attention correlation matrix.

As shown in Figure 4, the location attention matrix is the result processed by the SoftMax function row-wise. We denote

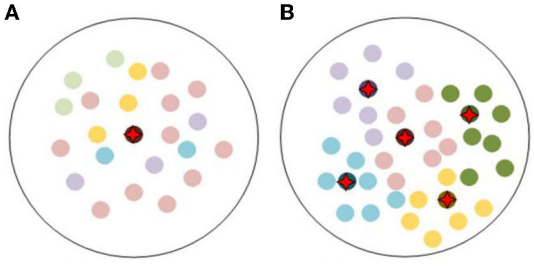


FIGURE 5
To understand the distribution of facial features in the feature space, the figure on the left shows the case of a single center **(A)**, each light color dot represents different family members' face samples, and the dark dot with the star represents the characteristics of the family id center, and the figure on the right shows the distribution of features with multiple centers **(B)**.

the operation of summing $\boldsymbol{A^i}$ by column as $\boldsymbol{S}$, and then reshape it into $\boldsymbol{A'_i} \in \mathbb{R}^{H \times W}$, as shown in Equation (4):

$$A'_i = R(S(A_l^i)) \tag{4}$$

The maximum value of A^i after threshold operation is set as the center of the attention matrix, which is expressed in Equations (5)–(7):

$$A''_i = T(A'_i) \tag{5}$$

$$T_\theta(a) = \begin{cases} a_i & if\ a_i \geq \theta \\ 0 & otherwise \end{cases} \tag{6}$$

$$CenterA_i = \max\left(A''_i\right) \tag{7}$$

where $\boldsymbol{T}$ is the threshold operation to make sure only the area with larger attention weight is kept. To make $\theta \in (0, 1)$ $\boldsymbol{A'_i}$ is normalized.

To gather attention to the center, we proposed the $\boldsymbol{L_{A(com)}}$. We introduce the reciprocal of the distance between each pixel in the matrix and the center is introduced to weight the point-to-point attention matrix value difference. So that pixels far from the center have smaller weights, and pixels closer to the center have larger weights. At the same time, for those pixels near the center point, their attention values are more close to that of the

FIGURE 6
Examples of Family in the Wild (FIW) dataset (Robinson et al., 2018).

center point. $\boldsymbol{L_{A(com)}}$ is represented as Equations (8) and (9):

$$L_{A(com)} = \sum_{x=0}^{H}\sum_{x=y}^{W}\left(\frac{1}{\left\|(x,y)-(C_x,C_y)\right\|_2^2}\bullet\left\|A''_i(C_x,C_y)-A''_i(x,y)\right\|_2^2\right) \quad (8)$$

$$and\ A''_i(x,y)\neq 0 \quad (9)$$

The purpose of $\boldsymbol{L_{A(dis)}}$ is to separate the attention centers from each other, which helps to extract the attention feature maps of different positions on the face. $\boldsymbol{L_{A(dis)}}$ is represented as Equation (10):

$$L_{A(dis)} = -\sum_{i=0}^{k}\sum_{j\neq i}^{k}\left\|(C_x^i,C_y^i),(C_x^j,C_y^j)\right\|_2^2 \quad (10)$$

Finally, the loss function $\mathbf{L_A}$ of the ALMACL part is obtained by adding the above two loss functions, as shown in Equation (11).

$$L_A = L_{A(com)}+L_{A(dis)} \quad (11)$$

Family-level multi-center loss

Different from the previous kinship verification method, in this paper we propose a family-level multi-center loss, which is a combination of the SoftMax function and the designed multi-center loss. Inspired by SoftTriple loss (Qian et al., 2019), as shown in Figure 5A, simply mapping the feature of father, mother, and child to the same feature center is improper, because, although children have latent similarities with their parents, fathers and mothers do not have such similarities. Single feature center will lead to improper intra/inter-class distance for kinship verification. To combat this issue, we design multiple feature centers for each family label, which we call family-level multi-center loss. As shown in Figure 5B, the features of different family members can be aggregated to the nearest center point by extending out multiple centers, which helps to separate the feature boundaries of different members. The family-level multi-center loss function $\boldsymbol{L_{fid-c}}$ is specified by the following Equation (12):

$$L_{fid-c} = \frac{1}{2Nm}\sum_{i=1}^{N}\sum_{k=1}^{m}\left\|\alpha_i - c_{\beta_i}^k\right\|_2^2 \quad (12)$$

where N is the number of samples in each minibatch, m is the number of each family center, and $\boldsymbol{c_{\beta_i}^k}$ is the category center, and the updated equations of the category center are as follows:

$$\frac{\partial Lfid-c}{\partial \alpha_i} = \frac{1}{Nm}\sum_{k=1}^{m}(\alpha_i - c_{\beta_i}^k) \quad (13)$$

TABLE 1 Ablation study results on family in the wild (FIW).

Method	B-B	S-S	SIBS	F-D	F-S	M-D	M-S	GF-GD	GF-GS	GM-GD	GM-GS	Avg
MTCNN + MLP	72.2	73.8	70.0	70.1	72.8	71.0	69.5	68.2	65.4	67.8	63.0	69.4
MTCNN + RCM	79.1	80.2	77.8	75.4	78.6	77.2	76.4	76.8	73.4	76.2	70.5	76.5
ACLMHA + RCM	81.3	82.1	78.6	77.3	80.5	78.4	77.4	78.5	74.1	76.4	72.2	77.5

$$\Delta C_j = \frac{\sum_{i=1}^{N} \delta_{(\beta_i = j)}(c_j - \alpha_i)}{\varepsilon + \sum_{i=1}^{N} \delta_{(\beta_i = j)}} \quad (14)$$

$$C^{'} = C_j - a\Delta C_j \quad (15)$$

where α_i denotes the sample, β_i denotes the corresponding label, ε is used to prevent the denominator from being zero when updating the calculation of categories with multiple feature centers, and $\delta()$ indicates that the value corresponding to the sample in the current training batch is 1 and the value corresponding to other samples is 0.

As shown in Figure 2, to ensure that the effective face features can be extracted, we propose a classification loss function $\boldsymbol{L_{cls}}$, which includes two parts: $\boldsymbol{L_{fid-CE}}$ and $\boldsymbol{L_{pid-CE}}$, as shown in Equations (16) and (17), and the total loss is shown in Equation (18).

$$L_{fid} = L_{fid-CE} + L_{fid-c} \quad (16)$$

$$L_{cls} = \lambda L_{fid-CE} + \beta L_{pid-CE} \quad (17)$$

$$L_{total} = L_{cls} + L_A + L_{fid} \quad (18)$$

Relation compare module

As shown in Figure 2, 2^*K groups of facial local features extracted from the input image pair are combined to generate K^*K features and then spliced with the global features. The obtained final features are fed into the perceptron layer, followed by an element-level addition operation, the output of which is used for family relationship learning. Finally, the kinship/non-kinship score of the face pair are acquired through the sigmoid activation function, as shown in Equation (19).

$$score = g(sum(f(cat(M_a^i(X_1), M_a^i(X_2), Z(X_1), Z(X_2))))) \quad (19)$$

Among them, $\boldsymbol{X_1}$ represents the input image of the child, $\boldsymbol{X_2}$ represents the input image of the parents, ***cat*** represents concatenation operation, and $\boldsymbol{Z}$ represents the mapping of the BaseNet network, which is used to extract global features. In addition, we use binary cross entropy (BCE) loss for training here. This module is similar to a learnable metric function. Through training, it can learn the feature relationships of faces among different family relationships. Therefore, it can overcome the limitations of hand-designed metric functions and learn the potential relationships between features better.

Experimental results

Datasets

The face kinship verification Family in the Wild (FIW) dataset (Robinson et al., 2018) is adopted for experiments in this paper. FIW is the largest dataset whose distribution is closest to the real data. As shown in Figure 6, the dataset contains 1,000 families and 10,676 individuals. It can be formed into 690 thousand pairs, including all the 11 kinds of kinship: B-B, S-S, SIBS, F-D, F-S, M-D, M-S; GF-GD, GF-GS, GM-GD, and GM-GS.

Training details

First, the CASIA-WebFace database is used to train the BaseNet, during which the combination of SoftMax and center loss is employed. We notice that, at the initial stage of training, if center loss is assigned with larger weight, it will lead to a very slow or difficult convergence. So, we propose to introduce a similar warm-up strategy that can dynamically adjust the weight of the center loss. Specifically, we start with a relatively small weight at the beginning of the training stage. In this paper, we set it to 0.5. After 200 thousand iterations, the weight of

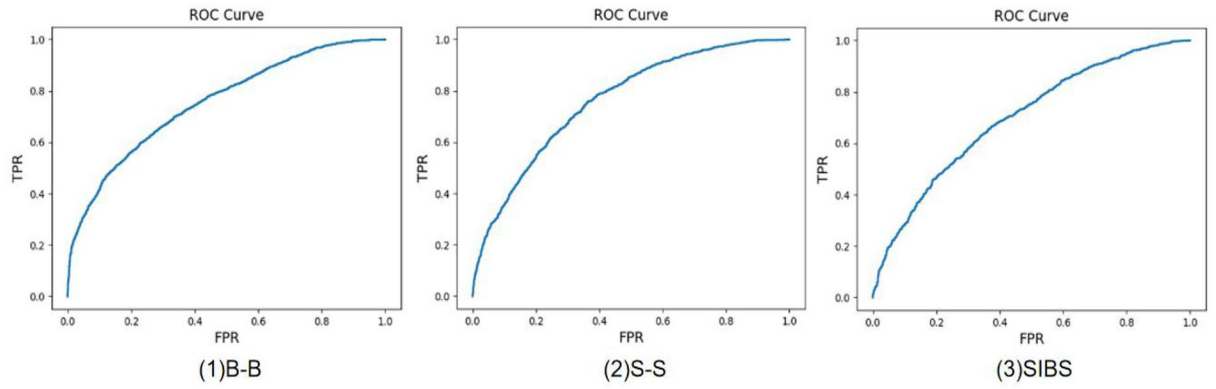


FIGURE 7
Receiver operating characteristic (ROC) curve of brother-brother (B-B), sister-sister (S-S), and brother-sister (SIBS).

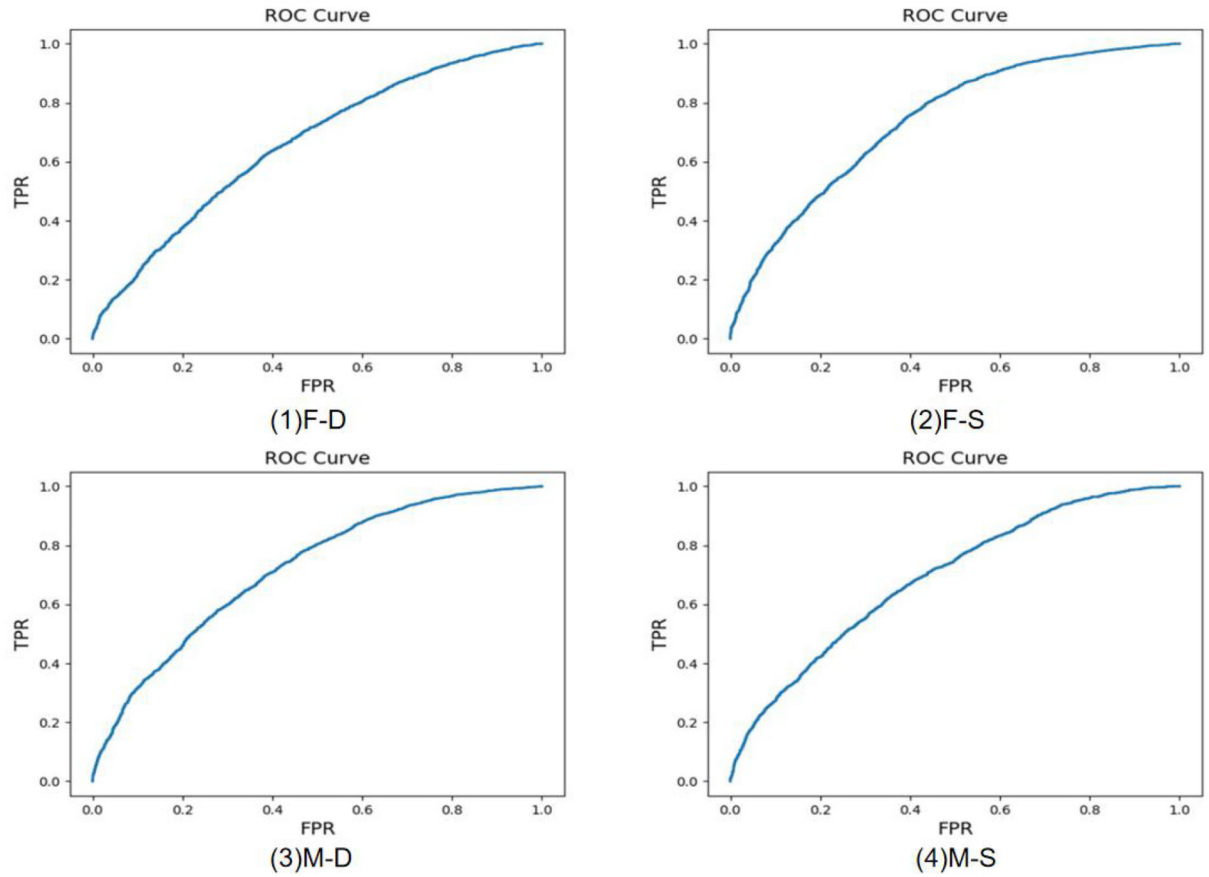


FIGURE 8
Receiver operating characteristic (ROC) curve of father-daughter (F-D), father-son (F-S), mother-daughter (M-D), and mother-son (M-S).

every five thousand iteration is 1.5 times of the original, until the iteration is completed. Second, we migrate the pretrained BaseNet on FIW. Only the last fully connected layer is left and all the parameters are frozen to learn the subsequent kinship model with a small learning rate. When the network is iterated to 200 thousand times, we unfreeze all the network layers and then fine-tune the entire network. Finally, in the verification phase, two face images are input into our model to verify their kinship.

Ablation experiments

To explore the effectiveness of our relationship model for latent feature learning among facial relatives, we design a group of comparative experiments between the multi-layer perceptron model (MLP), the relation compare model (RCM), and the relation compare module combined with ACLMHA. It should be noted that for the first two methods, the adopted features are the combination of the global and local features, which are extracted through MTCNN (Multi-task Cascaded Convolutional Networks)-based key points detection.

As shown in Table 1, the verification accuracy of RCM is increased by about 7% compared to the traditional MLP. In addition, ACLMHA combined with RCM achieves the highest result, which is a further 2% improvement. It shows that the proposed ACLMHA can enhance further the discrimination of local features compared to those. In addition, the average accuracy of each generation is 80.7, 78.4, and 75.3% separately, which shows that kinship verification of the second generation is the most challenging task.

Comparative experiments

To demonstrate further the advantages of our algorithm, the proposed algorithm is compared with other advanced algorithms published so far, and the specific comparison results are shown as follows.

Experiments of 11 different kinship verifications are conducted. Figures 7–9 show the receiver operating characteristic (ROC) curve of the proposed method on the FIW dataset. The higher the AUC (area under curve) value,

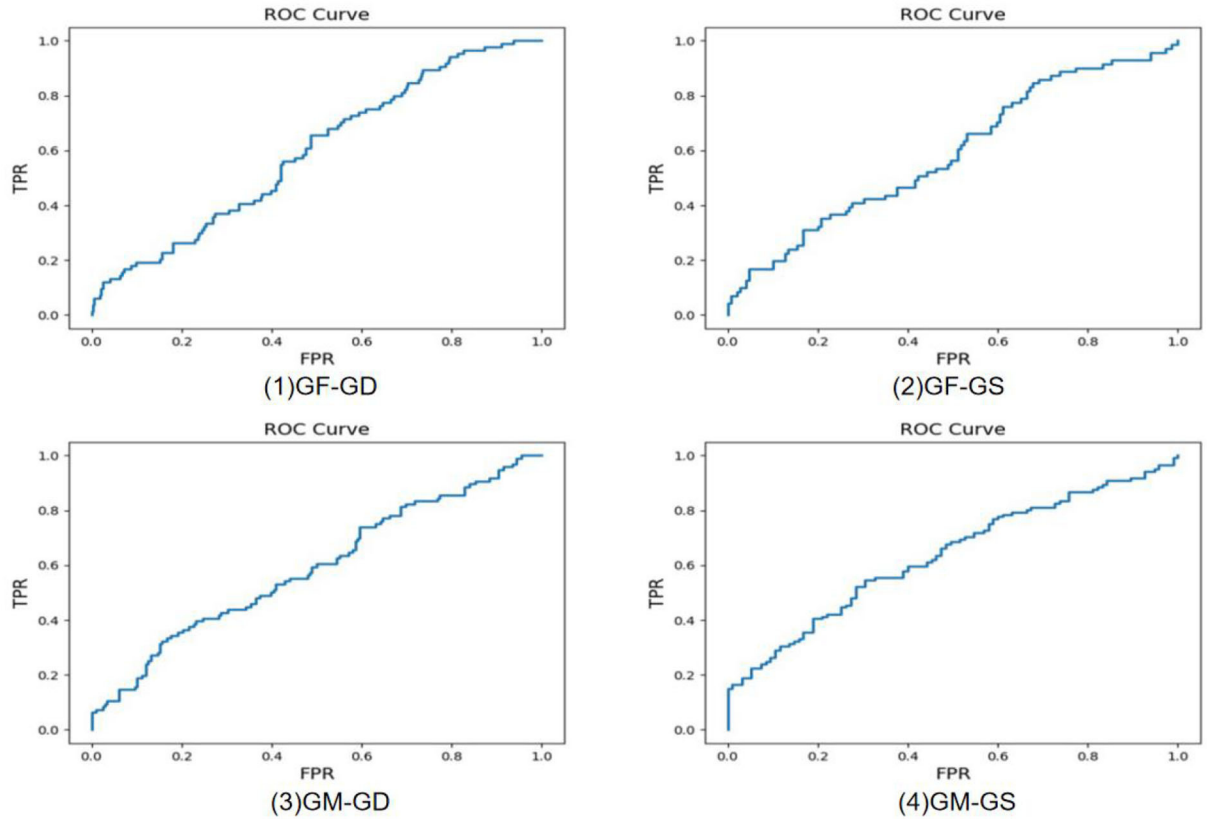


FIGURE 9
Receiver operating characteristic (ROC) curve of grandfather-granddaughter (GF-GD), grandfather-grandson (GF-GS), grandmother-granddaughter (GM-GD), and grandmother-grandson (GM-GS).

TABLE 2 Results of brother-brother (B-B), sister-sister (S-S), and brother-sister (SIBS) on family in the wild (FIW).

Method	B-B	S-S	SIBS	Avg
LBP (Ahonen et al., 2006)	55.5	57.5	55.4	56.1
SIFT (Dalal and Triggs, 2005)	57.9	59.3	56.9	58.0
VGG-face (Parkhi et al., 2015)	69.7	75.4	66.5	70.5
ResNet20 (Wen et al., 2016)	65.6	69.7	60.1	65.1
SphereFace (Liu et al., 2017)	71.9	77.3	70.2	73.1
ResNet50 (Hörmann et al., 2020)	66.4	65.3	76.0	69.2
ResNet50 + feature fusion (Yu et al., 2020)	75.1	74.4	72.0	73.8
InsightFace (Shadrikov, 2020)	80.2	80.4	77.3	79.3
Dual-VGGFace-v2 (Rachmadi et al., 2021)	66.3	73.2	67.2	68.9
AIAF + IFW (Liu et al., 2022)	73.8	85.5	77.6	78.9
Ours	81.3	82.1	78.6	80.7

the higher the prediction accuracy. It can be seen that the same-generation kinship verification achieves the best effect, followed by the first-generation kinship. The second-generation kinship verification is the most challenging task.

Tables 2–4 show the comparison results of the proposed method and the current state-of-the-art (SOTA) methods. As shown, for the proposed method, the average verification accuracy of the same-generation is 80.7%, which is 1.4% higher than the best results of other comparable algorithms. The average accuracy of the first-generation kinship is 78.4%, during which the M-D kinship verification achieves the highest result among all the mentioned methods. The average accuracy of the most challenging second-generation kinship is 75.3%, which is 3.5% higher than the best results of other

TABLE 3 Results of father-daughter (F-D), father-son (F-S), mother-daughter (M-D), and mother-son (M-S) on family in the wild (FIW).

Method	F-D	F-S	M-D	M-S	Avg
LBP (Ahonen et al., 2006)	55.1	53.8	55.7	54.7	54.8
SIFT (Dalal and Triggs, 2005)	56.4	56.2	55.1	56.5	56.1
VGG-face (Parkhi et al., 2015)	64.3	63.9	66.4	62.8	64.4
ResNet20 (Wen et al., 2016)	59.5	60.3	61.5	59.4	60.2
SphereFace (Liu et al., 2017)	69.3	68.5	71.8	69.5	69.8
ResNet50 (Hörmann et al., 2020)	76.9	80.1	76.7	78.2	78.0
ResNet50 + feature fusion (Yu et al., 2020)	75.5	81.8	74.7	75.2	76.8
InsightFace (Shadrikov, 2020)	75.2	80.8	77.7	74.4	77.0
Dual-VGGFace-v2 (Rachmadi et al., 2021)	65.3	64.1	67.3	66.3	65.8
AIAF + IFW (Liu et al., 2022)	79.1	78.2	76,1	86.5	79.9
Ours	77.3	80.5	78.4	77.4	78.4

TABLE 4 Results of grandfather-granddaughter (GF-GD), grandfather-grandson (GF-GS), grandmother-granddaughter (GM-GD), and grandmother-grandson (GM-GS) on family in the wild (fiw).

Method	GF-GD	GF-GS	GM-GD	GM-GS	Avg
LBP (Ahonen et al., 2006)	55.8	55.9	54.0	55.4	55.3
SIFT (Dalal and Triggs, 2005)	57.3	55.4	57.3	56.7	56.7
VGG-face (Parkhi et al., 2015)	62.1	63.8	57.4	61.6	61.2
ResNet20 (Wen et al., 2016)	55.4	58.1	59.7	59.7	58.2
SphereFace (Liu et al., 2017)	66.1	66.4	64.6	65.4	65.6
ResNet50 (Hörmann et al., 2020)	70.0	73.4	63.9	60.3	66.9
ResNet50 + feature fusion (Yu et al., 2020)	72.5	72.7	67.3	67.6	70.0
InsightFace (Shadrikov, 2020)	77.9	69.4	75.8	59.8	70.7
Dual-VGGFace-v2 (Rachmadi et al., 2021)	60.5	59.1	61.6	60.6	60.5
AIAF + IFW (Liu et al., 2022)	69.3	69.3	70.5	78.3	71.8
Ours	78.5	74.1	76.4	72.2	75.3

algorithms, which however achieves much higher performance compared with other optimal algorithms. The results show that the kinship verification evaluation of our method shows a greater improvement compared to the existing algorithm, especially for the most challenging second-generation kinship verification task.

Conclusions

In this paper, we propose a novel brain-inspired network with ACLMHA and FML to address the challenging feature expression, complex similarity measurement issues, and the misclassification due to single feature center in kinship verification. First, we propose an attention center learning guided multi-head attention mechanism to supervise the learning of attention weights and make different attention heads notice the characteristics of different regions to boost the deep model to capture various and abundant local features from different local face regions. Second, a family-level multi-center loss is proposed to ensure that the learned model can map different facial features of the same family to similar positions in the feature space. Finally, the feature relation compare module is introduced to measure the potential similarity of features among relatives. Extensive comparison experiments are conducted on the FIW dataset. Among them, the proposed method achieves a promising performance, especially in the verification of grandparents and grandchildren, which is significantly better than other state-of-art (SOTA) methods. The topic of how to combat data scarcity and better utilize the existing face dataset to improve the accuracy of facial kinship verification needs to be discussed in the future.

Data availability statement

Publicly available datasets were analyzed in this study. This data can be found here: http://smile-fiw.weebly.com/.

Author contributions

CL and MB: methodology, software, writing—original draft, review, and editing. LZ: methodology, software, writing—original draft, and review. KX: project administration, supervision, and conceptualization. WS: investigation, validation, and supervision. HZ: writing—review and formal analysis. All authors contributed to the article and approved the submitted version.

Funding

This work was supported by the Research Project of the Beijing Young Top-Notch Talents Cultivation Program (Grant No. CIT&TCD201904009), partially by the National Natural Science Foundation of China (Grant Nos. 62172006 and 61977001), and the Great Wall Scholar Program (CIT&TCD20190305).

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

Ahonen, T., Hadid, A., and Pietikainen, M. (2006). Face description with local binary patterns: application to face recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* 28, 2037–2041. doi: 10.1109/TPAMI.2006.244

Chen, T., Ding, S., Xie, J., Yuan, Y., Chen, W., Yang, Y., et al. (2019). "Abd-net: attentive but diverse person re-identification," in *Proceedings of the IEEE/CVF International Conference on Computer Vision* (Montreal, QC), 8351–8361.

Cohen, M. A., Cavanagh, P., Chun, M. M., and Nakayama, K. (2012). The attentional requirements of consciousness. *Trends Cogn. Sci.* 16, 411–417. doi: 10.1016/j.tics.2012.06.013

Dahan, E., and Keller, Y. (2020). A unified approach to kinship verification. *IEEE Trans. Pattern Anal. Mach. Intell.* 43, 2851–2857. doi: 10.1109/TPAMI.2020.3036993

Dalal, N., and Triggs, B. (2005). "Histograms of oriented gradients for human detection," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, Vol. 1 (San Diego, CA: IEEE), 886–893.

Dibeklioglu, H. (2017). "Visual transformation aided contrastive learning for video-based kinship verification," in *Proceedings of the IEEE International Conference on Computer Vision* (Venice), 2459–2468.

Feng, Y., and Ma, B. (2021). "Gender-based feature disentangling for kinship verification," in *2021 5th International Conference on Digital Signal Processing* (Kuala Lumpur), 320–325.

Fu, J., Liu, J., Tian, H., Li, Y., Bao, Y., Fang, Z., et al. (2019). "Dual attention network for scene segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (Long Beach, CA), 3146–3154.

Fu, J., Zheng, H., and Mei, T. (2017). "Look closer to see better: recurrent attention convolutional neural network for fine-grained image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Honolulu, HI), 4438–4446.

Goyal, A., and Meenpal, T. (2020). Patch-based dual-tree complex wavelet transform for kinship recognition. *IEEE Transact. Image Process.* 30, 191–206. doi: 10.1109/TIP.2020.3034027

Han, Y., Liu, H., Wang, Y., and Liu, C. (2022). A comprehensive review for typical applications based upon unmanned aerial vehicle platform. *IEEE J. Select. Top. Appl. Earth Observ. Remote Sens.* 15, 9654–9666. doi: 10.1109/JSTARS.2022.3216564

Higgins, I., Chang, L., Langston, V., Hassabis, D., Summerfield, C., Tsao, D., et al. (2021). Unsupervised deep learning identifies semantic disentanglement in single inferotemporal face patch neurons. *Nat. Commun.* 12, 1–14. doi: 10.1038/s41467-021-26751-5

Hörmann, S., Knoche, M., and Rigoll, G. (2020). "A multi-task comparator framework for kinship verification," in *2020 15th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2020)* (Buenos Aires: IEEE), 863–867.

Hu, J., Lu, J., Tan, Y. P., Yuan, J., and Zhou, J. (2017). Local large-margin multi-metric learning for face and kinship verification. *IEEE Transact. Circ. Syst. Video Technol.* 28, 1875–1891. doi: 10.1109/TCSVT.2017.2691801

Huang, Z., Wang, X., Huang, L., Huang, C., Wei, Y., and Liu, W. (2019). "Ccnet: criss-cross attention for semantic segmentation" in *Proceedings of the IEEE/CVF International Conference on Computer Vision* (Seoul), 603–612.

Kohli, N., Vatsa, M., Singh, R., Noore, A., and Majumdar, A. (2016). Hierarchical representation learning for kinship verification. *IEEE Transact Image Process.* 26, 289–302. doi: 10.1109/TIP.2016.2609811

Kumar, C., Ryan, R., and Shao, M. (2020). "Adversary for social good: protecting familial privacy through joint adversarial attacks," in *Proceedings of the AAAI Conference on Artificial Intelligence* (New York, NY), Vol. 34, 11304–11311.

Li, Y., Zeng, J., Zhang, J., Dai, A., Kan, M., Shan, S., et al. (2017). "Kinnet: fine-to-coarse deep metric learning for kinship verification," in *Proceedings of the 2017 Workshop on recognizing Families in the Wild* (California), 13–20.

Liu, F., Li, Z., Yang, W., and Xu, F. (2022). Age-invariant adversarial feature learning for kinship verification. *Mathematics* 10, 480. doi: 10.3390/math10030480

Liu, W., Wen, Y., Yu, Z., Li, M., Raj, B., and Song, L. (2017). "Sphereface: deep hypersphere embedding for face recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Honolulu, HI), 212–220.

Lu, J., Hu, J., and Tan, Y. P. (2017). Discriminative deep metric learning for face and kinship verification. *IEEE Transact. Image Process.* 26, 4269–4282. doi: 10.1109/TIP.2017.2717505

Nguyen, T. D. H., Nguyen, H. N. H., and Dao, H. (2020). "Recognizing families through images with pretrained encoder," in *2020 15th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2020)* (Buenos Aires: IEEE), 887–891.

Parkhi, O., Vedaldi, A., and Zisserman, A. (2015). "Deep face recognition," in *BMVC 2015 – Proceedings of the British Machine Vision Conference 2015* (British Machine Vision Association), 1–12.

Qian, Q., Shang, L., Sun, B., Hu, J., Li, H., and Jin, R. (2019). "Softtriple loss: deep metric learning without triplet sampling," in *Proceedings of the IEEE/CVF International Conference on Computer Vision* (Seoul), 6450–6458.

Rachmadi, R. F., Purnama, I. K. E., Nugroho, S. M. S., and Suprapto, Y. K. (2021). "Image-based kinship verification using dual vgg-face classifie," in *2020 IEEE International Conference on Internet of Things and Intelligence System (IoTaIS)* (Bali: IEEE), 123–128.

Rehman, A., Khalid, Z., Asghar, M. A., and Khan, M. J. (2019). "Kinship verification using deep neural network models," in *2019 International Symposium on Recent Advances in Electrical Engineering (RAEE)*. Vol. 4 (Islamabad: IEEE), 1–6.

Robinson, J. P., Shao, M., and Fu, Y. (2021). Survey on the analysis and modeling of visual kinship: A decade in the making. *IEEE Trans. Pattern Anal. Mach Intell.* 4, 4432–4453. doi: 10.1109/TPAMI.2021.3063078

Robinson, J. P., Shao, M., Wu, Y., Liu, H., Gillis, T., and Fu, Y. (2018). Visual kinship recognition of families in the wild. *IEEE Trans. Pattern Anal. Mach. Intell.* 40, 2624–2637. doi: 10.1109/TPAMI.2018.2826549

Robinson, J. P., Yin, Y., Khan, Z., Shao, M., Xia, S., Stopa, M., et al. (2020). "Recognizing families in the wild (RFIW): the 4th edition," in *2020 15th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2020)* (IEEE), 857–862.

Schroff, F., Kalenichenko, D., and Philbin, J. (2015). "Facenet: a unified embedding for face recognition and clustering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Boston, MA), 815–823.

Shadrikov, A. (2020). "Achieving better kinship recognition through better baseline," in *2020 15th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2020)* (Buenos Aires: IEEE), 872–876.

Wang, T., Hong, J., Han, Y., Zhang, G., Chen, S., Dong, T., et al. (2022). AOSVSSNet: attention-guided optical satellite video smoke segmentation network. *IEEE J. Select. Top. Appl. Earth Observ. Remote Sens.* 15, 8552–8566. doi: 10.1109/JSTARS.2022.3209541

Wang, X., Girshick, R., Gupta, A., and He, K. (2018). "Non-local neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Salt Lake City, UT), 7794–7803.

Wei, Z., Xu, M., Geng, L., Liu, H., and Yin, H. (2019). Adversarial similarity metric learning for kinship verification. *IEEE Access* 7, 100029–100035. doi: 10.1109/ACCESS.2019.2929939

Wen, Y., Zhang, K., Li, Z., and Qiao, Y. (2016). "A discriminative feature learning approach for deep face recognition," in *European Conference on Computer Vision* (Cham: Springer), 499–515.

Wu, H., Chen, J., Liu, X., and Hu, J. (2021). Component-based metric learning for fully automatic kinship verification. *J. Vis. Commun. Image Represent.* 79, 103265. doi: 10.1016/j.jvcir.2021.103265

Wu, Y., Ding, Z., Liu, H., Robinson, J., and Fu, Y. (2018). "Kinship classification through latent adaptive subspace," in *2018 13th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2018)* (Xi'an: IEEE), 143–149.

Xia, S., Shao, M., Luo, J., and Fu, Y. (2012). Understanding kin relationships in a photo. *IEEE Transact. Multimedia* 14, 1046–1056. doi: 10.1109/TMM.2012.2187436

Yarbus, A. L. (2013). *Eye Movements and Vision*. Springer.

Yi, D., Lei, Z., Liao, S., and Li, S. Z. (2014). Learning face representation from scratch. CoRR, vol. abs/1411.7923, 2014, doi: 10.48550/arXiv.1411.7923

Yu, J., Li, M., Hao, X., and Xie, G. (2020). "Deep fusion siamese network for automatic kinship verification," in *2020 15th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2020)* (Buenos Aires: IEEE), 892–899.

Yuan, Y., Huang, L., Guo, J., Zhang, C., Chen, X., and Wang, J. (2018). Ocnet: Object context network for scene parsing. *arXiv [Preprint]*. arXiv:1809.00916. doi: 10.48550/arXiv.1809.00916

Zhang, H., Wang, X., and Kuo, C. C. J. (2019). "Deep kinship verification via appearance-shape joint prediction and adaptation-based approach," in *2019 IEEE International Conference on Image Processing (ICIP)* (Taipei: IEEE), 3856–3860.

Zhang, K., Huang, Y., Song, C., Wu, H., Wang, L., and Intelligence, S. M. (2015). "Kinship verification with deep convolutional neural networks," in *British Machine Vision Conference* (Swansea: BMVA Press).

Zhang, L., Duan, Q., Zhang, D., Jia, W., and Wang, X. (2020). AdvKin: adversarial convolutional network for kinship verification. *IEEE Trans. Cybern.* 51, 5883–5896. doi: 10.1109/TCYB.2019.2959403

Zheng, H., Fu, J., Mei, T., and Luo, J. (2017). "Learning multi-attention convolutional neural network for fine-grained image recognition," in *Proceedings of the IEEE International Conference on Computer Vision* (Venice), 5209–5217.

Zheng, L., Chen, Y., Chen, X., and Zheng, F. (2021). "Age-uniform feature learning for image-based kinship verification," in *2021 5th International Conference on Computer Science and Artificial Intelligence* (Beijing), 65–71.

Zhou, C., Yang, G., Lu, Z., Liu, D., and Yang, Y. (2022). "A noise-robust feature fusion model combining non-local attention for material recognition," in *2022 the 5th International Conference on Image and Graphics Processing (ICIGP)* (Beijing), 132–138.

Zhou, X., Jin, K., Xu, M., and Guo, G. (2019). Learning deep compact similarity metric for kinship verification from face images. *Inf. Fus.* 48, 84–94. doi: 10.1016/j.inffus.2018.07.011

Zhu, X., Li, C., Chen, X., Zhang, X., and Jing, X. Y. (2022). Distance and direction based deep discriminant metric learning for kinship verification. *ACM Transact. Multimedia Comp. Commun. Appl.* 48, 1–20. doi: 10.1145/3531014

frontiers | Frontiers in Computational Neuroscience

TYPE Original Research
PUBLISHED 21 December 2022
DOI 10.3389/fncom.2022.1057439

OPEN ACCESS

EDITED BY
Yuqi Han,
Tsinghua University, China

REVIEWED BY
Yuxuan Zhao,
Institute of Automation (CAS), China
Alec Marantz,
New York University, United States

*CORRESPONDENCE
Li Su
l.su@sheffield.ac.uk

†These authors have contributed equally to this work and share first authorship

RECEIVED 29 September 2022
ACCEPTED 29 November 2022
PUBLISHED 21 December 2022

CITATION
Wingfield C, Zhang C, Devereux B, Fonteneau E, Thwaites A, Liu X, Woodland P, Marslen-Wilson W and Su L (2022) On the similarities of representations in artificial and brain neural networks for speech recognition.
Front. Comput. Neurosci. 16:1057439.
doi: 10.3389/fncom.2022.1057439

On the similarities of representations in artificial and brain neural networks for speech recognition

Cai Wingfield[1†], Chao Zhang[2†], Barry Devereux[3], Elisabeth Fonteneau[4], Andrew Thwaites[5], Xunying Liu[6], Phil Woodland[2], William Marslen-Wilson[5] and Li Su[7,8*]

[1]Department of Psychology, Lancaster University, Lancaster, United Kingdom, [2]Department of Engineering, University of Cambridge, Cambridge, United Kingdom, [3]School of Electronics, Electrical Engineering and Computer Science, Queens University Belfast, Belfast, United Kingdom, [4]Department of Psychology, University Paul Valéry Montpellier, Montpellier, France, [5]Department of Psychology, University of Cambridge, Cambridge, United Kingdom, [6]Department of Systems Engineering and Engineering Management, The Chinese University of Hong Kong, Shatin, Hong Kong SAR, China, [7]Department of Neuroscience, Neuroscience Institute, Insigneo Institute for in silico Medicine, University of Sheffield, Sheffield, United Kingdom, [8]Department of Psychiatry, University of Cambridge, Cambridge, United Kingdom

Introduction: In recent years, machines powered by deep learning have achieved near-human levels of performance in speech recognition. The fields of artificial intelligence and cognitive neuroscience have finally reached a similar level of performance, despite their huge differences in implementation, and so deep learning models can—in principle—serve as candidates for mechanistic models of the human auditory system.

Methods: Utilizing high-performance automatic speech recognition systems, and advanced non-invasive human neuroimaging technology such as magnetoencephalography and multivariate pattern-information analysis, the current study aimed to relate machine-learned representations of speech to recorded human brain representations of the same speech.

Results: In one direction, we found a quasi-hierarchical functional organization in human auditory cortex qualitatively matched with the hidden layers of deep artificial neural networks trained as part of an automatic speech recognizer. In the reverse direction, we modified the hidden layer organization of the artificial neural network based on neural activation patterns in human brains. The result was a substantial improvement in word recognition accuracy and learned speech representations.

Discussion: We have demonstrated that artificial and brain neural networks can be mutually informative in the domain of speech recognition.

KEYWORDS

automatic speech recognition, deep neural network, representational similarity analysis, auditory cortex, speech recognition

1. Introduction

Speech comprehension—the ability to accurately identify words and meaning in a continuous auditory stream—is a cornerstone of the human communicative faculty. Nonetheless, there is still limited understanding of the neurocomputational representations and processes in the human brain which underpin it. In this paper we approach a fundamental component of speech comprehension—namely the recognition of word identities from the sound of speech— in reverse: to find artificial systems which can accomplish the task, and use them to model and probe the brain's solution. In the domain of engineering, automatic speech recognition (ASR) systems are designed to identify words from recorded speech audio. In this way, ASR systems provide a computationally explicit account of how speech recognition *can* be achieved, so correspondences between the human and machine systems are of particular interest; specifically, the question of whether the learned representations in an ASR can be linked to those found in human brains. Modern advances in high-resolution neuroimaging and multivariate pattern-information analysis have made this investigation feasible.

In the present research, we took a bidirectional approach, relating machine-learned representations of speech to recorded brain representations of the same speech. First, we used the representations learned by an ASR system with deep neural network (DNN) acoustic models (Hinton et al., 2012) to probe the representations of heard speech in the brains of human participants undergoing continuous brain imaging. This provided a mechanistic model of speech recognition, and evidence of it matching responses in human auditory cortex. Then, in the opposite direction, we used the architectural patterns of neural activation we found in the brains to refine the DNN architecture and demonstrated that this improves ASR performance. This bidirectional approach was made possible by recently developed multivariate pattern analysis methods capable of comparing learned speech representations in living brain tissue and computational models.

ASR encompasses a family of computationally specified processes which perform the task of converting recorded speech sounds to the underlying word identities. Modern ASR systems employing DNN acoustic and language models now approach human levels of word recognition accuracy on specific tasks. For instance, regarding English, the word error rate (WER) of transcribing careful reading speech with no background noise can be lower than 2% (Luscher et al., 2019; Park et al., 2019), and the WER of transcribing spontaneous conversational telephone speech can be lower than 6% (Saon et al., 2017; Xiong et al., 2018).

For the present study, our ASR system was constructed based on a set of hidden Markov models (HMMs). For each, a designated context-dependent phonetic unit handled the transitions between the hidden states. A DNN model was used to provide the observation probability of a speech feature vector given each HMM state. This framework is often called a "hybrid system" in the ASR literature (Bourlard and Morgan, 1993; Hinton et al., 2012). The Hidden Markov Model Toolkit (HTK: Young et al., 2015; Zhang and Woodland, 2015a) represents a historical state-of-the-art ASR system, and is still among the most widely used. We used HTK to train the DNN-HMMs and construct the overall ASR pipeline of audio to text. A version of this model comprised a key part of the first-place winner of the multi-genre broadcast (MGB) challenge of the IEEE Automatic Speech Recognition and Understanding Workshop 2015 (Bell et al., 2015; Woodland et al., 2015). In this paper, all ASR systems were built in HTK using 200 h of training data from the MGB challenge. We designed the experimental setup carefully to use only British English speech and reduce the channel difference caused by different recording devices.

Of particular importance for the present study is the inclusion of a low-dimensional *bottleneck* layer in the DNN structure of our initial model. Each of the first five hidden layers contains 1,000 nodes, while the sixth hidden layer has just 26 nodes. Our choice to include six hidden layers in the DNN is not arbitrary. The performance of different DNN structures in the MGB challenge has previously been studied. Empirically, having a fewer hidden layers result in worse WERs, while more hidden layers result in unstable training performance due to the increased difficulty when optimizing deeper models. Similar structures were often adopted on different datasets and by different groups (e.g., Karafiát et al., 2013; Doddipatla et al., 2014; Yu et al., 2014; Liu et al., 2015). Since the layers in our DNN are feed-forward and fully connected, each node in each layer is connected only with the nodes from its immediately preceding layer, and as such the acoustic feature representations of the input speech are forced to pass through each layer in turn to derive the final output probabilities of the context-dependent phonetic units. The bottleneck layer representations are highly compressed and discriminative, and are therefore widely used as an alternative type of input features to acoustic models in ASR literature[1] (Grézl et al., 2007; Tüske et al., 2014; Woodland et al., 2015). In addition, the inclusion of this bottleneck layer greatly reduces the number of DNN parameters without significantly diminishing the accuracy of word recognition (Woodland et al., 2015), since it can prevent the model from over-fitting to the training data (Bishop, 2006). Thus, the bottleneck layer representation provides a learned, low-dimensional representation of speech which is both parsimonious and sufficient for high-performance speech recognition. This is especially interesting for the present study, given the inherently low-dimensional parameterization of speech that is given by articulatory features, which are a

1 Bottleneck layers which are trained alongside the other layers in a model have been shown to be superior to other methods of lowering dimensions, such as simple PCA (Grézl et al., 2007).

candidate characterization of responses to speech in human auditory cortex.

Recent electrocorticography (ECoG: Mesgarani et al., 2008, 2014; Chang et al., 2010; Di Liberto et al., 2015; Moses et al., 2016, 2018) and functional magnetic resonance imaging (fMRI: Arsenault and Buchsbaum, 2015; Correia et al., 2015) studies in humans show differential responses to speech sounds exhibiting different articulatory features in superior temporal speech areas. Heschl's gyrus (HG) and surrounding areas of the bilateral superior temporal cortices (STC) have also shown selective sensitivity to perceptual features of speech sounds earlier in the recognition process (Chan et al., 2014; Moerel et al., 2014; Saenz and Langers, 2014; Su et al., 2014; Thwaites et al., 2016). Building on our previous work investigating phonetic feature sensitivity in human auditory cortex (Wingfield et al., 2017), we focus our present analysis within language-related brain regions: STC and HG.

The neuroimaging data used in this study comes from electroencephalography and magnetoencephalography (EMEG) recordings of participants listening to spoken words in a magnetoencephalography (MEG) brain scanner. High-resolution magnetic resonance imaging (MRI) was acquired using a 3T MRI scanner for better source localization. As in our previous studies (Fonteneau et al., 2014; Su et al., 2014; Wingfield et al., 2017), the data (EMEG and MRI) has been combined to generate a source-space reconstruction of the electrophysiological activity which gave rise to the measurements at the electroencephalography (EEG) and MEG sensors. Using standard minimum-norm estimation (MNE) procedures guided by anatomical constraints from structural MRIs of the participants (Hämäläinen and Ilmoniemi, 1994; Gramfort et al., 2014), sources were localized to a cortical mesh at the gray-matter–white-matter boundary. Working with source-space activity allows us to retain the high temporal resolution of EMEG, while gaining access to resolved spatial pattern information. It also provides the opportunity to restrict the analysis to specific regions of interest on the cortex, where an effect of interest is most likely to be found.

Recent developments in multivariate neuroimaging pattern analysis methods have made it possible to probe the representational content of recorded brain activity patterns. Among these, representational similarity analysis (RSA: Kriegeskorte et al., 2008a) provides a flexible approach which is well-suited to complex computational models of rich stimulus sets. The fundamental principle of our RSA procedures was the computation of the similarity structures of the brain's response to experimental stimuli, and comparing the similarity structures with those derived from computational models. In a typical RSA study, this similarity structure is captured in a representational dissimilarity matrix (RDM), a square symmetric matrix whose rows and columns are indexed by the experimental stimuli, and whose entries give values for the dissimilarity of two conditions, as given by their correlation distance in the response space.

A key strength of RSA is that RDMs abstract away from the specific implementation of the DNN model or measured neural response, allowing direct comparisons between artificial and human speech recognition systems; the so-called "dissimilarity trick" (Kriegeskorte and Kievit, 2013). The comparison between RDMs computed from the ASR model and RDMs from human brains take the form of a Spearman's rank correlation ρ between the two (Nili et al., 2014).

RSA has been extended using the fMRI searchlight-mapping framework (Kriegeskorte et al., 2006; Nili et al., 2014) so that representations can be mapped through image volumes. Subsequently, searchlight RSA has been further extended into the temporal dimension afforded by EMEG data: spatiotemporal searchlight RSA (ssRSA: Su et al., 2012, 2014). Here, as in other studies using computational cognitive models (e.g., Khaligh-Razavi and Kriegeskorte, 2014; Mack et al., 2016), ssRSA facilitates the comparison to a machine representation of the stimulus space which may otherwise be incommensurable with a distributed brain response.

In the machine-to-human direction, using ssRSA and the ASR system as a reference, we found that the early layers of the DNN corresponded to early neural activation in primary auditory cortex, i.e., bilateral Heschl's gyrus, while the later layers of the DNN corresponded to late activation in higher level auditory brain regions surrounding the primary sensory cortex. This finding reveals that the neural network located within HG is likely to have a similar functional role as early layers of the DNN model, extracting basic acoustic features (though see Hamilton et al., 2021 for a recent contrasting study). The neurocomputational function of superior temporal gyrus regions is akin to later layers of the DNN, computing complex auditory features such as articulation and phonemic information.

