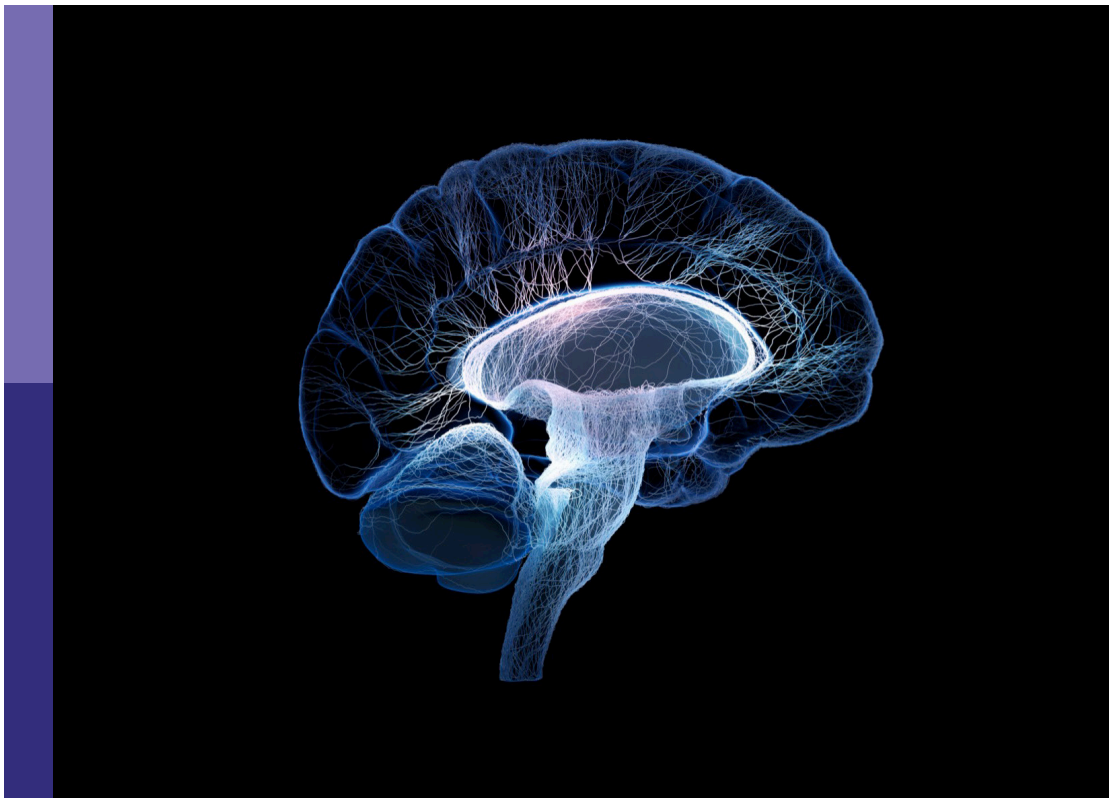
frontiers | Research Topics

Advanced methods and applications for neurointelligence

Edited by
Manning Wang and Alois C. Knoll

Coordinated by
Hu Cao

Published in
Frontiers in Neuroscience



FRONTIERS EBOOK COPYRIGHT STATEMENT

The copyright in the text of individual articles in this ebook is the property of their respective authors or their respective institutions or funders. The copyright in graphics and images within each article may be subject to copyright of other parties. In both cases this is subject to a license granted to Frontiers.

The compilation of articles constituting this ebook is the property of Frontiers.

Each article within this ebook, and the ebook itself, are published under the most recent version of the Creative Commons CC-BY licence. The version current at the date of publication of this ebook is CC-BY 4.0. If the CC-BY licence is updated, the licence granted by Frontiers is automatically updated to the new version.

When exercising any right under the CC-BY licence, Frontiers must be attributed as the original publisher of the article or ebook, as applicable.