In the reverse human-to-machine direction, using the pattern of results in the brain-image analysis, we improved the architecture of the DNN. The spatial extent of neural activation explained by the hidden-layer representations progressively reduced for higher layers, before expanding again for the bottleneck layer. This pattern, which mirrored the structure of the DNN itself, and (assuming an efficient and parsimonious processing stream in the brain) suggests that some pre-bottleneck layers might be superfluous in preparing the low-dimensional bottleneck compression. We restructured the DNN model with the bottleneck layer moved to more closely resemble the pattern of activation observed in the brain, hypothesizing that this would lead to a better transformation. With this simple, brain-inspired modification, we significantly improved the performance of the ASR system. It is notable that similar DNN structures have been developed independently elsewhere in order to optimize the low-dimensional speech feature representations from the DNN bottleneck layer. However, "reverse-engineer" human learning systems implemented in brain tissue in such a bidirectional fashion provides a

complementary approach in developing and refining DNN learning algorithms.

2. Study 1: Investigating ASR DNN representations

2.1. Materials and methods

2.1.1. Building DNN-HMM acoustic models for ASR

Here we construct a DNN which can each be included as a component in the hybrid DNN-HMM set-up of HTK. This is a widely used speech recognition set-up in both academic and industrial communities (Hinton et al., 2012), whose architecture is illustrated in Figure 1. Each network comprises an input layer, six hidden layers, and an output layer, which are all fully-connected feed-forward layers.

The DNN acoustic model was trained to classify each input frame into one of the triphone units at each time step. We used it as the acoustic model of our DNN-HMM ASR system to estimate the triphone unit likelihoods corresponding to each frame. The log-Mel filter bank (FBK) acoustic features were used throughout the paper, which were extracted with a 25 ms duration and 10 ms frame shift. The first order differentials of the FBK features were also included to extend the acoustic feature vectors. Each of these windows was transformed into a 40-dimensional FBK feature vector representing a speech frame with an offset of 10 ms. When being fed into the DNN input layer, the 40-dimensional feature vectors were augmented with their first-order time derivatives (also termed as *delta features* in the speech-recognition literature) to form an 80-dimensional vector o_t for the t-th frame. The final DNN input feature vector, x_t, was formed by stacking nine consecutive acoustic vectors around t, i.e., $x_t = \{o_{t-4}, o_{t-3}, \ldots, o_{t+4}\}$. Therefore, the DNN input layer (denoted as the FBK layer from Figure 2 to Figure 1) has 720 nodes and covers a 125 ms long input window starting at $(10 \times t - 50)$ ms and ending at $(10 \times t + 75)$ ms. Where this wider context window extended beyond the limits of the recording (i.e., at the beginning and end of the recording), boundary frames were duplicated to make up the nine consecutive frames.

Following the input layer FBK, there are five 1,000-node hidden layers (L2–L6), a 26-node "bottleneck" layer (L7), and the output layer (TRI). This network is therefore denoted as DNN-BN_7 since the bottleneck layer is the seventh layer (L7). All hidden nodes use a sigmoid activation function and the output layer uses a softmax activation function to estimate pseudo posterior probabilities for 6,027 output units. There are 6,026 such units corresponding to the tied triphone HMM states which are obtained by the decision tree clustering algorithm (Young et al., 1994). The last output unit is relevant to the non-speech HMM states. The DNN was trained on a corpus consisting of 200 h of British English speech selected from 7 weeks of TV broadcast shows by the BBC covering all genres. Using such a training set with a reasonably large amount of realistic speech samples guarantees our DNN model to be properly trained and close to the models used in real-world speech recognition applications. The DNN model was trained to classify each of the speech frames in the training set into one of the output units based on the cross-entropy loss function. All DNN-BN models were trained with the same configuration. The training was conducted using a modified NewBob learning rate scheduler (Zhang and Woodland, 2015a), with each minibatch having 800 frames, and with an initial learning rate of 2.0×10^{-3} and a momentum factor of 0.5. A layer-by-layer pre-training approach was adopted, which started by training a shallow artificial neural network with only one hidden layer for one epoch, and gradually adding in more hidden layers as the penultimate layer, one layer per epoch until the final DNN structure is achieved (Hinton et al., 2012). Afterwards the entire DNN model is jointly fine-tuned for 20 epochs. More details about the training configuration and data processing procedure can be found in (Woodland et al., 2015; Zhang and Woodland, 2015a).

When performing speech recognition at test-time, the posterior probabilities, $P(s_k \mid x_t)$, were converted to log-likelihoods to use as the observation density probabilities of the triphone HMM states. Specifically, the conversion was performed by

$$\ln p(x_t \mid s_k) = \ln P(s_k \mid x_t) + \ln p(x_t) - \ln P(s_k), \qquad (1)$$

where s_k is a DNN output for target k, and $P(s_k)$ is the frequency of frames corresponding to the units associated with target k in the frame-to-HMM-state alignments of the training set (Hinton et al., 2012).

2.1.2. Recorded speech stimuli

This study used speech stimulus recordings from Fonteneau et al. (2014), which consists of 400 English words spoken by a native British English female speaker. The set of words consists of nouns and verbs (e.g., *talk*, *claim*), some of which were past-tense inflected (e.g., *arrived*, *jumped*). We assume that the words' linguistic properties are independent of the acoustic–phonetic properties presently under investigation. We also assume that this sample of recorded speech provides a reasonable representation of naturally occurring phonetic variants of British English, with the caveat that the sampled utterances are restricted to isolated words and a single speaker.

Audio stimuli, which were originally recorded and presented to subjects with a 22.1 kHz sampling rate, were down-sampled to 16 kHz before building models, as the DNN was trained on a 16 kHz audio training set. After the DNN was first trained on the data from BBC TV programs, it was further adapted to fit the characteristics of the speaker and the recording channel

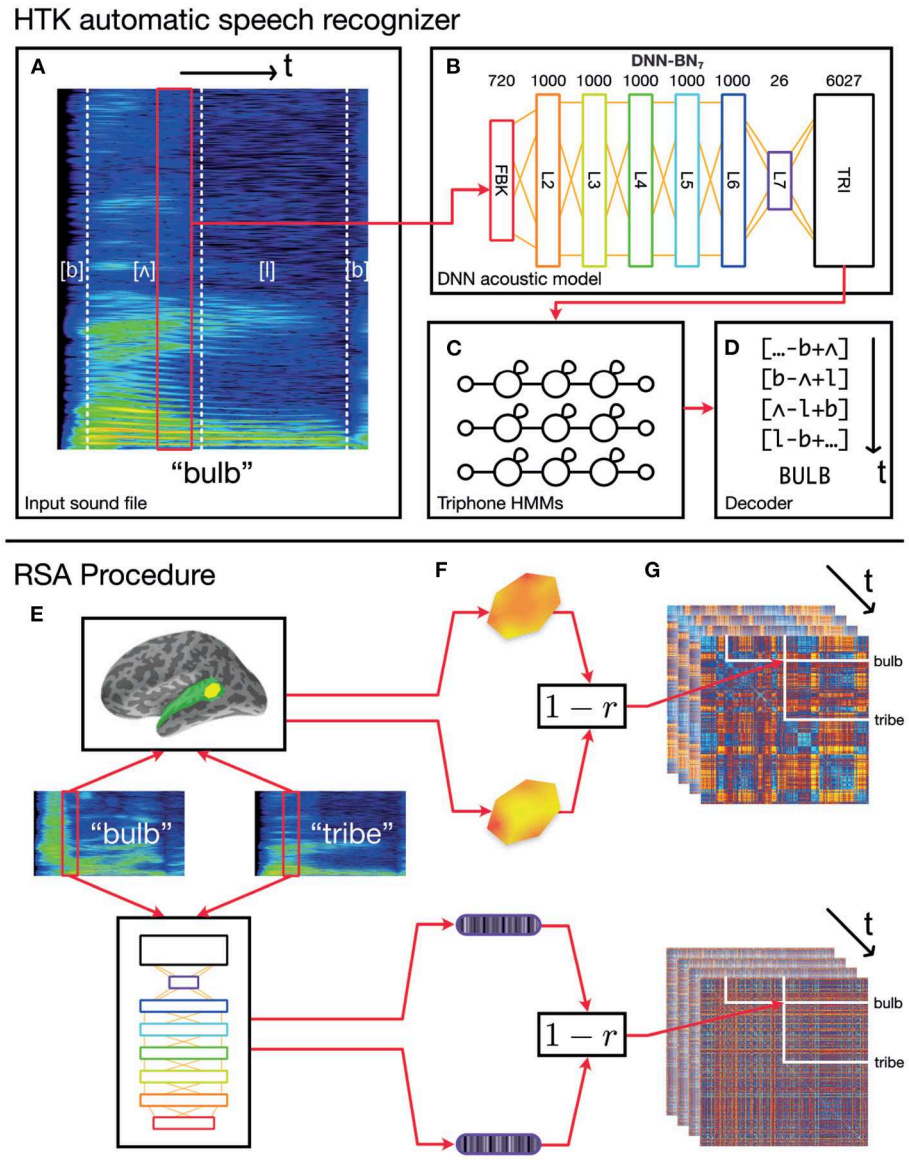


FIGURE 1
Schematic of the overall procedure. **(A–D)** Schematic representation of our automatic speech recognition system. Our ASR model is a hybrid DNN–HMM system built with HTK (Young et al., 2015; Zhang and Woodland, 2015a). **(A)** An acoustic vector is built from a window of recorded speech. **(B)** This is used as an input for a DNN acoustic model which estimates posterior probabilities of triphonetic units. Numbers above the figure indicate the size of each layer. Hidden layer L7 is the bottleneck layer for DNN-BN_7. **(C)** The triphone posteriors (TRI) are converted into log likelihoods, and used in a set of phonetic HMMs. **(D)** A decoder computes word identities from the HMM states. **(E–G)** Computing dynamic RDMs. **(E)** A pair of stimuli is presented to each subject, and the subjects' brain responses are recorded over time. The same stimuli are processed using HTK, and the hidden-layer activations recorded over time. **(F)** The spatiotemporal response pattern within a patch of each subject's cortex is compared using correlation distance. The same comparison is made between hidden-layer activation vectors. **(G)** This is repeated for each pair of stimuli, and distances entered into a pairwise comparison matrix called a representational dissimilarity matrix (RDM). As both brain response and DNN response evolve over time, additional frames of the dynamic RDM are computed.

of the stimuli data using an extra adaptation stage with 976 isolated words (see Zhang and Woodland, 2015b for details of the approach). This is to avoid any potential bias to our experimental results caused by the differences between the DNN model training set and the stimuli set, without requiring the collection of a large amount of speech samples in the same setting as the stimuli set to build a DNN model from scratch. There are no overlapping speech samples (words) between the adaptation and stimuli sets. This guarantees that the model RDM obtained using our stimuli set is not over-fitted into the

10.3389/fncom.2022.1057439

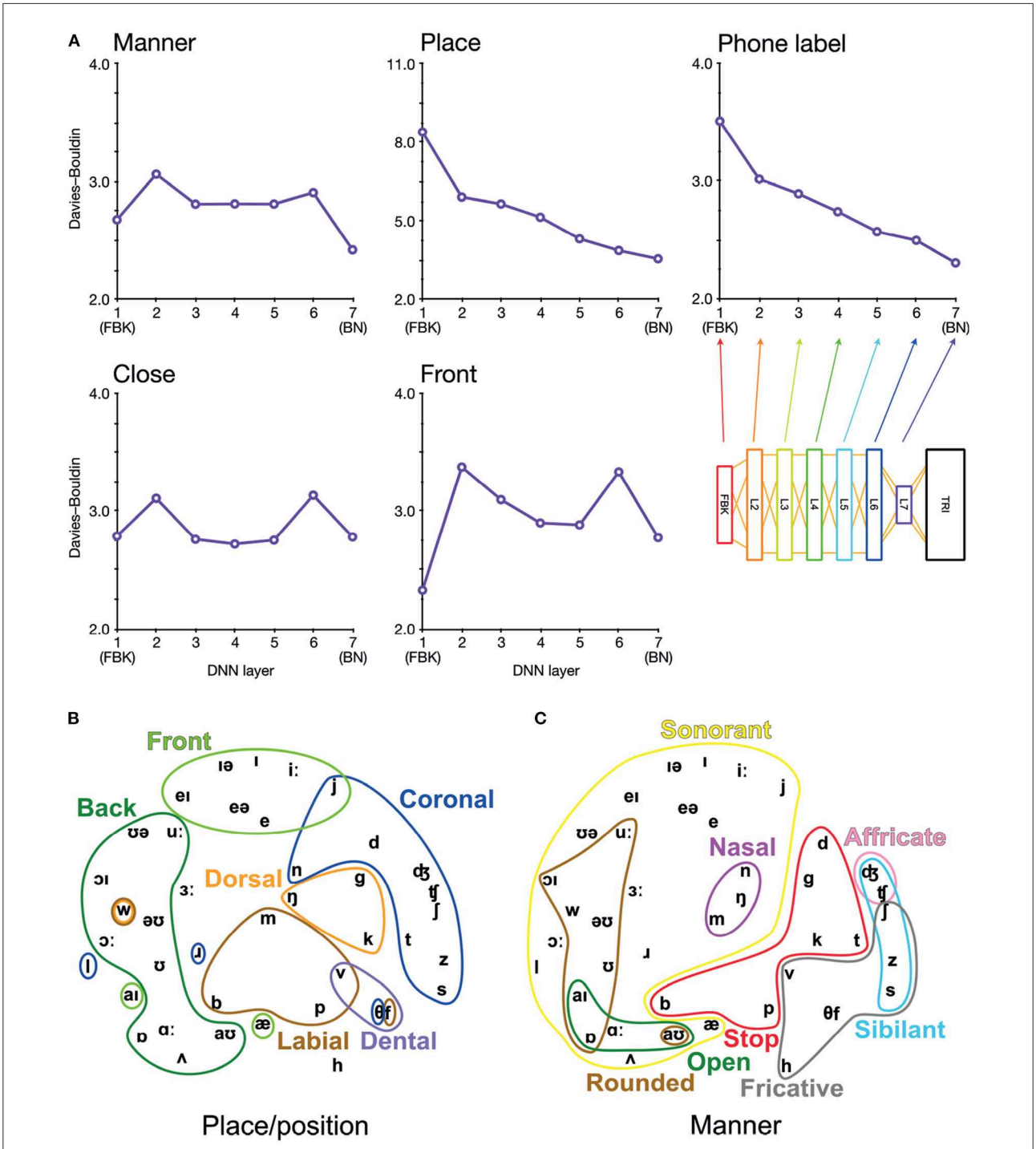


FIGURE 2
Arrangement of phonetic space represented in DNN-BN$_7$. **(A)** Davies–Bouldin clustering indices for hidden-layer representations. Each plot shows the Davies–Bouldin clustering index for the average hidden-layer representation for each phonetic segment of each stimulus. Lower values indicate better clustering. Indices were computed by labeling each segment by its phonetic label (top right panel), or by place, manner, frontness, or closeness features (other panels). **(B)** Average activation of phones for L7 Sammon non-linear multidimensional scaling (MDS) of average pattern of activation over phones, annotated with features describing place and position of articulation. **(C)** The same MDS arrangement annotated with features describing manner of articulation.

seen data, and guarantees our results and conclusions to be as general as possible.

2.1.3. Evaluating clustered representations

To investigate how the assignment of phonetic and featural labels to each segment of the stimuli could explain hidden-layer representations in DNN-BN_7, we computed Davies–Bouldin clustering indices for representational spaces at each layer.

Davies–Bouldin indices (Davies and Bouldin, 1979) are defined as the average ratio of within- and between-cluster distances for each cluster with its closest neighboring cluster. They indicate the suitability of category label assignment to clusters in high-dimensional data, with lower values indicating better suitability and with 0 the minimum possible value (obtained only if labels are shared only between identical points). This in turn serves as an indication of how suitably phonetic and feature labels might be assigned to hidden-layer representations. To compute Davies–Bouldin indices, we recorded the vector of hidden-layer activations elicited by each input time window of the stimuli for each layer in each DNN. There was a high level of correlation between many activation vectors resulting from overlapping adjacent input vectors. To minimize the effect of this, we used average vectors from each hidden layer over each contiguous phonetic segment. For example, in the word "bulb", the hidden-layer representations associated with each frame corresponding to the acoustic implementation of the first [b] were combined, and separately the representations for the final [b] were combined. Then, to each combined vector, we assigned a label under five separate labeling schemes: closeness features, frontness features, place features, manner features, and phonetic label. For place and manner features, we considered only phones which exhibited a place or manner feature (i.e., obstruents). For frontness and closeness features, we likewise considered only phones which exhibited frontness or closeness features (i.e., syllabic vowels). Where a phone had more than one appropriate feature assignment, we used the most appropriate feature. The full assignment of feature labels for phones used in the clustering analysis is given in Supplementary Figure 1.

We computed p-values for each Davies–Bouldin index calculation using a permutation procedure in which phone labels were randomized after averaging activation vectors for each segment of input (5,000 permutations). p-values were computed by randomizing the labels and recomputing Davies–Bouldin indices 5,000 times, building a distribution of Davies–Bouldin indices under the null hypothesis that phone and feature labels did not systematically explain differences in hidden-layer activations. In all cases, the observed Davies–Bouldin index was lower than the minimum value in the null distribution, yielding an estimated p-value of exactly 0.0002. Since the precision of this value is limited by the number of permutations performed, we report it as $p < 0.001$. All Davies–Bouldin index values reported were significant at the $p < 0.001$ level.

2.2. Results and discussion

Davies–Bouldin indices for each layer and categorization scheme are shown in Figure 2A. Of particular interest is the improvement of feature-based clustering in bottleneck layer L7 of DNN-BN_7, which shows that it is, in some sense, reconstructing the featural *articulatory* dimensions of the speaker. That is, though this was not included in the teaching signal, when forced to parsimoniously pass comprehension-relevant information through the bottleneck, DNN-BN_7 finds a representation of the input space which maps well onto the constraints on speech sounds inherent in the mechanics of the speaker. L7 showed the best clustering indices out of all layers for manner and place features and phone labels, and the second-best for frontness features. For closeness alone, L7 was not the best, but was still better than its adjacent layer L6. The general trend was that clustering improved for successively higher layers. Layers prior to the bottleneck tended to have larger clustering indices, indicating that their activations were not as well accounted for by phonetic or featural descriptions.

To further illustrate and visualize the representational space for L7, we used the phonetic partitioning of our stimuli provided by HTK, and averaged the activation across hidden nodes in L7 for each window of our 400 stimulus words which was eventually labeled with each phone. This gave us an average L7 response vector for each phone. We visualized this response space using the Sammon non-linear multidimensional scaling (MDS) technique in which true high-dimensional distances between points are compressed into two dimensions so as to minimize distortion (Sammon, 1969). Place/position features are highlighted in Figure 2B, and manner features are highlighted in Figure 2C.

To be clear, the presence of these feature clusters does not imply that there are individual nodes in L7 which track specific articulatory features. However, using the reasoning of RSA, we can see that articulatory features are descriptive of the overall arrangement of phones in the L7 response space. This ability to characterize and model an overall pattern ensemble in a way abstracted from the specific response format and distributed neural representations is one of the strengths of the RSA technique.

3. Study 2: Representational similarity mapping of auditory cortex with DNN representations

3.1. Materials and methods

3.1.1. Computing model RDMs from incremental machine states

To encapsulate the representational space of each of the DNN's hidden layer representations through time, we

computed model RDMs from the activation of each layer using the following procedure, illustrated in Figure 1. RSA computations were performed in Matlab using the RSA toolbox (Nili et al., 2014).

As described previously, the input layer of the DNN had access to 125 ms of audio input at each time step, to estimate the triphone-HMM-state likelihoods. Since we can only compute model RDMs where the DNN has activations for every word in the stimuli set, only the activations corresponding to the frames whose ending time is smaller than 285 ms (the duration of the shortest word) are used in our experiments Since each frame has a 25 ms duration and a 10 ms shift, only the activations of the first 27 frames of each word are reserved to construct our model RDMs (as the frame index t is required to satisfy $10 \times t + 25 \leqslant 285$).

For each fixed position of the sliding time window on each pair of our 400 stimulus words, we obtained the pattern of activation over the nodes in a particular layer of the DNN. By computing Pearson's correlation distance $(1 - r)$ between activation pattern for each pair of words, we built a 400 × 400 model RDM whose rows and columns were indexed by the stimulus words. Then, by moving the sliding time window in 10 ms increments and recomputing model RDM frames in this way, we produced a series of model RDMs which varied throughout the first 260 ms of the stimuli. We repeated this procedure for each hidden layer L2–L7, as well as the input and output layers FBK and TRI, producing in total eight series of model RDMs, or 216 individual model RDM frames. When building a model RDM frame from the input layer FBK, we used only the 40 log-mel filterbank values within the central 25 ms window (and did not include the first derivatives or overlapping context windows).

3.1.2. EMEG data collection

Sixteen right-handed native speakers of British English (six male, aged 19–35 years, self-reported normal hearing) participated in the study. For each participant, recordings of 400 English words, as spoken by a female native British English speaker were presented binaurally. Each word was repeated once. The study was approved by the Peterborough and Fenland Ethical Committee (UK). Continuous MEG data were recorded using a 306 channels VectorView system (Elektra-Neuromag, Helsinki, Finland). EEG was recorded simultaneously from 70 Ag-AgCl electrodes placed within an elastic cap (EASYCAP GmbH, Herrsching-Breitbrunn, Germany) according to the extended 10/20 system and using a nose electrode as the recording reference. All data Fonteneau et al. (2014).

3.1.3. EMEG source estimation

In order to track the cortical locations of brain–model correspondence, we estimated the location of cortical sources using the anatomically constrained MNE (Hämäläinen and Ilmoniemi, 1994) with identical parameters to those used in our previous work (Fonteneau et al., 2014; Su et al., 2014; Wingfield et al., 2017). MR structural images for each participant were obtained using a GRAPPA 3D MPRAGE sequence (TR = 2250 ms; TE = 2.99 ms; flip-angle = 9 deg; acceleration factor = 2) on a 3 T Trio (Siemens, Erlangen, Germany) with 1 mm isotropic voxels. From the MRI data, a representation of each participant's cerebral cortex was constructed using FreeSurfer software (https://surfer.nmr.mgh.harvard.edu/). The forward model was calculated with a three-layer boundary element model using the outer surface of the scalp as well as the outer and inner surfaces of the skull identified in the anatomical MRI. This combination of MRI, MEG, and EEG data provides better source localization than MEG or EEG alone (Molins et al., 2008).

The constructed cortical surface was decimated to yield approximately 12,000 vertices that were used as the locations of the dipoles. This was further restricted to the bilateral superior temporal mask as discussed previously. After applying the bilateral region of interest mask, 661 vertices remained in the left hemisphere and 613 in the right. To perform group analysis, the cortical surfaces of individual subjects were inflated and aligned using a spherical morphing technique implemented by MNE (Gramfort et al., 2014). Sensitivity to neural sources was improved by calculating a noise covariance matrix based on the 100 ms pre-stimulus period. The activations at each location of the cortical surface were estimated over 1 ms windows.

This source-reconstructed representation of the electrophysiological activity of the brain as the listeners heard the target set of 400 words was used to compute brain RDMs.

3.1.4. Computing brain RDMs in a spatiotemporal searchlight

To match the similarity structures computed from each layer of the DNN to those found in human participants, in the ssRSA procedure, RDMs were calculated from the EMEG data contained within a regular spatial searchlight patch and fixed-width sliding temporal window. We used a patch of vertices of radius 20 mm, and a 25 ms sliding window to match the 25 ms frames used in ASR. The searchlight patch was moved to center on each vertex in the masked source mesh, while the sliding window is moved throughout the epoch in fixed time-steps of 10 ms. From within each searchlight patch, we extracted the spatiotemporal response pattern from each subject's EMEG data. We computed word-by-word RDMs using Pearson's correlation distance $(1 - r)$ on the resulting response vectors. These RDMs were averaged across subjects, resulting in one brain RDM for

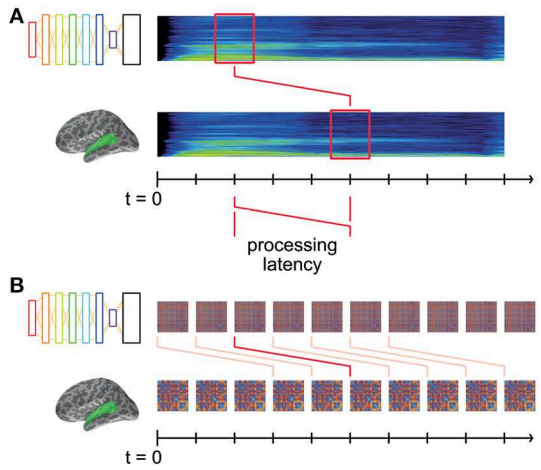


FIGURE 3
Matching model and data RDMs at systematic latencies. **(A)** Both DNN and brain representations change throughout the time-course of the stimulus, and are aligned to the start of the stimulus at $t = 0$. Some amount of time ("processing latency") elapses between the sound reaching the participants' eardrums and the elicited response in auditory cortex. Thus, the brain representations recorded at time t were elicited by the stimulus earlier in time. **(B)** For a given hypothesized processing latency, we RDMs from DNN layers and brain recordings are matched up, and an overall level of fit is computed. This modeled latency is systematically varied, the resultant level of fit thereby indicating how well the DNN's representation matches the brain's at that latency.

each within-mask vertex. Our 25 ms ssRSA sliding window moved in increments of 10 ms throughout an EMEG epoch of $[0, 540]$ ms, giving us a series of RDMs at each vertex for sliding windows $[t, t + 25]$ ms for each value of $t = 0, 10, \ldots, 510$. In total, this resulted in a total of 66,300 brain RDM frames. By using the ssRSA framework, we make this vast number of comparisons tractable by systematizing the comparison.

3.1.5. Systematic brain–model RDM comparisons

The model RDMs computed from the DNN layer activations describe the changing representational dissimilarity space of each layer throughout the duration of the stimulus words. We can think of this as a dynamic model timeline for each layer; a collection of RDMs indexed by time throughout the stimulus. Similarly, the brain data-derived RDMs computed from brain recordings describe the changing representational dissimilarity space of the brain responses at each searchlight location throughout the epoch, which we can think of as a dynamic data timeline. It takes non-zero time for vibrations at the eardrum to elicit responses in auditory cortex (Figure 3A). Therefore, it does not make sense to only compare the DNN RDM from a given time window to the precisely corresponding brain RDM for the same window of stimulus: to do so would be to hypothesize instantaneous auditory processing in auditory nerves and in the brain.

Instead, we offset the brain RDM's timeline by a fixed latency, k ms (Figure 3B). Then, matching corresponding DNN and brain RDMs at latency k tests the hypothesis that the DNN's representations explain those in auditory cortex k ms later. By systematically varying k, we are able to find the time at which the brain's representations are best explained by those in the DNN layers.

Thus, for each such potential processing latency, we obtain a spatial map describing the degree to which a DNN layer explains the brain's representations at that latency (i.e., mean Spearman's rank correlation coefficient between DNN and brain RDMs at that latency). Varying the latency then adds a temporal dimension to the maps of fit.

This process is repeated for each subject, and data combined by a t-test of the ρ values across subjects at each vertex within the mask and each latency. This resulted in one spatiotemporal t-map for each layer of the DNN. For this analysis, we used latencies ranging from 0 to 250 ms, in 10 ms increments.

3.1.6. Threshold-free cluster enhancement

We applied threshold-free cluster[2] enhancement (TFCE: Smith and Nichols, 2009) to the t-maps from each layer of the DNN. TFCE is an image-enhancement technique which enables the use of cluster-sensitive statistical methods without the requirement to make an arbitrary choice of initial cluster-forming threshold and is used as the standard statistical method by the FSL software package (Jenkinson et al., 2012).

TFCE transforms a statistical image in such a way that the value at each point becomes a weighted sum of local supporting clustered signal. Importantly, the shape of isocontours, and hence locations of local maxima, are unchanged by the TFCE transformation. For a t-map comprised of values $t_{v,k}$ for vertices v and latencies k, the TFCE transformation is given by

$$\mathrm{TFCE}\,(t_{v,k}) = \int_0^{t_{v,k}} h^2 \sqrt{e(h)}\,\mathrm{d}h \tag{2}$$

where $e(h)$ is the cluster extent of the connected component of (v, k) at threshold h. We approximated (2) with the sum

$$\sum_{i=0}^{i\Delta h \le t_{v,k} < (i+1)\Delta h} (i\Delta h)^2 \sqrt{e(i\Delta h)} \tag{3}$$

2 The term *cluster* here refers to spatiotemporally contiguous sets of datapoints in statistical maps of activation or model fit. This is a different term to *cluster* as used in the previous section to refer to sets of points located close-by in a high-dimensional abstract space. It is unfortunate that both of these concepts have the same name, but we hope their distinct meanings will be clear from the context.

where Δh was set to 0.1. The choice of Δh affects the accuracy of the approximation (3) but should not substantially bias the results.

All t-maps presented for the remainder of this paper have TFCE applied.

3.1.7. Group statistics and correction for multiple comparisons

To assess the statistical significance of the t-maps, we converted the t-values to p-values using a random-effects randomization method over subjects, under which p-values are corrected for multiple spatiotemporal comparisons (Nichols and Holmes, 2002; Smith and Nichols, 2009; Su et al., 2012). In the random-effects test, a null-distribution of t-values is simulated under the null hypothesis that Spearman's rank correlation values ρ are symmetrically distributed about 0 (i.e., no effect). By randomly flipping the sign of each individual subject's ρ-maps before computing the t-tests across subjects and applying the TFCE transformation, we simulate t-maps under the null hypothesis that experimental conditions are not differentially represented in EMEG responses. From each such simulated map, we record the map-maximum t-value, and collect these into a null distribution over all permutations. For this analysis we repeated the randomization 1,000 times, and collected separate null distributions for each hemisphere. To assess the statistical significance of a true t-value, we see in which quantile it lies in the simulated null distribution of map-maximum randomization t-values.

We performed this procedure separately for the models derived from each layer of the DNN, allowing us to obtain t-maps which could be easily thresholded at a fixed, corrected p-value.

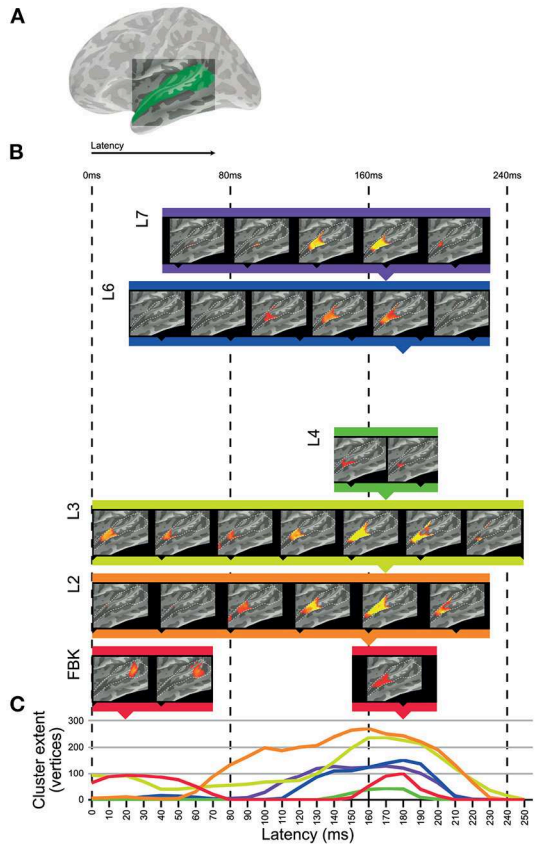


FIGURE 4
Clusters of significant fit of hidden-layer models to left-hemisphere EMEG data. **(A)** Location of region of interest mask for auditory cortex. **(B)** Maps describing fit of DNN layer models to EMEG data. Latency represents the time taken for the brain to exhibit neural representations that fit the DNN model prediction. All maps thresholded at $p < 0.01$ (corrected). **(C)** Line graphs showing the time-courses of cluster extents for each layer which showed significant fit.

3.2. Results

We used the dynamic representations from each layer of DNN-BN$_7$ to model spatiotemporal representations in the auditory cortices of human participants in an EMEG study by applying ssRSA. Areas of auditory cortex (Figure 4A) were defined using the Desikan–Killiany Atlas (STC and HG).

Figure 4 shows the left hemisphere results of this analysis. The brain maps in Figure 4B show threshold-free-cluster-enhanced t-maps (Smith and Nichols, 2009) computed from the model RDMs of each hidden layer, thresholded at $p < 0.01$. Model RDMs computed from all DNN layers except L5 showed significant fit in left STC and HG. Input layer FBK peaked early in left posterior STC at 0–70 ms, and later in left anterior STC and HG at 140–210 ms. Hidden-layer models L2–L4 and L6–L7 peaked later than FBK, achieving maximum cluster size at approximately 170 ms. Layers L5 and TRI showed no significant fit in the regions of interest. Overall, significant cluster size increased between layers FBK–L3, diminished for L4 and L5, and re-emerged for L6 and L7.

The line graphs in Figure 4C show the time-courses of each layer as they attain their maximum cluster extent. In general, there appeared to be two distinct clusters across the superior temporal region: an early cluster peaked in left posterior STC for the DNN input layer FBK, and another late cluster peaked in left anterior STC for DNN layers L1–L4 and L6–L7, throughout the whole epoch, but attaining a maximum cluster size at approx 170 ms. Details of timings for each layer are shown in Supplementary Table 1. Right hemisphere results are included in Supplementary Figure 2.

3.3. Discussion

The input layer FBK representing purely acoustic information (i.e., not a learned or task-relevant representation) showed a later and smaller effect (cluster in human posterior STC) than that of higher layers L2 and L3. The strongest cluster for FBK was early, and the later cluster appears to be a weaker version of those for higher hidden-layer models. The late cluster for FBK indicates that there is some involvement of both low-level acoustic features and higher-level phonetic information in the later neural processes at around 170 ms. However, since there is an intrinsic correlation between acoustic information and phonetic information, it is hard to completely dissociate them. Another explanation for the mixture of high and low levels of speech representations in a single brain region at the same time is the existence of feedback connections in human perceptual systems (However, the ASR systems used in this paper can achieve high degree of accuracy without the top-down feedback loop from higher to lower hidden layers.). It should be noted that while the FBK, L2 and L4 clusters all register as significant at a latency of 0 ms, timings correspond to a 25 ms window of EMEG data being matched against model state computed for the central 25 ms of 125 ms windows of audio, so only approximates the actual latency.

Moving up to hidden layers L2 and L3, we saw later clusters which fit the brain data more strongly than FBK in the left hemisphere. All hidden layers including L2 and L3 activate according to learned parameters. Progressively higher layers L4 and L5 fit with smaller clusters in human STC, with L5 showing no significant vertices at any time point ($p > 0.01$) in the left hemisphere but a very small cluster in the right hemisphere. However, the highest hidden layers L6 and L7 once again showed string fit with activations in left anterior STC.

Of particular interest is this re-emergence of fit in anterior STC to the representations in the bottleneck layer L7. In this layer of the DNN, the 1,000-node representation of L6 is substantially constrained by the reduced size of the 26-node L7. In particular, the fact that ASR accuracy is not greatly reduced by the inclusion of this bottleneck layer indicates that, for the machine solution, 26 nodes provide sufficient degrees of freedom to describe a phonetic space for purposes of word recognition. This, in conjunction with the re-emergence of fit for L7 to STC representations makes the representations of this layer of particular interest. The hidden layers in the DNN learn to sequentially transform acoustic information into phonetic probabilities in a way which generalizes across speakers and background acoustic conditions. There is no guarantee that the features the DNN learns to identify for recognition are comparable to those learned by the brain, so the fact that significant matches in the RDMs were found between machine and human solutions of the same problem is worthy of further consideration.

4. Study 3: Improving DNN design

4.1. Materials and methods

From the maximum cluster extents of the DNN layers shown in Figure 4, the activations of the DNN acoustic model significantly correspond to the activity in the left-hemisphere of human brain when listening to the same speech samples. This suggests that the DNN and human brain rely on similar mechanisms and internal representations for speech recognition.

Human speech recognition still has superior performance and robustness in comparison to even the most advanced ASR systems, so we reasoned that it could be possible to improve the DNN model structure based on the evidence recorded from the brain.

The overall minimal spatiotemporal clusters for L5 of DNN-BN_7 suggested that while early layers (L2–L3) were performing analogous transformations to early auditory cortex, and that the bottleneck (L7) was representing speech audio with a similarly parsimonious basis as left auditory cortex, there was a divergence of representation at intermediate layers (L4–L6). This indicates the possibility that the calculations in DNN layer L5 are less important for recognizing the speech accurately since brain does not appear to use such representations in the recognition process. On the other hand, although a bottleneck layer is positioned at L7, its strong correspondence to the brain reveals the importance of the calculations performed in that layer. Thus, it is natural to assume that more parameters and calculations in important layers can improve speech recognition performance, while fewer calculations can reduce the complexity of the model DNN structure without sacrificing the performance too much. With the supposition that the arrangement of auditory cortex would be adapted specifically to speech processing, we hypothesized that by moving the bottleneck layer into the positions occupied by divergent layers in DNN-BN_7, the network might learn representations that closer resemble those of human cortex, and thus improve the performance of the model.

To this end, we built and studied another DNN model, DNN-BN_5, which has the same number of parameters as DNN-BN_7 but has the bottleneck layer moved from L7 to L5. The details of the new DNN structures are shown in Figure 5C. For purposes of comparison, and following the same naming convention, we expanded our investigation with another two DNN models, DNN-BN_4 and DNN-BN_6 were also built for DNNs whose bottleneck layers are L4 and L6 respectively. In all models the number of parameters was kept to 5.0 million, matching the 4.9 million parameters of DNN-BN_7.

It may appear to the reader as if an alternative modification would be to re-locate the bottleneck layer relative to the input layer as we have done so, but attach it directly to the TRI layer (as in DNN-BN_7) without intermediate levels. However

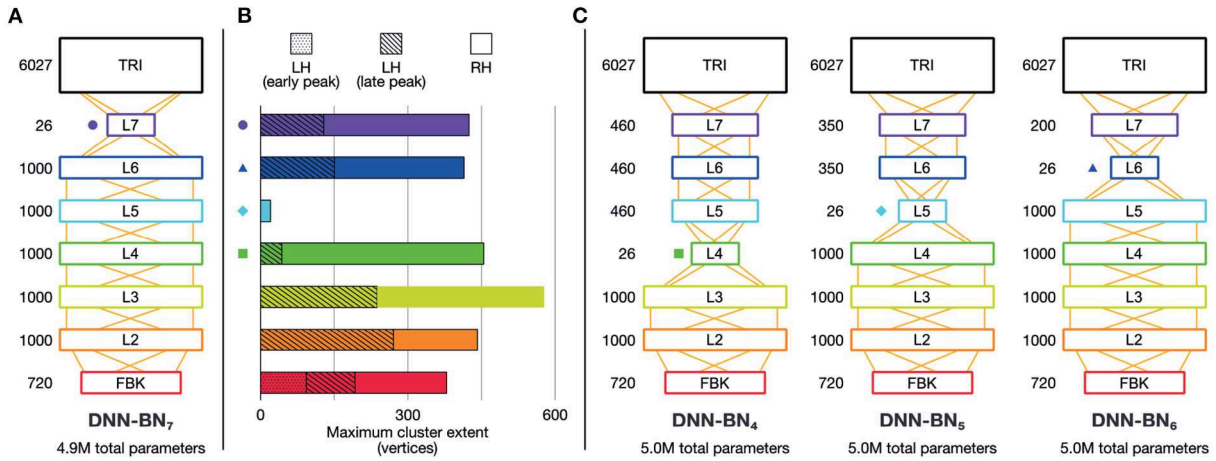


FIGURE 5
Brain-informed DNN design refinement. **(A)** Original DNN-BN_7 design. Numbers beside layers indicate number of nodes. **(B)** Maximum cluster extent indicates the degree of fit with EMEG brain representations. Where there is more than one spatiotemporally discontinuous cluster, we sum their contributions, with different segments indicated by different shading. Colored shapes on the DNN-layer axis and in other panels indicate the placement of the bottleneck layer for DNN-BN_{4-7}. **(C)** Candidates for adjusted DNN design: DNN-BN_4 (bottleneck at L4), DNN-BN_5 (bottleneck at L5) and DNN-BN_6 (bottleneck at L6).

we chose to fix the number of DNN layers and simply move the position of the bottleneck in order to keep the total number of parameters fixed at 5 million, since number of trainable parameters is a strong determiner of performance ceiling. We could have retained 5 million parameters by inflating the size of the hidden layers between the input and the bottleneck, but this would have forced upstream representations to change between models, making DNN-BN_7 harder to compare to DNN-BN_{4-6}. Additionally, early DNN studies demonstrated that, for a fixed number of parameters, deeper, thinner models (i.e., those with more layers containing fewer units) performed significantly better than shallower, wider models, and this is now a standard practice (Morgan, 2011; Hinton et al., 2012). Alternative DNN design choices may have different effects, and we hope to investigate this in future work.

We tested the derived DNN models with different bottleneck layer positions using two tasks: general large-vocabulary continuous speech recognition with recordings from BBC TV programs, and in-domain isolated-word recognition using the stimuli set. The MGB Dev set was derived as a subset of the official development set of the MGB speech recognition challenge (Bell et al., 2015), which includes 5.5 h of speech. Since the MGB testing set involves sufficient samples (8,713 utterances and 1.98M frames) from 285 speakers and 12 shows with diversified genres, and the related WER results are reliable metrics to evaluate the general performance of the DNN models for speech recognition. In contrast, the WERs on the stimuli set are much more noisier since it only consists of 400 isolated words from a single female speaker. However, the stimuli set WERs are still important metrics since the same 400 words are used to build the RDMs used in the key experiments. These results are presented in Table 1 and Figure 6C.

TABLE 1 The performance of DNN-HMM systems with different bottleneck layer positions.

System	Bottleneck layer	Accuracy%		WER%	
		Train	HV	MGB Dev	Stimuli
DNN-BN_7	L7	44.0	41.5	33.3	6.5
DNN-BN_6	L6	44.6	42.3	32.4	6.3
DNN-BN_5	L5	44.2	42.3	32.3	5.8
DNN-BN_4	L4	42.6	41.1	33.5	7.3

The WERs (the lower the better) were given on both the MGB challenge official development subset (MGB Dev), which is a general purpose large vocabulary continuous speech recognition testing set, as well as the 400 isolated words used as the stimuli in our listening experiments to derive the RDM (Stimuli). The MGB Dev WERs are reliable indicators for the general performance of the systems in realistic ASR tasks. The Stimuli WERs are the most direct indicators of the model performance on the data used in our brain-machine comparison experiments. The classification accuracy values (the higher the better) were obtained by classifying each frame into one of the 6,027 triphonetic DNN output units were obtained on both the training and held-out validation (HV) sets. For fair comparisons, DNN structures of all systems were constrained to have the same amount of model parameters (about 5M for each model, as shown in Figure 5). Accuracy can be considered as an auxiliary performance metric, which indicates that DNN-BN_6 suffered more from over-fitting compared to DNN-BN_5, since DNN-BN_6 is better in the training accuracy but not in the HV accuracy.

4.2. Results

As shown in Table 1 and Figure 6C, adjusting the design of the DNN structure to better fit with the representations

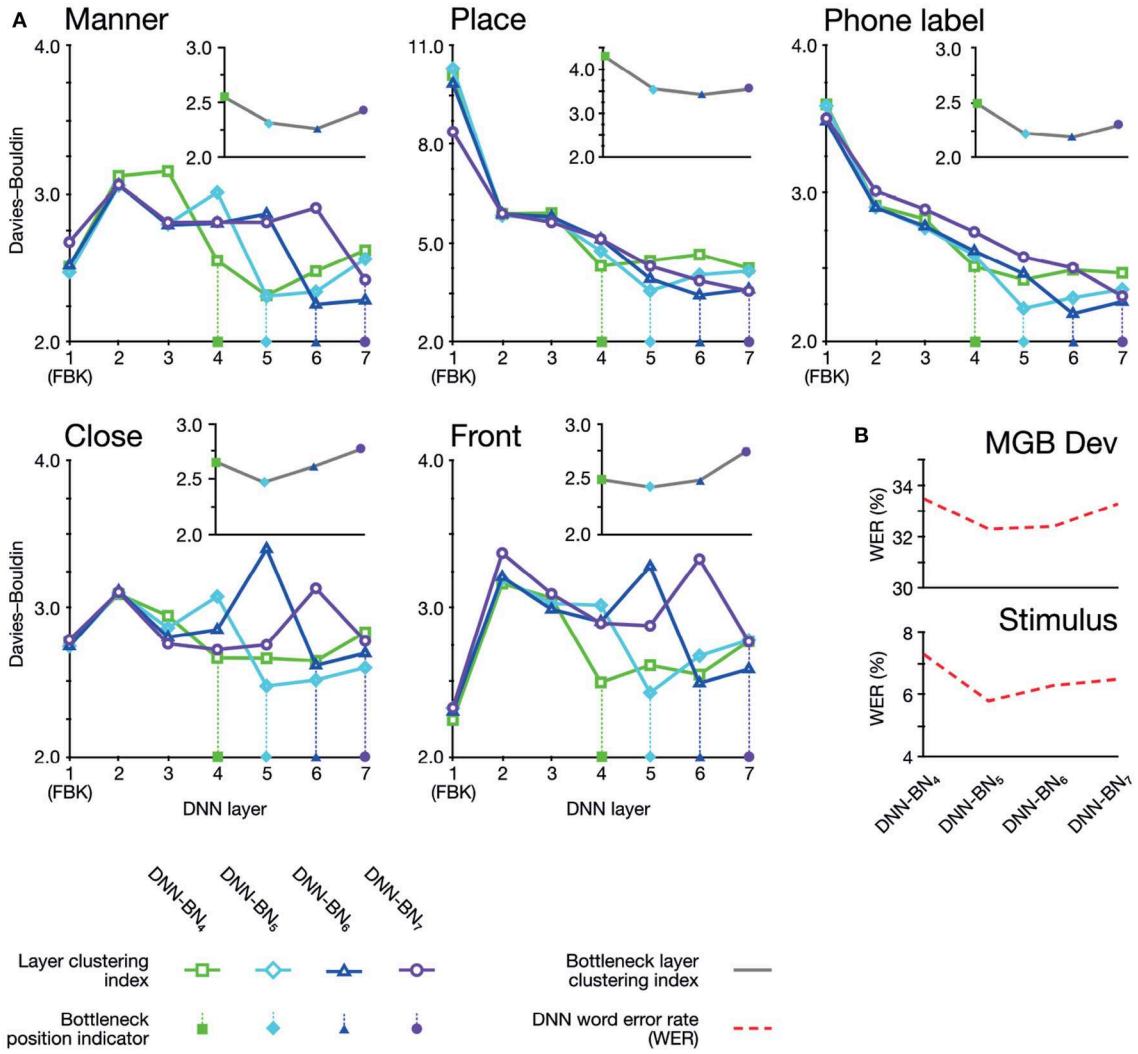


FIGURE 6
(A) Davies–Bouldin clustering indices for hidden-layer representations. Each plot shows the Davies–Bouldin clustering index for the average hidden-layer representation for each phonetic segment of each stimulus. Lower values indicate better clustering. Indices were computed by labeling each segment by its phonetic label (top right panel), or by place, manner, frontness, or closeness features (other panels). Colored shapes on the DNN-layer axis indicate the placement of the bottleneck layer for each System. Inset axes show clustering indices for bottleneck-layers only. Each plot shows the clustering index for the average bottleneck-layer representation for each phonetic segment of each stimulus. Indices were computed by labeling each segment by its phonetic label (top right), or by place, manner, frontness, or closeness features. Colored shapes on the DNN-layer axis indicate the placement of the bottleneck layer for each System. **(B)** WERs for each DNN system. Upper panel shows WERs on the MGB Dev set. Lower panel shows WERs for the stimuli.

exhibited in the human subjects led to improved DNN performance in terms of WER in DNN-BN_5 and DNN-BN_6. The MGB Dev set contains sufficient testing samples with diversified speaker and genre variations. When testing on MGB Dev, a 4-g language model (Woodland et al., 2015) was used to provide word-level contexts by rescoring each hypothesis in decoding as in general large vocabulary continuous speech recognition applications. The 1.0% absolute WER reduction (relatively 3.3%) obtained by comparing DNN-BN_7 with DNN-BN_5 is substantial (Bell et al., 2015; Woodland et al., 2015). Such an improvement was achieved without increasing the number of parameters, and hence demonstrates the superiority of the structure of DNN-BN_5. DNN-BN_6 also performed 0.9% (absolute WER) better than DNN-BN_7, but 0.1% WER worse than DNN-BN_5. This can also be observed from the frame classification accuracy values, as DNN-BN_6 has the same HV accuracy but better train accuracy compared with DNN-BN_5, indicating that placing the bottleneck layer at L6 results in over-fitting. Regarding the stimulus set, no language model was used since each stimulus utterance has only one word and the

recognition requires no word-level context. Still, the changes in WERs are consistent with those on the MGB Dev set. Comparing Table 1 to Figure 5, the WERs and the maximum cluster extent values of these DNN models are also consistent on the Stimulus test set.