Authors have the responsibility of ensuring that any graphics or other materials which are the property of others may be included in the CC-BY licence, but this should be checked before relying on the CC-BY licence to reproduce those materials. Any copyright notices relating to those materials must be complied with.

Copyright and source acknowledgement notices may not be removed and must be displayed in any copy, derivative work or partial copy which includes the elements in question.

All copyright, and all rights therein, are protected by national and international copyright laws. The above represents a summary only. For further information please read Frontiers' Conditions for Website Use and Copyright Statement, and the applicable CC-BY licence.

ISSN 1664-8714
ISBN 978-2-8325-4986-5
DOI 10.3389/978-2-8325-4986-5

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

Advanced methods and applications for neurointelligence

Topic editors
Manning Wang – Fudan University, China
Alois C. Knoll – Technical University of Munich, Germany

Topic coordinator
Hu Cao – Technical University of Munich, Germany

Citation
Wang, M., Knoll, A. C., Cao, H., eds. (2024). *Advanced methods and applications for neurointelligence*. Lausanne: Frontiers Media SA. doi: 10.3389/978-2-8325-4986-5

Table of contents

frontiers | Frontiers in Neuroscience

TYPE Original Research
PUBLISHED 11 April 2023
DOI 10.3389/fnins.2023.1168888

Check for updates

OPEN ACCESS

EDITED BY
Alois C. Knoll,
Technical University of Munich, Germany

REVIEWED BY
Chao Zeng,
University of Hamburg, Germany
Xianfa Xue,
South China University of Technology, China

*CORRESPONDENCE
Ning Wang
✉ Katie.Wang@brl.ac.uk

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

RECEIVED 18 February 2023
ACCEPTED 20 March 2023
PUBLISHED 11 April 2023

CITATION
Li J, Zhong J and Wang N (2023) A multimodal human-robot sign language interaction framework applied in social robots. *Front. Neurosci.* 17:1168888. doi: 10.3389/fnins.2023.1168888

COPYRIGHT
© 2023 Li, Zhong and Wang. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

A multimodal human-robot sign language interaction framework applied in social robots

Jie Li[1], Junpei Zhong[2] and Ning Wang[3]*

[1]School of Artificial Intelligence, Chongqing Technology and Business University, Chongqing, China, [2]Department of Rehabilitation Sciences, The Hong Kong Polytechnic University, Kowloon, Hong Kong SAR, China, [3]Bristol Robotics Laboratory, University of the West of England, Bristol, United Kingdom

Deaf-mutes face many difficulties in daily interactions with hearing people through spoken language. Sign language is an important way of expression and communication for deaf-mutes. Therefore, breaking the communication barrier between the deaf-mute and hearing communities is significant for facilitating their integration into society. To help them integrate into social life better, we propose a multimodal Chinese sign language (CSL) gesture interaction framework based on social robots. The CSL gesture information including both static and dynamic gestures is captured from two different modal sensors. A wearable Myo armband and a Leap Motion sensor are used to collect human arm surface electromyography (sEMG) signals and hand 3D vectors, respectively. Two modalities of gesture datasets are preprocessed and fused to improve the recognition accuracy and to reduce the processing time cost of the network before sending it to the classifier. Since the input datasets of the proposed framework are temporal sequence gestures, the long-short term memory recurrent neural network is used to classify these input sequences. Comparative experiments are performed on an NAO robot to test our method. Moreover, our method can effectively improve CSL gesture recognition accuracy, which has potential applications in a variety of gesture interaction scenarios not only in social robots.

KEYWORDS

social robots, sign language, gesture recognition, multimodal sensors, human-robot interaction

1. Introduction

According to statistics, there are over 70 million deaf people in the world.[1] For these people, communication with others through verbal language is impossible. Therefore, there are a great many difficulties in their daily communications. For instance, deaf people could not hear a horn when crossing the street. How to help the deaf community and those who have language impairment enjoy accessible social lives is very important. A service robot is a kind of intelligent robot dedicated to providing service for improving human life. With the development of robotics, information science, and sensor technology, service robots have been applied widely in many fields, such as medical rehabilitation, education, transportation, and entertainment to domestic service (Siciliano and Khatib, 2016; Yang et al., 2018a; Gonzalez-Aguirre et al., 2021).