As well as altering the position of the bottleneck layer, we also trained and tested a DNN without a bottleneck layer, but using the same 5.0M parameters. This DNN achieved 44.0% train accuracy and 42.3% HV accuracy, and 32.3% MGB Dev WER and 5.8% WER on the stimuli. In other words, close-to, but just falling short of (albeit insignificantly), the overall best model including a bottleneck layer: DNN-BN_5. The inclusion of a bottleneck layer was included in DNN-BN_5 was motivated both for machine-learning and computational-modeling reasons, as we have described. It is therefore notable that even though DNN-BN_5 contains a bottleneck layer, and thus forces a compression of the speech representation from 1,000 down to 26 dimensions, it was still able to achieve the overall best performance.

What is not immediately clear, however, is whether this improvement in performance arises from a corresponding improvement in the model's ability to extract a feature-based representation. In other words, if the bottleneck layer learns a representation akin to articulatory features, by moving the layer to improve performance does this enhance this learned representation? To answer this question, we investigated how the assignment of phonetic and featural labels to each segment of the stimuli could explain their hidden-layer representations. As before, we probed the organization of the representational space of each hidden layer according to phones and features using Davies–Bouldin clustering indices.

The clustering results exhibited two overall patterns of note. First, clustering (i.e., suitability of assignment of phonetic and featural labels to hidden layer representations) was improved on the DNNs whose design had been inspired by the human brains. Second, the optimum clustering level was often found in the bottleneck layer itself (highlighted on the graphs in Figure 6A). The clustering index at the bottleneck layers alone are separately graphed in inset panels in Figure 6A, and show that bottleneck layer clustering was also improved in DNN-BN_5 and DNN-BN_6.

In other words, the placement of the bottleneck layer in position 5 and 6 yielded, as predicted, the best clustering results both overall and in the bottleneck layer itself. Moving the bottleneck layer too far back (DNN-BN_4) yielded worse clustering results generally and in the bottleneck layer—indicated by the characteristic U-shaped curves in Figure 6B.

4.3. Discussion

Artificial Intelligence (AI) and machine learning have already been extensively applied in neuroscience primarily in analyzing and decoding large and complex neuroimaging or cell recording data sets. Here, DNN-based ASR systems were used as a model for developing and testing hypothesis and neuroscientific theories about how human brains perform speech recognition. This type of mechanistic or generative model—where the computational model can perform the behavioral task with realistic data (in this case, spoken word recognition)—can serve as a comprehensive framework for testing claims about neurocognitive functional organization (Kriegeskorte and Douglas, 2018). Moreover, insights can flow both ways; the neuroimaging data can also guide the exploration of the model space and lead to improvements in model performance, as we have seen.

While our use of neurological data only indirectly informed the improvements to ASR architecture, the present work can be seen as an initial step toward extracting system-level designs for neuromorphic computing from human auditory systems. This goal in itself is not new (see e.g., Toneva and Wehbe, 2019), however the key novel element of our approach is the ability to relate the machine and human solutions in complementary directions. The power of RSA, and in particular ssRSA, to relate the different forms of representations in these systems is key in this work. In summary, the methodology illustrated here paves the way for future integration of neuroscience and AI with the two fields driving each other forwards.

5. General discussion

We have used a DNN-based ASR system and spatiotemporal imaging data of human auditory cortex in a mutually informative study. In the machine-to-human direction, we have used a computational model of speech processing to examine representations of speech throughout space and time in human auditory cortex measured as source-localized EMEG data. In so doing, we have produced a functional map in human subjects for each part of the multi-stage computational model. We were able to relate dynamic states in the operating machine speech recognizer to dynamic brain states in human participants by using ssRSA, extended to account for a dynamically changing model. In a complementary analysis, we have improved the performance of the DNN-based ASR model by adapting the layered network architecture inspired by the staged neural activation patterns observed in human auditory cortex.

5.1. Relating dynamic brain and machine states: Comparing and contrasting computational models in vision and audition

There has been some recent successes in comparing machine models of perception to human neuroimaging data. This has primarily been in the domain of visual object perception (e.g.,

Kriegeskorte et al., 2008b; Cadieu et al., 2014; Clarke et al., 2014; Khaligh-Razavi and Kriegeskorte, 2014; Güçlü and van Gerven, 2015; Kriegeskorte, 2015; Cichy et al., 2016; Kheradpisheh et al., 2016; Devereux et al., 2018), with less progress made in speech perception (though see our previous work: Su et al., 2014; Wingfield et al., 2017).

The visual systems of humans and other primates are highly related, both in their architecture and in accounts of the neurocomputational processes they facilitate. There is evidence of a hierarchical organization of cortical regions in the early visual systems of human and non-human primates. There are also detailed accounts of process sequencing from early visual cortex through higher perceptual and semantic representation which exist for visual object perception in several primate models (e.g., Van Essen et al., 2001; Tootell et al., 2003; Denys et al., 2004; Orban et al., 2004; Kriegeskorte et al., 2008b). This is not so the case for speech processing and audition to the same degree.

In parallel, machine models for vision have often been designed based on theories of primate cortical processing hierarchies. This extends to recent work employing deep convolutional neural networks (CNN) for visual object processing, in particular those featuring layers of convolution and pooling. Furthermore, the convolutional layers in CNNs appear to learn features resembling those in the receptive fields of early visual cortex, and higher layers' representational spaces also match those found in higher visual cortex, and other regions in the visual object perception networks (Khaligh-Razavi and Kriegeskorte, 2014; Güçlü and van Gerven, 2015; Wen et al., 2018). Importantly, this means that the internal structures of machine vision systems are potentially informative and relevant to our understanding of the neurocomputational architecture of the natural system (and vice versa), and not just whether they generate equivalent outputs (for example in object classification tasks). To date, these common features are not well-established for DNNs or other type of acoustic models widely used for ASR systems.

Certain aspects of the human auditory processing system have resemblances to those in other primate models (Rauschecker and Scott, 2009; Baumann et al., 2013). However, no non-human primate supports anything like human speech communication, where intricately modulated sequences of speech sounds map onto hundreds of thousands of learned linguistic elements (words and morphemes), each with its own combination of acoustic-phonetic identifiers.

Perhaps due to this lack of neurocomputationally explicit models of spoken word recognition, the design of ASR systems has typically not been guided by existing biological models. Rather, by optimizing for engineering-relevant properties such as statistical learning efficiency, they have nonetheless achieved impressive accuracy and robustness.

It is striking, therefore, that we have been able to show that the regularities that successful ASR systems encode in the mapping between speech input and word-level phonetic labeling can indeed be related to the regularities extracted by the human system. In addition, like animal visual systems have inspired the field of computer vision, we have demonstrated that human auditory cortex can improve ASR systems using ssRSA.

6. Conclusion and future work

We have shown that our deep artificial neural network model of speech processing bears resemblance to patterns of activation in the human auditory cortex using the combination of ssRSA with multimodal neuroimaging data. The results also showed that the low-dimensional bottleneck layer in the DNN could learn representations that characterize articulatory features of human speech. In ASR research, although the development of systems based around the extraction of articulatory features has a long history (e.g., Deng and Sun, 1994), except for a small number of exemplars (e.g., Zhang et al., 2011; Mitra et al., 2013), recent studies mostly rely on written-form-based word piece units (Schuster and Nakajima, 2012; Wu et al., 2016) that are not directly associated with phonetic units. Our findings imply that developing appropriate intermediate representations for articulatory features may be central to speech recognition in both human and machine solutions. In human neuroscience studies, this account is consistent with previous findings of articulatory feature representation in the human auditory cortex (Mesgarani et al., 2014; Correia et al., 2015; Wingfield et al., 2017), but awaits further investigation and exploitation in machine solutions for speech recognition. In particular, previous work by Hamilton et al. (2021) has shown that—unlike our DNN architecture—the organization of early speech areas in the brain are not purely hierarchical, suggesting new potential avenues of model architectures including layer-bypassing connections.

The results we have presented here prompt further questions regarding how modifications to the design and training of DNN-based ASR models affects their representations, how to most effectively tailor a model to match the representational organization of the human brain, and which of these modifications lead to improved performance at the task. We hope to continue similar investigations to other types of artificial neural network models in our future work, such as different hidden activation functions, time-delay neural networks (Waibel et al., 1989; Peddinti et al., 2015), CNNs (LeCun et al., 1998; Krizhevsky et al., 2012), and recurrent neural networks (Rumelhart et al., 1986; Hochreiter and Schmidhuber, 1997), etc.

There is a difference between speech recognition (i.e., the extraction of word identities from speech audio) and speech comprehension (i.e., understanding and the elicitation of meaning). In this paper we have tackled only recognition. The HTK model we used is established and highly used in

the literature, and while it is able to incorporate context *via* the sliding window and hidden Markov language model, we certainly would not claim that it understands or comprehends speech as humans can. Recently, large deep artificial neural network models pre-trained on a massive amount of unlabeled waveform features (e.g., Baevski et al., 2020; Hsu et al., 2021; Chen et al., 2022), have demonstrated strong generalization abilities to ASR and many para-linguistic speech tasks (Mohamed et al., 2022). While we would not claim that these larger models are capable of true understanding, it would nonetheless be interesting to apply the methods used in this paper to study similar types of models and tasks. This may contribute to understanding the functional organization of human auditory cortex and improve such large scale speech-based computational models.

Data availability statement

Publicly available datasets were analyzed in this study. This data can be found here: The 200 h Multi-genre Broadcast (MGB) dataset used to train the ASR model was only available to the 2015 MGB-1 Challenge (http://www.mgb-challenge.org/) participants with copyright restrictions from BBC. The 26-dimensional hidden layer representations extracted from the L7 layer of the DNN-BN_7 model can be found in (http://mi.eng.cam.ac.uk/~cz277/stimuli). Masked, preprocessed human neuroimaging data used for this analysis is available from figshare (https://doi.org/10.6084/m9.figshare.5313484.v1). The DNN-based ASR system was created using an open-source toolkit, the HTK toolkit version 3.5 (https://htk.eng.cam.ac.uk/). The RSA procedure for this paper was performed using the open-source RSA toolbox (https://github.com/rsagroup/rsatoolbox_matlab), with the addition of specific extensions for ssRSA for EMEG (https://github.com/lisulab/rsatoolbox and https://github.com/lisulab/rsa-dnn-mapping). RDMs were computed from DNN layer representations using publicly available scripts (https://github.com/lisulab/htk-postprocessing).

Ethics statement

The studies involving human participants were reviewed and approved by Peterborough and Fenland Ethical Committee (UK). The patients/participants provided their written informed consent to participate in this study.

Author contributions

CW: conceptualization, formal analysis, software, methodology, writing, and editing. CZ: conceptualization, formal analysis, software, methodology, writing, editing, and data curation. BD: methodology and editing. EF: data acquisition, data curation, and editing. AT: software, data curation, and editing. XL: software, methodology, and editing. PW and WM-W: conceptualization, funding acquisition, supervision, and editing. LS: conceptualization, software, methodology, funding acquisition, supervision, writing, and editing. All authors contributed to the article and approved the submitted version.

Funding

This research was supported financially by a Senior Research Fellowship to LS from Alzheimer's Research UK (ARUK-SRF2017B-1), an Advanced Investigator grant to WM-W from the European Research Council (AdG 230570 NEUROLEX), by MRC Cognition and Brain Sciences Unit (CBSU) funding to WM-W (U.1055.04.002.00001.01), and by a European Research Council Advanced Investigator grant under the European Community's Horizon 2020 Research and Innovation Programme (2014-2020 ERC Grant agreement no 669820) to Lorraine K. Tyler.

Acknowledgments

The authors thank Anastasia Klimovich-Smith, Hun Choi, Lorraine Tyler, Andreas Marouchos, and Geoffrey Hinton for thoughtful comments and discussions. RSA computation was done in the RSA toolbox for Matlab (Nili et al., 2014) using custom EMEG and ssRSA extensions, to which Isma Zulfiqar, Fawad Jamshed, and Jana Klímová also contributed.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

Supplementary material

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fncom.2022.1057439/full#supplementary-material

References

Arsenault, J. S., and Buchsbaum, B. R. (2015). Distributed neural representations of phonological features during speech perception. *J. Neurosci.* 35, 634–642. doi: 10.1523/JNEUROSCI.2454-14.2015

Baevski, A., Zhou, Y., Mohamed, A., and Auli, M. (2020). "wav2vec 2.0: a framework for self-supervised learning of speech representations," in *Proceedings of the 34th International Conference on Neural Information Processing Systems: NIPS'20, Vol. 33* (Vancouver, BC), 12449–12460.

Baumann, S., Petkov, C. I., and Griffiths, T. D. (2013). A unified framework for the organization of the primate auditory cortex. *Front. Syst. Neurosci.* 7, 11. doi: 10.3389/fnsys.2013.00011

Bell, P., Gales, M., Hain, T., Kilgour, J., Lanchantin, P., Liu, X., et al. (2015). "The MGB challenge: evaluating multi-genre broadcast media transcription," in *Proc. ASRU* (Scotsdale, AZ), 687–693. doi: 10.1109/ASRU.2015.7404863

Bishop, C. (2006). *Pattern Recognition and Machine Learning*. New York, NY: Springer.

Bourlard, H., and Morgan, N. (1993). *Connectionist Speech Recognition: A Hybrid Approach*. Norwell, MA: Kluwer Academic Publishers. doi: 10.1007/978-1-4615-3210-1

Cadieu, C. F., Hong, H., Yamins, D. L., Pinto, N., Ardila, D., Solomon, E. A., et al. (2014). Deep neural networks rival the representation of primate IT cortex for core visual object recognition. *PLOS Comput. Biol.* 10, e1003963. doi: 10.1371/journal.pcbi.1003963

Chan, A. M., Dykstra, A. R., Jayaram, V., Leonard, M. K., Travis, K. E., Gygi, B., et al. (2014). Speech-specific tuning of neurons in human superior temporal gyrus. *Cereb. Cortex* 24, 2679–2693. doi: 10.1093/cercor/bht127

Chang, E. F., Rieger, J. W., Johnson, K., Berger, M. S., Barbaro, N. M., and Knight, R. T. (2010). Categorical speech representation in human superior temporal gyrus. *Nat. Neurosci.* 13, 1428–1432. doi: 10.1038/nn.2641

Chen, S., Wang, C., Chen, Z., Wu, Y., Liu, S., Chen, Z., et al. (2022). WavLM: large-scale self-supervised pre-training for full stack speech processing. *arXiv preprint arXiv:2110.13900*. doi: 10.1109/JSTSP.2022.3188113

Cichy, R. M., Khosla, A., Pantazis, D., and Oliva, A. (2016). Dynamics of scene representations in the human brain revealed by magnetoencephalography and deep neural networks. *Neuroimage*. 153, 346–358. doi: 10.1016/j.neuroimage.2016.03.063

Clarke, A., Devereux, B. J., Randall, B., and Tyler, L. K. (2014). Predicting the time course of individual objects with MEG. *Cereb. Cortex* 25, 3602–3612. doi: 10.1093/cercor/bhu203

Correia, J. M., Jansma, B. M., and Bonte, M. (2015). Decoding articulatory features from fMRI responses in dorsal speech regions. *J. Neurosci.* 35, 15015–15025. doi: 10.1523/JNEUROSCI.0977-15.2015

Davies, D. L., and Bouldin, D. W. (1979). A cluster separation measure. *IEEE Trans. Pattern Anal. Mach. Intell.* PAMI-1, 224–227. doi: 10.1109/TPAMI.1979.4766909

Deng, L., and Sun, D. X. (1994). A statistical approach to automatic speech recognition using the atomic speech units constructed from overlapping articulatory features. *J. Acoust. Soc. Am.* 95, 2702–2719. doi: 10.1121/1.409839

Denys, K., Vanduffel, W., Fize, D., Nelissen, K., Peuskens, H., Van Essen, D., et al. (2004). The processing of visual shape in the cerebral cortex of human and nonhuman primates: a functional magnetic resonance imaging study. *J. Neurosci.* 24, 2551–2565. doi: 10.1523/JNEUROSCI.3569-03.2004

Devereux, B. J., Clarke, A., and Tyler, L. K. (2018). Integrated deep visual and semantic attractor neural networks predict fMRI pattern-information along the ventral object processing pathway. *Nat. Sci. Rep.* 8, 10636. doi: 10.1038/s41598-018-28865-1

Di Liberto, G. M., O'Sullivan, J. A., and Lalor, E. C. (2015). Low-frequency cortical entrainment to speech reflects phoneme-level processing. *Curr. Biol.* 25, 2457–2465. doi: 10.1016/j.cub.2015.08.030

Doddipatla, R., Hasan, M., and Hain, T. (2014). "Speaker dependent bottleneck layer training for speaker adaptation in automatic speech recognition," in *Proc. Interspeech*, 2199–2203. doi: 10.21437/Interspeech.2014-492

Fonteneau, E., Bozic, M., and Marslen-Wilson, W. D. (2014). Brain network connectivity during language comprehension: interacting linguistic and perceptual subsystems. *Cereb. Cortex* 25, 3962–3976. doi: 10.1093/cercor/bhu283

Gramfort, A., Luessi, M., Larson, E., Engemann, D. A., Strohmeier, D., Brodbeck, C., et al. (2014). MNE software for processing MEG and EEG data. *Neuroimage* 86, 446–460. doi: 10.1016/j.neuroimage.2013.10.027

Grézl, F., Karafiát, M., Kontár, S., and Černocký, J. (2007). "Probabilistic and bottle-neck features for LVCSR of meetings," in *Proc. ICASSP* (Honolulu, HI), 757–760.

Güçlü, U., and van Gerven, M. A. (2015). Deep neural networks reveal a gradient in the complexity of neural representations across the ventral stream. *J. Neurosci.* 35, 10005–10014. doi: 10.1523/JNEUROSCI.5023-14.2015

Hämäläinen, M. S., and Ilmoniemi, R. (1994). Interpreting magnetic fields of the brain: minimum norm estimates. *Med. Biol. Eng. Comput.* 32, 35–42. doi: 10.1007/BF02512476

Hamilton, L. S., Oganian, Y., Hall, J., and Chang, E. F. (2021). Parallel and distributed encoding of speech across human auditory cortex. *Cell* 12, 4626.e13–4639.e13. doi: 10.1016/j.cell.2021.07.019

Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A.-R., Jaitly, N., et al. (2012). Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups. *IEEE Signal Process. Mag.* 29, 82–97. doi: 10.1109/MSP.2012.2205597

Hochreiter, S., and Schmidhuber, J. (1997). Long short-term memory. *Neural Comput.* 9, 1735–1780. doi: 10.1162/neco.1997.9.8.1735

Hsu, W.-N., Bolte, B., Tsai, Y.-H. H., K., L., Salakhutdinov, R., et al. (2021). HuBERT: self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM Trans. Audio Speech Lang. Process.* 29, 3451–3460. doi: 10.1109/TASLP.2021.3122291

Jenkinson, M., Beckmann, C. F., Behrens, T. E., Woolrich, M. W., and Smith, S. M. (2012). FSL. *Neuroimage* 62, 782–790. doi: 10.1016/j.neuroimage.2011.09.015

Karafiát, M., Grézl, F., Hannemann, M., Veselý, K., and Černocký, J. H. (2013). "BUT BABEL system for spontaneous Cantonese," in *Proc. Interspeech* (Lyon), 2589–2593. doi: 10.21437/Interspeech.2013-582

Khaligh-Razavi, S.-M., and Kriegeskorte, N. (2014). Deep supervised, but not unsupervised, models may explain IT cortical representation. *PLoS Comput. Biol.* 10, e1003915. doi: 10.1371/journal.pcbi.1003915

Kheradpisheh, S. R., Ghodrati, M., Ganjtabesh, M., and Masquelier, T. (2016). Deep networks can resemble human feed-forward vision in invariant object recognition. *Nat. Sci. Rep.* 6, 32672. doi: 10.1038/srep32672

Kriegeskorte, N. (2015). Deep neural networks: a new framework for modeling biological vision and brain information processing. *Annu. Rev. Vision Sci.* 1, 417–446. doi: 10.1146/annurev-vision-082114-035447

Kriegeskorte, N., and Douglas, P. K. (2018). Cognitive computational neuroscience. *Nat. Neurosci.* 21, 1148–1160. doi: 10.1038/s41593-018-0210-5

Kriegeskorte, N., Goebel, R., and Bandettini, P. (2006). Information-based functional brain mapping. *Proc. Natl. Acad. Sci. U.S.A.* 103, 3863–3868. doi: 10.1073/pnas.0600244103

Kriegeskorte, N., and Kievit, R. A. (2013). Representational geometry: Integrating cognition, computation, and the brain. *Trends Cogn. Sci.* 17, 401–412. doi: 10.1016/j.tics.2013.06.007

Kriegeskorte, N., Mur, M., and Bandettini, P. (2008a). Representational similarity analysis-connecting the branches of systems neuroscience. *Front. Syst. Neurosci.* 2, 4. doi: 10.3389/neuro.06.004.2008

Kriegeskorte, N., Mur, M., Ruff, D. A., Kiani, R., Bodurka, J., Esteky, H., et al. (2008b). Matching categorical object representations in inferior temporal cortex of man and monkey. *Neuron* 60, 1126–1141. doi: 10.1016/j.neuron.2008.10.043

Krizhevsky, A., Sutskever, I., and Hinton, G. (2012). "Imagenet classification with deep convolutional neural networks," in *Proc. NIPS* (New York, NY).

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proc. IEEE* 86, 2278–2324. doi: 10.1109/5.726791

Liu, X., Flego, F., Wang, L., Zhang, C., Gales, M., and Woodland, P. (2015). "The Cambridge university 2014 BOLT conversational telephone Mandarin Chinese LVCSR system for speech translation," in *Proc. Interspeech* (Dresden), 3145–3149. doi: 10.21437/Interspeech.2015-633

Luscher, C., Beck, E., Irie, K., Kitza, M., Michel, W., Zeyer, A., et al. (2019). "RWTH ASR systems for LibriSpeech: hybrid vs attention," in *Proc. Interspeech* (Graz), 231–235. doi: 10.21437/Interspeech.2019-1780

Mack, M. L., Love, B. C., and Preston, A. R. (2016). Dynamic updating of hippocampal object representations reflects new conceptual knowledge. *Proc. Natl. Acad. Sci. U.S.A.* 201614048, 113, 13203–13208. doi: 10.1073/pnas.1614048113

Mesgarani, N., Cheung, C., Johnson, K., and Chang, E. F. (2014). Phonetic feature encoding in human superior temporal gyrus. *Science* 343, 1006–1010. doi: 10.1126/science.1245994

Mesgarani, N., David, S. V., Fritz, J. B., and Shamma, S. A. (2008). Phoneme representation and classification in primary auditory cortex. *J. Acoust. Soc. Am.* 123, 899–909. doi: 10.1121/1.2816572

Mitra, V., Wang, W., Stolcke, A., Nam, H., Richey, C., Yuan, J., et al. (2013). "Articulatory trajectories for large-vocabulary speech recognition," in *Proc. ICASSP* (Vancouver, BC: IEEE), 7145–7149. doi: 10.1109/ICASSP.2013.6639049

Moerel, M., De Martino, F., and Formisano, E. (2014). An anatomical and functional topography of human auditory cortical areas. *Front. Neurosci.* 8, 225. doi: 10.3389/fnins.2014.00225

Mohamed, A., Lee, H.-Y., Borgholt, L., Havtorn, J., Edin, J., Igel, C., et al. (2022). Self-supervised speech representation learning: a review. *arXiv preprint arXiv:2205.10643*. doi: 10.1109/JSTSP.2022.3207050

Molins, A., Stufflebeam, S. M., Brown, E. N., and Hämäläinen, M. S. (2008). Quantification of the benefit from integrating MEG and EEG data in minimum ℓ_2-norm estimation. *Neuroimage* 42, 1069–1077. doi: 10.1016/j.neuroimage.2008.05.064

Morgan, N. (2011). Deep and wide: multiple layers in automatic speech recognition. *IEEE Trans. Audio Speech Lang. Process.* 20, 7–13. doi: 10.1109/TASL.2011.2116010

Moses, D. A., Leonard, M. K., and Chang, E. F. (2018). Real-time classification of auditory sentences using evoked cortical activity in humans. *J. Neural Eng.* 15, 036005. doi: 10.1088/1741-2552/aaab6f

Moses, D. A., Mesgarani, N., Leonard, M. K., and Chang, E. F. (2016). Neural speech recognition: Continuous phoneme decoding using spatiotemporal representations of human cortical activity. *J. Neural Eng.* 13, 056004. doi: 10.1088/1741-2560/13/5/056004

Nichols, T. E., and Holmes, A. P. (2002). Nonparametric permutation tests for functional neuroimaging: a primer with examples. *Hum. Brain Mapp.* 15, 1–25. doi: 10.1002/hbm.1058

Nili, H., Wingfield, C., Walther, A., Su, L., Marslen-Wilson, W., and Kriegeskorte, N. (2014). A toolbox for representational similarity analysis. *PLoS Comput. Biol.* 10, e1003553. doi: 10.1371/journal.pcbi.1003553

Orban, G. A., Van Essen, D., and Vanduffel, W. (2004). Comparative mapping of higher visual areas in monkeys and humans. *Trends Cogn. Sci.* 8, 315–324. doi: 10.1016/j.tics.2004.05.009

Park, D., Chan, W., Zhang, Y., Chiu, C.-C., Zoph, B., Cubuk, E., et al. (2019). "SpecAugment: a simple data augmentation method for automatic speech recognition," in *Proc. Interspeech* (Graz), 2613–2617. doi: 10.21437/Interspeech.2019-2680

Peddinti, V., Povey, D., and Khudanpur, S. (2015). "A time delay neural network architecture for efficient modeling of long temporal contexts," in *Proc. Interspeech* (Dresden), 3214–3218. doi: 10.21437/Interspeech.2015-647

Rauschecker, J. P., and Scott, S. K. (2009). Maps and streams in the auditory cortex: nonhuman primates illuminate human speech processing. *Nat. Neurosci.* 12, 718–724. doi: 10.1038/nn.2331

Rumelhart, D., McLelland, J., and PDP Research Group (1986). *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Volume 1: Foundations*. Cambridge, MA: MIT Press. doi: 10.7551/mitpress/5236.001.0001

Saenz, M., and Langers, D. R. (2014). Tonotopic mapping of human auditory cortex. *Hear. Res.* 307, 42–52. doi: 10.1016/j.heares.2013.07.016

Sammon, J. W. (1969). A nonlinear mapping for data structure analysis. *IEEE Trans. Comput.* 18, 401–409. doi: 10.1109/T-C.1969.222678

Saon, G., Kurata, G., Sercu, T., Audhkhasi, K., Thomas, S., Dimitriadis, D., et al. (2017). "English conversational telephone speech recognition by humans and machines," in *Proc. Interspeech* (Stockholm), 132–136. doi: 10.21437/Interspeech.2017-405

Schuster, M., and Nakajima, K. (2012). "Japanese and Korean voice search," in *Proc. ICASSP* (Kyoto), 5149–5152. doi: 10.1109/ICASSP.2012.6289079

Smith, S. M., and Nichols, T. E. (2009). Threshold-free cluster enhancement: addressing problems of smoothing, threshold dependence and localisation in cluster inference. *Neuroimage* 44, 83–98. doi: 10.1016/j.neuroimage.2008.03.061

Su, L., Fonteneau, E., Marslen-Wilson, W., and Kriegeskorte, N. (2012). "Spatiotemporal searchlight representational similarity analysis in EMEG source space," in *Proc. PRNI* (London), 97–100. doi: 10.1109/PRNI.2012.26

Su, L., Zulfiqar, I., Jamshed, F., Fonteneau, E., and Marslen-Wilson, W. (2014). Mapping tonotopic organization in human temporal cortex: representational similarity analysis in EMEG source space. *Front. Neurosci.* 8, 368. doi: 10.3389/fnins.2014.00368

Thwaites, A., Glasberg, B. R., Nimmo-Smith, I., Marslen-Wilson, W. D., and Moore, B. C. (2016). Representation of instantaneous and short-term loudness in the human cortex. *Front. Neurosci.* 10, 183. doi: 10.3389/fnins.2016.00183

Toneva, M., and Wehbe, L. (2019). "Interpreting and improving natural-language processing (in machines) with natural language-processing (in the brain)," in *33rd Conference on Neural Information Processing Systems (NeurIPS 2019)*, eds H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox and R. Garnett (Vancouver).

Tootell, R. B., Tsao, D., and Vanduffel, W. (2003). Neuroimaging weighs in: humans meet macaques in "primate" visual cortex. *J. Neurosci.* 23, 3981–3989. doi: 10.1523/JNEUROSCI.23-10-03981.2003

Tüske, Z., Golik, P., Schlüter, R., and Ney, H. (2014). "Acoustic modeling with deep neural networks using raw time signal for LVCSR," in *Proc. Interspeech* (Singapore), 890–894. doi: 10.21437/Interspeech.2014-223

Van Essen, D. C., Lewis, J. W., Drury, H. A., Hadjikhani, N., Tootell, R. B., Bakircioglu, M., et al. (2001). Mapping visual cortex in monkeys and humans using surface-based atlases. *Vision Res.* 41, 1359–1378. doi: 10.1016/S0042-6989(01)00045-1

Waibel, A., Hanazawa, T., Hinton, G., Shikano, K., and Lang, K. (1989). Phoneme recognition using time-delay neural networks. *IEEE Trans. Acoust. Speech Signal Process.* 37, 328–339. doi: 10.1109/29.21701

Wen, H., Shi, J., Zhang, Y., Lu, K. -H., Cao, J., and Liu, Z. (2018). Neural encoding and decoding with deep learning for dynamic natural vision. *Cereb. Cortex.* 28, 4136–4160. doi: 10.1093/cercor/bhx268

Wingfield, C., Su, L., Liu, X., Zhang, C., Woodland, P., Thwaites, A., et al. (2017). Relating dynamic brain states to dynamic machine states: human and machine solutions to the speech recognition problem. *PLoS Comput. Biol.* 13, e1005617. doi: 10.1371/journal.pcbi.1005617

Woodland, P., Liu, X., Qian, Y., Zhang, C., Gales, M., Karanasou, P., et al. (2015). "Cambridge University transcription systems for the Multi-genre Broadcast Challenge," in *Proc. ASRU* (Scotsdale, AZ), 639–646. doi: 10.1109/ASRU.2015.7404856

Wu, Y., Schuster, M., Chen, Z., Le, Q., Norouzi, M., Macherey, W., et al. (2016). Google's neural machine transltion system: bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Xiong, W., Wu, L., Droppo, J., Huang, X., and Stolcke, A. (2018). "The Microsoft 2016 conversational speech recognition system," in *Proc. ICASSP* (New Orleans), 5255–5259. doi: 10.1109/ICASSP.2017.7953159

Young, S., Evermann, G., Gales, M., Hain, T., Kershaw, D., Liu, X., et al. (2015). *The HTK Book (for HTK version 3.5)*. Cambridge: Cambridge University Engineering Department.

Young, S. J., Odell, J. J., and Woodland, P. C. (1994). "Tree-based state tying for high accuracy acoustic modelling," in *Proc. HLT* (Stroudsburg, PA: Association for Computational Linguistics), 307–312. doi: 10.3115/1075812.1075885

Yu, Z., Chuangsuwanich, E., and Glass, J. (2014). "Extracting deep neural network bottleneck features using low-rank matrix factorization," in *Proc. ICASSP* (Florence), 185–189. doi: 10.1109/ICASSP.2014.6853583

Zhang, C., Liu, Y., and Lee, C.-H. (2011). "Detection-based accented speech recognition using articulatory features," in *Proc. ASRU* (Waikoloa), 500–505. doi: 10.1109/ASRU.2011.6163982

Zhang, C., and Woodland, P. C. (2015a). "A general artificial neural network extension for HTK," in *Proc. Interspeech* (Dresden), 3581–3585. doi: 10.21437/Interspeech.2015-710

Zhang, C., and Woodland, P. C. (2015b). "Parameterised sigmoid and ReLU hidden activation functions for DNN acoustic modelling," in *Proc. Interspeech* (Dresden), 3224–3228. doi: 10.21437/Interspeech.2015-649

frontiers | Frontiers in Neuroscience

TYPE Original Research
PUBLISHED 22 December 2022
DOI 10.3389/fnins.2022.1075538

OPEN ACCESS

EDITED BY
Yuqi Han,
Tsinghua University, China

REVIEWED BY
Junhui Qian,
Chongqing University, China
Jun Wang,
Beihang University, China
Shuaifeng Zhi,
National University of Defense Technology, China

*CORRESPONDENCE
Liang Zhang
zhangliang@bit.edu.cn

SPECIALTY SECTION
This article was submitted to Perception Science, a section of the journal Frontiers in Neuroscience

RECEIVED 20 October 2022
ACCEPTED 28 November 2022
PUBLISHED 22 December 2022

CITATION
Wang Y, Han C, Zhang L, Liu J, An Q and Yang F (2022) Millimeter-wave radar object classification using knowledge-assisted neural network. *Front. Neurosci.* 16:1075538. doi: 10.3389/fnins.2022.1075538

Millimeter-wave radar object classification using knowledge-assisted neural network

Yanhua Wang[1,2,3,4], Chang Han[1,4], Liang Zhang[1,4]*, Jianhu Liu[5], Qingru An[5] and Fei Yang[2]

[1]Radar Research Laboratory, School of Information and Electronics, Beijing Institute of Technology, Beijing, China, [2]Beijing Institute of Technology Chongqing Innovation Center, Chongqing, China, [3]Advanced Technology Research Institute, Beijing Institute of Technology, Jinan, Shandong, China, [4]Electromagnetic Sensing Research Center of CEMEE State Key Laboratory, School of Information and Electronics, Beijing Institute of Technology, Beijing, China, [5]Beijing Rxbit Electronic Technology Co., Ltd., Beijing, China

To improve the cognition and understanding capabilities of artificial intelligence (AI) technology, it is a tendency to explore the human brain learning processing and integrate brain mechanisms or knowledge into neural networks for inspiration and assistance. This paper concentrates on the application of AI technology in advanced driving assistance system. In this field, millimeter-wave radar is essential for elaborate environment perception due to its robustness to adverse conditions. However, it is still challenging for radar object classification in the complex traffic environment. In this paper, a knowledge-assisted neural network (KANN) is proposed for radar object classification. Inspired by the human brain cognition mechanism and algorithms based on human expertise, two kinds of prior knowledge are injected into the neural network to guide its training and improve its classification accuracy. Specifically, image knowledge provides spatial information about samples. It is integrated into an attention mechanism in the early stage of the network to help reassign attention precisely. In the late stage, object knowledge is combined with the deep features extracted from the network. It contains discriminant semantic information about samples. An attention-based injection method is proposed to adaptively allocate weights to the knowledge and deep features, generating more comprehensive and discriminative features. Experimental results on measured data demonstrate that KANN is superior to current methods and the performance is improved with knowledge assistance.

KEYWORDS

millimeter-wave radar, object classification, knowledge-assisted, neural network, artificial intelligence

Introduction

Thanks to the complex structure and mechanisms of the brain, humans have the capability to continuously learn new knowledge, perceive complex environments, and make precise decisions (Cornelio et al., 2022; Kuroda et al., 2022). With the groundbreaking discovery of cells and continuous research in neuroscience, a variety of artificial neural networks have been proposed (van Dyck et al., 2022). Neural networks have promoted the development of artificial intelligence (AI) technologies in many fields, such as smart healthcare (Alsubai et al., 2022; Soenksen et al., 2022), intelligent transportation (Zhu et al., 2019; Zhu F. et al., 2020), etc. Similar to humans, networks acquire capabilities through learning. However, they learn things by brute force optimization based on input data, which limits their performance in various practical applications. To promote the next generation of AI technology, the neurology mechanism of the human brain learning process is studied, and the brain mechanism or knowledge is integrated into neural networks to help networks improve the perception and understanding of the world (Marblestone et al., 2016; Lindsay, 2020; Zhu J. et al., 2020).

This paper mainly focuses on the application of AI technology in advanced driving assistance system (ADAS), which has proved its effectiveness in safe driving and its evolution is in full swing. To elaborately capture the surroundings, multiple sensors are equipped on vehicles, such as cameras, LiDAR and millimeter-wave (MMW) radar. Cameras provide high-resolution optical images that are in line with human visual cognition and are widely applied in object detection (Redmon et al., 2016; Ren et al., 2017; Kim and Ro, 2019; Deng et al., 2022) and tracking (Danelljan et al., 2014; Smeulders et al., 2014; Nam and Han, 2016; Zhao et al., 2017; Han et al., 2019b) tasks. Although cameras offer optical images and give a semantic understanding of real-world scenarios, it is not robust facing adverse conditions, such as weak lighting or bad weather (Wang et al., 2021). As for LiDAR, it generates point cloud data and can be utilized for object detection and localization (Qi et al., 2018; Shi et al., 2019, 2021). However, these methods require dense point clouds to describe detailed information for accurate prediction, and LiDAR also has poor robustness to fog (Bijelic et al., 2018), rain or snow.

Compared with cameras and LiDAR, MMW radar is more reliable and robust in harsh environments. It is widely used in many practical scenarios, such as remote sensing target detection and classification (Liu et al., 2018; Wang et al., 2018; Liu et al., 2021; Tang et al., 2022) and intelligent transportation (Munoz-Ferreras et al., 2008; Felguera-Martin et al., 2012). Therefore, perception from pure radar data becomes a valuable alternative (Wang et al., 2021). Although it is widely used to obtain accurate location information about different objects (Prophet et al., 2019), it is still a challenge to extract discriminative semantic features from radar data for precise object classification. Great efforts have been made to advance MMW radar object classification performance. Existing researches are mainly based on three kinds of radar data, including micro-Doppler signatures (Villeval et al., 2014; Angelov et al., 2018; Held et al., 2019), point clouds (Feng et al., 2019; Zhao et al., 2020) and range-Doppler (RD) maps or range-azimuth maps (Major et al., 2019; Palffy et al., 2020). Since RD maps can be easily obtained in engineering and maintain rich Doppler and object motion information, this paper focuses on object classification based on RD maps.

Typical feature-based approaches (Rohling et al., 2010; Heuel and Rohling, 2012) extract hand-crafted features from RD maps, such as velocity, extension in range dimension, etc., which are physically interpretable. Then, a support vector machine (SVM) classifier is trained to classify the features. To extract these features, humans constantly learn and summarize laws from various objective things and construct feature extraction algorithms based on accumulated knowledge and experience. Therefore, these methods heavily rely on human experience and algorithm design, and their performance may degrade in complex practical application scenarios.

Recent advances in deep learning have promoted the development of automatic object classification. By learning and optimizing details from pure input data, neural networks can accomplish various specific tasks. A convolutional neural network (CNN) has been established to extract valuable features for automotive radar object classification (Patel et al., 2019; Shirakata et al., 2019). Recently, a radar object detection method was proposed (Gao et al., 2019), which combines a statistical constant false alarm rate (CFAR) detection algorithm with a visual geometry group network 16 (VGG-16) classifier (Simonyan and Zisserman, 2015). After that, RadarResNet (Zhang A. et al., 2021) was constructed for dynamic object detection based on range-azimuth-Doppler maps. Ouaknine et al. (2021) utilized a fully convolutional network (FCN) to accomplish object detection and classification tasks. A RODNet (Wang et al., 2021) was proposed for radar object detection based on cross-modal supervision approach. These methods automatically learn features from training data and obtain good results. However, they discard human knowledge, which means the information they obtain may be not comprehensive.

In order to promote the intelligence of radar object classification and achieve more accurate and stable performance, it is a trend to introduce prior knowledge generated from human brains and experience into neural networks for assistance and guidance. Recently, similar ideas and methods have been studied in many fields. Reference (Chen and Zhang, 2022) presented the concept of knowledge embedding in machine learning and summarized the current research results. In the field of radar target classification, physics-aware features were obtained from synthetic aperture radar (SAR) images and injected into the layer of a deep network to provide abundant prior information for training and classification

(Huang et al., 2022). In Zhang et al. (2022), azimuth angle and phase information were extracted from SAR images and served as domain knowledge to improve the performance of SAR vehicle classification. For polarimetric high-resolution range profile classification, a feature-guided network was proposed with state-of-the-art results (Zhang L. et al., 2021). In the driving assistance system, the information obtained by the tracker has been studied to improve object classification accuracy (Heuel and Rohling, 2011). A state-aware method was proposed to model the discrimination and reliability information synchronously into the tracking framework to ensure robust performance (Han et al., 2019a).

Following the idea, a knowledge-assisted neural network (KANN) using RD map sequence for automotive MMW radar object classification is proposed. The primary intention is to inject knowledge into the neural network to supplement the network with physical information and to improve the classification performance. The network imitates the structure of neural mechanisms in human brains, however, it achieves learning tasks through brute force optimization of input data and lacks perception of the practical physical world. While knowledge is generated based on how and what the human brain thinks when accomplishing complex tasks. It conforms to human brain cognition and is an objective and physical description of the objects in the practical world. By fusing the knowledge and high dimensional data fitting, the network will have some physical cognition capability and be more similar to the way the human brain perceives, which will improve the network performance in practical driving assistance applications.

Specifically, in the method, the RD map sequence is served as input, which consists of several frames of region-of-interest (RoI) about an object based on CFAR algorithm. To improve the performance of the network, two kinds of prior knowledge of RD maps based on human expertise are extracted and hierarchically integrated into the network for assistance. The first one is image knowledge which describes the explicit spatial information of the RD map. It is obtained from the algorithms consistent with the human brain visual mechanism and applied to the attention mechanism to help the network more accurately concentrate on object regions. The second one is object knowledge which represents the semantic attribute information of objects. It includes the ranges, velocities, azimuths, and RD map extension features, which are important when humans are classifying objects. Additionally, RD maps of the same object may vary with different ranges, velocities, etc. Therefore, object knowledge is injected into the network to assist its training and classification. It is combined with the deep features extracted from the network adaptively through an attention-based injection method to provide more comprehensive and discriminative features. Experimental results on measured data of four kinds of objects demonstrate that KANN can achieve advanced performance and the assistance of knowledge is helpful.

Knowledge-assisted neural network

The architecture of KANN is shown in **Figure 1**. KANN employs the RD map sequence containing several consecutive frames of RoIs in RD maps about an object as input data. The RoIs are cut out from RD maps based on CFAR algorithm. Different frames are fed into the network as different channels to provide temporal dimension information. Knowledge-guided attention module (KAM) and knowledge injection module (KIM) are proposed to generate the features for classification with knowledge assistance. The knowledge utilized is some prior information obtained from artificial algorithms, and it contains the physical cognition consistent with human brain when humans classify objects in traffic environments. Specifically, in KAM, an attention mechanism is established, and the prior image knowledge containing specific spatial information is applied to help make the attention assignment more reasonable and discriminative. KIM is utilized to extract spatiotemporal information about input data. Inspired by the human brain cognition when classifying objects, in this module, object knowledge containing semantic attribute information is adaptively injected into the network to provide more valuable information for classification. The rest of this section will first introduce the RD map sequence generation method in detail. Then, the specific contents of KAM and KIM are explained.

Range-Doppler map sequence generation

MMW radar dominantly transmits continuous chirps and receives the reflected echoes from objects. The workflow of RD map sequence generation is shown in **Figure 2**. First, the RD map is generated from the radar signal through the 2D-fast Fourier transform (FFT). The 2D-CFAR detection algorithm is then utilized to detect the objects. After that, inspired by Patel et al. (2019), a fixed-size RoI of the RD map is cut out for each detected object. Finally, an RD map sequence is constructed by stacking several RoIs about the same object in continuous frames of RD maps. The RoIs across frames are associated based on the range and velocity correlation, which means the detection results with the largest overlap are regarded as the same object. It should be noted that, to provide temporal information, in a sequence, the highest detected peak of the object is in the center of the first RoI, and the rest RoIs have the same location in the RD map as the first one. The ground truth categories of the RD map sequences are annotated according to the optical images. Specifically, before the data collection, the radar sensor and

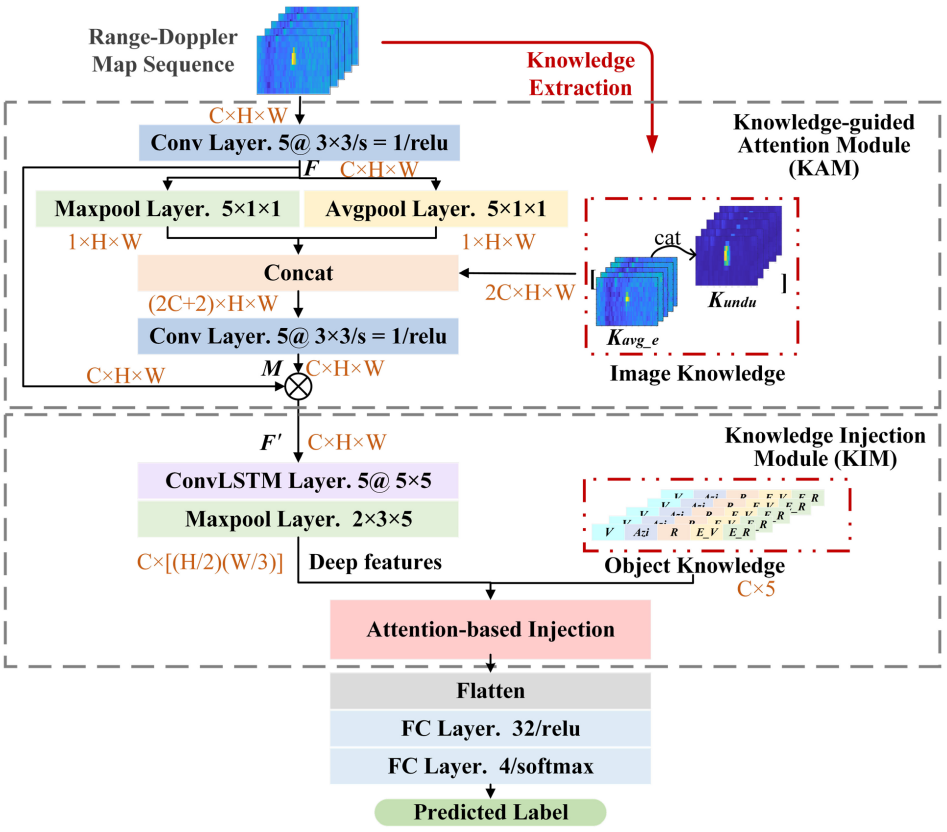


FIGURE 1
Illustration of knowledge-assisted neural network (KANN). (C, H, and W denote the channel number, height, and width of the data; ⊗ represents Hadamard product).

camera are calibrated in typical scenarios. First, the range and azimuth measurement results of the radar sensor are calibrated based on angle reflectors. Then, some cooperative pedestrians, cyclists, and cars are employed as detection objects on a test road. The radar data and optical images are recorded separately, and the locations and other information of the objects from the two sensors are compared and calibrated. Finally, after collecting the measured data, the RoIs in RD maps are labeled based on the optical images.