1 World Federation of the Deaf. (2023). E. coli. http://wfdeaf.org/our-work/. [Accessed February 8, 2023].

As a kind of service robot, the social robot is aimed at interacting with people in a human-centric way, which can provide a friendly way for interaction and services to meet the diverse demands of human beings (Breazeal et al., 2016; Yang et al., 2018b). Thus, social robots are expected to help the above-mentioned people communicate with others in a nonverbal way. In this sense, how to develop and design an intuitive, natural, easily interactive, and friendly interaction mode that can help these people communicate conveniently is a challenging topic for social robots.

Among various approaches to human-robot interaction (HRI), the way of using hand gestures for interaction facilitates more efficient communication between humans and robots. Since gesture interaction is a kind of non-contact way, which is more secure, friendly, and easy to accept by humankind. The gesture is one of the most widely used communicative manners. In the long-term social practice process, the gesture is endowed with a variety of specific meanings. At present, gesture has become the most powerful tool for expressing sentiment, intention, or attitude for humans. Hence, more and more researchers focus on gesture recognition and its applications. Many approaches are studied to recognize hand gestures by different modality sensors with various features. These approaches can be mainly categorized into three types: the wearable sensor-based approaches (Si et al., 2022), the vision sensor-based approaches (Mitra and Acharya, 2007; Oudah et al., 2020; Rastgoo et al., 2020; AI Farid et al., 2022), and the combination of the above-mentioned gesture recognition approaches (Wu et al., 2016; Xue et al., 2018; Roda-Sanchez et al., 2023). However, most of these studies were based on the single static or dynamic gestures to classification or recognition. Seldom of them focused on both dynamic and static recognition by using different modal information. Dynamic and static gestures are both needed to recognize under some specific circumstances, such as sign language recognition (SLR) for deaf or speech-impaired people.

Sign language is highly structural hand gestures, including static gestures and dynamic gestures. It serves as a useful tool for the deaf and hearing-impaired individuals in daily communication. The structural features of sign language make it very suitable for computer vision algorithms (Wu and Huang, 1999). Therefore, many relevant studies (such as SLR) are based on vision-based approaches (Cheok et al., 2019). The input data of vision-based SLR algorithms are usually divided into static gesture and dynamic gesture. Correspondingly, there are static-based and dynamic-based SLR approaches. For static sign language gestures, the approaches, such as K-nearest neighbor (Tharwat et al., 2015), support vector machine (Kurdyumov et al., 2011), and multilayer perceptron (Karami et al., 2011) are used to obtain better results. The vision-based dynamic sign language approaches include hidden Markov model (HMM; Wang et al., 2003), dynamic time wrapping (Lichtenauer et al., 2008), relevance vector machine (Wong and Cipolla, 2005), and finite state machine (Hong et al., 2000), etc.