Knowledge-guided attention module

Since an object only occupies a small region in the RD map, KAM establishes an attention mechanism that is inspired by the visual attention mechanism of human brains (Lindsay, 2020). It generates different weights to help networks focus on the discriminative regions in each RoI, while suppressing unnecessary ones. In KAM, as shown in Figure 1, image knowledge is prepared as the assistant knowledge to participate in the generation of the attention matrix for more precise attention assignment. Considering that the spatial information obtained by the network lacks the physical cognition of the practical world, introducing image knowledge can make the network assign attention in a way more similar to the human brain. The image knowledge is obtained from algorithms based on human expertise and is composed of the average energy (K^{avg_e}) and undulation feature (K^{undu}) which delineate the exact spatial information and distinguish the target and background clutter. Given a pixel s_{ij} whose location is (i, j) in an RoI, we consider the pixels in the surrounding region with the size of 3×3 to calculate the features:

$$K_{ij}^{avg\_e} = \frac{1}{n}\left(\sum_{i-1}^{i+1}\sum_{j-1}^{j+1} s_{ij}^2\right), \tag{1}$$

$$K_{ij}^{undu} = \frac{1}{n}\left(\sum_{i-1}^{i+1}\sum_{j-1}^{j+1} \left(s_{ij} - \bar{s}\right)^2\right), \tag{2}$$

where $n = 9$ and $\bar{s}$ denotes the number of pixels and the mean amplitude of the RoI, respectively. By stacking the two feature sequences in channel dimension, image knowledge of the RD map sequence can be obtained.

Figure 3 shows the two features extracted over the same RoI in an RD map. It can be observed that K^{avg_e} and K^{undu} can represent the spatial information consistent with the visual

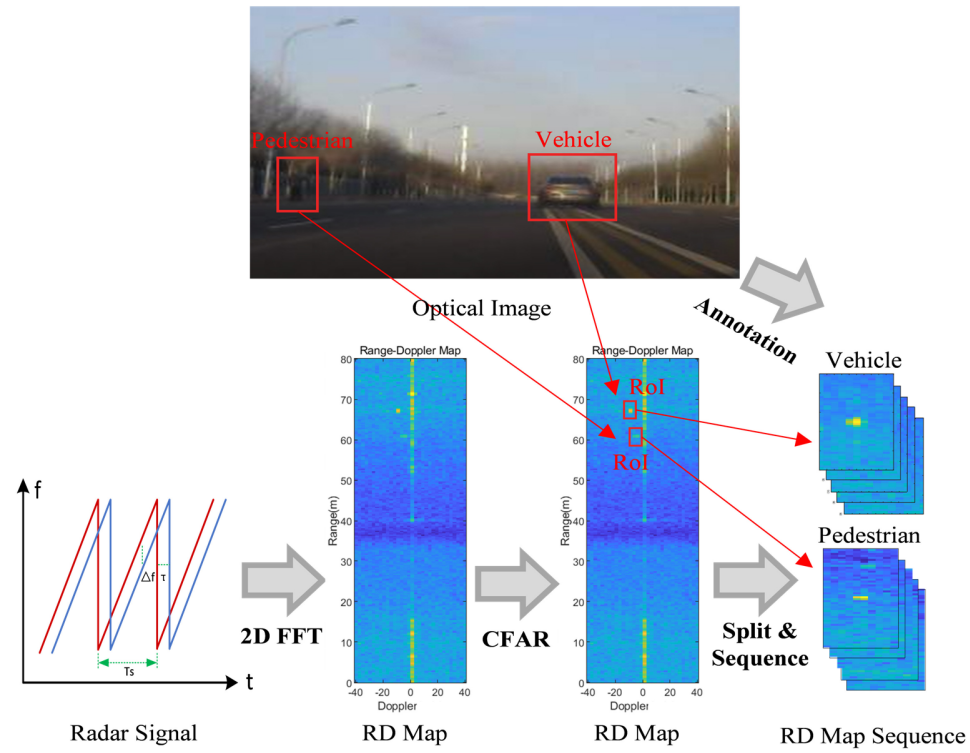


FIGURE 2
The workflow of range-Doppler (RD) map sequence generation.

cognition of the human brain. Concretely, K^{avg_e} describes the average energy of the region and highlights the target regions, while K^{undu} describes the amplitude undulation information.

In this module, given an RD map sequence, it is first processed with a convolutional layer whose kernel size is 3×3 to obtain the feature map $\boldsymbol{F}$. Then, a max pooling layer and an average pooling layer are applied to down-sampled $\boldsymbol{F}$ in two aspects to capture spatial information autonomously. The size of the layers is configured to $5 \times 1 \times 1$ to obtain compact spatial information. At this time, image knowledge is introduced to concatenated with the pooling results in channel dimension to generate the weight matrix $\boldsymbol{M}$:

$$\boldsymbol{M} = \sigma\left(f\left(concat\left(MaxPool(\boldsymbol{F});\ AvgPool(\boldsymbol{F});\ IMK\right)\right)\right), \quad (3)$$

where σ denotes the "relu" activation function, f represents the convolution operation, $MaxPool\ (\cdot)$ and $AvgPool\ (\cdot)$ denote the max pooling and average pooling operation respectively. Next, according to $\boldsymbol{M}$, the redefined feature map $\boldsymbol{F}'$ can be obtained:

$$\boldsymbol{F}' = \boldsymbol{M} \otimes \boldsymbol{F}, \quad (4)$$

where ⊗ denotes Hadamard product.

Compared with most existing attention mechanisms, KAM improves the physical perception ability of the network and can explore more accurate attention distribution by embedding image knowledge which is obtained from human expertise and contains precise spatial information of samples.

Knowledge injection module

Since the RoI from a whole RD map only represents a portion of the radar field-of-view, the network trained with the data will lack the radial velocity, range, and other information of objects in the real world. However, for the same object, the shape or extension in the RD map may vary with its velocity, range, and azimuth relative to the radar sensor. Missing this information can lead to poor classification performance of the network. Therefore, to generate more discriminative features for classification, in KIM, object knowledge is injected into the network by combining with the deep features. Object knowledge includes the velocity, range, azimuth, range profile (P_r), and velocity profile (P_v) of the object in each RoI. These five kinds of information not only offer real-world information about the objects, but also have the capability of classification (Prophet et al., 2018). In this way, the network can improve the overall perception of samples, which is more similar to the cognition of the human brain and can enhance the performance of the semantic classification task.

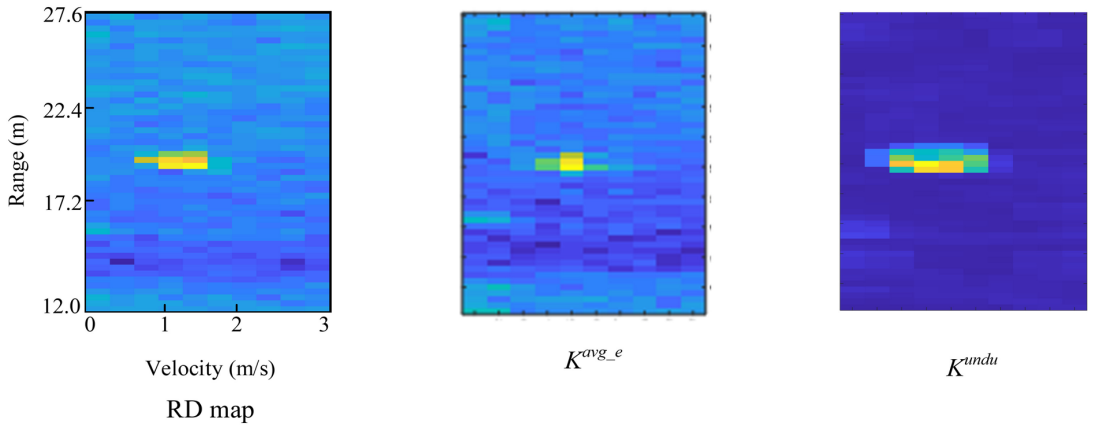


FIGURE 3
K^{avg_e} and K^{undu} extracted from the same region-of-interest (RoI) in an range-Doppler (RD) map.

The velocity, range, and azimuth can be obtained by 3D-FFT. From the RD map obtained by 2D-FFT, the radial range and relative velocity can be captured. Then, FFT is performed on the range-velocity bins to estimate the azimuth. P_r and P_v describe the target extensions in range and velocity dimensions, as shown in Figure 4. P_r and P_v are the maximum length of detected points in range and velocity dimensions of the object, respectively, and can be calculated by:

$$P_r = (r_e - r_s + 1) \cdot \Delta R, \tag{5}$$

$$P_v = (v_e - v_s + 1) \cdot \Delta v, \tag{6}$$

where r_s and r_e denote the starting and ending points detected, ΔR represents the range resolution. In (6), v_s, v_e, and Δv denote the similar meanings in velocity dimension.

The structure of KIM is given in Figure 1, a ConvLSTM (Shi et al., 2015) layer is employed to extract the deep features containing spatiotemporal information. ConvLSTM network is a recurrent structure owing good capability of modeling sequential data and extracting temporal information. Meanwhile, it can learn the spatial information of each individual time step due to the convolution operation. Therefore, considering that the input is a sequence, ConvLSTM network is suitable for extracting deep features from both temporal and spatial dimensions simultaneously.

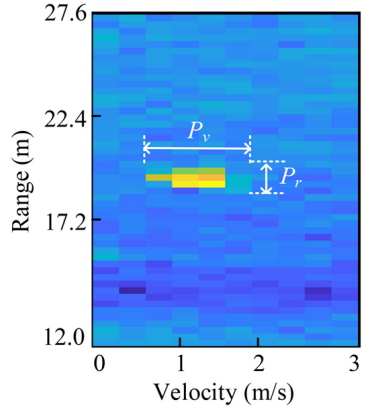


FIGURE 4
The schematic diagram of P_r and P_v.

Then, object knowledge is combined with deep features. Considering that there is a gap between the two features and the same feature may have different contributions in different classification tasks, inspired by squeeze-and-excitation networks (Hu et al., 2018), we adopt an attention-based method to combine object knowledge and deep features in an adaptive way, as shown in Figure 5. Specifically, the deep features and object knowledge are first mapped to the same dimension through an fully connected (FC) layer and scaled to 0∼1 by sigmoid activation, respectively, making them similar and conducive for combination. Then, the mapped features, $\boldsymbol{F}_d$ and $\boldsymbol{F}_{OBK}$, are connected in channel dimension and $\boldsymbol{F}_c$ can be acquired. After that, the global pooling operation is utilized to squeeze $\boldsymbol{F}_c$, and two FC layers are adopted to learn the attention weight vector $\boldsymbol{a}$ containing two elements:

$$\boldsymbol{a} = [a_{OBK}, a_{deep}] = \delta\left(\boldsymbol{W}_2 \cdot \sigma\left(\boldsymbol{W}_1 \cdot AvgPool\left(\boldsymbol{F}_c\right)\right)\right), \tag{7}$$

where a_{deep} and a_{OBK} are the weights of the deep features and object knowledge, respectively, δ and σ are "sigmoid" and "relu" activation functions, $\boldsymbol{W}_2$ and $\boldsymbol{W}_1$ are parameter matrices, $AvgPool\ (\cdot)$ denotes the average pooling operation. Subsequently, object knowledge and the deep features are redefined by multiplying with the corresponding weights. Next, $\boldsymbol{F}_d$ and $\boldsymbol{F}_{OBK}$ are added to their redefined results to preserve original information from different sources. Finally, the

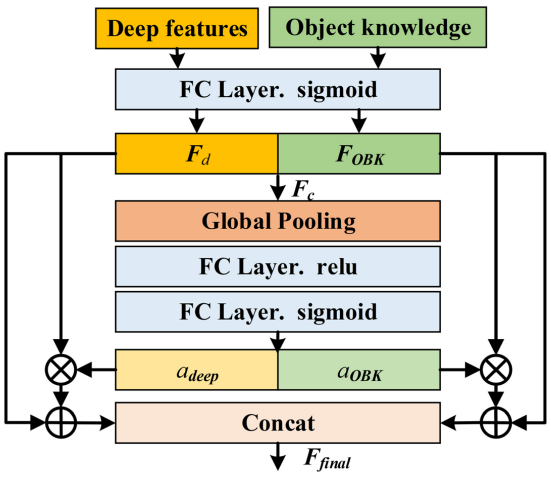


FIGURE 5
Illustration of attention-based injection method (⊗ represents Hadamard product; ⊕ denotes element-wise addition).

concatenated features are used for classification:

$$F_{final} = concat\left(F_d + a_{deep}F_d, \quad F_{OBK} + a_{OBK}F_{OBK}\right), \quad (8)$$

In this module, by injecting object knowledge, the network trains with sufficient information about samples, which improves its learning capability and classification performance.

Experiment

In this section, to evaluate the performance of KANN, we conduct a variety of experiments based on a measured dataset. The dataset is first introduced in detail. Then, the classification performance of KANN is assessed by comparative experiments. Additionally, we analyze the influence of the knowledge assistance and network structure on the performance of KANN through experiments.

Dataset preparation and implementation details

The measured dataset is collected by an automotive MMW multiple-input multiple-output (MIMO) radar with 4 Tx and 8 Rx producing a total of 32 virtual antennas. It uses the Frequency Modulated Continuous Waveform (FMCW) which is widely used in automotive radar (Hu et al., 2019). The specific configurations of radar are provided in **Table 1**.

The radar sensor is assembled and mounted on the front of the car as shown in **Figure 6**. The data is collected under different lighting conditions in different scenarios, such as city streets, elevated roads, and tunnels. Some sample scenarios are shown in **Figure 7**. Four kinds of objects are considered, including pedestrian, runner, vehicle, and cyclist, with overlapping speed ranges.

TABLE 1 The specific configurations of radar.

Parameter	Value
Center frequency and bandwidth	77 GHz and 600 MHz
Maximum range, resolution	80 m, 0.3 m
Maximum radial velocity, resolution	40 m/s, 0.3 m/s
Field of view	−45°∼45°
Number of chirps per frame	64
Number of samples per chirp	256

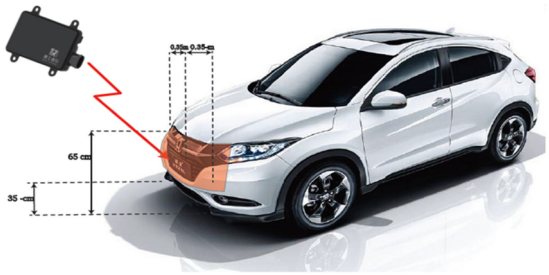


FIGURE 6
Radar installation diagram.

After collecting the original radar echoes, we perform the sequence generation method and knowledge extraction algorithms to obtain the RD map sequences and two kinds of knowledge. It should be noted that in the experiments we stack the RoIs of the same object in the RD maps of five continuous frames to construct an RD map sequence, and the RoIs in different RD map sequences are completely different. There are some samples given in **Figure 8**. Then, the samples are randomly divided into training and testing datasets. The detailed settings are listed in **Table 2**.

Moreover, the implementation details are shown. The experiments are conducted on a server cluster with a 64-bit Linux operating system. In the training phase, the batch size is set to 64, the learning rate is 0.01, and the network is optimized with adaptive moment estimation (Adam) algorithm.

Evaluation metrics

In order to evaluate the performance of different methods, the average accuracy (AA) of all classes is applied. This metric takes into account the imbalance of the data and can provide a more objective assessment of performance. It can be calculated by:

$$\mathrm{AA} = \frac{1}{C}\sum_{c=1}^{C}\frac{N_{TP}}{N_c}, \quad (9)$$

FIGURE 7
Samples of different data collection scenarios.

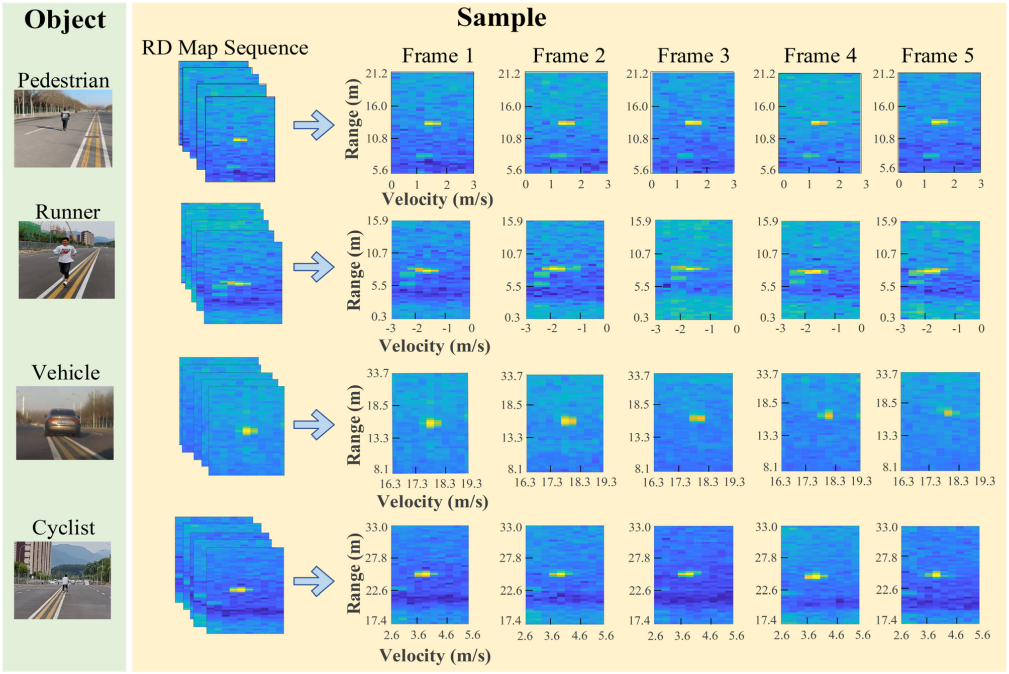


FIGURE 8
Samples of four kinds of objects.

where C is the number of classes, N_{TP} is the number of samples classified correctly in class c, and N_c is the total number of samples in class c.

Quantitative analysis

Object classification

In this part, we assess the performance of KANN based on the measured dataset. Additionally, six methods that have been studied in this field are served as comparisons. Two of them are feature-based methods that extract the features contained in object knowledge and then utilize SVM (Heuel and Rohling, 2012) and random forest (RF) classifiers to predict the object labels, respectively. The remaining four comparison methods establish different neural networks to accomplish the task, containing Three Layer-CNN (TL-CNN) (Patel et al., 2019), VGG-16 (Gao et al., 2019), FCN (Ouaknine et al., 2021), and RadarResNet (Zhang A. et al., 2021).

The per-class accuracy and AA of different methods are shown in **Table 3**. The values in bold in the table are the highest accuracy among the methods. We can observe that KANN

can achieve advanced performance, and its AA and per-class accuracy is all above 90%. These results demonstrate that KANN is effective in the MMW radar object classification task. Since the two kinds of knowledge are obtained from the human brain's wisdom and logic, integrating them can inspire the network to learn the samples in a way more similar to the human brain and extract more discriminative and comprehensive features.

In order to assess the performance of KANN in practical application, the runtime of different methods is analyzed, and the results are listed in Table 4. It can be observed that the feature-based methods cost the shortest time because their input is a set of prepared artificial feature vectors but not raw data. Among the deep learning methods, the runtime of KANN is the longest, which is 2.053 s. It can be inferred that the structure of ConvLSTM contained in KANN costs more time compared with convolution operation due to its serial units. As for the application, in general, it is an acceptable efficiency for the proposed method and its accuracy is the highest.

TABLE 2 The detailed settings of the dataset.

Object	Number of training data	Number of testing data
Pedestrian	840	279
Runner	873	350
Vehicle	1,085	368
Cyclist	714	246
Total	3,512	1,243

TABLE 3 Experimental results on different methods.

Method	Accuracy (%)				
	Pedestrian	Runner	Vehicle	Cyclist	AA
Features+SVM (Villeval et al., 2014)	60.93	76.13	91.70	91.60	80.09
Features+RF	78.49	67.90	91.70	88.55	81.66
TL-CNN (Patel et al., 2019)	83.52	71.19	91.27	90.08	84.02
VGG-16 (Gao et al., 2019)	88.53	79.42	93.01	93.13	88.52
FCN (Ouaknine et al., 2021)	87.81	**92.59**	94.32	94.66	92.35
RadarResNet (Zhang A. et al., 2021)	**94.98**	85.60	**96.51**	90.84	91.98
KANN	93.19	91.35	**96.51**	**95.42**	**93.39**

TABLE 4 Computational costs on different methods.

Model	Runtime (s)
Features+SVM (Villeval et al., 2014)	0.013
Features+RF	0.007
TL-CNN (Patel et al., 2019)	0.081
VGG-16 (Gao et al., 2019)	0.864
FCN (Ouaknine et al., 2021)	0.079
RadarResNet (Zhang A. et al., 2021)	0.293
KANN	2.053

Ablation study

To investigate the advantage of knowledge assistance, we conduct the ablation experiment. The basic network of KANN without knowledge assistance is regarded as the baseline. Then, image knowledge and object knowledge, are separately added to the baseline for assistance. KANN is compared with these three models. Moreover, to assess the structure of KANN, we exchange the positions of KAM and KIM to conduct the experiments. Besides, considering that the spatial information can also be obtained by convolutional operation, in KANN, we remove the image knowledge and apply two convolution kernels with randomly initialized parameters to extract spatial information to compare the effect of artificial image knowledge and the automatically obtained spatial information. The results are listed in Table 5. The values in bold in the table are the highest accuracy among the methods.

It is shown that the baseline performs worst. When one kind of knowledge is added, the accuracy improves. The integration of image knowledge increases AA by 3.63%, and object knowledge makes the network achieve an 8.06% increase in AA. KANN with image knowledge and object knowledge achieves the best performance, whose AA is more than 10% higher than the baseline. Additionally, we can observe that when KIM lies in the front, AA drops by about 3%. As for the comparison of artificial image knowledge and spatial information from convolution operation, it can be seen that the AA of the model with artificial image knowledge is approximately 5% higher than the model with learnable spatial information. It can be inferred that though the network can automatically extract the information, the artificial features can supplement

TABLE 5 Experimental results of the ablation study.

Model	Accuracy (%)				
	Pedestrian	Runner	Vehicle	Cyclist	AA
Baseline	78.92	68.91	87.77	83.97	79.89
Baseline + image knowledge	79.93	73.25	90.83	90.08	83.52
Baseline + object knowledge	81.83	86.83	90.39	92.76	87.95
Baseline + image knowledge + object knowledge (KANN)	93.19	**91.35**	**96.51**	**95.42**	**93.39**
KIM + KAM	83.87	81.84	90.83	92.37	90.53
KANN with convolution operation	**98.57**	84.77	87.77	82.44	88.38

the network with physical and discriminative information which the network lacks.

From the results, we can conclude that with the knowledge assistance, the network is inspired to learn sample information no longer solely by optimizing data. It can explore information in a way more like the human brain. Although an attention mechanism is built to help the network focus on the object regions first, it is trained by the network learning mechanism. Image knowledge contains sample spatial information, which is acquired based on human brain wisdom and logic. By introducing image knowledge into the attention mechanism, the network can assign attention not only based on the network learning results, but also according to the visual cognition of the human brain. As a result, the network concentrates more precisely on the object region and achieves more accurate classification performance. As for object knowledge, it provides semantic information about objects in the real world, which is in line with the human brain perception when classifying objects. Adding this information supplements more information for the network and can improve the accuracy. Besides, by contrast, object knowledge plays a more significant role in KANN. It can be inferred that object knowledge offers the semantic information which the network lacks, while image knowledge modifies the attention weight matrix. On the other hand, from the module location experiment, it can be inferred that the network can achieve better performance by focusing on the object regions in the early stage and delicately combining object knowledge and deep features through the attention-based method just before classification.

Comparison of combination methods

In this part, to evaluate the performance of the attention-based combination method in KIM, different combination methods are applied to the fusion of object knowledge and deep features, and the results are discussed. Two common approaches, including concatenation and element-wise addition are served as the comparison methods. The results are listed in Table 6, we can observe that three combination methods can all achieve good results with AA all above 90%. The values in bold in the table are the highest accuracy among the methods. The experiment proves that the object knowledge definitely supplements physical and discriminative information for the network and improves the classification performance. Among them, the attention-based method performs best because the network can adaptively assign weights of different features, and the features generated are more suitable for this classification task. As for the other two, they are just the simple combination of the two features, and their accuracy is lower than our adaptive attention-based combination method.

TABLE 6 Experimental results of different combination methods.

Combination method	Accuracy (%)				
	Pedestrian	Runner	Vehicle	Cyclist	AA
Concatenation	90.32	90.53	92.14	93.13	91.53
Element-wise addition	89.96	86.83	94.76	90.84	90.59
Attention-based method	**93.19**	**91.35**	**96.51**	**95.42**	**93.39**

Conclusion

In this paper, we propose a knowledge-assisted network KANN based on RD map sequence for automotive MMW radar object classification. We introduce two kinds of prior knowledge to help the network learn information from samples in a way more similar to the human brain. In this way, the neural network can generate more discriminative features for semantic classification tasks. Specifically, image knowledge helps the network more accurately focus on the object regions. Object knowledge is fused with the deep feature from the network to provide more comprehensive information for classification. To effectively combine the two aspects of information, an attention-based injection method is employed to achieve the adaptive combination. Experiments based on measured data of four classes of objects verify the effectiveness of KANN and demonstrate that knowledge assistance can improve the performance of the network. Our research is continuing, and the data in more complex traffic scenarios, e.g., the crowded situation and strong interference conditions, is still being collected and processed. Since some researches show that introducing knowledge into the network can mitigate the network data size dependence (Zhang L. et al., 2021; Huang et al., 2022), in future research, based on our expanded dataset, we will conduct further experiments to assess the effect of knowledge injection with the training data size as the main topic. Simultaneously, the practical application value will be further evaluated with the data in more complex traffic scenarios.

Data availability statement

The datasets presented in this article are not readily available because the dataset is part of ongoing work. Requests to access the datasets should be directed to LZ, zhangliang@bit.edu.cn.

Author contributions

YW first proposed the idea that introducing prior knowledge into neural networks for assistance, participated in the construction of KANN, analyzed

the effectiveness of KANN, and wrote the original manuscript. CH established the network, carried out the experiments based on the measured dataset, and participated in the writing of the original manuscript. LZ organized a study in the field of automotive MMW radar object classification based on knowledge assistance, proposed the detailed framework of KANN, investigated the feasibility of the method, and reviewed the manuscript and made valuable suggestions. JL constructed the measured dataset, arranged the data collection experiments, and reviewed the manuscript. QA and FY conducted the automotive MMW radar data collection experiments. All authors contributed to the article and approved the submitted version.

Funding

This work was supported by the National Key R&D Program of China (Grant Nos. 2018YFE0202102 and 2018YFE0202103), the China Postdoctoral Science Foundation (Grant No. 2021M690412), the Natural Science Foundation of Chongqing, China (Grant No. cstc2020jcyj-msxmX0812), and project ZR2021MF134 supported by the Shandong Provincial Natural Science Foundation.

Conflict of interest

JL and QA were employed by Beijing Rxbit Electronic Technology Co., Ltd.

The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

References

Alsubai, S., Khan, H. U., Alqahtani, A., Sha, M., Abbas, S., and Mohammad, U. G. (2022). Ensemble deep learning for brain tumor detection. *Front. Comput. Neurosci.* 16:1005617. doi: 10.3389/fncom.2022.1005617

Angelov, A., Robertson, A., Murray-Smith, R., and Fioranelli, F. (2018). Practical classification of different moving targets using automotive radar and deep neural networks. *IET Radar Sonar Navig.* 12, 1082–1089. doi: 10.1049/iet-rsn.2018.0103

Bijelic, M., Gruber, T., and Ritter, W. (2018). "A benchmark for lidar sensors in fog: Is detection breaking down?," in *Proceedings of the 2018 IEEE intelligent vehicles symposium (IV)* (Piscataway, NJ: IEEE), 760–767. doi: 10.1109/IVS.2018.8500543

Chen, Y., and Zhang, D. (2022). Integration of knowledge and data in machine learning. *arXiv* [preprint] arXiv: 2202.10337v1

Cornelio, P., Haggard, P., Hornbaek, K., Georgiou, O., Bergström, J., Subramanian, S., et al. (2022). The sense of agency in emerging technologies for human–computer integration: A review. *Front. Neurosci.* 16:949138. doi: 10.3389/fnins.2022.949138

Danelljan, M., Khan, F. S., Felsberg, M., and Van De Weijer, J. (2014). "Adaptive color attributes for real-time visual tracking," in *Proceedings of the 2014 IEEE conference on computer vision and pattern recognition (CVPR)* (Piscataway, NJ: IEEE), 1090–1097. doi: 10.1109/CVPR.2014.143

Deng, C., Jing, D., Han, Y., Wang, S., and Wang, H. (2022). FAR-Net: Fast anchor refining for arbitrary-oriented object detection. *IEEE Geosci. Remote Sens. Lett.* 19, 1–5. doi: 10.1109/LGRS.2022.3144513

Felguera-Martin, D., Gonzalez-Partida, J.-T., Almorox-Gonzalez, P., and Burgos-García, M. (2012). Vehicular traffic surveillance and road lane detection using radar interferometry. *IEEE Trans. Veh. Technol.* 61, 959–970. doi: 10.1109/TVT.2012.2186323

Feng, Z., Zhang, S., Kunert, M., GmbH, R. B., and Wiesbeck, W. (2019). "Point cloud segmentation with a high-resolution automotive radar," in *Proceedings of the AmE 2019–automotive meets electronics; 10th GMM-Symposium* (Piscataway, NJ: IEEE), 1–5.

Gao, X., Xing, G., Roy, S., and Liu, H. (2019). "Experiments with mmWave automotive radar test-bed," in *Proceedings of the 53rd Asilomar Conference on Signals, Systems, and Computers, Pacific Grove, CA, USA* (Piscataway, NJ: IEEE), 1–6. doi: 10.1109/IEEECONF44664.2019.9048939

Han, Y., Deng, C., Zhao, B., and Zhao, B. (2019b). Spatial-temporal context-aware tracking. *IEEE Signal Process. Lett.* 26, 500–504. doi: 10.1109/LSP.2019.2895962

Han, Y., Deng, C., Zhao, B., and Tao, D. (2019a). State-Aware anti-drift object tracking. *IEEE Trans. Image Process.* 28, 4075–4086. doi: 10.1109/TIP.2019.2905984

Held, P., Steinhauser, D., Kamann, A., Koch, A., Brandmeier, T., and Schwarz, U. T. (2019). "Normalization of micro-doppler spectra for cyclists using high-resolution projection technique," in *Proceedings of the 2019 IEEE international conference on vehicular electronics and safety (ICVES)* (Piscataway, NJ: IEEE), 1–6. doi: 10.1109/ICVES.2019.8906495

Heuel, S., and Rohling, H. (2011). "Two-Stage pedestrian classification in automotive radar systems," in *Proceedings of the 2011 12th international radar symposium (IRS)* (Piscataway: IEEE), 8.

Heuel, S., and Rohling, H. (2012). "Pedestrian classification in automotive radar systems," in *Proceedings of the 2012 13th international radar symposium (IRS)* (Piscataway: IEEE), 39–44. doi: 10.1109/IRS.2012.6233285

Hu, J., Shen, L., and Sun, G. (2018). "Squeeze-and-excitation networks," in *Proceedings of the 2018 IEEE/CVF conference on computer vision and pattern recognition* (Piscataway, NJ: IEEE), 7132–7141. doi: 10.1109/CVPR.2018.00745

Hu, X., Li, Y., Lu, M., Wang, Y., and Yang, X. (2019). A multi-carrier-frequency random-transmission chirp sequence for TDM MIMO automotive radar. *IEEE Trans. Veh. Technol.* 68, 3672–3685. doi: 10.1109/TVT.2019.2900357

Huang, Z., Yao, X., Liu, Y., Dumitru, C. O., Datcu, M., and Han, J. (2022). Physically explainable CNN for SAR image classification. *arXiv* [preprint] arXiv: 2110.14144v1

Kim, J. U., and Ro, Y. M. (2019). "Attentive layer separation for object classification and object localization in object detection," in *Proceedings of the 2019 IEEE international conference on image processing (ICIP)* (Piscataway, NJ: IEEE), 3995–3999. doi: 10.1109/ICIP.2019.8803439

Kuroda, N., Ikeda, K., and Teramoto, W. (2022). Visual self-motion information contributes to passable width perception during a bike riding situation. *Front. Neurosci.* 16:938446. doi: 10.3389/fnins.2022.938446

Lindsay, G. W. (2020). Attention in psychology, neuroscience, and machine learning. *Front. Comput. Neurosci.* 14:29. doi: 10.3389/fncom.2020.00029

Liu, A., Wang, F., Xu, H., and Li, L. (2018). N-SAR: A new multichannel multimode polarimetric airborne SAR. *IEEE J. Selected Top. Appl. Earth Observ. Remote Sens.* 11, 3155–3166. doi: 10.1109/JSTARS.2018.2848945

Liu, Q., Zhang, X., Liu, Y., Huo, K., Jiang, W., and Li, X. (2021). Multi-polarization fusion few-shot HRRP target recognition based on meta-learning framework. *IEEE Sens. J.* 21, 18085–18100. doi: 10.1109/JSEN.2021.3085671

Major, B., Fontijne, D., Ansari, A., Sukhavasi, R. T., Gowaikar, R., Hamilton, M., et al. (2019). "Vehicle detection with automotive radar using deep learning on range-azimuth-doppler tensors," in *Proceedings of the 2019 IEEE/CVF international conference on computer vision workshop (ICCVW), Seoul, Korea (South)* (Piscataway, NJ: IEEE), 924–932. doi: 10.1109/ICCVW.2019.00121

Marblestone, A. H., Wayne, G., and Kording, K. P. (2016). Toward an integration of deep learning and neuroscience. *Front. Comput. Neurosci.* 10:94. doi: 10.3389/fncom.2016.00094

Munoz-Ferreras, J. M., Perez-Martinez, F., Calvo-Gallego, J., Asensio-Lopez, A., Dorta-Naranjo, B. P., and Blanco-del-Campo, A. (2008). Traffic surveillance system based on a high-resolution radar. *IEEE Trans. Geosci. Remote Sens.* 46, 1624–1633. doi: 10.1109/TGRS.2008.916465

Nam, H., and Han, B. (2016). "Learning multi-domain convolutional neural networks for visual tracking," in *Proceedings of the 2016 IEEE conference on computer vision and pattern recognition (CVPR)* (Piscataway, NJ: IEEE), 4293–4302. doi: 10.1109/CVPR.2016.465

Ouaknine, A., Newson, A., Rebut, J., Tupin, F., and Pérez, P. (2021). "CARRADA dataset: Camera and automotive radar with range-angle-doppler annotations," in *Proceedings of the 25th international conference on pattern recognition (ICPR)* (Milan), 5068–5075. doi: 10.1109/ICPR48806.2021.9413181

Palffy, A., Dong, J., Kooij, J. F. P., and Gavrila, D. M. (2020). CNN based road user detection using the 3D radar cube. *IEEE Robot. Autom. Lett.* 5, 1263–1270. doi: 10.1109/LRA.2020.2967272

Patel, K., Rambach, K., Visentin, T., Rusev, D., Pfeiffer, M., and Yang, B. (2019). "Deep learning-based object classification on automotive radar spectra," in *Proceedings of the 2019 IEEE radar conference, Boston, MA, USA* (Piscataway, NJ: IEEE), 1–6. doi: 10.1109/RADAR.2019.8835775

Prophet, R., Hoffmann, M., Vossiek, M., Sturm, C., Ossowska, A., Malik, W., et al. (2018). "Pedestrian classification with a 79 GHz automotive radar sensor," in *Proceedings of the 2018 19th international radar symposium (IRS)* (Piscataway, NJ: IEEE), 1–6. doi: 10.23919/IRS.2018.8448161

Prophet, R., Li, G., Sturm, C., and Vossiek, M. (2019). "Semantic segmentation on automotive radar maps," in *Proceedings of the 2019 IEEE intelligent vehicles symposium (IV)* (Piscataway, NJ: IEEE), 756–763. doi: 10.1109/IVS.2019.8813808

Qi, C. R., Liu, W., Wu, C., Su, H., and Guibas, L. J. (2018). "Frustum PointNets for 3D object detection from RGB-D data," in *Proceedings of the 2018 IEEE/CVF conference on computer vision and pattern recognition (CVPR)* (Piscataway, NJ: IEEE), 918–927. doi: 10.1109/CVPR.2018.00102

Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. (2016). "You only look once: Unified, real-time object detection," in *Proceedings of the 2016 IEEE conference on computer vision and pattern recognition (CVPR)* (Piscataway, NJ: IEEE), 779–788. doi: 10.1109/CVPR.2016.91

Ren, S., He, K., Girshick, R., and Sun, J. (2017). Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.* 39, 1137–1149. doi: 10.1109/TPAMI.2016.2577031

Rohling, H., Heuel, S., and Ritter, H. (2010). "Pedestrian detection procedure integrated into an 24 GHz automotive radar," in *Proceedings of the 2010 IEEE radar conference* (Piscataway, NJ: IEEE), 1229–1232. doi: 10.1109/RADAR.2010.5494432

Shi, S., Wang, X., and Li, H. (2019). "PointRCNN: 3D object proposal generation and detection from point cloud," in *Proceedings of the 2019 IEEE/CVF conference on computer vision and pattern recognition (CVPR)* (Piscataway, NJ: IEEE), 770–779. doi: 10.1109/CVPR.2019.00086

Shi, S., Wang, Z., Shi, J., Wang, X., and Li, H. (2021). From points to parts: 3D object detection from point cloud with part-aware and part-aggregation network. *IEEE Trans. Pattern Anal. Mach. Intell.* 43, 2647–2664. doi: 10.1109/TPAMI.2020.2977026

Shi, X., Chen, Z., Wang, H., Yeung, D.-Y., Wong, W., and Woo, W. (2015). Convolutional LSTM network: A machine learning approach for precipitation nowcasting. *arXiv* [preprint] arXiv: 1506.04214v1

Shirakata, N., Iwasa, K., Yui, T., Yomo, H., Murata, T., and Sato, J. (2019). "Object and direction classification based on range-doppler map of 79 GHz MIMO radar using a convolutional neural network," in *Proceedings of the 2019 12th global symposium on millimeter waves (GSMM)* (Piscataway, NJ: IEEE), 1–3. doi: 10.1109/GSMM.2019.8797649

Simonyan, K., and Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. *arXiv* [preprint] arXiv:1409.1556

Smeulders, A. W. M., Chu, D. M., Cucchiara, R., Calderara, S., Dehghan, A., and Shah, M. (2014). Visual tracking: An experimental survey. *IEEE Trans. Pattern Anal. Mach. Intell.* 36, 1442–1468. doi: 10.1109/TPAMI.2013.230

Soenksen, L. R., Ma, Y., Zeng, C., Boussioux, L., Villalobos Carballo, K., Na, L., et al. (2022). Integrated multimodal artificial intelligence framework for healthcare applications. *NPJ Digit. Med.* 5, 149. doi: 10.1038/s41746-022-00689-4

Tang, L., Tang, W., Qu, X., Han, Y., Wang, W., and Zhao, B. (2022). A scale-aware pyramid network for multi-scale object detection in SAR images. *Remote Sens.* 14, 973. doi: 10.3390/rs14040973

van Dyck, L. E., Denzler, S. J., and Gruber, W. R. (2022). Guiding visual attention in deep convolutional neural networks based on human eye movements. *Front. Neurosci.* 16:975639. doi: 10.3389/fnins.2022.975639

Villeval, S., Bilik, I., and Gürbuz, S. Z. (2014). "Application of a 24 GHz FMCW automotive radar for urban target classification," in *Proceedings of the 2014 IEEE radar conference* (Piscataway, NJ: IEEE), 1237–1240. doi: 10.1109/RADAR.2014.6875787

Wang, J., Zheng, T., Lei, P., and Bai, X. (2018). Ground target classification in noisy SAR images using convolutional neural networks. *IEEE J. Selected Top. Appl. Earth Observ. Remote Sens.* 11, 4180–4192. doi: 10.1109/JSTARS.2018.2871556

Wang, Y., Jiang, Z., Li, Y., Hwang, J.-N., Xing, G., and Liu, H. (2021). RODNet: A real-time radar object detection network cross-supervised by camera-radar fused object 3D localization. *IEEE J. Sel. Top. Signal Process.* 15, 954–967. doi: 10.1109/JSTSP.2021.3058895

Zhang, A., Nowruzi, F. E., and Laganiere, R. (2021). "RADDet: Range-azimuth-doppler based radar object detection for dynamic road users," in *Proceedings of the 18th conference on robots and vision (CRV)* (Piscataway, NJ: IEEE), 95–102. doi: 10.1109/CRV52889.2021.00021

Zhang, L., Han, C., Wang, Y., Li, Y., and Long, T. (2021). Polarimetric HRRP recognition based on feature-guided Transformer model. *Electron. Lett.* 57, 705–707. doi: 10.1049/ell2.12225

Zhang, L., Leng, X., Feng, S., Ma, X., Ji, K., Kuang, G., et al. (2022). Domain knowledge powered two-stream deep network for few-shot SAR vehicle recognition. *IEEE Trans. Geosci. Remote Sens.* 60, 1–15. doi: 10.1109/TGRS.2021.3116349

Zhao, Z., Han, Y., Xu, T., Li, X., Song, H., and Luo, J. (2017). A reliable and real-time tracking method with color distribution. *Sensors* 17:2303. doi: 10.3390/s17102303

Zhao, Z., Song, Y., Cui, F., Zhu, J., Song, C., Xu, Z., et al. (2020). Point cloud features-based kernel SVM for human-vehicle classification in millimeter wave radar. *IEEE Access* 8, 26012–26021. doi: 10.1109/ACCESS.2020.2970533

Zhu, F., Lv, Y., Chen, Y., Wang, X., Xiong, G., and Wang, F.-Y. (2020). Parallel transportation systems: Toward IoT-enabled smart urban traffic control and management. *IEEE Trans. Intell. Transport. Syst.* 21, 4063–4071. doi: 10.1109/TITS.2019.2934991

Zhu, J., Su, H., and Zhang, B. (2020). Toward the third generation of artificial intelligence. *Sci. Sin. Inf.* 50:1281. doi: 10.1360/SSI-2020-0204

Zhu, L., Yu, F. R., Wang, Y., Ning, B., and Tang, T. (2019). Big data analytics in intelligent transportation systems: A survey. *IEEE Trans. Intell. Transport. Syst.* 20, 383–398. doi: 10.1109/TITS.2018.2815678

frontiers | Frontiers in Neuroscience

TYPE Original Research
PUBLISHED 28 December 2022
DOI 10.3389/fnins.2022.1070645

OPEN ACCESS

EDITED BY
Chenwei Deng,
Beijing Institute of Technology, China

REVIEWED BY
Yunkai Li,
Tianjin University, China
Ling Weng,
Hebei University of Technology, China

*CORRESPONDENCE
Huaping Liu
hpliu@tsinghua.edu.cn

SPECIALTY SECTION
This article was submitted to Perception Science, a section of the journal Frontiers in Neuroscience

RECEIVED 15 October 2022
ACCEPTED 21 November 2022
PUBLISHED 28 December 2022

CITATION
Zheng W, Liu H, Guo D and Sun F (2022) Robust tactile object recognition in open-set scenarios using Gaussian prototype learning. *Front. Neurosci.* 16:1070645. doi: 10.3389/fnins.2022.1070645

Robust tactile object recognition in open-set scenarios using Gaussian prototype learning

Wendong Zheng[1,2], Huaping Liu[1,2]*, Di Guo[1,2] and Fuchun Sun[1,2]

[1]Department of Computer Science and Technology, Tsinghua University, Beijing, China, [2]State Key Laboratory of Intelligent Technology and Systems, Beijing National Research Center for Information Science and Technology, Tsinghua University, Beijing, China

Tactile object recognition is crucial for effective grasping and manipulation. Recently, it has started to attract increasing attention in robotic applications. While there are many works on tactile object recognition and they also achieved promising performances in some applications, most of them are usually limited to closed world scenarios, where the object instances to be recognition in deployment are known and the same as that of during training. Since robots usually operate in realistic open-set scenarios, they inevitably encounter unknown objects. If automation systems falsely recognize unknown objects as one of the known classes based on the pre-trained model, it can lead to potentially catastrophic consequences. It motivates us to break the closed world assumption and to study tactile object recognition in realistic open-set conditions. Although several open-set recognition methods have been proposed, they focused on visual tasks and may not be suitable for tactile recognition. It is mainly due to that these methods do not take into account the special characteristic of tactile data in their models. To this end, we develop a novel Gaussian Prototype Learning method for robust tactile object recognition. Particularly, the proposed method converts feature distributions to probabilistic representations, and exploit uncertainty for tactile recognition in open-set scenarios. Experiments on the two tactile recognition benchmarks demonstrate the effectiveness of the proposed method on open-set tasks.

KEYWORDS

tactile perception, object recognition, open-set recognition, Gaussian prototype learning, tactile object recognition

1. Introduction

Object recognition is a prerequisite for robotic dexterous manipulations, which is the cornerstone of many robotic applications (Li et al., 2018; Qiao et al., 2021). For example, a robot needs to know the category of an object for selecting a suitable interaction pattern or manipulation strategy during exploring the surroundings or performing manipulation (He et al., 2020; Zheng et al., 2020a). Therefore, how to effectively realize object recognition has recently attracted widespread attention in robotic research fields.

Since tactile sensing is an effective way of perceiving some physical properties of the manipulated objects through physical interaction (Luo et al., 2017), it has been extensively used in robotic tasks involving object recognition, material identification (Zheng et al., 2019), texture recognition and robotic grasp detection (Guo et al., 2021). Liu and Sun (2017) proposed a tactile recognition method for classify material identification. Xu et al. (2013) proposed a tactile identification method with Bayesian exploration. Kerr et al. (2018) used tactile data to classify the materials with the BioTAC sensor. In addition, tactile information is used as an effective complement of visual information for robotic tasks. In Liu et al. (2016), a novel visual-tactile fusion method was proposed for object recognition using joint group kernel sparse coding. Guo et al. (2017) adopted tactile information as an important complement of visual information for the robotic grasp detection task. These works have shown that tactile perception plays a significant role in robotic recognition tasks.

While there are many works on tactile recognition and they have been demonstrated to be effective for some specific applications (Yi et al., 2020), they have mainly focused on constructing predictive models to classify predefined and fixed object classes in closed-set scenarios, assuming that the classes seen in testing must have appeared in training. In fact, such an assumption is usually violated in actual robotic applications (Zheng et al., 2020b). This is mainly due to that robots are commonly deployed in realistic unconstrained environments, where objects of unknown classes are regularly encountered. When observing an unknown object, these closed-set classification methods incorrectly categorize it as one of the known classes with high confidence. As classifier prediction in robotic applications can trigger some kind of costly robotic action, such misclassification can be catastrophic and is often not acceptable. Thus, it is necessary to investigate robust tactile recognition in open-set scenarios, which is also referred to open-set tactile recognition. The schematic is shown in Figure 1, where robots should have the dual ability of unknown detection and known classification.

To the best of our knowledge, tactile object recognition of open-set scenarios is still unexplored research in the robotic field. Similar to other open-set recognition, open-set tactile recognition also faces the core challenge of how to not only correctly classify samples from the known classes but also effectively detect and flag unknown examples as the novel. Many methods have been proposed to handle this problem in the literature. The mainstream methods attempt to utilize thresholding to reformulate open-set recognition as a closed-set classifier. As feature distribution of training samples is not explicitly considered in their learning objectives, the learned features generally have excessive intra-class variance. The inter-class distance can even be smaller than the intra-class distance in the learned feature space. This makes it difficult to set an appropriate threshold that well separates known from unknown.

In addition, another technical solution aims to collect unknown samples for training a $(K + 1)$-class model, where K is the number of known categories and all unknowns are treated as an additional category. The strategy is simple and intuitive, but it usually requires large-scale training data to represent the large numbers of unknowns in open scenarios. However, collecting sufficient tactile data is difficult for training due to the complex collection process and constraints of robot-object physical interactions. Hence, constructing an effective model for open-set tactile recognition is still an open question.

As we know, humans can effectively recognize objects in open environments based on template or prototype matching. Motivated by the recognition mechanism, we propose an uncertainty estimation model for open-set tactile object recognition in this work. The framework consists of two main components, which are the feature extractor and the class prototypes. The feature extractor simulates the perception system of humans for transforming the raw sensing data into abstract representations. Moreover, the prototypes for each category serve as abstract memories of the corresponding category in the brain. By matching the tactile features (abstract representation) with prototypes (classes memories), the proposed model performs object recognition. During inference, if the feature of a test tactile sample can not match well with all the prototypes of the known classes, it will be considered as the unknown.

To this end, the learned features of each known class are characterized by a Gaussian distribution in our framework. As known samples follow the prior distributions, those test samples located in low probability regions will be recognized as unknown by the model. Meanwhile, for the test samples from known classes, the model will compute its probabilities over all known classes and classify it as the class with the highest probability. To explicitly enforce training samples following Gaussian distributions, we introduce a likelihood regularization term to the classification discriminant function during training. In addition, we further add a classification margin to make each cluster more compact and further improve the generalization of the model. The main contributions are summarized as follows:

1. In this paper, we specifically address tactile object recognition in open-set scenarios. To this end, a novel Gaussian Prototype Learning method is proposed, which is suitable for both unknown detection and known classification.
2. We introduce a likelihood regularization term to explicitly enforce training samples following Gaussian distributions. In addition, we further introduce a classification margin to make each cluster more compact, which is more beneficial for unknown detection.
3. We perform comprehensive experimental evaluations of our proposed method on publicly available tactile datasets. The

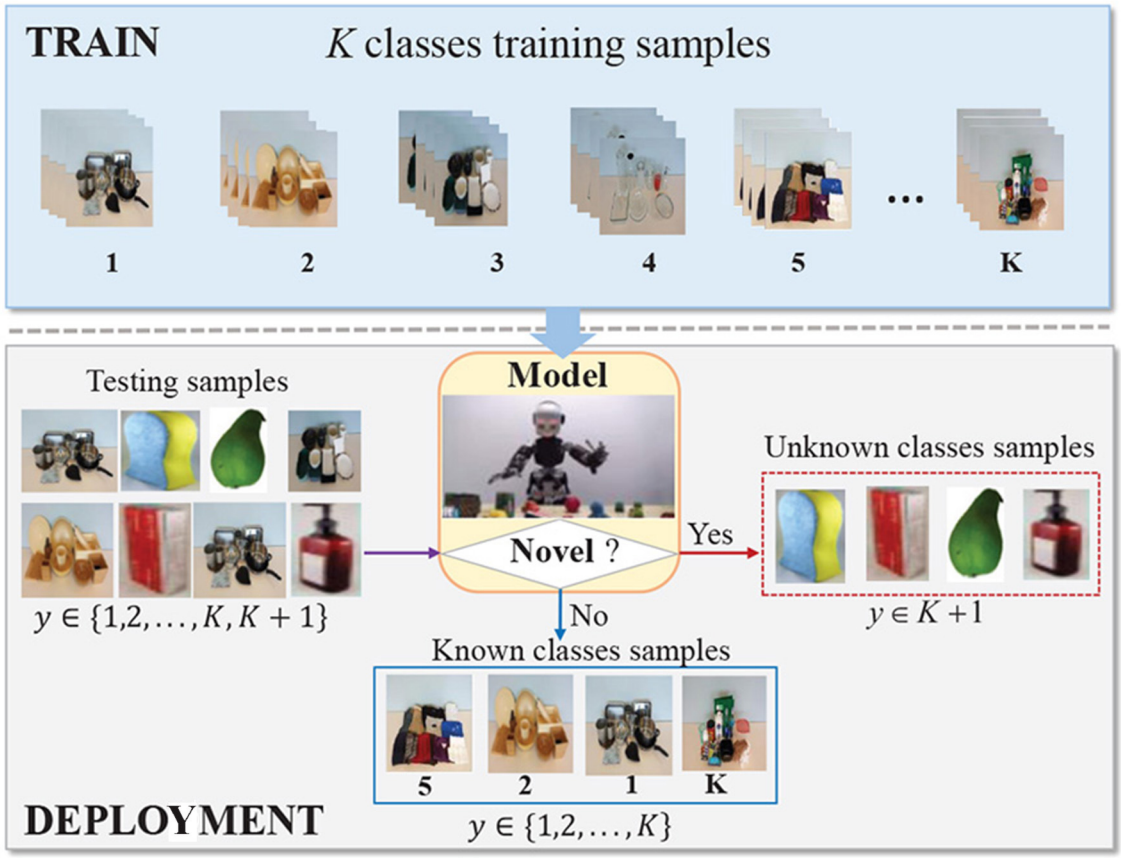


FIGURE 1
The schematic of open-set tactile object recognition. Some images are from https://sites.gatech.edu/hrl/mr-gan/.

experimental results demonstrate the effectiveness of the proposed method.

Please note that our proposed open-set tactile recognition is not just a matter of the robot filling in gaps in its knowledge base. Instead, we aim to enable robots will be able to continually expand the scope of the knowledge to learn new unknown classes over time in an active learning manner. That is to say, at any particular point in time the model needs to be able to detect and reject unseen data belonging to unknown tasks or classes. These unknown data could be utilized and learned with another algorithm in some human-in-the-loop system at a later stage. We believe that this research will aid in active learning and continual learning in open-set conditions, which can serve as the first step toward building lifelong robot tactile recognition systems.

In the following, related works are briefly reviewed in Section 2. In Section 3, we describe the problem of tactile open-set object recognition. Section 4 details the framework architecture and learning model of the proposed method. The experimental results and analysis are given in Section 5.

2. Related work

In this section, the main related works are briefly reviewed from two aspects: *tactile object recognition* and *open-Set Recognition*.

2.1. Tactile object recognition

Object recognition is a fundamental perceptual capability for many robot applications (Meyer et al., 2020). While vision enables robots to have excellent visual recognition capabilities (Deng et al., 2019; Han et al., 2019), it is not always effective for object recognition in practical tasks (Yang et al., 2019). This is mainly due to that objects of similar appearance can have very different physical properties, which may not be easily obtained visually (Deng et al., 2022). Tactile sensing is an important perception modality, of which the interactive nature allows it to convey rich and diverse tactile information, such as texture, roughness, or stiffness (Li et al., 2020). It is crucial for robots to explore and learn the mechanical properties of manipulated

objects, especially when interacting with unknown objects in practical environments.

Considering its effectiveness in perceiving environments, tactile information has been extensively adopted in a variety of robot recognition tasks. Liu and Sun (2017) proposed a tactile material recognition model with semantic labels, which improved the identification performance. Kerr et al. (2018) utilized BioTAC sensor to collect tactile data, and then these data are used to classify the materials. Yuan et al. (2018) used GelSight tactile sensor to recognize 11 properties of the clothes, which aim to help the robot understand their material properties. Based on a hybrid touch approach, Taunyazov et al. (2019) developed an effective tactile identification framework for texture classification. More recently, Gu et al. (2020) proposed an event-based tactile object recognition method with a spiking graph neural network using electronic skins.

Although the mentioned tactile-based recognition methods have been successfully applied in some specific robotic tasks, most of them are deployed under a closed-set condition. Such a closed-set scenario is practically unfeasible in robotic applications. Robots commonly are deployed in open environments, where they will often come across new types of objects. Recently, Abderrahmane et al. (2018, 2019) proposed a tactile recognition framework, which can recognize both known as well as novel objects. Nevertheless, this framework still did not explicitly consider the nature of open-set. In particular, the set of novel classes that can be recognized must be known in advance in the framework. Moreover, it relied on the hypothesis that attributes learned from the training seen-classes are shared by the testing unseen-classes. Obviously, they are potential drawbacks in practice applications. Consequently, existing methods are not suitable for open-set tactile object recognition.

2.2. Open-set recognition

Open-set tactile recognition faces the core challenge is how to not only correctly classify samples from the known classes but also effectively detect and flag unknown examples as the novel. Traditional closed-set classification models may not work in open-set problems because they often predict high confidence for inputs that are significantly different from the training classes (Wang et al., 2022). To tackle this challenge, a variety of related methods have been proposed in the literature. An intuitive method is to use closed-set classifier to solve open-set recognition by setting rejection threshold, such as 1-vs-set SVM (Scheirer et al., 2012), SROSR (Zhang and Patel, 2016), NNO (Bendale and Boult, 2015), DOC (Shu et al., 2017), and CROSR (Dhamija et al., 2018). Exploring this idea, Scheirer et al. (2012) proposed 1-vs-Set model based on SVM to detect unknown samples by adding an extra hyper-line. Bendale and Boult (2015) extended Nearest Class Mean (NCM) classifier to open-set conditions, establishing a Nearest Non-Outlier (NNO) algorithm. Recently, Bendale and Boult (2016) proposed to use the Openmax layer to replace the Softmax layer in deep neural networks. This method redistributes the probability distribution of Softmax to obtain the class probability of unknown samples.

As most of these models ignore constructing reasonable feature distribution for different classes, the learned features generally have excessive intra-class variance (Han et al., 2017). The inter-class distance can even be smaller than the intra-class distance in the learned feature space. As a consequence, it is hard to select an appropriate threshold that well separates known from unknowns. Moreover, feature distribution of training samples is not explicitly considered in their learning objectives, which will limit the performance of the model to detect unknown samples.

Another technical route is to collect or synthesize examples of extra classes for representing unknowns. Along this line, G-OpenMax (Ge et al., 2017) proposed to train a generator for synthesizing examples that represent all unknown classes for model training. Neal et al. (2018) developed counterfactual image generation, which aimed to generate extra class image samples that cannot be classified into any known class. Since the complex collection process and operation constraints, it is difficult to acquire large amounts of tactile data for unknown. Therefore, it is unfeasible to learn an effective model with limited training data for generating sufficient samples to represent unknowns.

3. Problem formulation

In this work, we aim to realize robotic tactile object recognition in open-set scenarios. The goal is to endow robots with an effective mechanism to detect samples from unknown classes that may be encountered during testing, which are not available to be seen in training. To accomplish this goal, the tactile open-set recognition model is able to (i) correctly classify known tactile inputs (i.e., classes from the training set) and (ii) effectively detect unknown tactile classes (i.e., classes not exposed in the training set).

Let us formalize the problem described above. Given a tactile training dataset $D_{tr} = \{(t_i, y_i)\}_{i=1}^{M}$, where $t_i \in R^d$ denotes a training tactile sample, $y_i \in Y = \{1, 2, ..., K\}$ is the corresponding class label and M denotes the number of training samples. The testing dataset $D_{te} = \{(t_j, y_j)\}_{i=1}^{N}$ where $t_j \in R^d$, $y_j \in Y' = \{1, 2, ..., K, K+1, ..., k'\}$ $(k' > K)$ and N is the number of testing samples. Here, $\{k+1, ..., k'\}$ denotes the set of unknown categories, which is referred to as novelty and uniformly denoted as Y_{K+1} in this paper. Therefore, $Y' = Y \cup Y_{K+1}$ and $Y \cap Y_{K+1} = \varnothing$. Our task is that the tactile recognition system need to determine whether a tactile observation $t_j \in Y'$ is from the known classes Y or the unknown classes Y_{K+1}. If t_j is from Y, the classifier should predict a class label $\hat{y} \in Y$, otherwise it can be judged as the novel class Y_{K+1}.

The primary challenge of solving this problem is how to enable the model to classify tactile examples of seen classes into their respective classes and meantime detect tactile data of unseen classes. Traditional classifiers predict the class of the input instance with the highest Softmax probability. Since the model is impossible to know in advance unknown classes that may be encountered in practice, it tends to predict the lowest probability on the unknown classes. As a consequence, directly using closed-set classifiers for open-set recognition would classify unknown instances into known categories with improperly high confidence, yielding poor performance in open-set recognition. What is more, it is hard to collect sufficient tactile data in practice. These factors make the existing open-set methods unsuitable for tactile recognition. Therefore, it needs to be investigated carefully.

As discussed above, the open-set tactile recognition is a non-trivial task due to the following two major challenges:

1. Similar to other open-set recognition problems, open-set tactile recognition also faces the core challenge is how to not only correctly classify samples from the known classes but also effectively detect and flag unknown examples as the novel.
2. Different from other open-set visual recognition tasks, collecting sufficient tactile data is difficult for training due to the complex collection process and constraints of robot-object physical interactions. This makes it difficult to migrate some existing open-set recognition methods with a complicated network to the tactile open-set recognition task.
3. Moreover, the tactile signals for object recognition are commonly high-dimensional dynamic time-series, which exhibit many challenges. Firstly, it is impossible to directly use high-dimensional signals into the existing machine learning methods without any preprocessing techniques. Additionally, there is the nature of misalignment among different tactile measurements. It makes tactile open-set recognition more difficult.

4. The proposed method

In this section, we first expound the framework architecture of the proposed method, and then we elaborate the details of the Gaussian prototype learning model in the method. Finally, we describe the algorithm optimization of the model.

4.1. Framework architecture

The framework of our proposed model is shown in Figure 2, which can be structurally disentangled into two main modules: *feature extraction module* $f(\theta, t)$ and *Gaussian prototype learning module*. The feature extraction module is used to transform the raw tactile inputs into abstract feature representations, where t is a tactile input and θ denotes the parameters of the feature extraction module. Different from the traditional softmax layer for classification on the learned features, we adopt a prototype learning module to learn class prototypes $\mu_{y_i}^l$ on the extracted features for each class $y_i \in Y$, where the superscript $l \in \{1, 2, ..., L\}$ is the number of prototypes in each category. Finally, we apply these prototypes for classification by template matching. When the extracted feature $f(\theta, t)$ of an input t can not match well with all prototypes of all known classes, it can be viewed as unknown. In this model, a feature extraction module and prototype module are jointly learned from data during training, thus forming a unified end-to-end deep framework, which is beneficial to improve the performance of recognition.

Previous experiments demonstrated that when the number of prototypes l in each class is large, it can not promote the classification accuracy and on the contrary will degrade the performance of the model. In fact, the deep neural network is very powerful for feature representation. Although the initial feature distribution is complex and scattered, the features of each class can be compacted to fit a single class centroid with some appropriate constraints after transformation. As such, we maintain one prototype for each category in our model. For convenience, $\mu_{y_i}^l$ is denoted as μ_{y_i}, of which the superscript is omitted in the following description.

4.2. Gaussian prototype learning model

Given a tactile input t_j, we firstly extract its abstract representation through the feature extraction module $f(\theta, t_j)$, and then search the nearest prototype based on the Euclidean distance between the extracted feature with all prototypes in the feature space. Finally, we assign the class label of this prototype to the tactile input. The process can be described as:

$$\hat{y} = \begin{cases} \arg\max\limits_{i=1}^{K} g_i(t_j), & if\ g_i(t_j) > \delta \\ Unknown\ Y_{K+1}, & if\ g_i(t_j) \leq \delta \end{cases} \tag{1}$$

where $g_i(x)$ is the class discriminant function that denotes the matching score of tactile sample t_j with class i, δ is a rejection threshold.

To train the framework, we introduce the three optimization objectives, which are *discriminative classification loss*, *feature distribution loss* and *learning to detect unknowns*.

4.2.1. Discriminative classification loss

Intuitively, an ideal class prototype should effectively discriminate and classify samples from different categories. To achieve the goal, we propose a discriminative classification loss. It aims to make the prototype of the corresponding class closer to $f(\theta, t_i)$ while the prototypes of other classes stay away from $f(\theta, t_i)$, ensuring tactile input is correctly classified.

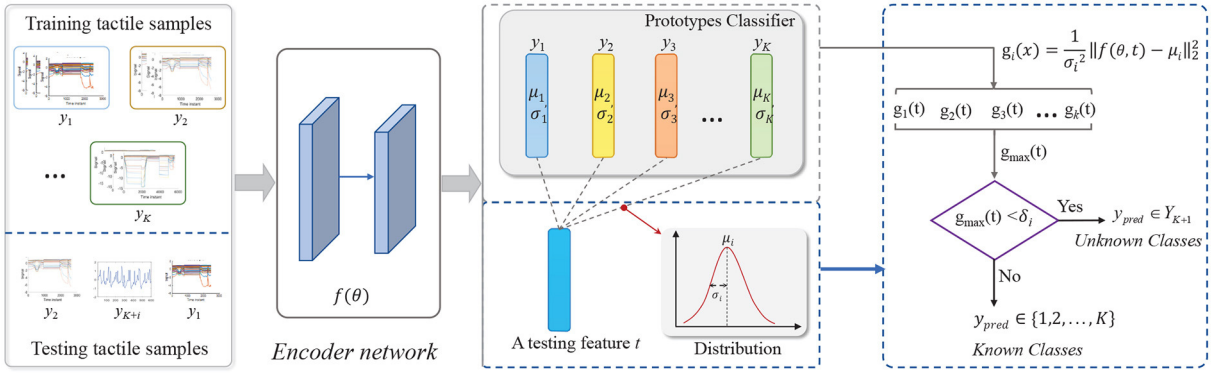


FIGURE 2
The framework of our proposed method for open-set recognition.

Essentially, the discriminative classification loss is a novel distance-based cross-entropy loss. Similar to traditional cross-entropy loss, it calculates cross-entropy loss with class probabilities obtained from the distances between samples feature and all prototypes. Specifically, given a sample t_i and its class label y_i, the probability of belonging to the corresponding prototype can be measured by the distance, and the probabilities are normalized in a similar way of Softmax. With this definition, the loss is defined as:

$$L_{cls}(\theta, \mu_i) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{K}\Gamma(\hat{y} = y_j)\log P_{y_j}(\hat{y}|t_i). \quad (2)$$

where $\Gamma(\cdot)$ is symbolic function, and p_{y_i} is class-specific probability, of which the definition can be expressed as:

$$P_{y_j}(\hat{y}|t_i) = \frac{e^{-\frac{1}{T}\left\|f(\theta,t_i)-\mu_{y_j}\right\|_2^2}}{\sum_{i\in Y} e^{-\frac{1}{T}\|f(\theta,t_i)-\mu_i\|_2^2}} \quad (3)$$

where T is a temperature coefficient that is used to control the characteristics of the classifier. We set the value of T as the variance σ^2 in the feature space, in order to normalize the representation space and increase the stability of the system. All classes prototypes μ_i with $i \in Y$ and the variance σ^2 are updated in an online manner.

By minimizing $L_{cls}(\theta, \mu_i)$, the loss aims to encourage separating the samples from different categories in learned feature space. In particular, this objective is to decrease the distance between samples of the same category and the corresponding prototype, and increase the distance between the sample and all other incorrect prototypes. Since the objective considers all prototypes in each updating step, it can better guarantee the convergence of training.

4.2.2. Feature distribution loss

For open-set recognition, the learned features need not only to be separable in different classes but also be compact in the same class. However, the above classification loss only makes the features of different categories separable. As a result, a feature t_i is far away from the corresponding category centroid μ_{y_i}, but it still is correctly classified if it is relatively closer to μ_{y_i} than to the feature centroids of the other classes. To tackle this issue, we further introduce a feature distribution loss to learn discriminative and compact representation, making it more applicable for our task.

The feature distribution loss is essentially the maximum likelihood regularization term on the assumption of Gaussian distribution. Specifically, we assume that the extracted feature on the training set conforms the Gaussian mixture distribution, viewing class prototype μ_{y_i} as the mean of a Gaussian component, which can be expressed as:

$$p(t_i) = \sum_{i=1}^{k} N(f(\theta, t_i), \mu_{y_i}, \sigma_{y_i})p(y_i). \quad (4)$$

where σ_{y_i} is covariance of class y_i in the feature space, and $p(y_i)$ is the prior probability of class y_i. For the convenience of calculation, the likelihood regularization term is defined as the negative log-likelihood. By reasonably setting constant prior probabilities $p(y_i)$, the likelihood regularization term L_{lkd} is simplified to Equation (5).

$$L_{lkd}(\theta, \mu_{y_i}) = -\sum_{i=1}^{k} \log N(f(\theta, t_i), \mu_{y_i}, \sigma_{y_i}). \quad (5)$$

The objective of the regularization term aims to maximize the log-likelihood of sample t_i for its corresponding class.

By minimizing L_{lkd}, the model can effectively reduce the within-class variance and constrain the feature distribution of

known classes, so it can reserve more feature space for unknown classes and improve the performance of the proposed method for detecting unknowns.

4.2.3. Learning to detect unknowns

The threshold-based rejection is frequently used in open-set recognition tasks. Most of the existing methods directly adopt the predefined threshold to detect unknowns, which is not suitable in practical applications. In order to make our model effective on the open set tasks, we explicitly consider adopting class-specific rejection criteria. In particular, we use an adaptive strategy by letting the value threshold δ to be proportional to maximal distance Δ_{y_i} between samples specific class y_i and the corresponding class centroid μ_{y_i}, i.e., $\delta = \alpha\Delta_{y_i}$ where α is proportional coefficient. Formally, Equation (1) can be expressed as:

$$\hat{y} = \begin{cases} t \in class \ \arg\max_{i=1}^{k} g_i(t), & if \ g_i(t) > \alpha\Delta_{y_i} \\ Unknown, & if \ g_i(t) \leq \alpha\Delta_{y_i} \end{cases}, \quad (6)$$

where $g_i(t) = \frac{1}{\sigma_{y_i}^2} \left\| f(\theta, t) - \mu_{y_i} \right\|_2^2$. Instead of adopting the predefined threshold, we explicitly learn specific threshold of each category by minimizing the following objective:

$$L_{thr}(\theta, \mu_{y_i}) = \sum_{i \in Y} \max(0, m(\frac{1}{\sigma^2} \left\| f(t_i, \theta) - \mu_{y_i} \right\|^2 - \alpha\Delta_i)). \quad (7)$$

where $m = -1$ if $i = y_i$ and $m = 1$ otherwise.

By minimizing L_{thr}, the model can obtain class-specific rejection thresholds, instead of presetting a global threshold as in prior works. It makes the proposed model effective to detect unknown samples.

4.3. Algorithm optimization

With the above-mentioned analysis, the optimization process of our proposed method is structurally divided into two components: *optimization of feature representation* and *optimization of rejection threshold*.

(1) In this optimization of feature representation, the trainable parameters in the proposed method are composed of two parts, i.e., parameters of encoder network for feature transformation $f(\theta, t_i)$ and all classes prototypes μ_i. To this end, we combine discriminative classification loss and feature distribution loss. The formally objective function is expressed as:

$$L(\theta, \mu_{y_i}) = L_{cls}(\theta, \mu_{y_i}) + \lambda L_{lkd}(\theta, \mu_{y_i}) \quad (8)$$

where $\lambda \geq 0$ is weighting coefficients, which controls the trade-off of the two loss terms to optimal performance.

Require:
(1): Training data $D_{tr} = \{(t_i, y_i)\}_{i=1}^{M}$, and the associated class label $y_i \in Y = \{1, 2, ..., k\}$;
(2): Hyperparameter: α, λ, the learning rate η;
(3): Testing dataset $D_{te} = \{(t_j, y_j)\}_{i=1}^{N}$, and the associated class label $y_j \in Y' = \{1, 2, ..., K, K+1, ..., k'\}$ $(k' > K)$.

Ensure:
Learned encoder network $f(\theta)$, class prototypes μ_{y_i} and corresponding covariance σ_{y_i}.

for number of iterations **do**;
  Update parameters θ, σ_{y_i} and μ_{y_i} by descending their stochastic gradients by Equation (8).
  $\theta \leftarrow \theta - \eta \cdot \nabla_\theta (L_{cls}(\theta, \mu_{y_i}) + \lambda L_{lkd}(\theta, \mu_{y_i}))$
  $\mu_{y_i} \leftarrow \mu_{y_i} - \eta \cdot \nabla_{\mu_{y_i}} (L_{cls}(\theta, \mu_{y_i}) + \lambda L_{lkd}(\theta, \mu_{y_i}))$
end for
return $f(\theta)$ and $\{(\sigma_i, \mu_i)\}_{i=1}^{k}$
$g_i(t) = \frac{1}{\sigma_{y_i}^2} \left\| f(\theta, t) - \mu_{y_i} \right\|_2^2$
$g_{max}(t) = sort(g_i(t))$
if $g_{max}(t) > \delta_i$
  Predict t as known classes with label y_{pred}
else do
  detect t as unknown with label Y_{K+1}
end if

Algorithm 1. The program flowchart of the proposed method.

For the hybrid optimization objective function $L(\theta, \mu_{y_i})$ in Equation (5), we can directly calculate the gradients of $\partial L/\partial f$ and $\partial L/\partial \mu_{y_i}$. According to the error back propagation, we can calculate the gradient of $\partial L/\partial \theta$. With the gradients of L over all parameters, we can jointly optimize both feature extractor and all classes prototypes using a gradient descent (SGD) optimization algorithm in an end-to-end way.

(2) For optimization of rejection threshold, it aims to achieve optimal class-specific thresholds $\alpha\Delta_i$. In this process, we held-out set of samples from the training set to learn the optimal thresholds. Hence, we split the samples into two parts, one part used for learning the feature extractor $f(\theta)$ and the classes prototypes μ_{y_i} and the remaining part for learning the value of $\alpha\Delta_{y_i}$.

In summary, the optimization process of our proposed model is elaborated in Algorithm 1.

5. Experiments

In this section, the proposed open-set tactile recognition method is comprehensively evaluated on two publicly available datasets. Firstly, the adopted datasets, evaluation metrics, comparison methods, and implementation details are described. Then experiments results and their analysis are provided.

FIGURE 3
All material images of the data set. The numbers, respectively, denote the beginning of each category. The original images are from Strese et al. (2016). It has been reproduced with permission from IEEE, available at https://zeus.lmt.ei.tum.de/downloads/texture/.

Finally, we further analyze the sensitivity of hyperparameters in the model.

5.1. Dataset splits

We demonstrate our proposed method on two publicly available data sets, which are Haptic Texture Database (LTM_108) dataset (Strese et al., 2016) and Penn Haptic Adjective Corpus (PHAC-2) (Chu et al., 2015) dataset. They have been used to evaluate a model's ability to recognize objects or textures by tactile modality (Liu and Sun, 2017). In these two data sets, their tactile data, respectively, represent two typical types of tactile information. Different from closed-set recognition, open-set tactile recognition needs a special setup and experiments. The splitting of the dataset is described as follows:

LTM_108: The LTM_108 dataset consists of 108 different surface material instances, which are divided into 9 categories based on the material properties. These material images of the dataset are shown in Figure 3. In this dataset, it provides multimodal data for each material instance, namely visual images, tactile acceleration traces and sound signals generated from the surface-tool interaction. The dataset provides a training set and a testing set. They both contain 108 material instances and every instance has ten tactile samples. In this experiment, we only use the tactile acceleration traces as tactile data for object recognition.

Although this dataset has been directly used for some closed-set tasks of tactile recognition, we use this dataset to tackle more challenging the open-set tactile recognition task. To provide a suitable test platform, a new dataset split is proposed based on the original dataset. In particular, we randomly select $K < 9$ categories tactile samples from the train set to train our models and use totally 9 categories of tactile samples from the test set for test evaluation. This setting ensures that the test set appears some material categories that are not in the training set. The Table 1 show a case of the dataset splits when $K = 6$.

TABLE 1 The details of the dataset splits on LTM_108.

Material category	Training samples	Testing samples
Mesh	13 × 10	13 × 10
Stones	9 × 10	9 × 10
Glossy	9 × 10	9 × 10
Wood	13 × 10	13 × 10
Rubbers	5 × 10	5 × 10
Fibers	15 × 10	15 × 10
Foams	-	12 × 10
Foils and paper	-	15 × 10
Textile and fabrics	-	17 × 10
Total	64 × 10	108 × 10

PHAC-2: There are 60 objects in the PHAC-2 dataset. The visual images of the dataset are shown in Figure 4. According to the physical properties, these objects are divided into eight categories. In this data set, each object contains tactile signals and visual images. The tactile signals are collected by two SynTouch BioTacs tactile sensors, which are installed to the grippers of a PR2 robot. In order to mimic the process of humans exploring the tactile properties of objects, the robot used four exploratory procedures to acquire five types of tactile data. According to the specified procedures, ten trials are performed on each object, resulting in totally 600 tactile samples. Although the joint data on gripper during exploratory movements are available, we focused on the tactile signals for classification in this experiment. In particular, each tactile sample consists of five components P_{DC}, P_{AC}, T_{DC}, T_{AC} and E_{19}.

Similar to the above dataset setting, the PHAC-2 dataset also needs to be reorganized and split. Firstly, we randomly select an object instance from each material category as test objects

FIGURE 4
The PHAC-2 contains 60 objects, which are organized by their primary material. The original images are from Chu et al. (2015). It has been reproduced with permission from Elsevier, available at https://hi.is.mpg.de/research_projects/learning-haptic-adjectives-from-tactile-data.

and remain other 52 object instances. Then, we randomly select $K < 8$ categories from eight categories from the remaining object instances. When $K = 5$, the details of the dataset splits are shown in Table 2. Please note that according to the above setting, not only does the test set contains some categories that are not in the training set, but also the training set and the testing set do not share the same object instance even from the same category. Different from instance-level recognition, this experiment can be referred to as categorization-level open-set recognition. To this need, we need the proposed model to have generalization and robustness for unseen object instances.

5.2. Data preprocessing and network architecture

Considering the difference between the two types of tactile signals, we adopt two different feature extraction methods and network architectures for classification. The specific details are as follows.

LTM_108: In the LTM_108 dataset, the recorded tactile signals are three-axis acceleration traces. Firstly, the three-axis acceleration traces are conversed to a one-dimensional signal by the DFT321 algorithm (Kuchenbecker et al., 2010). Considering the effectiveness of short-time Fourier transform (STFT) extracting features of time-series signals, we adopt STFT to convert a one-dimensional DFT321 signal into a spectrogram. These spectrograms are in the log domain, where the length of a frame length is 500 and the increment of frame and frame is 250. By the predefined configuration settings mentioned above, there are 100 spectrogram samples of size 50 x 250 extracted from each tactile acceleration trace.

As convolutional neural network (CNN) has proven to be effective in visual classification, which has achieved good performance on many tasks. Moreover, some CNN models pre-trained on ImageNet (Deng et al., 2009) have shown generalization and discrimination. In this experiment, we use the pre-trained Resnet18 (He et al., 2016) model on ImageNet as the network backbone of the proposed method.

PHAC-2: As in Abderrahmane et al. (2019), we firstly normalize the five components (P_{DC}, P_{AC}, T_{DC}, T_{AC}, E_{19}) in each signal sample, respectively. As the sample rate of P_{AC} is higher than other components of a tactile sample, we downsample it to match the other signals' sample rate of 100 Hz. For some exploratory movements, the length of tactile signals varies considerably from objects. In order to resolve the length difference of signal, we downsample the signal of each exploratory movement to a fixed length of 150. Principal Component Analysis is used independently on the E_{19} data from each exploratory movement to capture the four most principal components across all objects. Thus, we obtain 64 tactile signals for each object in each trial.

Recently, Ji et al. (2015) has demonstrated the effectiveness of CNN on temporal signals with limited amounts. In this experiment, we adopt Convolutional neural networks (CNN) to perform tactile object recognition. The specific network structure is the same as the Haptic CNN model in Gao et al. (2016). Every tactile sample per object has 64 tactile signals. We concatenate the 64 features along the channel axis, which is used as the input of our model.

TABLE 2 The details of the dataset splits on the PHAC-2.

Material category	Original samples	Training samples	Testing samples
Foam	16 × 10	15 × 10	1 × 10
Organic	5 × 10	4 × 10	1 × 10
Fabric	7 × 10	6 × 10	1 × 10
Plastic	13 × 10	12 × 10	1 × 10
Paper	12 × 10	11 × 10	1 × 10
Stone	2 × 10	-	1 × 10
Glass	2 × 10	-	1 × 10
Metal	3 × 10	-	1 × 10
Total	60 × 10	48 × 10	8 × 10

5.3. Evaluation metric

In this experiment, we use the three metrics to evaluate the classification performance, including *Accuracy* and *F-measure* and *AUC*.

- Accuracy: As a common metric method to evaluate classifiers on a closed set task, recognition accuracy *Acc* is defined as:

$$Accuracy = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TN} + \mathrm{TP} + \mathrm{FP} + \mathrm{FN}} \tag{9}$$

  where TP, TN, FN, and FP, respectively, denote true positive, true negative false negative, and false positive. The sum of the three quantities is equal to the total number of samples.
- F-measure: F-measure is commonly evaluation metric, which is defined as a harmonic mean of Precision *P* and Recall *R*:

$$F - measure = 2 \times \frac{P \times R}{P \times R} = \frac{2TP}{2TP + FP + FN} \tag{10}$$

  As suggested in Bendale and Boult (2016) and Geng et al. (2020), we use macro-averaged F1-score. It is denoted as macro-F1.
- AUC: It denotes area under the ROC curve (AUC), which measures the performance of detecting unknown between known and unknown data.

5.4. Comparison methods

To validate the advantages of our proposed method, several classical methods were also implemented for comparison. A brief description of the methods is as follows:

- Softmax: It used the highest probability from the softmax layer of networks as the confidence score for classification.
- τ-Softmax (Hendrycks and Gimpel, 2016): It aims to use a global threshold on the softmax probability to determine whether an input sample belongs to an unknown class. We refer to this method as τ-Softmax.
- τ-Center (Wen et al., 2016): It can be combined with cross-entropy loss to encourage the training data to form better-behaved class structures, which may be easier to model and facilitate greater distinction of open-set inputs. To this end, we also use it to detect unknown classes by a predefined threshold, which is denoted as τ-Center loss.
- OpenMax (Bendale and Boult, 2016): It proposed replacing the softmax layer with OpenMax, which calibrates the confidence score with Weibull distribution. It proposed an inference method for detecting novel classes.

We note that some advanced methods, such as Yoshihashi et al. (2019) and Sun et al. (2020), have also been proposed to deal with open-set visual recognition. However, we do not take them for comparison, because the networks of these methods are too complex to work on the limited training data of tactile tasks.

5.5. Implementation details

For open-set recognition, the ratio of seen and unseen is an important factor, which quantifies the openness of the problem. As in Zhou et al. (2021), the openness is defined as:

$$openness = 1 - \sqrt{\frac{N_{train}}{N_{test}}} \tag{11}$$

where N_{train} and N_{test}, respectively, denote the number of categories in training set and testing set. As we described in the preliminaries, $N_{train} = K$.

In this experiment, we empirically set the likelihood regularization parameter λ to 0.01 in experiments. For the margin parameter α, the optimization of the objective function becomes more difficult as the value increases. Therefore, α needs to be smaller when the number of classes gets more. In our experiments, we empirically set α to 0.4 and 0.3 for LTM_108 and PHAC-2, respectively.

5.6. Experimental results and analysis

Experimental results on the LTM_108 and PHAC-2 datasets are reported in this subsection. In this experiment, we randomly select 6 categories as known classes for LTM_108. Considering the instance imbalance of categories in PHAC-2, the first five categories are used as known classes. Ten trials are performed on each experiment, and the averaged results are used as final metric

TABLE 3 Experimental results of different method.

Model	LTM_108			PHAC-2		
	Accuracy	Macro-F1	AUC	Accuracy	Macro-F1	AUC
Softmax	59.1%	0.491	0.878	62.5%	0.518	0.871
τ-Softmax (Hendrycks and Gimpel, 2016)	61.7%	0.567	0.970	70.01%	0.625	0.975
τ-Center (Wen et al., 2016)	64.9%	0.613	0.975	71.3%	0.634	0.977
OpenMax (Bendale and Boult, 2016)	62.5%	0.574	0.978	58.7%	0.536	0.928
Proposed method	70.76%	0.669	0.986	75.5%	0.703	0.986

scores. In this setting, the corresponding experimental results on different methods are shown in Table 3.

From Table 3, it can be seen that our proposed method achieves the highest Acc, macro-F1, and AUC on the two datasets. It indicates that the proposed model outperforms the compared methods, which also demonstrates that the proposed method is able to effectively improve the ability to detect unknowns while ensuring the accuracy of the known classification simultaneously.

As mentioned above, the open-set recognition on PHAC-2 is more challenging, as its test set and training set does not share the same object instance. Besides, we do not perform any data augmentation or employ some specific and complex networks in these experiments. Even so, our proposed method still achieves optimal performance. This further verifies the effectiveness of the proposed method.

Additionally, it is clear that τ-Center exhibits a better performance among all these compared methods because it explicitly encourages stronger compactness of feature, which is beneficial for open-set recognition. However, it mainly aims at improving the softmax loss and feature distribution is not explicitly modeled. Therefore, it can not achieve optimal performances dealing with the tactile OSR problem. Since integrating the advantages from both classification discrimination with Gaussian Prototype Learning and likelihood estimation of feature distribution, our proposed method performs better in open-set conditions. It highlights the importance of considering the likelihood of feature distribution in the tactile OSR problem.

In particular, we can observe that as a state-of-art open-set recognition method, Openmax shows low performance, especially on the PHAC-2 dataset. It is mainly due to low recall on known classes with a few training instances since test instances from smaller classes are usually projected farther from the mean activation vector of the corresponding class. This demonstrates that Openmax may be also hardly infer the class probability of unknown inputs by the probability distribution of Softmax. Moreover, our experiments indicate that merely thresholding the output probabilities of softmax helps, but is still relatively weak for open set recognition.

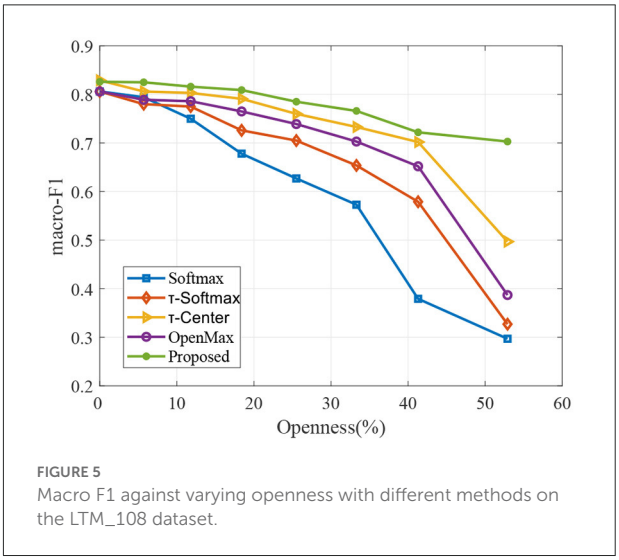


FIGURE 5
Macro F1 against varying openness with different methods on the LTM_108 dataset.

5.7. Effectiveness of different openness

To valid the robustness of our proposed model to different openness, we evaluate performance over multiple openness values in the experiments. In particular, we vary the openness of Equation (11) by varying the number of classes in the training sample, while the number of test classes remains the same. We evaluate the performance by macro F1-scores. The corresponding results are shown in Figures 5, 6.

As to be expected, when more known classes are available during training, the performances of classifiers are better for all methods in Figures 5, 6. We can observe that the proposed approach remains relatively stable over a wide range of openness, which produces better results compared to other methods.

5.8. Parameter sensitivity analysis

In the proposed model, α and λ are important parameters, and their values affect on the model's performance. To obtain optimal values for these parameters, we conduct extensive experiments to perform grid search for α and

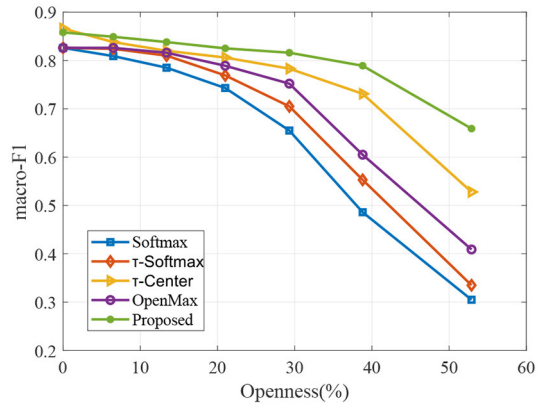


FIGURE 6
Macro F1 against varying openness with different methods on the PHAC-2 dataset.

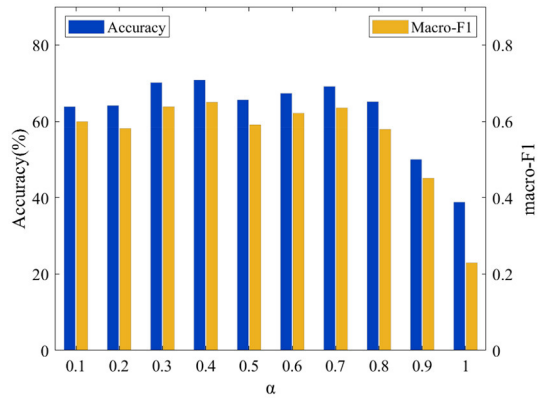


FIGURE 7
Acc and macro F1 for different α on the LTM_108 dataset.

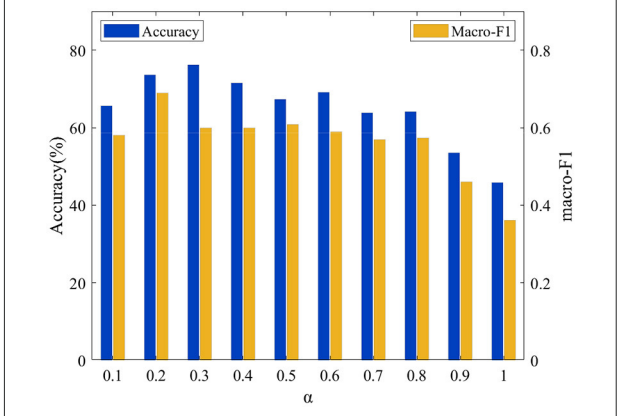


FIGURE 8
Acc and macro F1 for different α on the PHAC-2 dataset.

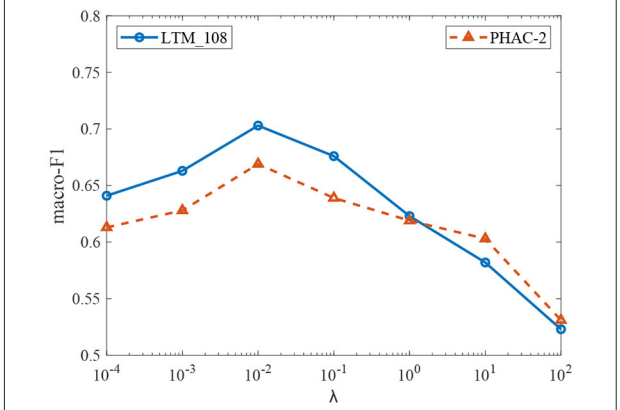


FIGURE 9
The performance of proposed model in terms of λ.

λ within the set $\{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$ and $\{0.0001, 0.001, 0.01, 0.1, 1, 10, 100\}$. The experimental results show that the model can achieve optimal performance when $\alpha = 0.4$ and $\lambda = 0.01$ on the LTM_108 dataset. For the PHAC-2 dataset, the model shows the best performance where $\alpha = 0.3$ and $\lambda = 0.01$. For the convenience of explanation, the sensitivity analysis of these two parameters is divided into two parts for illustration.

To analyze the effect of these parameters α on the proposed model's performance, we set the value of λ to 0.01 and perform grid search of the parameter α within the set $\{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$. The relationships between Accuracy and the macro-F1 and of the value of α are shown on the two datasets in Figures 7, 8, respectively. It can be observed that the performance of the model is very sensitive to the value of the parameter α, and the model performs well when $\alpha \in [0.1, 0.8]$ on both datasets.

Then, we conduct experiments to study the effect of the parameter λ on the performance of the model. Fixing parameters $\alpha = 0.4$ on the LTM_108 dataset and $\alpha = 0.3$ on the PHAC-2 dataset, we tune the parameter λ within the set $\{0.0001, 0.001, 0.01, 0.1, 1, 10, 100\}$ and the corresponding experimental results are given in Figure 9. It can be observed that our model achieves good performances in the range of $\lambda \in (0, 1]$. When $\lambda > 1$, the model's performance on the contrary degrades. This is mainly because the likelihood regularization starts to play a role when the training accuracy is close to saturation, and a strong regularization weakens the discrimination effect of the model. Hence, there is a need to find the optimal balance of the two terms in the optimization process.

6. Conclusion

In this work, we specifically address tactile object recognition in open-set scenarios, which aims to enable robots to exploit

tactile explorations in unstructured environments. To this end, we proposed a novel Gaussian prototype learning model, which incorporates classification and novel class detection into a unified framework. In particular, a likelihood regularization term is introduced to explicitly consider the feature distribution of tactile data. In addition, we further develop an adaptive classification margin to improve the performance of the model. Experimental results validate the effectiveness of the proposed method, which has the potential to improve the performance of open-set tactile perception. We believe that it makes the first step to formulate lifelong tactile recognition in the real world. In the future, we will explore the generalization of the proposed method to realize active continual learning in the open world.

Data availability statement

The raw data supporting the conclusions of this article will be made available by the authors, without undue reservation.

Author contributions

WZ and HL proposed the basic idea of this method and completed theoretical modeling. DG performed the experiments analysis and revised the manuscript. FS provided overall supervision of this research. All authors contributed to the article and approved the submitted version.

Funding

This work was supported by the National Natural Science Fund for Key International Collaboration (62120106005) and partially funded by the China Postdoctoral Science Foundation under Grant 2022M711825.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

Abderrahmane, Z., Ganesh, G., Crosnier, A., and Cherubini, A. (2018). Haptic zero: recognition of objects never touched before. *Rob. Auton. Syst.* 105, 11–25. doi: 10.1016/j.robot.2018.03.002

Abderrahmane, Z., Ganesh, G., Crosnier, A., and Cherubini, A. (2019). A deep learning framework for tactile recognition of known as well as novel objects. *IEEE Trans. Ind. Inform.* 16, 423–432. doi: 10.1109/TII.2019.2898264

Bendale, A., and Boult, T. (2015). "Towards open world recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Boston, MA: IEEE), 1893–1902.