Recently, with the advent of deep neural networks (DNN, Cao et al., 2022a), various deep learning algorithms are applied to SLR (Camgoz et al., 2018; Cui et al., 2019; Qi et al., 2021). Pu et al. presented a dynamic convolutional neural network (CNN) SLR model based on RGB video input (Pu et al., 2018). Wei et al. combined the 3D convolutional residual network and bidirectional long short-term memory (LSTM) network to recognize dynamic sign language gestures (Wei et al., 2019). Similarly, Cui et al. developed a dynamic SLR framework by combining CNN and bidirectional LSTM networks (Cui et al., 2019). Ye et al. proposed a 3D Recurrent CNN to classify gestures and localize joints (Ye et al., 2018). With the development of sensor technology (Chen et al., 2020; Cao et al., 2022b), many high accuracy and low cost sensors appears, such as Kinect and Leap Motion Controller (LMC) sensors. These sensors can capture hand or arm information more conveniently. The combination of new emerging sensors and deep learning approaches brings more new possibilities for SLR. Chong and Lee used the features recorded from the LMC sensor to classify 26 letters in American Sign Language (ASL). The recognition accuracy reaches 93.81% with DNN algorithms (Chong and Lee, 2018). Naglot et al. used a deep learning method to achieve 96.15% based on LMC gesture samples (Naglot and Kulkarni, 2016). Kumar et al. (2017a) presented a multimodal framework combining the HMM and bidirectional LSTM networks. The framework can recognize isolated sign language gesture datasets from Kinect and LMC sensors. To improve the accuracy of SLR, researchers fused different features to achieve the expected results. Kumar et al. (2017a) classified 25 Indian sign language (ISL) gestures by employing the coupled HMM to fuse the Leap Motion and Kinect sign language information. Bird et al. (2020) presented a late fusion approach to multimodality in SLR by fusing RGB and 3D hand data with a deep convolutional network. In the above research works, the sign language gestures involve both isolated static and dynamic hand gestures, based on Chinese sign language (CSL), ISL, ASL, and other sign languages from different countries, etc. The SLR approaches include traditional machine learning, deep learning, and the combination of both algorithms. However, these studies seldom take into account both static and dynamic sign language gestures in a classifier at the same time. Moreover, most of the researchers focus on using depth or RGB information as the input data of the classifier. Generally, the fusion of different modal input data also often uses these two data. The SLR framework proposed by Bird et al. fused two modalities of gesture datasets both captured from one sensor (LMC; Bird et al., 2020). Hence, inspired by the previous work (Naglot and Kulkarni, 2016; Kumar et al., 2017a,b; Bird et al., 2020), we propose a multimodal SLR framework that combines CSL features from several sensors to recognize static and dynamic hand gestures. This framework uses the deep learning method to fuse two modalities features from two different sensors to improve the recognition accuracy. The proposed multimodal framework can not only recognize singular CSL gestures but also recognize gestures consisting of two singular gestures.

Sign language mainly use the human hands to convey information. In some cases, other body parts such as fingers, arms, and head also used to convey information (Wu and Huang, 1999). CSL gestures mainly use human hands and arms. Therefore, the focus of this paper is to use human hand and arm information to classify corresponding gestures. Different from most of the vision-based input data, this work fuses the information from the visual sensor and surface electromyography (sEMG) signals by using Leap Motion and a wearable Myo armband. Though some research works in the gesture recognition area use human arm sEMG captured by Myo armband or other similar devices. Sometimes, higher recognition accuracy is also

achieved. But for SLR, seldom research applies sEMG signals to classify different sign language gestures. In this work, considering the characteristic of CSL, we apply the advantages of information fusion to fuse two modalities of data to improve recognition accuracy. It combines the advantages of arm sEMG information in gesture recognition and the complementary for different modal sensor information. Besides, SLR is mainly applied to the daily communication between deaf, speech-impaired, and autism spectrum disorders (ASD) communities, the proposed CSL recognition framework is applied to social robots. Thus, it can promote communication between these communities and entertainment with robots.

The main contribution of this work is that an HRI system by integrating two modalities of CSL data is developed for deaf and speech-impaired people, which enables the social robots to communicate with target people efficiently and friendly. Most importantly, the proposed system can be applied in other interaction scenarios between robots and autistic children. The remainder of the paper is organized as follows. The CSL gestures classification method is presented in section 2. Section 3 provides the simulations and case studies on the real-world robot. Section 4 concludes this work and discusses the further potential applications.

2. Methodology

The overview of the proposed HRI system is shown in Figure 1. It includes three phases: data collection, data classification, and robot response.