Bendale, A., and Boult, T. E. (2016). "Towards open set deep networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Las Vegas, NV: IEEE), 1563–1572.

Chu, V., McMahon, I., Riano, L., McDonald, C. G., He, Q., Perez-Tejada, J. M., et al. (2015). Robotic learning of haptic adjectives through physical interaction. *Rob. Auton. Syst.* 63, 279–292. doi: 10.1016/j.robot.2014.09.021

Deng, C., Han, Y., and Zhao, B. (2019). High-performance visual tracking with extreme learning machine framework. *IEEE Trans. Cybern.* 50, 2781–2792. doi: 10.1109/TCYB.2018.2886580

Deng, C., Jing, D., Han, Y., Wang, S., and Wang, H. (2022). Far-net: fast anchor refining for arbitrary-oriented object detection. *IEEE Geosci. Remote Sens. Lett.* 19, 1–5. doi: 10.1109/LGRS.2022.3144513

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). "Imagenet: a large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition* (Miami, FL: IEEE), 248–255.

Dhamija, A. R., Günther, M., and Boult, T. E. (2018). Reducing network agnostophobia. *arXiv preprint arXiv:1811.04110*. doi: 10.48550/arXiv.1811.04110

Gao, Y., Hendricks, L. A., Kuchenbecker, K. J., and Darrell, T. (2016). "Deep learning for tactile understanding from visual and haptic data," in *2016 IEEE International Conference on Robotics and Automation (ICRA)* (IEEE), 536–543.

Ge, Z., Demyanov, S., Chen, Z., and Garnavi, R. (2017). Generative openmax for multi-class open set classification. *arXiv preprint arXiv:1707.07418*. doi: 10.5244/C.31.42

Geng, C., Huang, S.-J., and Chen, S. (2020). Recent advances in open set recognition: a survey. *IEEE Trans. Pattern Anal. Mach. Intell.* 43, 3614–3631. doi: 10.1109/TPAMI.2020.2981604

Gu, F., Sng, W., Taunyazov, T., and Soh, H. (2020). Tactilesgnet: a spiking graph neural network for event-based tactile object recognition. *arXiv preprint arXiv:2008.08046*. doi: 10.1109/IROS45743.2020.9341421

Guo, D., Liu, H., Fang, B., Sun, F., and Yang, W. (2021). Visual affordance guided tactile material recognition for waste recycling. *IEEE Trans. Autom. Sci. Eng.* 19, 2656–2664. doi: 10.1109/TASE.2021.3065991

Guo, D., Sun, F., Liu, H., Kong, T., Fang, B., and Xi, N. (2017). "A hybrid deep architecture for robotic grasp detection," in *2017 IEEE International Conference on Robotics and Automation (ICRA)* (Singapore: IEEE), 1609–1614.

Han, Y., Deng, C., Zhang, Z., Li, J., and Zhao, B. (2017). "Adaptive feature representation for visual tracking," in *2017 IEEE International Conference on Image Processing* (Beijing: IEEE), 1867–1870.

Han, Y., Deng, C., Zhao, B., and Tao, D. (2019). State-aware anti-drift object tracking. *IEEE Trans. Image Process.* 28, 4075–4086. doi: 10.1109/TIP.2019.2905984

He, K., Zhang, X., Ren, S., and Sun, J. (2016). "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Las Vegas, NV: IEEE), 770–778.

He, W., Xue, C., Yu, X., Li, Z., and Yang, C. (2020). Admittance-based controller design for physical human-robot interaction in the constrained task space. *IEEE Trans. Autom. Sci. Eng.* 17, 1937–1949. doi: 10.1109/TASE.2020.2983225

Hendrycks, D., and Gimpel, K. (2016). A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv preprint arXiv:1610.02136*. doi: 10.48550/arXiv.1610.02136

Ji, M., Fang, L., Zheng, H., Strese, M., and Steinbach, E. (2015). "Preprocessing-free surface material classification using convolutional neural networks pretrained by sparse autoencoder," in *2015 IEEE 25th International Workshop on Machine Learning for Signal Processing (MLSP)* (Boston, MA: IEEE), 1–6.

Kerr, E., McGinnity, T. M., and Coleman, S. (2018). Material recognition using tactile sensing. *Expert. Syst. Appl.* 94, 94–111. doi: 10.1016/j.eswa.2017.10.045

Kuchenbecker, K. J., McMahan, W., Landin, N., and Romano, J. M. (2010). "Dimensional reduction of high-frequencey accelerations for haptic rendering," in *Proceedings of EuroHaptics* (Amsterdam; Berlin: springer), 79–86.

Li, Q., Kroemer, O., Su, Z., Veiga, F. F., Kaboli, M., and Ritter, H. J. (2020). A review of tactile information: perception and action through touch. *IEEE Trans. Rob.* 36, 1619–1634. doi: 10.1109/TRO.2020.3003230

Li, Z., Huang, B., Ajoudani, A., Yang, C., Su, C.-Y., and Bicchi, A. (2018). Asymmetric bimanual control of dual-arm exoskeletons for human-cooperative manipulations. *IEEE Trans. Rob.* 34, 264–271. doi: 10.1109/TRO.2017.2765334

Liu, H., and Sun, F. (2017). Material identification using tactile perception: a semantics-regularized dictionary learning method. *IEEE/ASME Trans. Mechatron.* 23, 1050–1058. doi: 10.1109/TMECH.2017.2775208

Liu, H., Yu, Y., Sun, F., and Gu, J. (2016). Visual-tactile fusion for object recognition. *IEEE Trans. Autom. Sci. Eng.* 14, 996–1008. doi: 10.1109/TASE.2016.2549552

Luo, S., Bimbo, J., Dahiya, R., and Liu, H. (2017). Robotic tactile perception of object properties: a review. *Mechatronics* 48, 54–67. doi: 10.1016/j.mechatronics.2017.11.002

Meyer, J., Eitel, A., Brox, T., and Burgard, W. (2020). "Improving unimodal object recognition with multimodal contrastive learning," in *IEEE/RSJ International Conference on Intelligent Robots and Systems, Vol. 2* (Las Vegas, NV: IEEE), 4.

Neal, L., Olson, M., Fern, X., Wong, W.-K., and Li, F. (2018). "Open set learning with counterfactual images," in *Proceedings of the European Conference on Computer Vision* (Munich; Berlin: Springer), 613–628.

Qiao, H., Chen, J., and Huang, X. (2021). A survey of brain-inspired intelligent robots: integration of vision, decision, motion control, and musculoskeletal systems. *IEEE Trans. Cybern.* 52, 11267–11280. doi: 10.1109/TCYB.2021.3071312

Scheirer, W. J., de Rezende Rocha, A., Sapkota, A., and Boult, T. E. (2012). Toward open set recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* 35, 1757–1772. doi: 10.1109/TPAMI.2012.256

Shu, L., Xu, H., and Liu, B. (2017). "Doc: deep open classification of text documents," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing* (Copenhagen, Stroudsburg: ACL), 2911–2916.

Strese, M., Schuwerk, C., Iepure, A., and Steinbach, E. (2016). Multimodal feature-based surface material classification. *IEEE Trans. Haptics*. 10, 226–239. doi: 10.1109/TOH.2016.2625787

Sun, X., Yang, Z., Zhang, C., Ling, K. -V., and Peng, G. (2020). "Conditional gaussian distribution learning for open set recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (Seattle, NJ: IEEE), 13480–13489.

Taunyazov, T., Koh, H. F., Wu, Y., Cai, C., and Soh, H. (2019). "Towards effective tactile identification of textures using a hybrid touch approach," in *2019 International Conference on Robotics and Automation (ICRA)* (Montreal, QC: IEEE), 4269–4275.

Wang, W., Han, Y., Deng, C., and Li, Z. (2022). Hyperspectral image classification via deep structure dictionary learning. *Remote Sens.* 14, 2266. doi: 10.3390/rs14092266

Wen, Y., Zhang, K., Li, Z., and Qiao, Y. (2016). "A discriminative feature learning approach for deep face recognition," in *European Conference on Computer Vision* (Glasgow; Berlin: Springer), 499–515.

Xu, D., Loeb, G. E., and Fishel, J. A. (2013). "Tactile identification of objects using bayesian exploration," in *2013 IEEE International Conference on Robotics and Automation* (Karlsruhe: IEEE), 3056–3061.

Yang, C., Luo, J., Liu, C., Li, M., and Dai, S.-L. (2019). Haptics electromyography perception and learning enhanced intelligence for teleoperated robot. *IEEE Trans. Autom. Sci. Eng.* 16, 1512–1521. doi: 10.1109/TASE.2018.2874454

Yi, Z., Xu, T., Guo, S., Shang, W., and Wu, X. (2020). Tactile surface roughness categorization with multineuron spike train distance. *IEEE Trans. Autom. Sci. Eng.* 18, 1835–1845. doi: 10.1109/TASE.2020.3021742

Yoshihashi, R., Shao, W., Kawakami, R., You, S., Iida, M., and Naemura, T. (2019). "Classification-reconstruction learning for open-set recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (Long Beach, CA: IEEE), 4016–4025.

Yuan, W., Mo, Y., Wang, S., and Adelson, E. H. (2018). "Active clothing material perception using tactile sensing and deep learning," in *2018 IEEE International Conference on Robotics and Automation (ICRA)* (Brisbane, QLD: IEEE), 4842–4849.

Zhang, H., and Patel, V. M. (2016). Sparse representation-based open set recognition. *IEEE Trans. Pattern. Anal. Mach. Intell.* 39, 1690–1696. doi: 10.1109/TPAMI.2016.2613924

Zheng, W., Liu, H., and Sun, F. (2020a). Lifelong visual-tactile cross-modal learning for robotic material perception. *IEEE Trans. Neural Netw. Learn. Syst.* 32, 1192–1203. doi: 10.1109/TNNLS.2020.2980892

Zheng, W., Liu, H., Wang, B., and Sun, F. (2019). Cross-modal surface material retrieval using discriminant adversarial learning. *IEEE Trans. Ind. Inform.* 15, 4978–4987. doi: 10.1109/TII.2019.2895602

Zheng, W., Liu, H., Wang, B., and Sun, F. (2020b). Cross-modal material perception for novel objects: a deep adversarial learning method. *IEEE Trans. Autom. Sci. Eng.* 17, 697–707. doi: 10.1109/TASE.2019.2941230

Zhou, D.-W., Ye, H.-J., and Zhan, D.-C. (2021). "Learning placeholders for open-set recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (Nashville, TN: IEEE), 4401–4410.

frontiers | Frontiers in Neuroscience

TYPE Original Research
PUBLISHED 10 January 2023
DOI 10.3389/fnins.2022.1080521

OPEN ACCESS

EDITED BY
Yuqi Han,
Tsinghua University, China

REVIEWED BY
Tiancheng Dong,
Wuhan University, China
Wendong Zheng,
Tsinghua University, China

*CORRESPONDENCE
Liangyu Zhao
zhaoly@bit.edu.cn

SPECIALTY SECTION
This article was submitted to Perception Science, a section of the journal Frontiers in Neuroscience

RECEIVED 26 October 2022
ACCEPTED 22 November 2022
PUBLISHED 10 January 2023

CITATION
Cui J, Wu J and Zhao L (2023) Learning channel-selective and aberrance repressed correlation filter with memory model for unmanned aerial vehicle object tracking. *Front. Neurosci.* 16:1080521. doi: 10.3389/fnins.2022.1080521

Learning channel-selective and aberrance repressed correlation filter with memory model for unmanned aerial vehicle object tracking

Jianjie Cui[1], Jingwei Wu[2] and Liangyu Zhao[1]*

[1]School of Aerospace Engineering, Beijing Institute of Technology, Beijing, China, [2]The Second Academy of CASIC, Beijing, China

To ensure that computers can accomplish specific tasks intelligently and autonomously, it is common to introduce more knowledge into artificial intelligence (AI) technology as prior information, by imitating the structure and mindset of the human brain. Currently, unmanned aerial vehicle (UAV) tracking plays an important role in military and civilian fields. However, robust and accurate UAV tracking remains a demanding task, due to limited computing capability, unanticipated object appearance variations, and a volatile environment. In this paper, inspired by the memory mechanism and cognitive process in the human brain, and considering the computing resources of the platform, a novel tracking method based on Discriminative Correlation Filter (DCF) based trackers and memory model is proposed, by introducing dynamic feature-channel weight and aberrance repressed regularization into the loss function, and by adding an additional historical model retrieval module. Specifically, the feature-channel weight integrated into the spatial regularization (SR) enables the filter to select features. The aberrance repressed regularization provides potential interference information to the tracker and is advantageous in suppressing the aberrances caused by both background clutter and appearance changes of the target. By optimizing the aforementioned two jointly, the proposed tracker could restrain the potential distractors, and train a robust filter simultaneously by focusing on more reliable features. Furthermore, the overall loss function could be optimized with the Alternative Direction Method of Multipliers (ADMM) method, thereby improving the calculation efficiency of the algorithm. Meanwhile, with the historical model retrieval module, the tracker is encouraged to adopt some historical models of past video frames to update the tracker, and it is also incentivized to make full use of the historical information to construct a more reliable target appearance representation. By evaluating the method on two challenging UAV benchmarks, the results prove that this tracker shows superior performance compared with most other advanced tracking algorithms.

KEYWORDS

unmanned aerial vehicle, object tracking, discriminative correlation filter, channel regularization, aberrance repressed, historical memory

1. Introduction

The thinking ability endowed by the brain is fundamental. Due to its existence, human beings are more intelligent than animals. It is also the premise that humans have the capability to conduct scientific research (Kuroda et al., 2022). People rely on their brains to recognize the world, learn knowledge, and summarize rules. Their brains also allow them to use memory systems to store the information generated when experiencing different events (Atkinson and Shiffrin, 1968; Cornelio et al., 2022). In turn, the information serves as prior knowledge, helping people in dealing with similar problems better and adapting to new complex scenes faster. The core of artificial intelligence (AI) is to enable the machine to complete specific tasks independently through learning and using prior information (Connor et al., 2022; Foksinska et al., 2022; Nofallah et al., 2022; Pfeifer et al., 2022; Wang et al., 2022). The original scientific research mainly adopts the following two methods. The first is to mathematically represent the law (i.e. prior information) that people summarize when perceiving things, and then use mathematical expressions and logical frameworks to construct modules and methods for computers, just like teachers teach students what they know. The second is to build a variety of artificial neural networks based on the neural structure of the human brain and then use large-scale data to train and fit the network (Deng et al., 2022; Liu et al., 2022), aiming to enable the computer to automatically learn the characteristics of various things from the data itself, just like the students read books and learn by themselves. Although scientists have put a lot of effort into the research and utilization of the human brain, it is still a difficult task to determine how to endow computers with more and better prior knowledge through algorithms.

This paper mainly concentrates on visual object tracking on the UAV platform, which plays an important role in the field of computer vision, and is widely used in many tasks, such as collision avoidance (Baca et al., 2018), traffic monitoring (Elloumi et al., 2018), military surveillance (Shao et al., 2019), and aerial cinematography (Gschwindt et al., 2019). By adopting this technology, it aims to predict the precise status of the target in a video sequence captured by an onboard camera only with the information given in the first frame (Han et al., 2022). Over the past few years, a lot of effort has been put into the tracking field. However, it is still a challenging task to design a robust and efficient tracker, when considering the various complex UAV tracking scenarios, e.g., occlusion, change of viewpoint, and limited power capacity.

In the past decade, the research on visual object tracking mainly adopted the two methods below, namely the discriminative correlation filter (DCF)-based method and the Siamese-based method. The Siamese-based method (Bertinetto et al., 2016b; Li et al., 2018a; Wang et al., 2019; Voigtlaender et al., 2020; Javed et al., 2022) aims at the offline learning of the similarity measurement function between image patches, by maximizing the distance between the target and the background patches while minimizing the distance between the different image patches belonging to the same target. Such a method consists of two identical subchannels that are used to process the target template and the current frame search area, respectively. The target location is determined by computing the partial similarity between the target template and each location in the search area. Moreover, the Siamese-based method uses neural network architecture and numerous training data to obtain excellent feature extraction capability, so it needs to occupy a large number of computing resources in the tracking process. DCF-based methods are based on the correlation theory in the field of signal processing, and it computes the correlation between different image patches by convolution. Such a method usually adopts the hand-crafted features carefully designed with prior information and aims at training a correlation filter online in the region around the target by minimizing a least squares loss. Due to the convolution theorem, DCF-based methods can track objects at hundreds of frames per second (FPS) with only one CPU. Considering that the computing resources of the UAV platform are very limited, and the speed is a key issue in addition to the tracking performance, this paper mainly concentrates on target tracking based on DCF methods.

The development history of the DCF-based method is the process by which people integrated more and better prior information into the tracking framework. As people add their understanding of tracking tasks as regular constraints to the loss function (Mueller et al., 2017; Han et al., 2019b), the trained correlation filter becomes more and more discriminative and robust. Mosse (Bolme et al., 2010), as the originator of correlation filtering, deemed target tracking as a problem of binary classification, and trained the filter by randomly sampling a fixed number of background samples as negative samples. This greatly limits its discriminative power. To effectively increase the number of training samples, which was critical to the performance of the trained classifier, KCF (Henriques et al., 2014) introduced the circulant matrix into the tracking framework and obtained a large number of virtual negative samples by circularly shifting the target samples. The cyclic shifting greatly increased the training samples and caused boundary effects that seriously limited the improvement of tracking performance simultaneously. To mitigate the boundary effect, SRDCF (Danelljan et al., 2015) added the SR term into the loss function, aiming at penalizing the non-zero value near the template boundaries. BACF (Kiani Galoogahi et al., 2017) generated lots of real background samples, by expanding the search area and introducing a binary mask for middle elements cropping. To solve the scale change of the target, DSST (Danelljan et al., 2014a) introduced an independent scale filter, in addition to the classical correlation filter used for locating, as well as SAMF (Li and Zhu, 2014) sampled multiscale images, thereby building image pyramids. For the improvement of the feature representation, CN (Danelljan et al.,

2014b) brought in color features, while ECO (Danelljan et al., 2017) added depth features obtained from off-line training of the neural network. STRCF (Li et al., 2018b) brought in additional temporal constraints to the SRDCF to limit the variation of the filter in consecutive frames. This effectively reduced the risk of filter degradation in case of sudden large appearance variations. SAT (Han et al., 2019a) advocated a kurtosis-based updating scheme to guarantee a high-confidence template updating. ASRCF (Dai et al., 2019) realized the adaptive suppression of clutter in different regions by regarding the SR term, introduced in SRDCF, as a variable. MUSTer (Hong et al., 2015) built the short-term and long-term memory stores, thereby processing the target appearance memories. Autotrack (Li et al., 2020) reformulated the loss function by introducing the change of response maps into temporal regularization (TR) and SR terms, thereby realizing adaptive adjustment. Regardless of the great progress in DCF-based tracking methods, there are still some issues to solve. (1) Most original trackers treat the features of different dimensions equally. Features of different dimensions play different roles in tracking different scenarios and different kinds of targets. The tracker is easily biased by similar interference due to ignorance of the feature channel information. (2) Most original trackers have insufficient ability to suppress potential interference. Most of the original methods merely utilize the same and fixed bowl-shaped SR term centered on the target, aiming at giving more weight to the background area for suppression. Additional suppression is not applied to the potential interference according to the actual tracking situation, thus leading to limited anti-aberrance capability. (3) Most original trackers do not effectively use historical information. Most of the original methods updated the filter with a constant update rate, thereby causing the waste of historical information and the risk of filter degradation. Historical information is one of the most important factors in the tracking process and should be efficiently used to enhance the discriminant capacity of the tracker.

The brain can perceive the interference information in the background, independently select the optimal features to describe the target, and use historical memory to achieve an accurate target location in the current frame. When considering the above, a UAV tracking algorithm with repressed dynamic aberrance, a channel selective correlation filter, and a historical model retrieval module is proposed to solve the aforementioned problems. Moreover, by formulating the dynamic feature channel weight and the aberrance repressed regularization into the integral loss function, the tracking algorithm is built, thereby enabling the filter to highlight valuable features in the channel domain and using response maps to sense and suppress background interference in advance. Meanwhile, the model retrieval module, by imitating brain memory realizes the adaptive update of the tracker. This paper has the main contributions as follows.

i) A novel tracking method, that integrates the aberrance repressed regularization and dynamic feature channel weight into the loss function of the DCF framework, is proposed. For joint modeling of the two factors, the tracker obtains the ability to screen target features based on actual background interference and learns more differentiated target appearance representation. Thus, the loss function could be solved in very few iterations by employing an efficient ADMM algorithm.

ii) A model retrieval module is employed which can realize the adaptive update of the tracker by saving the history filters. This module can also enhance the tracker's learning of the appearance of the trusted targets with historical information and reduce the pollution of unreliable samples for the tracker.

iii) By giving the experimental validation conducted on two public UAV datasets, the effectiveness of this method is demonstrated.

2. Proposed methodologies

2.1. Revisted autotrack

In this section, the baseline Autotrack of this tracker shall be revised.

Most original trackers, based on the discriminative correlation filters (DCF), attempt to add a variety of regularization terms such as spatial regularization (SR) and temporal regularization (TR), thereby improving the discrimination ability to target and background. Such regularization terms are usually predefined fixed parameters, so flexibility and adaptability are lacking in cluttered and challenging scenarios. To realize automatic adjustment of the hyper-parameters of the SR and TR terms during tracking, Autotrack constructs them with the response maps obtained during detection. Specifically, Autotrack introduces the partial response variation Λ to the SR parameter $\tilde{\boldsymbol{u}}$, and the global response variation $\|\boldsymbol{\Lambda}\|_2$ to the reference value $\tilde{\theta}$ of the coefficient of the TR term. The partial response variation Λ is defined as the variation of response maps between two continuous frames, with the Equation as below.

$$\boldsymbol{\Lambda} = \frac{\boldsymbol{R}_t[\psi_\Delta] - \boldsymbol{R}_{t-1}}{\boldsymbol{R}_{t-1}} \tag{1}$$

Where, $\boldsymbol{R}$ refers to the response map calculated in the detection phase. $[\psi_\Delta]$ represents the shift operator which makes the response peaks in response maps of two continuous frames coincide with each other. As for Autotrack, the integral objective

loss function is shown below:

$$\begin{aligned} \boldsymbol{E}(\boldsymbol{H}_t, \theta_t) &= \frac{1}{2}\left\| \boldsymbol{y} - \sum_{k=1}^{K} \boldsymbol{x}_t^k \circledast \boldsymbol{h}_t^k \right\|_2^2 + \frac{1}{2}\sum_{k=1}^{K}\left\| \tilde{\boldsymbol{u}} \odot \boldsymbol{h}_t^k \right\|_2^2 \\ &+ \frac{\theta_t}{2}\sum_{k=1}^{K}\left\| \boldsymbol{h}_t^k - \boldsymbol{h}_{t-1}^k \right\|_2^2 + \frac{1}{2}\left\| \theta_t - \tilde{\boldsymbol{\theta}} \right\|_2^2 \\ \text{s.t.} \quad \tilde{\boldsymbol{u}} &= \boldsymbol{P}^{\top}\delta log(\boldsymbol{\Lambda}+1) + \boldsymbol{u} \\ \tilde{\boldsymbol{\theta}} &= \frac{\zeta}{1 + log(\nu \left\| \boldsymbol{\Lambda} \right\|_2 + 1)}, \left\| \boldsymbol{\Lambda} \right\|_2 \leq \phi \end{aligned} \tag{2}$$

Where, $\boldsymbol{X}_t = [\boldsymbol{x}_t^1, \boldsymbol{x}_t^2, ..., \boldsymbol{x}_t^K]$ and $\boldsymbol{H}_t = [\boldsymbol{h}_t^1, \boldsymbol{h}_t^2, ..., \boldsymbol{h}_t^K]$ represent the trained filter and the extracted target feature matrix at t frame, respectively. K is the total number of feature channels. $\boldsymbol{x}_t^k \in \boldsymbol{R}^{T\times T}$ indicates the sample feature vector with length T in frame t in k channel and $y \in \boldsymbol{R}^{T\times T}$ is the desired corresponding label set in the Gaussian shape. $\tilde{\boldsymbol{u}}$ and θ_t represent the coefficients of SR and TR, respectively. $\tilde{\theta}$ is the reference value of θ_t used for measuring the change in the tracking response map between two continuous frames. $\mathbf{P}^{\mathrm{T}} \in R^{T\times T}$ is a binary matrix, used in cropping the central elements of the training sample $\boldsymbol{X}_t$. δ is a constant that can be used in balancing the weights of partial response variations. $\boldsymbol{u}$ represents a fixed bowl-shaped matrix of SR which is identical to the STRCF tracker. $\circledast$ and $\odot$ represent the convolution operation and the elemental multiplication, respectively. $\|\|_2^2$ is the Euclidean norm.

SR and TR, constructed by response maps variation, enable the trained filter in Autotrack to adjust automatically while flying and be more adaptable to different scenarios. Although this method has achieved outstanding performance, it does have two limitations. a) This method uses the response map generated by the filter in the previous frame, rather than the learned filter in this frame, thus leading to insufficient suppression of interference. Sudden changes in response maps give important information regarding the similarity of the current object and the appearance model and reveal potential aberrances. The tracker should reduce the learning of irrelevant objects according to the changes during the training phase. b) The weight of each feature channel is equivalent. Different channels describe the objects in different dimensions. There may be many similar features between the target and the background, which are useless or even have a negative effect on the discriminatory ability of trackers. Thus, the filter selects partial distinctive features based on the actual situation for training and updating.

2.2. Loss function construction

To solve the above problems and enhance the discrimination ability and anti-interference ability of the tracker, the weight of the feature channel and aberrance suppression are introduced together to restrain the filter. Specifically, feature channel weight, which is treated as an optimization variable, updates simultaneously with the filter. Also, the variation of two continuous response maps, as an aberrance suppression regularization, is integrated into the training process. The loss function is shown below.

$$\begin{aligned} \boldsymbol{E}(\boldsymbol{H}_t, \theta_t, \boldsymbol{v}_t) &= \frac{1}{2}\left\| \boldsymbol{y} - \sum_{k=1}^{K} \boldsymbol{x}_t^k \circledast \boldsymbol{h}_t^k \right\|_2^2 + \frac{1}{2}\sum_{k=1}^{K}\left\| \upsilon_t^k \tilde{\boldsymbol{u}} \odot \boldsymbol{h}_t^k \right\|_2^2 \\ &+ \frac{\lambda_1}{2}\sum_{k=1}^{K}\left\| \upsilon_t^k - \upsilon_0^k \right\|_2^2 \\ &+ \frac{\theta_t}{2}\sum_{k=1}^{K}\left\| \boldsymbol{h}_t^k - \boldsymbol{h}_{t-1}^k \right\|_2^2 + \frac{1}{2}\left\| \theta_t - \tilde{\boldsymbol{\theta}} \right\|_2^2 \\ &+ \frac{\lambda_2}{2}\left\| \boldsymbol{Q}_{t-1} - \sum_{k=1}^{K} \boldsymbol{x}_t^k \circledast \boldsymbol{h}_t^k \right\|_2^2 \end{aligned} \tag{3}$$

Where υ_t^k is the weight coefficient of feature channel k at t frame. It should be noted that υ_t^k is not a fixed parameter, but a variable that changes with the target appearance during the tracking. The constant υ_0^k is regarded as the reference of υ_t^k, which represent the advance distributions of targets in the different feature channels. υ_0^k is set to 1, thereby ensuring that each feature channel has the same weight in the initial state. $\boldsymbol{Q}_{t-1}$ refers to the response map generated from the previous frame, and is equivalent to $\sum_{k=1}^{K} \boldsymbol{x}_{t-1}^k \circledast \boldsymbol{h}_{t-1}^k$. Thus, it can be treated as a constant signal during the optimization stage. λ_1, and λ_2 are parameters that control model overfitting.

Equation 3 consists of six items that can be divided into four parts. The first part constitutes the first item, the regression term. The second part, including the second and third items, is the SR integrated with channel selection. The third part, consisting of the fourth and fifth items, is the TR borrowed from Autotrack. The fourth part, made up of the last item, is the regularization term, aiming at restricting and counteracting the aberrances created by the background information. For the introduction of channel weight υ_t^k, the feature sifting of the filter is realized in the channel domain by mitigating the impact of features having no relation to the targets and by excluding needless information. By introducing aberrance repressed regularization, which gives greater penalties for interference, the ability of the tracker to identify the aberrance in the background, and suppress the subsequent changes of response maps on the basis of the baseline, is further improved. The fusion of these two factors enables the filter to find the aberrance in time, and utilize the best features, thereby maximizing the differentiation between the target and background.

2.3. Optimization

As observed from Equation 3, the optimization of the overall loss function involves the complex correlation operation between matrices. Therefore, to reduce computational complexity, and reduce sufficient computing efficiency, the Parseval theorem is used to convert complex correlation operations into simple elemental multiplication operations and move the loss function from the time domain to the Fourier domain as $\boldsymbol{E}(\boldsymbol{H}_t, \hat{\boldsymbol{G}}_t, \theta_t, \boldsymbol{\upsilon}_t)$. Besides, the constraint parameter $\hat{\boldsymbol{g}}_t^k = \sqrt{T}\boldsymbol{F}\boldsymbol{P}^{\mathrm{T}}\boldsymbol{h}_t^k$ is used in constituting the Augmented Lagrangian function $\boldsymbol{L}(\boldsymbol{H}_t, \hat{\boldsymbol{G}}_t, \boldsymbol{\upsilon}, \theta_t, \hat{\boldsymbol{M}}_t)$ as follows:

$$\begin{aligned}\boldsymbol{L}(\boldsymbol{H}_t, \hat{\boldsymbol{G}}_t, \boldsymbol{\upsilon}, \theta_t, \hat{\boldsymbol{M}}_t) &= \boldsymbol{E}(\boldsymbol{H}_t, \hat{\boldsymbol{G}}_t, \theta_t, \boldsymbol{\upsilon}_t) \\ &\quad + \sum_{k=1}^{K} (\hat{\boldsymbol{g}}_t^k - \sqrt{T}\boldsymbol{F}\boldsymbol{P}^{\mathrm{T}}\boldsymbol{h}_t^k)\hat{\boldsymbol{m}}_t^k \\ &\quad + \frac{\mu}{2}\sum_{k=1}^{K}\left\|\hat{\boldsymbol{g}}_t^k - \sqrt{T}\boldsymbol{F}\boldsymbol{P}^{\mathrm{T}}\boldsymbol{h}_t^k\right\|_2^2 \\ \boldsymbol{E}(\boldsymbol{H}_t, \hat{\boldsymbol{G}}_t, \theta_t, \boldsymbol{\upsilon}_t) &= \frac{1}{2}\left\|\hat{\boldsymbol{y}} - \sum_{k=1}^{K}\hat{\boldsymbol{x}}_t^k \odot \hat{\boldsymbol{g}}_t^k\right\|_2^2 \\ &\quad + \frac{1}{2}\sum_{k=1}^{K}\left\|\upsilon_t^k\tilde{\boldsymbol{u}} \odot \boldsymbol{h}_t^k\right\|_2^2 + \frac{\theta_t}{2}\sum_{k=1}^{K}\left\|\hat{\boldsymbol{g}}_t^k - \hat{\boldsymbol{g}}_{t-1}^k\right\|_2^2 \\ &\quad + \frac{1}{2}\left\|\theta_t - \tilde{\boldsymbol{\theta}}\right\|_2^2 + \frac{\lambda_1}{2}\sum_{k=1}^{K}\left\|\upsilon_t^k - \upsilon_0^k\right\|_2^2 \\ &\quad + \frac{\lambda_2}{2}\left\|\hat{\boldsymbol{Q}}_{t-1} - \sum_{k=1}^{K}\hat{\boldsymbol{x}}_t^k \odot \hat{\boldsymbol{g}}_t^k\right\|_2^2\end{aligned} \tag{4}$$

Where symbol $\hat{}$ represents the discrete Fourier transformation (DFT), for example, $\hat{\boldsymbol{y}} = \sqrt{N}\boldsymbol{F}\boldsymbol{y}$ and $\boldsymbol{F}$ called the Fourier matrix is the orthonormal $N \times N$ matrix of complex basis vectors. $\boldsymbol{m}$ refers to the Lagrangian multiplier, and $\boldsymbol{\mu}$ represents the penalty parameter. For simplification, $\hat{\boldsymbol{G}}_t = [\hat{\boldsymbol{g}}_t^1, \hat{\boldsymbol{g}}_t^2, \hat{\boldsymbol{g}}_t^3, ..., \hat{\boldsymbol{g}}_t^K]$ and $\hat{\boldsymbol{M}}_t = [\hat{\boldsymbol{m}}_t^1, \hat{\boldsymbol{m}}_t^2, \hat{\boldsymbol{m}}_t^3, ..., \hat{\boldsymbol{m}}_t^K]$ are defined. By assigning $\hat{\boldsymbol{s}}_t^k = \frac{1}{\mu}\hat{\boldsymbol{m}}_t^k$ the optimization of Equation (4) is equivalent to solving equation (5).

$$\begin{aligned}\boldsymbol{L}(\boldsymbol{H}_t, \hat{\boldsymbol{G}}_t, \boldsymbol{\upsilon}, \theta_t, \hat{\boldsymbol{S}}_t) &= \boldsymbol{E}(\boldsymbol{H}_t, \hat{\boldsymbol{G}}_t, \boldsymbol{\upsilon}, \theta_t) \\ &\quad + \frac{\mu}{2}\sum_{k=1}^{K}\left\|\hat{\boldsymbol{g}}_t^k - \sqrt{T}\boldsymbol{F}\boldsymbol{P}^{\mathrm{T}}\boldsymbol{h}_t^k + \hat{\boldsymbol{s}}_t^k\right\|_2^2\end{aligned} \tag{5}$$

Considering the complexity of the above-mentioned function, the alternative direction method of multipliers (ADMM) (Lin et al., 2010) is applied to speed up the calculation. Specifically, the function of optimization can be divided into a few sub-problems to be solved iteratively. During the solution of every subproblem, only one variable is contained to be optimized, while the others are regarded as fixed constants temporarily. In this way, each subproblem and its relevant closed-form solution can be given in detail below.

Subproblem for $\hat{\boldsymbol{G}}_t$: By giving $\boldsymbol{H}_t$, $\boldsymbol{\upsilon}$, θ_t, $\hat{\boldsymbol{S}}_t$, the optimal $\hat{\boldsymbol{G}}_t^*$ could be obtained by solving the optimization problem:

$$\begin{aligned}\hat{\boldsymbol{G}}_t^* &= \arg\min_{\hat{\boldsymbol{G}}_t^*}\{\frac{1}{2}\left\|\hat{\boldsymbol{y}} - \sum_{k=1}^{K}\hat{\boldsymbol{x}}_t^k \odot \hat{\boldsymbol{g}}_t^k\right\|_2^2 + \frac{\theta_t}{2}\sum_{k=1}^{K}\left\|\hat{\boldsymbol{g}}_t^k - \hat{\boldsymbol{g}}_{t-1}^k\right\|_2^2 \\ &\quad + \frac{\lambda_2}{2}\left\|\hat{\boldsymbol{Q}}_{t-1} - \sum_{k=1}^{K}\hat{\boldsymbol{x}}_t^k \odot \hat{\boldsymbol{g}}_t^k\right\|_2^2 \\ &\quad + \frac{\mu}{2}\sum_{k=1}^{K}\left\|\hat{\boldsymbol{g}}_t^k - \sqrt{T}\boldsymbol{F}\boldsymbol{P}^{\mathrm{T}}\boldsymbol{h}_t^k + \hat{\boldsymbol{s}}_t^k\right\|_2^2\}\end{aligned} \tag{6}$$

However, it is still very difficult to solve Equation 6 directly, because this subproblem containing $\hat{\boldsymbol{X}}_k\hat{\boldsymbol{g}}_k$ shows a high computation complexity and needs multiple iterations in ADMM. Fortunately, $\hat{\boldsymbol{X}}_k$ is sparse, which means that each element of $\hat{\boldsymbol{y}}(\hat{\boldsymbol{y}}(n), n = 1, 2, \ldots, N)$ is merely related to $\hat{\boldsymbol{x}}_k(n) = [\hat{\boldsymbol{x}}_k(n)^1, \hat{\boldsymbol{x}}_k(n)^2, \ldots, \hat{\boldsymbol{x}}_k(n)^D]$ and $\hat{\boldsymbol{g}}_k(n) = [conj(\hat{\boldsymbol{g}}_k(n)^1), conj(\hat{\boldsymbol{g}}_k(n)^2), ..., conj(\hat{\boldsymbol{g}}_k(n)^D)]$, where $conj()$ refers to the complex conjugate operation. Thus, this subproblem can be divided into N simpler problems across K channels as follows.

$$\begin{aligned}\Gamma_j^*(\hat{\boldsymbol{G}}_t) &= \arg\min_{\Gamma_j(\hat{\boldsymbol{G}}_t)}\{\left\|\hat{\boldsymbol{y}}_j - \Gamma_j(\hat{\boldsymbol{X}}_{\mathbf{t}})^{\top}\Gamma_j(\hat{\boldsymbol{G}}_t)\right\|_2^2 \\ &\quad + \mu\left\|\Gamma_j(\hat{\boldsymbol{G}}_t) + \Gamma_j(\hat{\boldsymbol{s}}_t) - \Gamma_j(\sqrt{T}\boldsymbol{F}\boldsymbol{P}^{\top}\boldsymbol{H}_t)\right\|_2^2 \\ &\quad + \theta_t\left\|\Gamma_j(\hat{\boldsymbol{G}}_t) - \Gamma_j(\hat{\boldsymbol{G}}_{t-1})\right\|_2^2 \\ &\quad + \frac{\lambda_2}{2}\left\|\hat{\boldsymbol{Q}}_{t-1} - \Gamma_j(\hat{\boldsymbol{X}}_{\mathbf{t}})^{\top}\Gamma_j(\hat{\boldsymbol{G}}_t)\right\|_2^2\}\end{aligned} \tag{7}$$

Where, $\Gamma_j(\hat{\boldsymbol{G}}_t) \in C^{(}K \times 1)$ indicates the vector including all K channel value of $\hat{\boldsymbol{G}}_t$ on pixel $j(j = 1, 2, \ldots, N)$. By introducing the Sherman-Morrison formula $(uv^H + A)^{-1} = A^{-1} - \frac{A^{-1}uv^HA^{-1}}{v^HA^{-1}u+1}$, the inverse operation in the derivation can be further simplified and accelerated.Then, the closed-form solution of this subproblem can be obtained as follows.

$$\Gamma_j^*(\hat{\boldsymbol{G}}_t) = \frac{1}{\mu + \theta_t}(\mathbf{I} - \frac{(1+\lambda_2)\Gamma_j(\hat{\mathbf{X}}_t)\Gamma_j(\hat{\mathbf{X}}_t)^{\top}}{\theta_t + \mu + (1+\lambda_2)\Gamma_j(\hat{\mathbf{X}}_t)^{\top}\Gamma_j(\hat{\mathbf{X}}_t)})\boldsymbol{\rho} \tag{8}$$

Where $\boldsymbol{\rho}$ is merely an intermediate variable for simple representation and $\boldsymbol{\rho} = \Gamma_j(\hat{\mathbf{X}}_t)\hat{\mathbf{y}}_j + \theta_t\Gamma_j(\hat{\boldsymbol{G}}_{t-1}) - \mu\Gamma_j(\hat{\mathbf{S}}_t) + \mu\Gamma_j(\sqrt{T}\mathbf{F}\mathbf{P}^{\top}\boldsymbol{H}_t) + \lambda_1\Gamma_j(\hat{\mathbf{X}}_t)\hat{\mathbf{Q}}_{t-1}$

Subproblem for $\boldsymbol{H}_t$: By fixing $\hat{\boldsymbol{G}}_t$,$\boldsymbol{\upsilon}$,θ_t,$\hat{\mathbf{S}}_t$, $\boldsymbol{H}_t$ can be solved with the equation below:

$$\begin{aligned}\boldsymbol{h}_t^{k*} &= \arg\min_{\boldsymbol{h}_t^k}\{\frac{1}{2}\left\|\upsilon_t^k\tilde{\boldsymbol{u}} \odot \boldsymbol{h}_t^k\right\|_2^2 + \frac{\mu}{2}\left\|\hat{\boldsymbol{g}}_t^k - \sqrt{T}\boldsymbol{F}\boldsymbol{P}^{\mathrm{T}}\boldsymbol{h}_t^k + \hat{\boldsymbol{s}}_t^k\right\|_2^2\} \\ &= \frac{\mu T\mathbf{p} \odot (\boldsymbol{s}_t^k + \boldsymbol{g}_t^k)}{\lambda_1(\upsilon_t^k\tilde{\boldsymbol{u}} \odot \upsilon_t^k\tilde{\boldsymbol{u}}) + \mu T\mathbf{p}}\end{aligned} \tag{9}$$

Where, $\mathbf{p} = [\mathbf{P}_{11}, \mathbf{P}_{22}, \ldots, \mathbf{P}_{TT}]^\top$ represents the column vector, that composed of the diagonal elements of $\mathbf{P}$. As observed in Equation 9, the computational cost on $\boldsymbol{h}_t^{k*}$ solution is very low, because it only involves the element-wise operation and an inverse fast Fourier transform.

Subproblem for θ_t: By treating $\hat{\boldsymbol{G}}_t$,$\boldsymbol{\upsilon}$,$\boldsymbol{H}_t$,$\hat{\boldsymbol{S}}_t$ as constants, the optimal θ_t can be obtained by solving the problem of optimization below:

$$\begin{aligned}\theta_t^* &= \arg\min_{\theta_t} \{\frac{\theta_t}{2}\sum_{k=1}^{K}\left\|\hat{\boldsymbol{g}}_t^k - \hat{\boldsymbol{g}}_{t-1}^k\right\|_2^2 + \frac{1}{2}\left\|\theta_t - \tilde{\theta}\right\|_2^2\} \\ &= \tilde{\theta} - \frac{\sum_{k=1}^{K}\left\|\hat{\boldsymbol{g}}_t^k - \hat{\boldsymbol{g}}_{t-1}^k\right\|_2^2}{2}\end{aligned} \tag{10}$$

Subproblem for $\boldsymbol{\upsilon_t}^*$ Given $\hat{\boldsymbol{G}}_t$,θ_t,$\boldsymbol{H}_t$,$\hat{\boldsymbol{S}}_t$, $\boldsymbol{\upsilon_t^k}$ can be optimized with the following equation.

$$\begin{aligned}\upsilon_t^{k*} &= \arg\min_{\upsilon_t^k} \frac{1}{2}\sum_{k=1}^{K}\left\|\upsilon_t^k\tilde{\boldsymbol{u}} \odot \boldsymbol{h}_t^k\right\|_2^2 + \frac{\lambda_1}{2}\sum_{k=1}^{K}\left\|\upsilon_t^k - \upsilon_0^k\right\|_2^2 \\ &= \frac{\lambda_1\upsilon_0^k}{(\tilde{\boldsymbol{u}} \odot h_t^k)^\top(\tilde{\boldsymbol{u}} \odot h_t^k) + \lambda_1}\end{aligned} \tag{11}$$

Lagrangian multiplier updating:

$$\hat{\boldsymbol{S}}_t^{i+1} = \hat{\boldsymbol{S}}_t^{\,i} + \mu^i(\hat{\boldsymbol{G}}_t^{\,i+1} - \hat{\boldsymbol{H}}_t^{\,i+1}) \tag{12}$$

Where, i and $i+1$ represent the previous and current iterations. The new $\hat{\boldsymbol{G}}_t$ $\hat{\boldsymbol{H}}$ obtained from the above optimization solution is used to update the Lagrangian multiplier. The regularization constant observes the updating laws of $\mu^{i+1} = min(\mu_{max}, \beta\mu^i)$, thereby ensuring the convergence of the integral model according to ADMM.

2.4. Historical model retrieval module

Most original tracking methods use linear interpolation with a constant learning rate β, like Equation 13, to update the filter. However, such an updating method not only causes the tracker to indiscriminately treat all the historical information but also results in filter pollution and degradation. The tracking result is poor when faced with complex scenes, such as partial occlusion, and camera defocus. Too high a learning rate causes the tracker to easily overfit and then neglect historical information, while too low a learning rate disenables the tracker from effectively learning the change of targets. Considering that the human brain can recall historical memory to make the best choice when identifying targets and HMTS tracker, the history filter, namely the historical model retrieval module is retrieved, and the best filter of the current frame is obtained by selecting and linear interpolating several effective filters. Specifically, historical filters are saved first, and a filter library is built. After the training phase of each frame, the correlation between each template and the current sample image is calculated. Several historical templates with the highest scores are selected and the scores are used as weights to linear interpolate them, thereby obtaining a tracking template for the next frame object location. This module is described below in detail with mathematical symbols.

$$\boldsymbol{h}_t = \beta\boldsymbol{h} + (1-\beta)\boldsymbol{h}_{t-1} \tag{13}$$

Similiar to HMTS tracker (Chen et al., 2022), this method retains the filter for each frame as the historical model $\boldsymbol{H}_{hist}$. However, the HMTS tracker builds the filters library with all historical filters, which causes much computing burden and redundancy. For example, when tracking to the end of a long video, there are numerous historical filters, and there is great similarity in target appearance between the current filter and the front filter. Therefore, the size is fixed to ϕ_{hist} and the filters library is constructed as $\boldsymbol{H}_{hist} = \{(\boldsymbol{h}_i, s_i)\}_{i=1}^{\phi_{hist}}$. s_i refers to the score of each historical model.

As expressed by the regression term in the loss function Equation 3, the convolution results of the trained filter and sample should ideally present a Gaussian shape centered on the target, namely the label $\boldsymbol{y}$. The basis of correlation filtering theory is as below: the more similar the two signals are, the greater the correlation between them is. Thus, like the HMTS tracker, the s_i is defined as the correlation between the label $\boldsymbol{y}$ and the convolution results $\boldsymbol{R}_i$ of different historical filters $\boldsymbol{H}_i, i \in [1, \phi_{hist}]$ and the current frame target samples $\boldsymbol{X}_t$. The equation of s_i is as follows:

$$\begin{aligned}s_i &= max(\mathscr{F}^{-1}(\boldsymbol{y}^H\boldsymbol{R}_i)) \\ \boldsymbol{R}_i &= \left\|\sum_{k=1}^{K}\boldsymbol{x}_t^k \circledast \boldsymbol{h}_i^k\right\|_2^2\end{aligned} \tag{14}$$

Where $\mathscr{F}^{-1}$ represents the inverse Fourier transform,^{H} indicates the conjugate transpose, and $max(\cdot)$ refers to the maximum of the vector.

After the tracker training phase in accordance with Section 2.3, Equation (14) is adopted to calculate the scores of the trained filter in the current frame and historical filters in the filters library. Next, the historical model with the lowest score in the filter library is replaced by the filter trained from the current frame, thereby ensuring no change in the number of filters in the library. It needs to be noted that, since the first frame is the most accurate manually labeled target information, the filter of the first frame shall always remain in the filter library. The filter $\boldsymbol{h}_t$ used for object detection in the next frame can be obtained by a linear weighting of the filters with the top ϕ_{scores} scores.

$$\begin{aligned}\boldsymbol{h}_t &= \sum_i s_i\boldsymbol{h}_i \\ s.t.Rank(s_i) &\geq \phi_{scores}\end{aligned} \tag{15}$$

Where, $Rank(s_i)$ represents the index of s_i in the set $\{s_i\}_{i=1}^{\phi_{hist}}$, which is ranked in descending and $i \in [1, \phi_{hist}]$. It needs to be noted that the filter trained in the first frame always participates in the calculation of Equation 15 and it is given the lowest weight in ϕ_{scores} filters if $Rank(s_1) \leq \phi_{scores}$.

3. Experiments

In this section, the tracking performance of the proposed tracker is evaluated against the nine state-of-the-art trackers, namely AutoTrack, ASRCF, ECO-HC, STRCF, SRDCF, BACF, LADCF (Xu et al., 2019), MCCT-H (Wang et al., 2018) and Staple (Bertinetto et al., 2016a) on two difficult UAV benchmarks (UAV123 Mueller et al., 2016 and VisDrone2018-test-dev Zhu et al., 2018). For the measurement of the performance of the aforementioned trackers, the employed evaluation metric named one-pass evaluation(OPE) contained two indicators, namely Precision Rate and Success Rate. It needs to be noted that the precision plot threshold is set to 10 pixels in UAV123 and to 21 pixels in VisDrone2018-test-dev, when considering the different target sizes from different UAV datasets.

3.1. Implementation details

Our tracker was used in MATLAB-2017a with an Intel i7-9750H CPU, and 16GB of RAM, and runs at a 25 FPS average with hand-crafted characteristics for target representation. The common hyper-parameters are kept to the same values as the baseline Autotrack, namely $\delta = 0.2$, $\nu = 2 \times 10^{-5}$, and $\zeta = 13$. The SR constraint coefficient λ_1 and the response aberrance regularization constraint coefficient λ_2 which are unique to the proposed tracker, are set as 0.71 and 0.001, respectively. In the historical model retrieval module, $\phi_{hist} = 30$ and $\phi_{scores} = 20$ are determined. As for the ADMM algorithm, the number of iterations is set as 4, $\beta = 10$, and $\mu_{max} = 10^4$, which also shares the same parameters as in Autotrack.

3.2. Quantitative evaluation

UAV123 is the most commonly used dataset in UAV object tracking, with 123 videos with more than 110K frames composed. In these sequences, 12 of the challenging attributes involved, such as background clutter, aspect ratio change, and similar object, required a more accurate and stable tracking algorithm. The quantitative comparison of different trackers is shown in Figure 1, and it can be observed that our tracker shows the best precision with the second success rate, slightly lower than ECO-HC. However, the proposed method achieves a remarkable advantage of 2.6% in precision and 1.5% in success rate, compared with the baseline tracker Autotrack.

VisDrone2018-test-dev is a dataset that is especially proposed for aerial object tracking competition. It consists of 35 videos captured from 14 different cities and covers various aspects including such as shooting position, tracking scene, target type, and object density. Different scenarios, weather conditions, and illumination changes are primarily addressed in this dataset. As shown in Figure 2, the proposed tracker is superior to all other evaluated trackers, and it can achieve 81.1% and 60.7% in the distance precision (DP) and the area under the curve (AUC), respectively. By comparing with the baseline tracker, Autotrack, our tracker accomplishes 2.3% and 3.4% of performance gains in precision and success rate, respectively.

3.3. Parametric sensitivity

As presented in Section 3.1, some hyper-parameters of the proposed tracker need to be set, namely the spatial-channel regularization constraint coefficient λ_1 and the response aberrance regularization constraint coefficient λ_2 in the loss function. In this section, the influence of different configurations on tracking results is investigated. When evaluating each hyper-parameter for a fair comparison, the common parameters are maintained at the same value as in Autotrack and all other parameters are fixed. Considering the operation speed, ϕ_{hist} is set as a constant of 30 and $\phi_{scores} = 20$ is set as a constant of 20 to ensure the efficient use of historical information and the effective reduction of redundancy. Table 1 exhibits the tracking results under different λ_1,λ_2 in VisDrone2018-test-dev, where ϕ_{scores} is fixed to 20. It can be observed that this tracker yields the best performance with $\lambda_1 = 0.001$ and $\lambda_2 = 0.71$.

3.4. Ablation experiments

As described in Section 2, in our method loss function is reconstructed by introducing the feature channel weight and aberrance repressed regularization, and an additional historical memory model is added to the baseline Autotrack. To prove the effectiveness of each module, ablation experiments were conducted. The results are shown in Table 2. AutoTrack_csar only reconstructs the loss function, while AutoTrack_hist only adds the historical memory model. As observed, by adding the two modules separately, the performance of the baseline tracker can be improved effectively. Moreover, by joining these two modules simultaneously, our method can achieve excellent performance against the baseline. This is mainly because the fusion of the two enables the tracker to effectively use historical information to prevent background clutter during tracking while establishing a more robust target appearance representation.

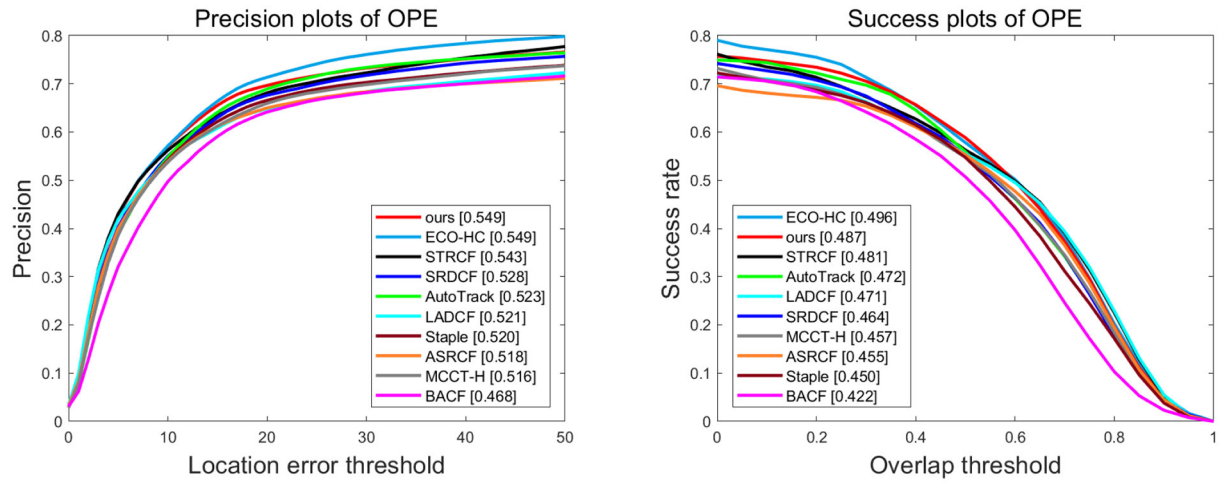


FIGURE 1
Overall performance of ten trackers on UAV123. The legend of success plot represents the area under the curve score of each tracker. The legend of the precision plot refers to the threshold scores at 10 pixels.

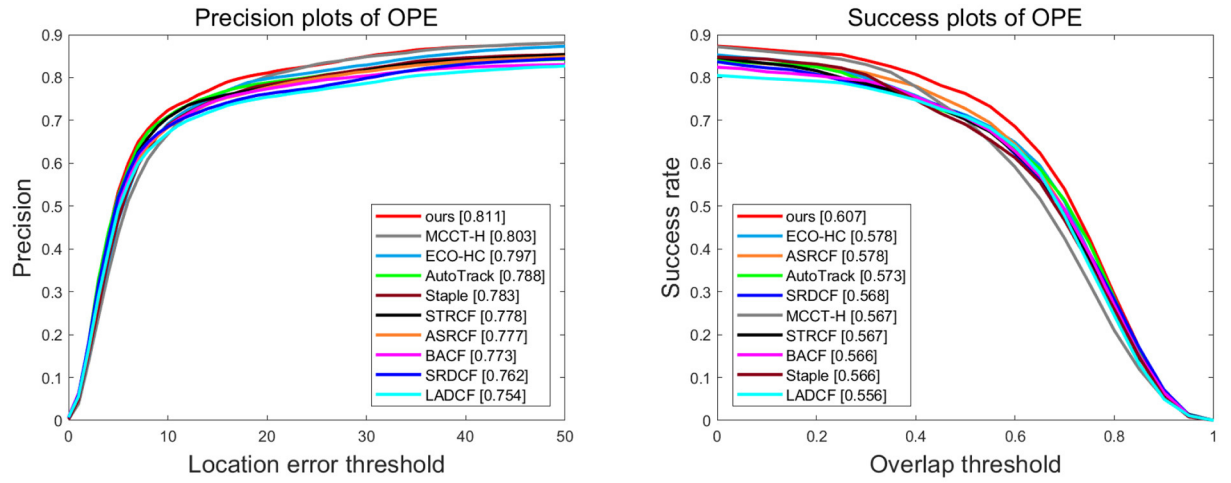


FIGURE 2
Overall performance of ten trackers on Visdrone2018-test-dev. The legend of success plot represents the area under the curve score of each tracker. The legend of the precision plot refers to threshold scores at 21 pixels.

TABLE 1 The success rate and precision rate (percentage) related to the varying number of regularization constraint coefficients on VisDrone2018-test-dev.

Parameter	λ_1				λ_2			
Value	**0.001**	0.1	0.5	1	**0.71**	0.01	0.1	1
Success Rate	**60.7**	58.4	59.2	59.0	**60.7**	58.5	59.1	59.0
Precision Rate	**81.1**	78.4	79.5	80.2	**81.1**	79.1	80.1	79.5

In historical models, $\phi_{scores} = 20$ and $\phi_{hist} = 30$. The threshold of precision rate is set to 21 pixels.
Bold values refer to first place in the experiments.

TABLE 2 The success rate and precision rate (percentage) of ablation experiments on UAV123.

Tracker	AutoTrack	Two regularization	Historical memory	Precision rate	Success rate
AutoTrack	✓			52.3	47.2
AutoTrack_csar	✓	✓		53.7	47.4
AutoTrack_hist	✓		✓	54.2	47.5
Ours	✓	✓	✓	**54.9**	**48.7**

The precision rate threshold is set as 10 pixels.
Bold values refer to first place in the experiments.

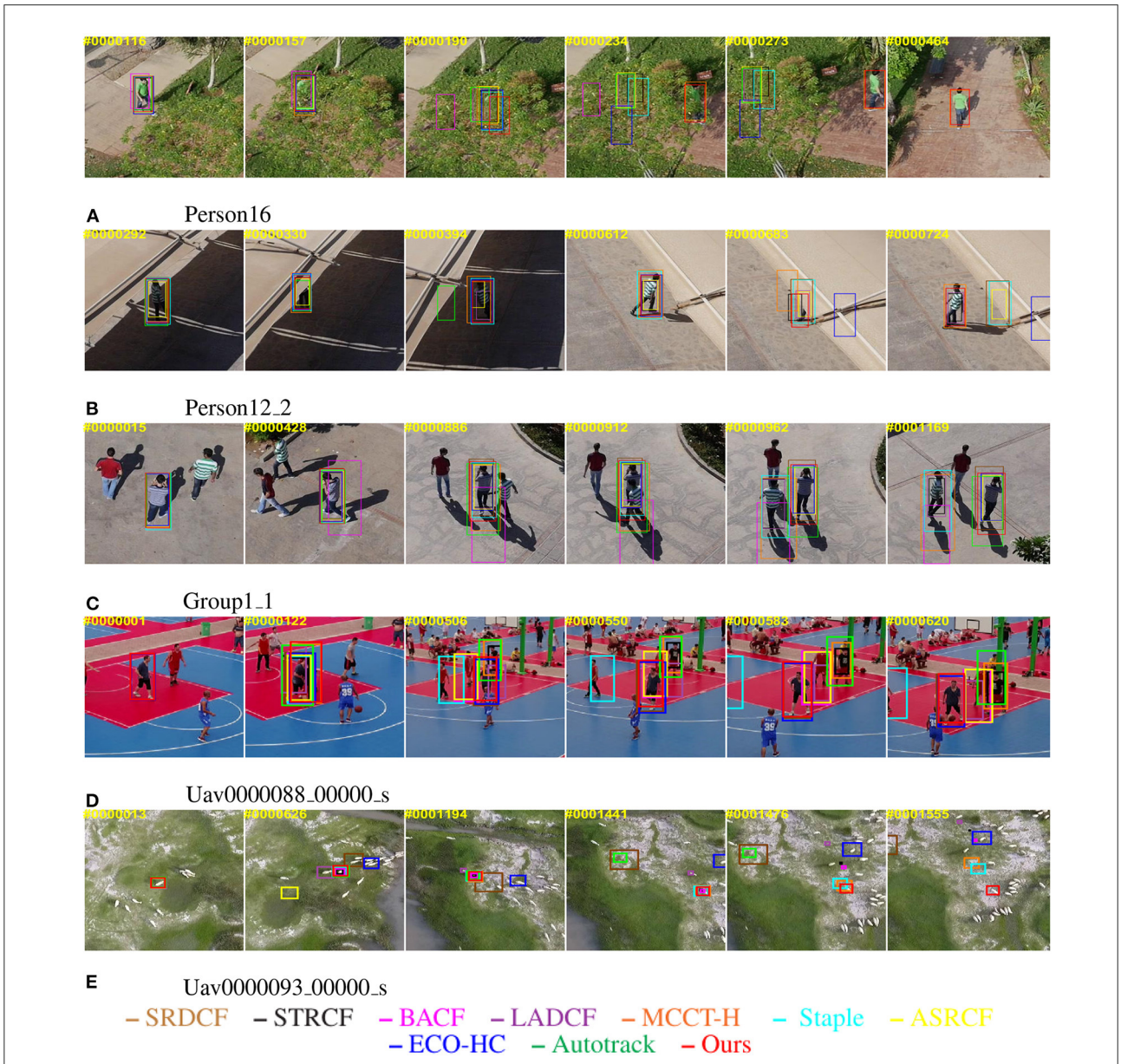


FIGURE 3
Qualitative performance evaluation of the proposed tracker and the other nine most advanced trackers on the typical UAV videos. The number in the upper left corner refers to the frame number. The tracking boxes in different colors represent the tracking results of different trackers in the frame. **(A)** Person16. **(B)** Person12_2. **(C)** Group1_1. **(D)** Uav0000088_00000_s. **(E)** Uav0000093_00000_s. The photos appearing in this figure have been reused from: 'A benchmark and simulator for uav tracking' and 'Vision meets drones: a challenge'. The corresponding website are 'https://cemse.kaust.edu.sa/ivul/uav123' and 'http://aiskyeye.com'.

3.5. Qualitative evaluation

In this subsection, the qualitative comparison is given to the proposed method and the aforementioned 9 state-of-the-art algorithms to better demonstrate the performance of each tracker in Figure 3. The above image sequences (containing person16, person12_2, group1_1 in UAV123 and Uav0000088_00000_s, and Uav0000093_00000_s in VisDrone2018-test-dev) mainly include three challenging attributes, namely similar object (SOB), background clutters (BC), and occlusion (OC). It can be observed that our tracker is effective in solving these difficult issues, and can locate the targets accurately.

When facing a similar object and background clutter, aberrance repressed regularization can help the tracker in accurately perceiving and fully restraining the interference regions in advance. Simultaneously, dynamic feature channel weight realizes the independent filtering of different dimensional features, thereby encouraging the filters to focus on more reliable and discriminative features between the target and a cluttered background. By jointly modeling the above two constraints, the tracker can learn the robust features of the target according to the environment and the interference from a cluttered background.

When there is an occlusion, the trackers can learn the features of the block and lose the target information, thus leading to model drift and a failure of tracking. With the introduction of a historical model retrieval module in our method, the tracker has a memory function similar to the human brain by saving a history template. The method of dynamic updating of the template encourages the tracker to reduce the learning rate when the training sample is abnormal, thereby effectively reducing the probability of template pollution. The memory function of the tracker also guarantees that the method can accurately lock the target again after the disappearance of the occlusion.

In summary, when challenging attributes occur during tracking, the addition of the two constraints endows the tracker with the ability to select the most distinguishing feature for sensing and suppressing the interference around the target, while the historical model retrieval module effectively reduces the pollution of interference and noise to the tracker. However, when meeting viewpoint change and rotation, the performance of our tracker is reduced because of rapid changes in the appearance of the target. In the future, we will explore how to refine tracking results to solve such problems.

4. Conclusion

Based on the idea that the brain can perceive interference information in the background, select the optimal features independently to describe the target, and use historical memory to achieve accurate target location in the current frame, in this paper, we propose a UAV tracking algorithm on the basis of repressed dynamic aberrance and a channel selective correlation filter with a historical model retrieval module combined. By jointly modeling feature channel weight and the aberrance repressed regularization, our tracker could restrain the potential distractors, and highlight the valuable features in the channel domain, thereby constructing a robust target appearance. With a historical model retrieval module, our tracker can make full use of the historical information to update the tracker, while effectively avoiding tracking drift. The experimental results on the two public UAV benchmarks demonstrate that the proposed method achieves better tracking results than the other advanced algorithms.

Data availability statement

The original contributions presented in the study are included in the article/supplementary material, further inquiries can be directed to the corresponding author.

Author contributions

JC and JW proposed the basic idea of this method and wrote the code together. JC completed theoretical modeling. JC and LZ performed the experiments and data analysis. JC wrote the first draft of the manuscript. JW and LZ revised the manuscript. All authors contributed to the article and approved the submitted version.

Funding

This work was supported by the Natural Science Foundation of China (Grant No. 12072027).

Acknowledgments

The authors thank Małgorzata Siemiątkowska, Anna Olichwer, Magdalena Żukowska, Maksymilan Koc, and Karolina Adaszewska, for their significant help with the organization of the study, and the psychophysiological data collection. We also would like to thank Marta Grześ – for their help with the participants' recruitment and contact. We also thank Piotr Kałowski, Dominika Pruszczak, Małgorzata Hanć, and Michał Lewandowski for their help with the misophonia interviews.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

References

Atkinson, R. C., and Shiffrin, R. M. (1968). Human memory: a proposed system and its control processes. *Psychol. Learn. Motivat.* 2, 89–195. doi: 10.1016/S0079-7421(08)60422-3

Baca, T., Hert, D., Loianno, G., Saska, M., and Kumar, V. (2018). "Model predictive trajectory tracking and collision avoidance for reliable outdoor deployment of unmanned aerial vehicles," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Madrid: IEEE), 6753–6760.

Bertinetto, L., Valmadre, J., Golodetz, S., Miksik, O., and Torr, P. H. (2016a). "Staple: complementary learners for real-time tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Las Vegas, NV: IEEE), 1401–1409.

Bertinetto, L., Valmadre, J., Henriques, J. F., Vedaldi, A., and Torr, P. H. (2016b). "Fully-convolutional siamese networks for object tracking," in *European Conference on Computer Vision* (Cham: Springer), 850–865.

Bolme, D. S., Beveridge, J. R., Draper, B. A., and Lui, Y. M. (2010). "Visual object tracking using adaptive correlation filters," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (San Francisco, CA: IEEE), 2544–2550.

Chen, S., Wang, T., Wang, H., Wang, Y., Hong, J., Dong, T., et al. (2022). Vehicle tracking on satellite video based on historical model. *IEEE J. Select. Top. Appl. Earth Observat. Remote Sens.* 15, 7784–7796. doi: 10.1109/JSTARS.2022.3195522

Connor, S., Li, T., Roberts, R., Thakkar, S., Liu, Z., and Tong, W. (2022). The adaptability of using ai for safety evaluation in regulatory science: a case study of assessing drug-induced liver injury (dili). *Front. Artif. Intel.* 5, 1034631. doi: 10.3389/frai.2022.1034631

Cornelio, P., Haggard, P., Hornbaek, K., Georgiou, O., Bergström, J., Obrist, M., et al. (2022). The sense of agency in emerging technologies for human-computer integration: a review. *Front. Neurosci.* 16, 949138. doi: 10.3389/fnins.2022.949138

Dai, K., Wang, D., Lu, H., Sun, C., and Li, J. (2019). "Visual tracking via adaptive spatially-regularized correlation filters," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (Long Beach, CA: IEEE), 4670–4679.

Danelljan, M., Bhat, G., Shahbaz Khan, F., and Felsberg, M. (2017). "ECO: efficient convolution operators for tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Honolulu, HI: IEEE), 6638–6646.

Danelljan, M., Häger, G., Khan, F., and Felsberg, M. (2014a). "Accurate scale estimation for robust visual tracking," in *British Machine Vision Conference, Nottingham* (Nottingham: Bmva Press), 1–5.

Danelljan, M., Hager, G., Shahbaz Khan, F., and Felsberg, M. (2015). "Learning spatially regularized correlation filters for visual tracking," in *Proceedings of the IEEE International Conference on Computer Vision* (Santiago: IEEE), 4310–4318.

Danelljan, M., Shahbaz Khan, F., Felsberg, M., and Van de Weijer, J. (2014b). "Adaptive color attributes for real-time visual tracking," *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Columbus, OH: IEEE), 1090–1097.

Deng, C., Jing, D., Han, Y., Wang, S., and Wang, H. (2022). Far-net: fast anchor refining for arbitrary-oriented object detection. *IEEE Geosci. Remote Sens. Lett.* 19, 1–5. doi: 10.1109/LGRS.2022.3144513

Elloumi, M., Dhaou, R., Escrig, B., Idoudi, H., and Saidane, L. A. (2018). "Monitoring road traffic with a uav-based system," in *2018 IEEE Wireless Communications and Networking Conference (WCNC)* (Barcelona: IEEE), 1–6.

Foksinska, A., Crowder, C., Crouse, A., Henrikson, J., Byrd, W., Rosenblatt, G., et al. (2022). The precision medicine process for treating rare disease using the artificial intelligence tool medikanren. *Front. Artif. Intell.* 5, 910216. doi: 10.3389/frai.2022.910216

Gschwindt, M., Camci, E., Bonatti, R., Wang, W., Kayacan, E., and Scherer, S. (2019). "Can a robot become a movie director? learning artistic principles for aerial cinematography," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Macau: IEEE), 1107–1114.

Han, Y., Deng, C., Zhao, B., and Tao, D. (2019a). State-aware anti-drift object tracking. *IEEE Trans. Image Process.* 28, 4075–4086. doi: 10.1109/TIP.2019.2905984

Han, Y., Deng, C., Zhao, B., and Zhao, B. (2019b). Spatial-temporal context-aware tracking. *IEEE Signal Process. Lett.* 26, 500–504. doi: 10.1109/LSP.2019.2895962

Han, Y., Liu, H., Wang, Y., and Liu, C. (2022). A comprehensive review for typical applications based upon unmanned aerial vehicle platform. *IEEE J. Select. Top. Appl. Earth Observat. Remote Sens.* 15, 9654–9666. doi: 10.1109/JSTARS.2022.3216564

Henriques, J. F., Caseiro, R., Martins, P., and Batista, J. (2014). High-speed tracking with kernelized correlation filters. *IEEE Trans. Pattern Anal. Mach. Intell.* 37, 583–596. doi: 10.1109/TPAMI.2014.2345390

Hong, Z., Chen, Z., Wang, C., Mei, X., Prokhorov, D., and Tao, D. (2015). "Multi-store tracker (muster): a cognitive psychology inspired approach to object tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Boston, MA: IEEE), 749–758.

Javed, S., Danelljan, M., Khan, F. S., Khan, M. H., Felsberg, M., and Matas, J. (2022). Visual object tracking with discriminative filters and siamese networks: a survey and outlook. *IEEE Trans. Pattern Anal. Mach. Intell.* 1–20. doi: 10.1109/TPAMI.2022.3212594

Kiani Galoogahi, H., Fagg, A., and Lucey, S. (2017). "Learning background-aware correlation filters for visual tracking," in *Proceedings of the IEEE International Conference on Computer Vision* (Venice: IEEE), 1135–1143.

Kuroda, N., Ikeda, K., and Teramoto, W. (2022). Visual self-motion information contributes to passable width perception during a bike riding situation. *Front. Neurosci.* 16, 938446. doi: 10.3389/fnins.2022.938446

Li, B., Yan, J., Wu, W., Zhu, Z., and Hu, X. (2018a). "High performance visual tracking with siamese region proposal network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Salt Lake City, UT: IEEE), 8971–8980.

Li, F., Tian, C., Zuo, W., Zhang, L., and Yang, M.-H. (2018b). "Learning spatial-temporal regularized correlation filters for visual tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Salt Lake City, UT: IEEE), 4904–4913.

Li, Y., Fu, C., Ding, F., Huang, Z., and Lu, G. (2020). "Autotrack: Towards high-performance visual tracking for uav with automatic spatio-temporal regularization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Seattle, 11923–11932.

Li, Y., and Zhu, J. (2014). "A scale adaptive kernel correlation filter tracker with feature integration," in *European Conference on Computer Vision* (Cham: Springer), 254–265.

Lin, Z., Chen, M., and Ma, Y. (2010). The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. *arXiv preprint arXiv:1009.5055*. doi: 10.48550/arXiv.1009.5055

Liu, C., Ding, W., Chen, P., Zhuang, B., Wang, Y., Zhao, Y., et al. (2022). Rb-net: training highly accurate and efficient binary neural networks with reshaped point-wise convolution and balanced activation. *IEEE Trans. Circ. Syst. Video Technol.* 32, 6414–6424. doi: 10.1109/TCSVT.2022.3166803

Mueller, M., Smith, N., and Ghanem, B. (2016). "A benchmark and simulator for uav tracking," in *European Conference on Computer Vision* (Cham: Springer), 445–461.

Mueller, M., Smith, N., and Ghanem, B. (2017). "Context-aware correlation filter tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Honolulu, H: IEEE), 1396–1404.

Nofallah, S., Wu, W., Liu, K., Ghezloo, F., Elmore, J. G., and Shapiro, L. G. (2022). Automated analysis of whole slide digital skin biopsy images. *Front. Artif. Intell.* 5, 1005086. doi: 10.3389/frai.2022.1005086

Pfeifer, L. D., Patabandige, M. W., and Desaire, H. (2022). Leveraging r (levr) for fast processing of mass spectrometry data and machine learning: applications analyzing fingerprints and glycopeptides. *Front. Anal. Sci.* 2, 961592. doi: 10.3389/frans.2022.961592

Shao, J., Du, B., Wu, C., and Zhang, L. (2019). Tracking objects from satellite videos: a velocity feature based correlation filter. *IEEE Trans. Geosci. Remote Sens.* 57, 7860–7871. doi: 10.1109/TGRS.2019.2916953

Voigtlaender, P., Luiten, J., Torr, P. H., and Leibe, B. (2020). "Siam r-CNN: visual tracking by re-detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Seattle, 6578–6588.

Wang, N., Zhou, W., Tian, Q., Hong, R., Wang, M., and Li, H. (2018). "Multi-cue correlation filters for robust visual tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Salt Lake City, UT: IEEE), 4844–4853.

Wang, Q., Zhang, L., Bertinetto, L., Hu, W., and Torr, P. H. (2019). "Fast online object tracking and segmentation: a unifying approach," in *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition* (Long Beach, CA: IEEE), 1328–1338.

Wang, W., Han, Y., Deng, C., and Li, Z. (2022). Hyperspectral image classification via deep structure dictionary learning. *Remote Sens.* 14, 2266. doi: 10.3390/rs14092266

Xu, T., Feng, Z.-H., Wu, X.-J., and Kittler, J. (2019). Learning adaptive discriminative correlation filters via temporal consistency preserving spatial feature selection for robust visual object tracking. *IEEE Trans. Image Process.* 28, 5596–5609. doi: 10.1109/TIP.2019.2919201

Zhu, P., Wen, L., Bian, X., Ling, H., and Hu, Q. (2018). Vision meets drones: a challenge. *arXiv preprint arXiv:1804.07437*. doi: 10.48550/arXiv.1804.07437

frontiers | Frontiers in Neuroscience

TYPE Original Research
PUBLISHED 16 January 2023
DOI 10.3389/fnins.2022.1117134

OPEN ACCESS

EDITED BY
Chenwei Deng,
Beijing Institute of Technology, China

REVIEWED BY
Jian Jia,
University of Chinese Academy of Sciences, China
Anca Marginean,
Technical University of Cluj-Napoca, Romania
Wenzhen Haung,
Tsinghua University, China
Qiaozhe Li,
Chinese Academy of Sciences (CAS), China

*CORRESPONDENCE
Youxin Chen
✉ chenyx@pumch.cn

SPECIALTY SECTION
This article was submitted to Perception Science, a section of the journal Frontiers in Neuroscience

RECEIVED 06 December 2022
ACCEPTED 28 December 2022
PUBLISHED 16 January 2023

CITATION
Xu J, Yang K, Chen Y, Dai L, Zhang D, Shuai P, Shi R and Yang Z (2023) Reliable and stable fundus image registration based on brain-inspired spatially-varying adaptive pyramid context aggregation network. *Front. Neurosci.* 16:1117134. doi: 10.3389/fnins.2022.1117134

Reliable and stable fundus image registration based on brain-inspired spatially-varying adaptive pyramid context aggregation network

Jie Xu[1], Kang Yang[2], Youxin Chen[3]*, Liming Dai[2], Dongdong Zhang[2], Ping Shuai[4,5], Rongjie Shi[2] and Zhanbo Yang[2]

[1]Beijing Institute of Ophthalmology, Beijing Tongren Eye Center, Beijing Tongren Hospital, Capital Medical University, Beijing Key Laboratory of Ophthalmology and Visual Sciences, Beijing, China, [2]Beijing Zhizhen Internet Technology Co. Ltd., Beijing, China, [3]Department of Ophthalmology, Peking Union Medical College Hospital, Beijing, China, [4]Department of Health Management and Physical Examination, Sichuan Provincial People's Hospital, University of Electronic Science and Technology of China, Chengdu, China, [5]School of Medicine, University of Electronic Science and Technology of China, Chengdu, China

The task of fundus image registration aims to find matching keypoints between an image pair. Traditional methods detect the keypoint by hand-designed features, which fail to cope with complex application scenarios. Due to the strong feature learning ability of deep neural network, current image registration methods based on deep learning directly learn to align the geometric transformation between the reference image and test image in an end-to-end manner. Another mainstream of this task aims to learn the displacement vector field between the image pair. In this way, the image registration has achieved significant advances. However, due to the complicated vascular morphology of retinal image, such as texture and shape, current widely used image registration methods based on deep learning fail to achieve reliable and stable keypoint detection and registration results. To this end, in this paper, we aim to bridge this gap. Concretely, since the vessel crossing and branching points can reliably and stably characterize the key components of fundus image, we propose to learn to detect and match all the crossing and branching points of the input images based on a single deep neural network. Moreover, in order to accurately locate the keypoints and learn discriminative feature embedding, a brain-inspired spatially-varying adaptive pyramid context aggregation network is proposed to incorporate the contextual cues under the supervision of structured triplet ranking loss. Experimental results show that the proposed method achieves more accurate registration results with significant speed advantage.

KEYWORDS

retinal image analysis, fundus image registration, deep learning, context aggregation, structured triplet ranking loss

1. Introduction

Fundus image analysis has been widely researched, due to its significant advantage of non-invasive observation. The purpose of image registration (Hill et al., 2001; Sotiras et al., 2013) is to deform the test image to the coordinate system of the reference image, so that the same point can be imaged at the same coordinate of the two images (Oliveira and Tavares, 2014).

Registration of medical images is a crucial step in the image processing. Image registration can trace the progression of the same patient through time, providing a basis for clinical diagnosis, lowering physician effort, and aiding in the investigation of disease prognosis and outcome. In order to accurately learn the deformation coefficient to transform the test image, the matching keypoints between the test image and reference image should be obtained. To this end, previous methods rely on human-designed features to distinguish among visually similar keypoints, by encoding the texture, shape or intensity gradient with particularly designed computing pattern. Recently, deep neural network (DNN) (Krizhevsky et al., 2012; Simonyan and Zisserman, 2014; He et al., 2016) based image registration has made rapid progress due to its strong feature learning ability. Some current DNN based methods propose to directly learn the geometric transformation, such as homography transformation, between the test image and reference image. Other works also aim to learn the dense pixel-level displacement vector filed between the image pair (Cao et al., 2017; Krebs et al., 2017). However, due to the complex and variable retinal vascular structure, previous methods fail to achieve **reliable** and **stable** registration performance, which severely limits downstream applications. Considering that the vessel crossing and branching points are able to reliably and stably characterize the fundus image (Deng et al., 2010; Chen et al., 2011), we propose to choose all the crossing and branching points as the keypoints. To this end, a single deep neural network is utilized to learn to simultaneously locate and match all the keypoints.

Since the lower-level spatial details and higher-level semantic cues of fundus image are both critical for learning accurate keypoint detection and corresponding discriminative feature embedding for keypoint matching, we employ the widely used encoder-decoder architecture (Ronneberger et al., 2015) as the basic network. Moreover, due to the large intra-class variability and small inter-class difference of fundus image, the non-matching keypoints are prone to be misclassified. It is natural for human being to gain the knowledge of contextual consistency, which is helpful for alleviating this issue. As a result, contextual cues should be incorporated into the vanilla encoder-decoder architecture to handle these critical issues. To this end, on the basis of the encoder-decoder architecture, we propose a brain-inspired spatially-varying adaptive pyramid context aggregation network. Concretely, with the proposed spatially-varying adaptive pyramid context aggregation module, every pixel location of the feature map is reweighted with the learned weight factor guided by the aggregated global contextual cues. Feature vectors of any two pixel locations are explicitly interacted by the form of matrix multiplication between the reshaped two-dimensional feature maps, leading to the spatially-varying feature weight factors. The generated weight factors are then utilized as the dilated depth-wise convolution kernels with different dilation factors to aggregate the contextual cues in receptive fields with multiple scales. In this way, the contextual cues are integrated into the feature maps with predictable and spatially-varying depth-wise convolutions. In addition, we employ a structured triplet ranking loss, whose aim is to supervise the network to enlarge the distance of feature embedding between non-matching keypoints and narrow the distance of feature embedding between the matching keypoints, leading to compactness between matching keypoints and dispersion between non-matching keypoints.

In order to verify the effectiveness of the proposed method, proper dataset and evaluation metric should be elaborately designed. However, current FIRE dataset (Hernandez-Matas et al., 2017) only labels a small part of the keypoints. Meanwhile, some keypoints of FIRE dataset are not located at branching or crossing points. So this dataset can't be used for training our proposed model. To this end, we collect 200 retinal images of 50 patients taken with fundus camera by RetCam3 and Canon. Concretely, 100 neonatal fundus images of 27 patients with low imaging quality are taken from RetCam3. Another 100 high-quality retinal images of 23 patients taken from Canon are also included. Meanwhile, different imaging angles and diverse overlapping areas between the image pair are also considered during the construction of dataset. In order to quantitatively evaluate the proposed method, following previous methods (Hernandez-Matas et al., 2017), we choose the Area Under Curve (AUC) value as the registration score. Experimental results demonstrate that our proposed method achieves significant performance improvement over the vanilla encoder-decoder network. Our method achieves the best registration performance among the deep learning based methods. Meanwhile, our proposed method also surpasses most of the traditional registration methods with significantly faster execution speed by an order of magnitude.

Our contributions are summarized into three parts:

- We propose to achieve **reliable** and **stable** keypoint detection and registration results for fundus image. Considering that the vessel crossing and branching points can reliably and stably characterize the key components of fundus image, we propose to learn to detect and match all the crossing and branching points of the input image pair with a single deep neural network.
- In order to cope with the large intra-class variability and small inter-class difference of retinal image, we propose a brain-inspired spatially-varying adaptive pyramid context aggregation based on the widely used encoder-decoder architecture. In this way, long-range contextual cues are incorporated into the feature maps with predictable and input-variant convolutions. Moreover, a structured triplet ranking loss is employed to enforce the network to produce similar feature embedding for matching keypoints in the input image pair, and dissimilar feature embedding for non-matching keypoints.
- Since there is no proper fundus image registration dataset for method evaluation, we construct a large-scale dataset which covers diverse application scenarios. Quantitative and qualitative results show that our proposed method is able to reliably and stably locate and match keypoints.

We organize our paper as follows. Section 2 reviews related work. Section 3 shows the detail of our method. Section 4 demonstrates experimental results. Finally, Section 5 presents our conclusion.

2. Related work

2.1. Deep learning based image registration

Since the learning based image registration is mainly considered in this paper, we provide a brief review of related works on deep learning based image registration in this part. In recent years, several

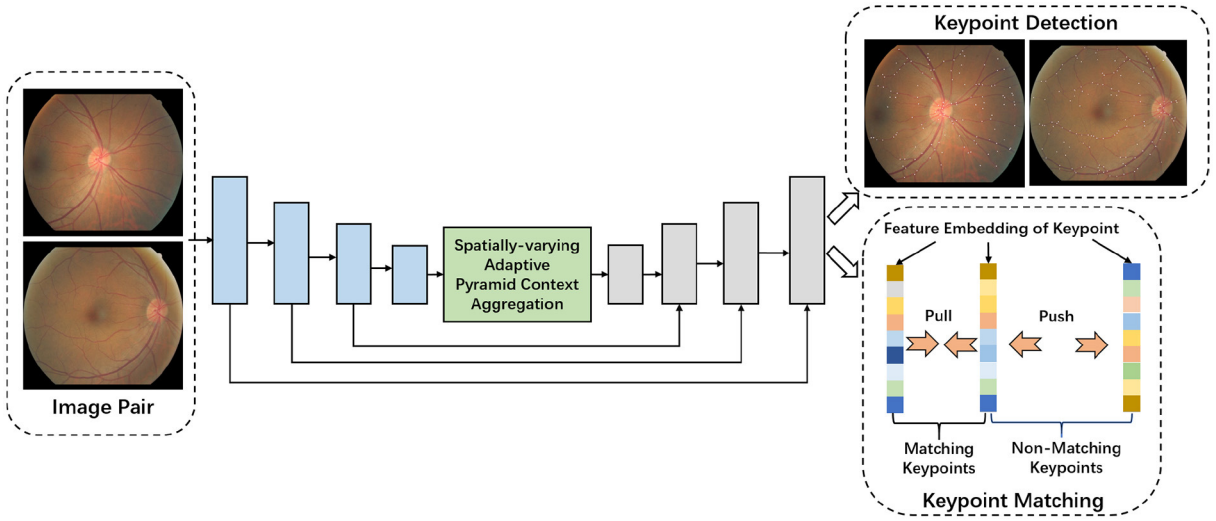


FIGURE 1
Overview of the proposed network for simultaneous keypoints detection and keypoints matching.

methods (Cao et al., 2017; Krebs et al., 2017) have proposed to employ the DNN to directly learn the warp field between the test image and reference image. Ground truth warp fields are required in the above methods (Rohé et al., 2017; Sokooti et al., 2017; Yang et al., 2017) to supervise the learning of DNN. In order to obtain the ground truth warp field, several methods propose to simulate the deformation operation and generate deformed images. Some other methods employ the classical registration method, which rely on hand-designed feature. However, the above methods are difficult to obtain ground truth warp field as the ground reality, which severely limit the application in real scenario. Recently, several unsupervised learning based image registration methods (Li and Fan, 2017; Vos et al., 2017; Zou et al., 2020) are also proposed. However, these methods fail to cope with complex image registration application, such as large transformations (Vos et al., 2017).

Compared to images collected in our daily life, the retinal image registration is a much more challenging problem. First, there are large differences in illumination, color, contrast and imaging angles of the input image pair in diverse scenarios. The overlapping areas between the test image and the reference image may be also diverse. Furthermore, significant changes in retinal structure may be caused by the progression of retinopathy. As a result, current deep learning based image registration methods fail to achieve reliable and stable registration results, which are not applicable for the challenging fundus image task.

2.2. Deep metric learning

Deep metric learning aims to learn the distance metric to compare and measure similarity between pairs of examples, which is important for various tasks, such as image retrieval (Sohn, 2016; Movshovitz-Attias et al., 2017), clustering (Hershey et al., 2016). One of the main task of deep metric learning is to design proper loss function. Contrastive loss (Chopra et al., 2005; Hadsell et al., 2006) aims to encode the pair-wise relations between the anchor example and one similar(positive) or dissimilar(negative) example, which is first proposed to learn the feature embedding for image search task. Triplet loss (Wang et al., 2014; Schroff et al., 2015; Cui et al., 2016) is used to learn feature embedding for face recognition task. A triplet is composed of the anchor example, a positive example and a negative example. The triplet loss is to learn a distance metric by which the anchor point is closer to the similar point than the dissimilar one by a margin. Recently, richer structural relations among multiple examples are considered by ranking-motivated methods (Schroff et al., 2015; Oh Song et al., 2016; Sohn, 2016; Law et al., 2017; Movshovitz-Attias et al., 2017). Some other methods propose to design clustering-motivated structured losses (Hershey et al., 2016; Oh Song et al., 2017). However, since clustering-motivated losses are more difficult to optimize, the ranking-motivated loss function is mainly considered in this paper.

3. Method details

This section presents details of our method for reliable and stable fundus image registration. We show the overview of the proposed model in Figure 1. We start by introducing the encoder-decoder network, which is the baseline of our model. Then we introduce the proposed network architecture and employed loss function.

3.1. Encoder-decoder network architecture

For the fundus image registration method based on deep neural network (DNN), in order to achieve accurate pixel-level image registration results, robust global semantic information

and rich local spatial details are required. Current DNN stacks successive convolutional and pooling layers to obtain roust feature representations. However, due to the multiple pooling operations, the feature spatial resolution is largely reduced. As a result, local spatial details are severely lost for the features in deeper-level layers. On the contrary, due to fewer pooling layers, spatial resolution of features in lower-level layers are larger. In this way, the features in lower-level layers encode rich local spatial details. However, the lack of semantic and discriminative cues make the lower-level features fail to effectively model long-range information. Since both the local spatial details and global semantic cues are essential for accurate image registration performance, a balanced fusion of the lower-level features and the deeper-level features is required.

As shown in Figure 1, current widely used encoder-decoder architecture employs the encoder sub-network to extract the multi-scale features by multiple stacked convolutional and pooling operations. The later decoder sub-network then combines the extracted multi-level features by multiple feature fusion operations. Concretely, with the input image pair, the successive convolutional and pooling layers of encoder sub-network extract multi-scale features, similar to ResNet (He et al., 2016) or VGGNet (Simonyan and Zisserman, 2014). The decoder sub-network consists of multiple feature fusion operations, which are employed to fuse the multi-scale features generated by the encoder sub-network progressively. For every fusion operation, $\hat{F}_i$, the feature in current layer i, is first upsampled to match the resolution of the feature map F_{i-1} from the lower neighbor layer $i-1$. The feature concatenation along the channel dimension is applied, which is followed by another convolution for further feature abstraction. This operation can be formulated as:

$$\hat{F_{i-1}} = Conv(Concat(Up((\hat{F}_i)), F_{i-1})). \quad (1)$$

The above fusion operation is iterated until the lowest layer, where the generated feature F_1 has the same spatial resolution as the input image, which is used to produce the final prediction.

3.2. Spatially-varying context aggregation module

Due to the large intra-class variability and small inter-class difference of fundus image, the non-matching keypoints are prone to be misclassified. As a result, contextual cues should be incorporated into the vanilla encoder-decoder architecture to handle this critical issue (Liu et al., 2020). To this end, with the deepest feature map generated by the encoder, a novel context aggregation module is applied to incorporate the contextual cues in a spatially-varying manner. The details are illustrated below.

In order to model the long-range contextual cues, previous methods are mainly designed to generate global-consistent feature re-weighting coefficient. For example, SE-Net (Jie et al., 2019) is proposed to produce channel-wise feature re-weighting factor of global distribution by a squeeze-and-excitation mechanism. Differently, we propose to aggregate the global contextual cues by generating spatially-varying feature re-weighting factors. In this way, the long-range relations are more effectively mined in a spatially-varying manner.

Figure 2 shows the overall architecture of the proposed Spatially-varying Context Aggregation (SCA) module. First, we explicitly model the long-range relations between any two pixel locations by matrix multiplication, generating spatially-varying context kernel prediction. Then, the predicted context kernels are applied on the original feature map, leading to aggregated context enhanced feature. Following are the detailed processing pipeline.

With the feature map $X \in R^{H\times W\times C}$ generated by the last feature block of the encoder, we first transform it into two forms with two independent convolutional operations: the *key* and *query*. The H, W and C refer to the hight, width and channel number, respectively. The *key* feature map $K \in R^{H\times W\times C}$ and the *query* feature map $Q \in R^{H\times W\times s^2}$ are then used to aggregate the contextual cues. Here, s is the kernel size of the learned context kernel.

In order to effectively model the global contextual cues between pixels, the relation within any pixel locations should be explicitly interacted. To this end, the *key* feature map $K \in R^{H\times W\times C}$ and the *query* feature map $Q \in R^{H\times W\times s^2}$ are first reshaped into 2D form, $K \in Q^{H\times W\times C}$ and $Q^{'} \in R^{(H\times W)\times s^2}$, respectively. In this way, our aim is to make each column of K effectively encodes the channel-wise characteristics of original feature map X along the channel dimension C. The length of each of the $C-$dimensional feature vector is $H \times W$. Meanwhile, each column of $Q^{'}$ models one of the s^2-dimensional feature vectors with the length of $H \times W$.

Afterwards, in order to explicitly model the interactions between each column of $K \in Q^{H\times W\times C}$ and $Q^{'} \in R^{(H\times W)\times s^2}$ for all the $(H \times W)$ pixel locations, we employ following operations:

$$S^{'}(i,j) = \sum_{q=1}^{H\times W} Q^{'}(q,i) \times K^{'}(q,j), \quad (2)$$

Where $i = 1, 2, ..., s^2, j = 1, 2, ..., C$. Since the number of *query* vectors is s^2, s^2 feature vectors encoded the interactions between all the pixel locations can be thus obtained. The length of each of the feature vector is C. We can also rewrite the above operation of dot product form as a form of matrix multiplication:

$$S^{'} = Q^{'T} \times K^{'}, \quad (3)$$

Where $Q^{'T}$ refers to the transpose of matrix $Q^{'}$, $S^{'} \in R^{s^2\times C}$ is the union of all the obtained cues about spatial location relation.

Then, the generated two-dimensional $S^{'} \in R^{s^2\times C}$ is reshaped into 3D form $S \in R^{s\times s\times C}$. We then employ a batch normalization operation to modulate S, generating the predicted spatially-varying context kernel. The generated kernel effectively encodes the relation cues between pixels of all spatial locations, which can be used to produce spatially-varying weight factor $F \in R^{H\times W\times C}$ for all $H \times W$ spatial locations.

In order to fully exploit the information encoded in the spatially-varying context kernel, the depth-wise convolution is applied on the original feature map X with the context kernel S as the depth-wise convolution kernel. In this way, each channel of S is able to modulate one specific channel of X in an independent manner. The spatially-varying context guided modulation can thus be implemented. Concretely, as shown in Figure 3, we first split the context kernel $S \in R^{s\times s\times C}$ into C two-dimensional kernels along the channel dimension. Each of the 2D C kernels has a spatial dimension of $s \times s$. These C kernels are then applied on each channel of the original feature map $X \in R^{H\times W\times C}$ in an independent manner,

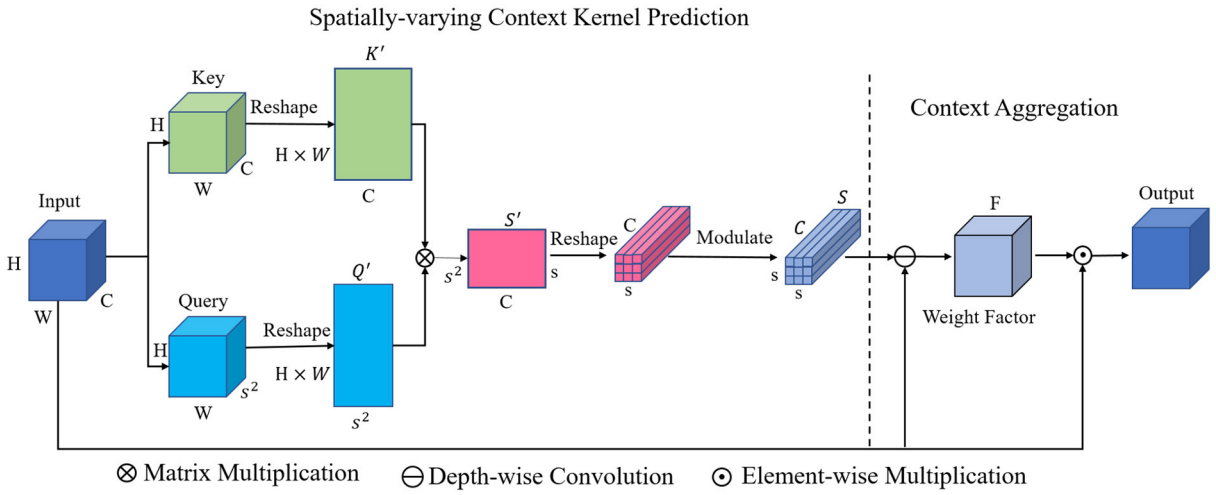


FIGURE 2
Details of the proposed spatially-varying context aggregation module, which first predicts the spatially-varying context kernel and then aggregates the context with the predicted re-weight kernels.

generating intermediate feature. A $1 \times 1 \times 1$ convolution is then used to transform the generated intermediate feature map for further feature abstraction. The obtained feature is then processed with one Sigmoid activation function, which produces the spatially-varying weight factor $F \in R^{H \times W \times C}$. Finally, an element-wise multiplication between M and X is performed to achieve the output feature map, which is then passed through the decoder part for multi-scale feature fusion.

3.3. Spatially-varying adaptive pyramid context aggregation module

3.3.1. Dilated convolution

Standard convolution is characterized by its property of local receptive field. However, large receptive field is essential for enhancing deep neural network's discriminative feature learning ability. Hence, pooling layer is used after several convolutional layers to enlarge the receptive field. However, the adoption of pooling layer leads to the loss of spatial details and lower-resolution feature map, which is unfavorable for accurate pixel-level keypoint location and matching. Dilated convolution is able to effectively alleviate this challenging issue by sparsifying the standard convolution separated by zero with specific interval (dilation rate), which allows us to enlarge the receptive field without loss of spatial resolution of the feature map.

3.3.2. Depth-wise dilated convolution

Depth-wise separable convolution transforms the standard convolution into a depth-wise convolution followed by a point-wise convolution. In this way, the computation complexity is thus drastically reduced. Concretely, the depth-wise convolution is applied on each channel of the feature map independently. The point-wise convolution is then used to fuse the output from the depth-wise convolution.

On the basis of the context aggregation module above, a dilation pyramid based context aggregation module is incorporated for further context aggregation of multi-scale field-of-view, as shown in Figure 3. Concretely, with the predicted spatially-varying context kernel S, we employ three parallel dilated convolutions with different dilation rates to model contextual cues in a context-adaptive manner. The three different dilation rates are set as 1, 3, 5 in our paper. In this way, a dilation pyramid context aggregation block is obtained.

With these operations, three context kernels ($S1$, $S2$, and $S3$) with different context aggregation fields are obtained. The three context kernels are then applied over the original feature map X, leading to three different weight factors R_1, R_2, and R_3. The three generated weight factors are then fused by element-wise sum:

$$R = R_1 + R_2 + R_3. \tag{4}$$

With the final fused weight kernel R, similar to the above SCA module, an element-wise multiplication is operated between R and X to ensure each channel of R can independently modulate the corresponding channel of X.

3.4. Loss function

In order to supervise the above network to effectively locate and match the keypoints, specifically designed loss functions are utilized.

3.4.1. Keypoint location loss

We convert the keypoint location task into a pixel-level binary classification problem. In order to accurately locate the keypoints, the widely used cross-entropy loss is first utilized to supervise the learning of the transformed feature map of the last feature block of the above adaptive pyramid context aggregation network:

$$CE(y_i, p_i) = -[y_i log(p_i) + (1 - y_i) log(1 - p_i)], \tag{5}$$

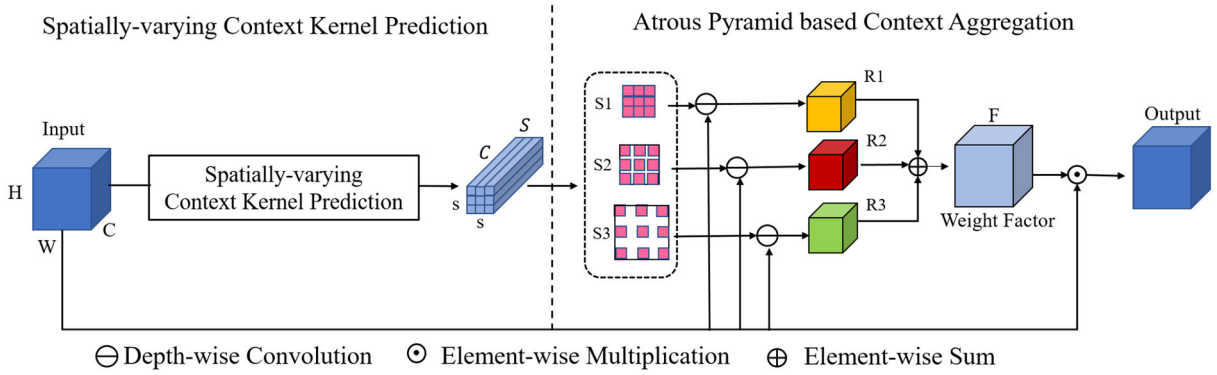


FIGURE 3
Details of the proposed spatially-varying adaptive pyramid context aggregation module. The adaptive pyramid mechanism aggregates the contextual cues with multi-scale field-of-views via convolution pyramid with multiple atrous rates.

Where y_i means the label of pixel i (1 and 0 means the keypoint and background, respectively), p_i refers to the predicted probability of pixel i to be the keypoint.

We also use the Dice loss for more accurate keypoint location:

$$Dice(X, Y) = 1 - \frac{2|P \cap Y|}{|P| + |Y|}, \tag{6}$$

Where P means the pixel set of the predicted keypoints, Y means the pixel set of ground truth keypoints. $|P \cap Y|$ refers to the sum of the element-wise production between P and Y. $|P| + |Y|$, $|P|$ refers to the sum of all the elements of P, $|Y|$ refers to the sum of all the elements of Y.

3.4.2. Keypoint matching loss

In order to supervise the network to enhance the discriminative power of learned feature embedding of keypoints, proper keypoint matching loss should be designed. The ideal keypoint matching loss should reduce the gap between matching keypoints and enlarge the gap between non-matching keypoints.

To this end, with the feature map in the last feature block of decoder before generating keypoint detection prediction, we transform this feature map into three-dimensional feature embedding. Thus, every fundus image keypoint has its corresponding one-dimensional feature embedding. Following Huang et al. (2016) and Opitz et al. (2017), we set the feature embedding dimension as 512. In this way, our task is to enlarge the distance of feature embedding between non-matching keypoints and narrow the distance of feature embedding between the matching keypoints, leading to compactness between matching keypoints and dispersion between non-matching keypoints. Metric learning mechanism is employed to tackle the above problem in this paper. Concretely, we use the ranking loss to compute the relative distance between the one dimensional feature embedding of every two keypoints in the input image pair.

3.4.2.1. Pair-wise ranking loss

This widely used loss is also called contrastive loss. Positive and negative pairs of the one-dimensional feature embedding of keypoints in input image pair are both required for computing the pair-wise ranking loss. One positive pair consists of an anchor keypoint k_a and the matching keypoint k_p. One negative pair consists of an anchor keypoint and a non-matching keypoint k_n. The one-dimensional feature embedding of the anchor keypoint k_a, the matching keypoint k_p and the non-matching keypoint k_n are f_a, f_p, and f_n, respectively. For positive pairs, the aim of the pair-wise ranking loss is to guide the network to learn proper feature embedding with a small distance. On the contrary, for negative pairs, the pair-wise ranking loss aims to supervise the network to learn feature embedding with a large distance. We choose the Euclidian distance as the distance computing function to measure the similarity between the feature embedding. The above operations can be formulated as:

$$L(f_a, f_p, f_n) = \begin{cases} d(f_a, f_p), & if\ PostivePair, \\ max(0, m - d(f_a, f_n)), & if\ NegativePair. \end{cases} \tag{7}$$

As shown in the Equation 7, for one positive pair, if the distance between f_a and f_p are larger than 0, the loss value will also be positive. Hence, the network is guided to reduce the distance to be 0. In this way, this pair-wise ranking loss guides the network to produce similar feature embedding for matching keypoints. On the other hand, for negative pair, when the distance between the feature embedding of the anchor keypoint and negative (non-matching) keypoint is larger than a specific margin threshold, the loss will be 0. When the distance is reduced below the margin value, the loss value will be positive. When the distance between f_a and f_p, the loss value is the largest value m. In this way, the pair-wise ranking loss supervises the network to produce dissimilar feature embedding for non-matching keypoints. When the distance for a negative pair is distant enough (larger than the default threshold), the network will focus on the learning of feature embedding for more difficult pairs.

3.4.2.2. Triplet ranking loss

Instead of using only one pair of keypoints for every computation of pair-wise ranking loss, the triplet ranking loss considers the relations of a triplet, which consists of an anchor keypoint k_a, a positive keypoint k_p and a negative keypoint k_n. The aim of the triplet ranking loss is to guide the network to produce separable feature embedding: the distance between the feature embedding of

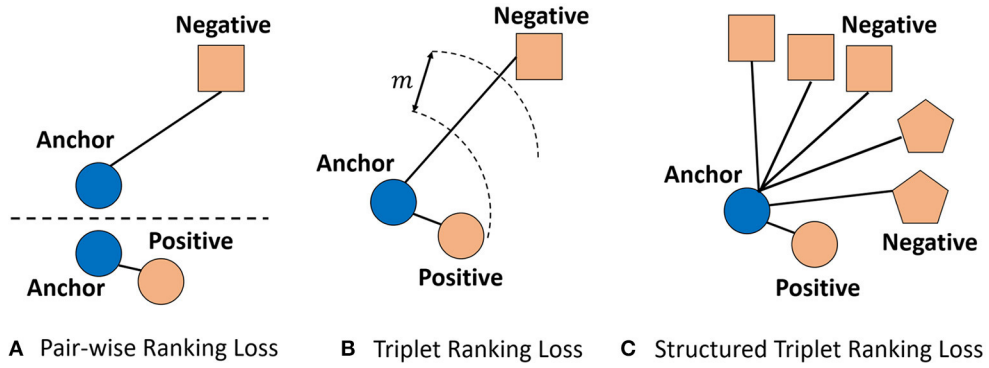


FIGURE 4
Illustration of the **(A)** Pair-wise Ranking loss, **(B)** Triplet Ranking loss, and **(C)** Structured Triplet Ranking loss. Different shapes represent different classes. The blue circle is an anchor. For Pair-wise Ranking loss, the anchor and one positive example or one negative example are considered for every loss computation. For Triplet Ranking loss, the anchor is compared with only one negative example and one positive example. For the Structured Triplet Ranking loss, the anchor is compared with all negative examples.

the anchor keypoint and negative keypoint $d(f_a, f_n)$ is larger than the distance between the feature embedding of anchor keypoint and the positive keypoint $d(f_a, r_p)$ by a specific margin m). The above operations can be rewritten as:

$$L(f_a, f_p, f_n) = max(0, m + d(f_a, f_p) - d(f_a, f_n)). \quad (8)$$

We note that the difference between the pair-wise ranking loss and triplet ranking loss is that pair-wise ranking loss only considers pair of keypoints for one loss computation, however, a triplet of anchor keypoint, positive keypoint and negative keypoint is considered for the triplet ranking loss.

3.4.2.3. Structured triplet ranking loss

Triplet loss (Weinberger and Saul, 2009; Schroff et al., 2015) is proposed to pull the learned feature embedding of anchor keypoint closer to the positive keypoint than to the negative keypoint by a fixed margin. However, the triplet loss only considers one triplet for every loss computation, neglecting the relations among multiple keypoints. To this end, inspired from Oh Song et al. (2016); Wang X. et al. (2019), we propose to employ the structured triplet ranking loss to supervise the feature embedding learning of our network, which explores the structured relationship among multiple keypoints.

Concretely, the structured triplet ranking loss encourages the interaction between more negative keypoints. On the basis of triplet loss, the employed structured triplet ranking loss aims to supervise the learned feature embedding between the anchor keypoint and one positive keypoint is as similar as possible. Moreover, the feature embedding between the anchor keypoint and all negative keypoints as dissimilar as possible. Formally, the structured triplet ranking loss aims to pull the anchor keypoint closer to one positive keypoint than all negative keypoints than a margin m.

$$L = \frac{1}{2|\mathbf{P}|} \sum_{(i,j)\in\mathbf{P}} [d(f_i, f_j) + log(\sum_{(i,p)\in\mathbf{N}} exp(m - d(f_i, f_p)) + \sum_{(j,l)\in\mathbf{N}} exp(m - d(f_j, f_l))]_+, \quad (9)$$

Where $\mathbf{P}$ and $\mathbf{N}$ are the set of positive pairs and negative pairs respectively, f_i, f_p, f_j, and f_l refer to the feature embedding of pixel i, pixel p, pixel j, and pixel l, respectively. $[\cdot]_+$ is the hinge function. Illustration of the Pair-wise Ranking loss, Triplet Ranking loss, and Structured Triplet Ranking loss are shown in Figure 4.

TABLE 1 Details of our constructed AN-200 dataset.

	Camera	Number of image pairs	Number of patients
Adult	Canon	100	27
Neonatus	RetCam3	100	23

We collect and label 200 fundus image pairs of adult and neonatus, which are taken from Canon and RetCam3.

3.5. Implementation details

The hyperparameters of batch-size, weight decay are set to 1, $1e-3$ respectively. The monmentum is set as 0.9. We use pytorch (Paszke et al., 2017) as the basic implement architecture. The widely used stochastic gradient descent strategy is used for training the proposed model.

4. Experiments

In this section, we present extensive experiments to validate the proposed model for fundus image registration. First, we show our evaluation dataset and metric. Then we present a detailed analysis of our model on the constructed large-scale dataset.

4.1. Datasets and metrics

4.1.1. Dataset

Current widely used funds image registration dataset, FIRE, consists of 134 image pairs from 39 patients, which are acquired with Nidek AFC-210 fundus camera. The keypoints of images in FIRE dataset are randomly labeled in a sparse manner. There is not

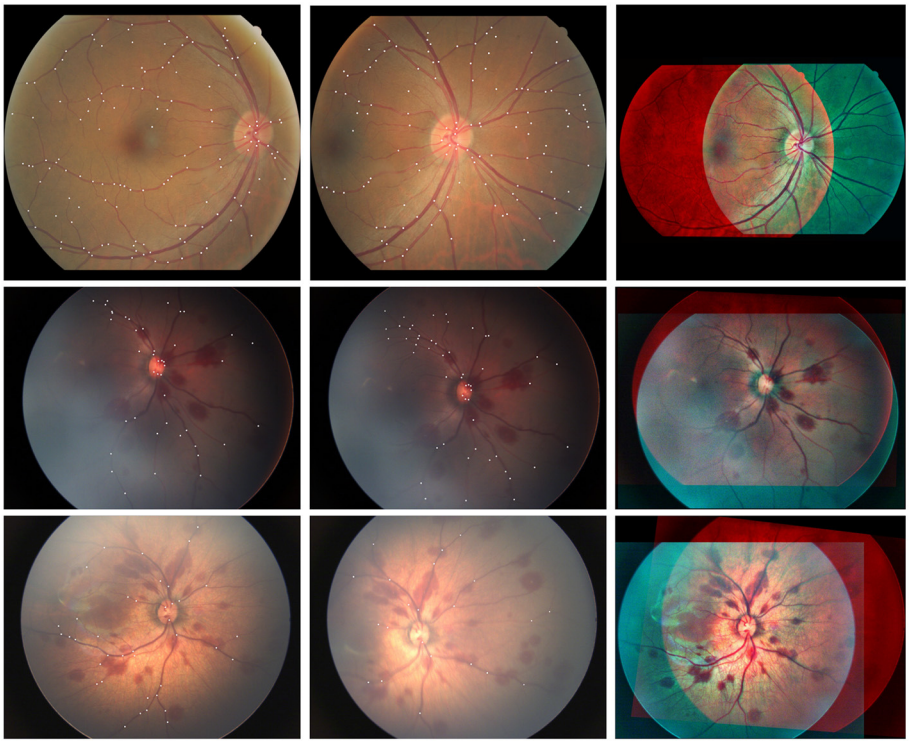


FIGURE 5
Example of the fundus images from diverse applications, including adult and neonatus patients acquired under good or bad imaging conditions. Moreover, different imaging angles and overlapping areas between the image pairs are also considered.

a guarantee that all the vessel branching and crossing points are labeled as keypoints. In this case, these sparse ground-truth keypoint labelings fail to train our proposed model. As a result, a large-scale fundus image registration dataset, which labels all the keypoints in a reliable and stable manner, is required for further research.

To this end, we collect 200 pairs of fundus images under various imaging conditions (illumination, angle etc.) taken from different fundus cameras, such as Canon and RetCam3, as shown in Table 1. The constructed dataset is termed as AN-200 dataset. Concretely, 100 high-quality retinal images of 27 adult patients are acquired from Canon. Moreover, the neonatal fundus images are often with low image quality, due to the uncooperative image acquiring process. We collect 100 neonatal fundus images taken from 23 patients with RetCam3 to support various neonatal applications. In addition, different imaging angles and lighting conditions are considered during the construction of the dataset. Example of the fundus images are shown in Figure 5. For every image pair, all the branching and crossing points are labeled as keypoints. All the matched keypoints are then labeled as ground truth matching keypoints. In this way, a reliable and stable fundus image registration dataset is constructed.

4.1.2. Evaluation metric

First, we choose the widely used FIRE dataset to quantitatively evaluate the proposed method and compare with state of the art methods. Since FIRE dataset only labels part of the crossing and branching points, our model cannot be trained on this dataset. Following Rivas-Villar et al. (2022), we train the models on the training set of our constructed dataset. The trained models are then evaluated on FIRE dataset with the registration score proposed by Hernandez-Matas et al. (2017), which calculate the success ratio between the fixed and moving image pairs after the transformation of the moving image with the learned transformation parameters.

Concretely, pixels of moving image are first transformed into the coordinate space of fixed image. We then calculate the averaged distance between the transformed pixels and the ground-truth points of fixed image as the registration error of this image pair. If the registration error is below a threshold, the registration of this image pair is successful. With larger threshold, more image pairs are deemed successful registrations. By varying the threshold from 0 to larger value, the percentage of successful registration pairs enlarges gradually. In this way, we can plot the registration curve, where the X axis corresponds to the setting threshold, the Y axis refers to the percentage of successfully registered images. With the plotting curve, the Area Under Curve (AUC) can be calculated as the final registration score. The original FIRE dataset (Hernandez-Matas et al., 2017) is divided into three sub-datasets based on the overlapping and anatomical similarity between an image pair. The sub-dataset *S* consists of 71 image pairs with more than 75% overlapping and no anatomical differences. The sub-dataset *P* contains 49 image pairs with less than 75% overlapping. Finally, the sub-dataset *A* is composed of 14 image pairs with anatomical differences. Similar to Rivas-Villar et al. (2022), we calculate the AUC score on the *S*, *P*, and *A* sub-datasets and the whole FIRE dataset.

In addition, we also calculate the AUC value as the registration score on our constructed AN-200 dataset with the same computing manner. Concretely, 60%, 20% and 20% of the original dataset are randomly divided into the training, validation and test set, respectively. The final registraction score is reported on the test set.

TABLE 2 The evaluation results of methods with different network settings on AN-200 and FIRE datasets.

Method	AN-200(%)	FIRE(%)
U-Net	70.5	68.1
SCA-Net	72.2	69.5
SAPCA-Net	**72.9**	**71.1**

The bold values mean the best performance.

4.2. Ablation study on the network architecture

Based on the constructed dataset, in order to obtain better understanding of the proposed network, we evaluate following methods with different network settings. The experimental results are summarized in Table 2:

- Baseline: We first choose the vanilla encoder-decoder architecture (U-Net) as the backbone network to simultaneously learn the detection of keypoint and the generation of feature embedding, under the supervision of the above cross-entropy loss, Dice loss and the proposed structured triplet ranking loss. As shown in Table 2, the Baseline achieves an AUC of 70.5 and 68.1% on AN-200 and FIRE datasets, respectively.
- Spatially-varying context aggregation network (SCA-Net): Then we enhance the simple U-Net with the proposed spatially-varying context aggregation module. Concretely, over the last stage of the encoder sub-network of U-Net, the generated feature map of encoder sub-network is enhanced with the SCA module. The global contextual cues are thus incorporated. The loss functions are kept the same with the Baseline. The AUC on AC-200 of SCA-Net is 72.2%, and the AUC on FIRE is enlarged to 69.5%. The performance improvement is 1.7 and 1.4%, respectively.
- Spatially-varying adaptive pyramid context aggregation network (SAPCA-Net): Finally, we test our overall network, SAPCA-Net, by changing the SCA-module with the SAPCA module to incorporate context-adaptive cues. Compared to original U-Net, the SAPCA-Net largely improves the AUC of AD-200 by 2.4%, the AUC of FIRE by 3.0%. Concretely, the AUC of AD-200 is significantly enlarged from 70.5 to 72.9%, and the AUC of FIRE is improved from 68.1 to 71.1%. These results effectively show the effectiveness of the proposed SAPCA module.

As shown in Figure 6, we plot the curve of the successful registration ratio as the change of different error thresholds. In addition to the above quantitative comparisons, we also show the visualized results of our method. Figure 7 demonstrates the visualized keypoint detection and keypoint matching results from two typical scenarios. The last row also shows the final fused results with the matching keypoints. The first column of Figure 7 shows the ground truth keypoint detection and fused result. As shown in Figure 7, the baseline method is able to effectively locate and match keypoints. However, there exist a number of wrong keypoint matching results. The SCA-Net is able to remove some false positive predictions, leading to better keypoint matching result. Finally, the SAPCA-Net further removes more false positive keypoint matching predictions. Meanwhile, the number of true keypoint matching is also increased.

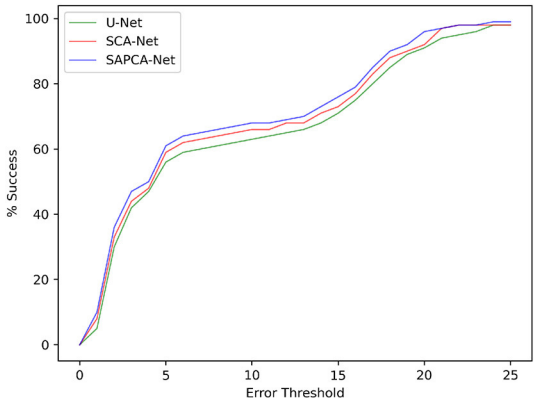


FIGURE 6
The registration success with different error thresholds for the model with different network settings on FIRE dataset.

As a result, the final fused result with the matching keypoints generated by the SAPCA-Net is visually better than other methods. These qualitative comparisons further demonstrate the effectiveness of the proposed network architecture.

4.3. Ablation study on the loss function

On the basis of the above best performing SAPCA-Net, we also conduct further ablation study for further understanding of the loss function. We evaluate the SAPCA-Net with following different loss functions, the results are summarized in Table 3:

- SAPCA-net-pairwise: We first replace the keypoint matching loss function of SAPCA-Net with the simple pairwise ranking loss. Pairwise ranking loss guides the SAPCA-Net to learn the pairwise relationship between the feature embedding of the anchor keypoint and one positive/negative keypoint. As shown in Table 3, the SAPCA-Net-Pairwise achieves the AUC of 71.4 and 69.7% on AN-200 and FIRE, respectively.
- SAPCA-net-triplet: Then we replace the keypoint matching loss function with the triplet ranking loss. The triplet loss helps the network to pull the anchor point closer to the similar keypoint than the dissimilar one by a margin. The AUC of SAPCA-Net-Triplet on AN-200 is 72.2%, and the AUC on FIRE is improved to 70.3%.
- SAPCA-net-structured-triplet: We further replace the keypoint matching loss function with structured triplet ranking loss. The structured triplet ranking loss supervise the network to learn the structured relationship among multiple keypoints. Compared to original pair-wise ranking loss, the AUC of AN-200 is enlarged from 71.4 to 72.9%, and the AUC of FIRE is improved from 69.7 to 71.1%. These results effectively show the effectiveness of the employed structured triplet ranking loss.

Among the SAPCA-Net with the above three different loss functions, the SAPCA-Net-Structured-Triplet achieves significantly better results, which effectively demonstrates the superiority of the structured triplet ranking loss for the learning of matching keypoints.

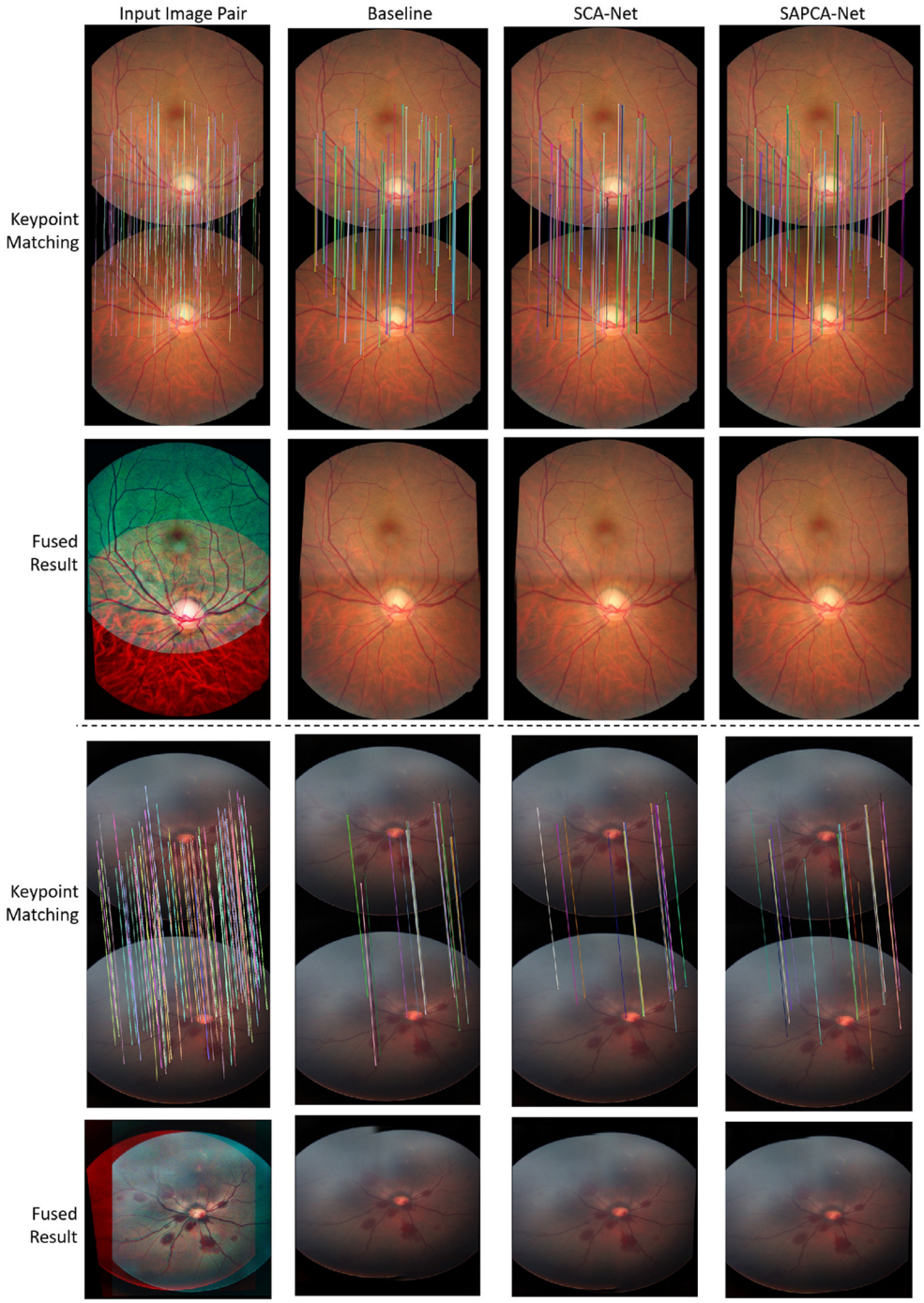


FIGURE 7
Example of the keypoint detection and matching results of normal adult and neonatal fundus images. We also show the fused image with the matching keypoints.

The change curve of registration success ratio under different error thresholds is shown in Figure 8.

4.4. Comparison to state-of-arts

In order to compare our proposed best-performing SAPCA-Net with state of the art methods, the widely used FIRE dataset is employed for evaluation. We first focus on the deep learning based methods. As shown in Table 4, compared to previous two-stage UNet + RANSAC (Rivas-Villar et al., 2022), our end-to-end registration method achieves consistently better results on the S, P, A sub-datasets and the whole FIRE dataset. Concretely, on the four dataset settings, our SAPCA-Net achieves the registration score of 93.9, 36.2, 71.9, and 71.1%, significantly outperforming UNet + RANSAC by 3.1, 6.9, 5.9, and 5.4%, respectively. Moreover, our model accomplishes the two steps of keypoint detection and matching with a single network. However, for previous UNet + RANSAC model, the

keypoint detection is first accomplished by a U-Net, which is followed by traditional RANSAC (Fischler and Bolles, 1981) for the keypoint matching step. In this way, the execution time of our proposed SAPCA-Net is much shorter.

TABLE 3 Ablation study on the loss function.

Method	AN-200(%)	FIRE(%)
SAPCA-Net-Pairwise	71.4	69.7
SAPCA-Net-Triplet	72.2	70.3
SAPCA-Net-Structured-Triplet	**72.9**	**71.1**

With the structured triplet loss, SAPCA-Net-Structured-Triplet achieves the best result. The bold values mean the best performance.

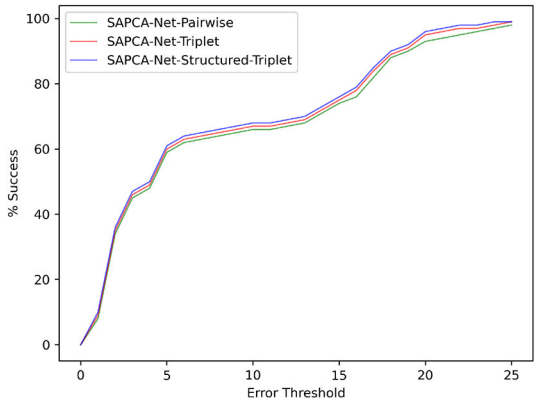


FIGURE 8
The curve of successful registration percentage under different error thresholds for the model with different loss functions on FIRE dataset.

Then, we compare our method with traditional registration methods. As shown in Table 4, our SAPCA-Net obtains the best registration score on the A sub-dataset, by achieving 71.1% AUC. This result is 3.8% better than previous best performing VOTUS. On the S sub-dataset, our method obtains the registration score of 93.9%, slightly better than VOTUS, while is 1.9% lower than the REMPE. On the whole FIRE dataset, our method outperforms most of the traditional methods. Although VOTUS and REMPE achieve better registration scores than our SAPCA-Net, the execution time of these two methods are two orders of magnitude slower than our method. Concretely, the execution time of our method is only 0.32s, which shows significant advantage compared to the VOTUS (106s) and REMPE (198s). This is a big advantage for applications in clinical scenarios.

5. Conclusion

Current deep learning based image registration methods directly learn to align the geometric transformation or the dense displacement vector field between the input image pair. These previous modeling paradigms fail to achieve keypoint detection and registration results in a reliable and stable way. To this end, in this paper, we aim to tackle this challenging issue. First, considering that the vessel crossing and branching points can reliably and stably characterize the key components for fundus image, a single network is employed to simultaneously learn to detect and match all the crossing and branching points of the input image pair in an end-to-end manner. Moreover, a spatially-varying adaptive pyramid context aggregation network is proposed to aggregate contextual cues in multi-scale field-of-view, which are much beneficial for accurate keypoint detection and matching. Furthermore, a structured triplet ranking loss is employed to guide

TABLE 4 Comparison to state-of-arts on FIRE dataset.

Method	S	P	A	FIRE	Execution time
SIFT + WGTM (Lowe, 2004)	83.7	54.4	40.7	68.5	–
GDB-ICP (Yang et al., 2007)	81.4	30.3	30.3	57.6	19
Harris-PIIFD (Yang et al., 2007)	90.0	9.0	44.3	55.3	13
SURF + WGTM (Bay et al., 2008)	83.5	6.1	6.9	47.2	–
ED-DB-ICP (Tsai et al., 2009)	60.4	44.1	49.7	55.3	44
RIR-BS (Chen et al., 2011)	77.2	0.49	12.4	44.0	-
ATS-RGN (Serradell et al., 2014)	36.9	0.0	14.7	21.1	-
EyeSLAM (Braun et al., 2018)	30.8	22.4	26.9	27.3	7
GFEMR (Wang J. et al., 2019)	81.2	60.7	47.4	70.2	10
RIFT + NTG (Zhou et al., 2022)	90.7	51.2	81.0	71.7	-
VOTUS (Motta et al., 2019)	93.4	67.2	68.1	81.2	106
REMPE (Hernandez-Matas et al., 2020)	95.8	54.2	66.0	77.3	198
U-Net + RANSAC (Rivas-Villar et al., 2022)	90.8	29.3	66.0	65.7	0.65
Our SAPCA-Net	93.9	36.2	71.9	71.1	0.32

the learning of similar feature embedding for matching keypoint and dissimilar feature embedding for non-matching keypoints. The proposed model is trained on a new constructed large-scale dataset with well-labeled ground-truths. Both quantitative and qualitative results show the effectiveness of the proposed method.

Data availability statement

The raw data supporting the conclusions of this article will be made available by the authors, without undue reservation.

Author contributions

JX and YC pointed out the problem of current methods and provided new solution. YC and PS designed the dataset constructing scheme. JX, YC, PS, LD, RS, and ZY collected and labeled the dataset. YC, PS, and DZ cleaned the dataset. JX, KY, LD, DZ, RS, and ZY performed the experiments. JX, YC, and PS evaluated the experimental results. KY wrote the first draft of the manuscript. All the authors revised the manuscript, contributed to the article, and approved the submitted version. All the authors approve the final version to be published and agree to be accountable for all aspects of the work in ensuring that questions related to the accuracy or integrity of any part of the work are appropriately investigated and resolved.

Funding

This work was supported by the Sichuan Provincial People's Hospital Fund Project No. 2021LY15 and the Chengdu Science and Technology Bureau Project No. 2021-YF05-00498-SN.

Conflict of interest

KY, LD, DZ, RS, and ZY were employed by the company Beijing Zhizhen Internet Technology Co. Ltd.

The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

Bay, H., Ess, A., Tuytelaars, T., and Van Gool, L. (2008). Speeded-up robust features (surf). *Comput. Vis. Image Understand.* 110, 346–359. doi: 10.1016/j.cviu.2007.09.014

Braun, D., Yang, S., Martel, J. N., Riviere, C. N., and Becker, B. C. (2018). Eyeslam: Real-time simultaneous localization and mapping of retinal vessels during intraocular microsurgery. *Int. J. Med. Rob. Comput. Assist. Surg.* 14, e1848. doi: 10.1002/rcs.1848

Cao, X., Yang, J., Zhang, J., Nie, D., Kim, M., Wang, Q., et al. (2017). "Deformable image registration based on similarity-steered CNN regression," in *International Conference on Medical Image Computing and Computer-Assisted Intervention* (Québec City, QC: Springer), 300–308.

Chen, L., Xiang, Y., Chen, Y., and Zhang, X. (2011). "Retinal image registration using bifurcation structures," in *IEEE International Conference on Image Processing* (Brussels: IEEE), 2169–2172.

Chopra, S., Hadsell, R., and LeCun, Y. (2005). "Learning a similarity metric discriminatively, with application to face verification," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), Vol. 1* (San Diego, CA: IEEE), 539–546.

Cui, Y., Zhou, F., Lin, Y., and Belongie, S. (2016). "Fine-grained categorization and dataset bootstrapping using deep metric learning with humans in the loop," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Las Vegas, NV: IEEE), 1153–1162.

Deng, K., Tian, J., Zheng, J., Zhang, X., Dai, X., and Xu, M. (2010). Retinal fundus image registration via vascular structure graph matching. *Int. J. Biomed. Imaging* 2010, 906067. doi: 10.1155/2010/906067

Fischler, M. A., and Bolles, R. C. (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM* 24, 381–395. doi: 10.1145/358669.358692

Hadsell, R., Chopra, S., and LeCun, Y. (2006). "Dimensionality reduction by learning an invariant mapping," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), Vol. 2* (New York, NY: IEEE), 1735–1742.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition* (Las Vegas, NV: IEEE), 770–778.

Hernandez-Matas, C., Zabulis, X., and Argyros, A. A. (2020). Rempe: registration of retinal images through eye modelling and pose estimation. *IEEE J. Biomed. Health Inform.* 24, 3362–3373. doi: 10.1109/JBHI.2020.2984483

Hernandez-Matas, C., Zabulis, X., Triantafyllou, A., Anyfanti, P., Douma, S., and Argyros, A. A. (2017). Fire: fundus image registration dataset. *Model. Artif. Intell. Ophthalmol.* 1, 16–28. doi: 10.35119/maio.v1i4.42

Hershey, J. R., Chen, Z., Le Roux, J., and Watanabe, S. (2016). "Deep clustering: Discriminative embeddings for segmentation and separation," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (Shanghai: IEEE), 31–35.

Hill, D. L., Batchelor, P. G., Holden, M., and Hawkes, D. J. (2001). Medical image registration. *Phys. Med. Biol.* 46, R1. doi: 10.1088/0031-9155/46/3/201

Huang, C., Loy, C. C., and Tang, X. (2016). "Local similarity-aware deep feature embedding," in *Advances in Neural Information Processing Systems, Vol. 29* (Barcelona).

Jie, H., Shen, L., Samuel, A., and Gang, S. (2019). "Squeeze-and-excitation networks," in *IEEE Transactions on Pattern Analysis and Machine Intelligence* (Salt Lake City, UT: IEEE).

Krebs, J., Mansi, T., Delingette, H., Zhang, L., Ghesu, F. C., Miao, S., et al. (2017). "Robust non-rigid registration through agent-based action learning," in *International Conference on Medical Image Computing and Computer-Assisted Intervention* (Québec City, QC: Springer), 344–352.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). "Imagenet classification with deep convolutional neural networks," in *NIPS* (Nevada).

Law, M. T., Urtasun, R., and Zemel, R. S. (2017). "Deep spectral clustering learning," in *International Conference on Machine Learning* (Lugano: PMLR), 1985–1994.

Li, H., and Fan, Y. (2017). Non-rigid image registration using fully convolutional networks with deep self-supervision. *arXiv preprint* arXiv:1709.00799. doi: 10.1109/ISBI.2018.8363757

Liu, J., He, J., Qiao, Y., Ren, J. S., and Li, H. (2020). "Learning to predict context-adaptive convolution for semantic segmentation," in *European Conference on Computer Vision* (Glasgow: Springer), 769–786.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.* 60, 91–110. doi: 10.1023/B:VISI.0000029664.99615.94

Motta, D., Casaca, W., and Paiva, A. (2019). Vessel optimal transport for automated alignment of retinal fundus images. *IEEE Trans. Image Process.* 28, 6154–6168. doi: 10.1109/TIP.2019.2925287

Movshovitz-Attias, Y., Toshev, A., Leung, T. K., Ioffe, S., and Singh, S. (2017). "No fuss distance metric learning using proxies," in *Proceedings of the IEEE International Conference on Computer Vision* (Venice: IEEE), 360–368.

Oh Song, H., Jegelka, S., Rathod, V., and Murphy, K. (2017). "Deep metric learning via facility location," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Honolulu, HI: IEEE), 5382–5390.

Oh Song, H., Xiang, Y., Jegelka, S., and Savarese, S. (2016). "Deep metric learning via lifted structured feature embedding," in *Proceedings of the IEEE conference on computer vision and pattern recognition* (Las Vegas, NV: IEEE), 4004–4012.

Oliveira, F. P., and Tavares, J. M. R. (2014). Medical image registration: a review. *Comput. Methods Biomech. Biomed. Eng.* 17, 73–93. doi: 10.1080/10255842.2012.670855

Opitz, M., Waltner, G., Possegger, H., and Bischof, H. (2017). "Bier-boosting independent embeddings robustly," in *Proceedings of the IEEE International Conference on Computer Vision* (Venice: IEEE), 5189–5198.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., et al. (2017). "Automatic differentiation in pytorch," in *ICLR*.

Rivas-Villar, D., Hervella, Á. S., Rouco, J., and Novo, J. (2022). Color fundus image registration using a learning-based domain-specific landmark detection methodology. *Comput. Biol. Med.* 140, 105101. doi: 10.1016/j.compbiomed.2021.105101

Rohé, M.-M., Datar, M., Heimann, T., Sermesant, M., and Pennec, X. (2017). "Svf-net: learning deformable image registration using shape matching," in *International Conference on Medical Image Computing and Computer-Assisted Intervention* (Québec City, QC: Springer), 266–274.

Ronneberger, O., Fischer, P., and Brox, T. (2015). U-Net: "Convolutional networks for biomedical image segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention* (Munich).

Schroff, F., Kalenichenko, D., and Philbin, J. (2015). "Facenet: a unified embedding for face recognition and clustering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Boston, MA: IEEE), 815–823.

Serradell, E., Pinheiro, M. A., Sznitman, R., Kybic, J., Moreno-Noguer, F., and Fua, P. (2014). Non-rigid graph registration using active testing search. *IEEE Trans. Pattern Anal. Mach. Intell.* 37, 625–638. doi: 10.1109/TPAMI.2014.2343235

Simonyan, K., and Zisserman, A. (2014). "Very deep convolutional networks for large-scale image recognition," in *ICLR* (Banff, AB).

Sohn, K. (2016). "Improved deep metric learning with multi-class n-pair loss objective," in *Advances in Neural Information Processing Systems, Vol. 29* (Barcelona).

Sokooti, H., Vos, B., d., Berendsen, F., Lelieveldt, B. P., Išgum, I., et al. (2017). "Nonrigid image registration using multi-scale 3D convolutional neural networks," in *International Conference on Medical Image Computing and Computer-Assisted Intervention* (Québec City, QC: Springer), 232–239.

Sotiras, A., Davatzikos, C., and Paragios, N. (2013). Deformable medical image registration: a survey. *IEEE Trans. Med. Imaging* 32, 1153–1190. doi: 10.1109/TMI.2013.2265603

Tsai, C.-L., Li, C.-Y., Yang, G., and Lin, K.-S. (2009). The edge-driven dual-bootstrap iterative closest point algorithm for registration of multimodal fluorescein angiogram sequence. *IEEE Trans. Med. Imaging* 29, 636–649. doi: 10.1109/TMI.2009.2030324

Vos, B. D. D., Berendsen, F. F., Viergever, M. A., Staring, M., and Išgum, I. (2017). "End-to-end unsupervised deformable image registration with a convolutional neural network," in *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support* (Québec City, QC: Springer), 204–212.

Wang, J., Chen, J., Xu, H., Zhang, S., Mei, X., Huang, J., et al. (2019). Gaussian field estimator with manifold regularization for retinal image registration. *Signal Process.* 157, 225–235. doi: 10.1016/j.sigpro.2018.12.004

Wang, J., Song, Y., Leung, T., Rosenberg, C., Wang, J., Philbin, J., et al. (2014). "Learning fine-grained image similarity with deep ranking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Columbus, OH: IEEE), 1386–1393.

Wang, X., Hua, Y., Kodirov, E., Hu, G., Garnier, R., and Robertson, N. M. (2019). "Ranked list loss for deep metric learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (Long Beach, CA: IEEE), 5207–5216.

Weinberger, K. Q., and Saul, L. K. (2009). Distance metric learning for large margin nearest neighbor classification. *J. Mach. Learn. Res.* 10, 207–244.

Yang, G., Stewart, C. V., Sofka, M., and Tsai, C.-L. (2007). Registration of challenging image pairs: initialization, estimation, and decision. *IEEE Trans. Pattern Anal. Mach. Intell.* 29, 1973–1989. doi: 10.1109/TPAMI.2007.1116

Yang, X., Kwitt, R., Styner, M., and Niethammer, M. (2017). Quicksilver: fast predictive image registration-a deep learning approach. *Neuroimage* 158, 378–396. doi: 10.1016/j.neuroimage.2017.07.008

Zhou, J., Jin, K., Gu, R., Yan, Y., Zhang, Y., Sun, Y., et al. (2022). Color fundus photograph registration based on feature and intensity for longitudinal evaluation of diabetic retinopathy progression. *Front. Phys.* 10, 978392. doi: 10.3389/fphy.2022.978392

Zou, B., He, Z., Zhao, R., Zhu, C., Liao, W., and Li, S. (2020). Non-rigid retinal image registration using an unsupervised structure-driven regression network. *Neurocomputing* 404, 14–25. doi: 10.1016/j.neucom.2020.04.122

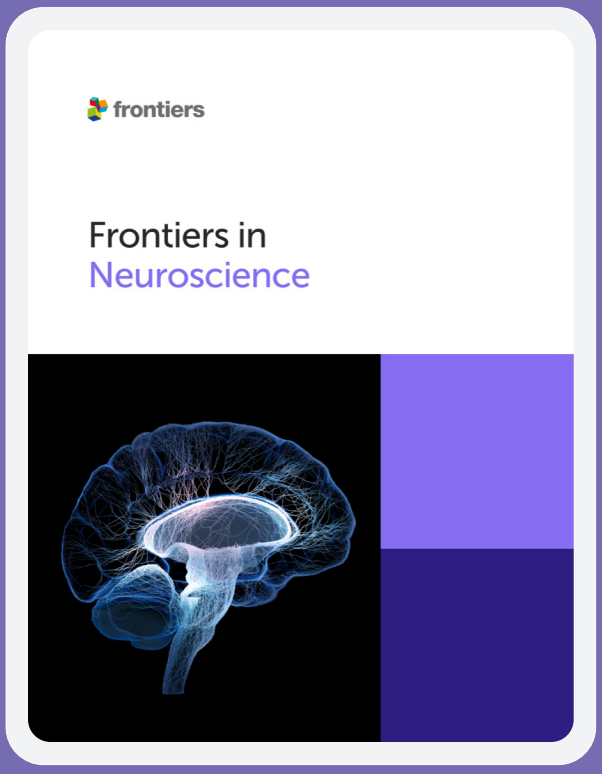
Frontiers in Neuroscience

Provides a holistic understanding of brain function from genes to behavior

Part of the most cited neuroscience journal series which explores the brain - from the new eras of causation and anatomical neurosciences to neuroeconomics and neuroenergetics.

Discover the latest Research Topics

See more →



Frontiers

Avenue du Tribunal-Fédéral 34
1005 Lausanne, Switzerland
frontiersin.org

Contact us

+41 (0)21 510 17 00
frontiersin.org/about/contact