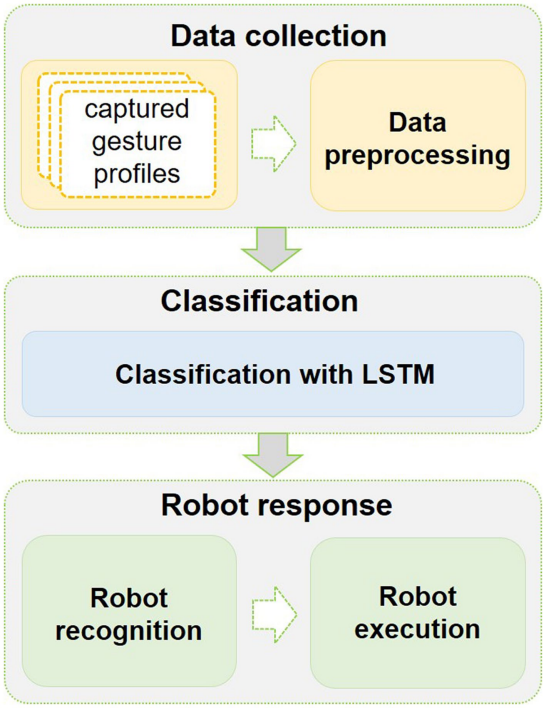


FIGURE 1
The framework of the proposed multimodal human-robot interaction (HRI) system.

2.1. System overview

In this data collection phase, we mainly collect four different kinds of common CSL gestures. Here, two modalities of hand action data are collected from two different modal sensors. Leap Motion is applied to capture human hand 3D features. Meanwhile, the human arm sEMG signals are captured by the Myo armband.

In the data classification phase, the collected gesture data from two sensors are preprocessed, respectively. Then, the features of two modalities datasets are fused as one dataset, which serves as input to the LSTM classifiers.

In the robot response phase, after the gesture data is recognized by the LSTM classifier, the results are transformed into executable commands of the social robot. Later, the robot makes a response to the recognition results.

2.2. Data collection and preprocessing

Figure 2 presents the overall steps of the data collection, preprocessing, and feature fusion. As aforementioned, the Myo armband and LMC are used to capture arm sEMG signals and human hand movements, respectively. As shown in Figure 2, a participant wears the Myo armband on the forearm and puts his/her hand onto the Leap Motion sensor within viewing range to capture sEMG signals and hand movement information synchronously. When a participant is performing a certain sign language, the data are recorded synchronously from the Myo armband and LMC. That is, the sEMG signals and human hand 3D vectors from both sensors are timely collected. In this paper, four daily CSL gestures are considered.

2.2.1. Hand 3D information captured by leap motion sensor

Leap motion is an optical hand tracking sensor that captures the movements of human hands with sub-millimeter accuracy. The sketch of LMC is shown in Figure 3A. The core of the device consists of three infrared LEDs placed at equal distances from each other, and two stereo cameras placed between each pair of IR sensors (Li et al., 2019). With these devices, LMC can detect the bones and joints of the human hand accurately by combining stereoscopy and depth-sensing. The view of a 3D representation of the hand translated by the two cameras is shown in Figure 3B. Compared with the Microsoft Kinect sensor, LMC is more portable, smaller ($L \times W \times H = 8 \times 3 \times 1.1\,cm^3$), and lower-cost (Weichert et al., 2013). Here, the Leap Motion sensor is applied to collect 3D vectors of the human hand.

Two healthy participants aged 22–35 years contributed to a dataset of CSL gestures. They are asked to repeat each gesture 50 times comfortably. The length of each gesture is recorded within 5 s to avoid muscle fatigue and affect data quality. During the data recording, participants are asked to take a break for each repeat. They told the details of data collection in advance. Four different gestures of the right hand are recorded at a frequency of 50 Hz. The LMC data are recorded by the deep cameras located on the sensor facing the participants' hand. It is worth noting that both participants placed their palms at the same height above the LMC sensor. Also, the positions of the Myo armband for them are the same.

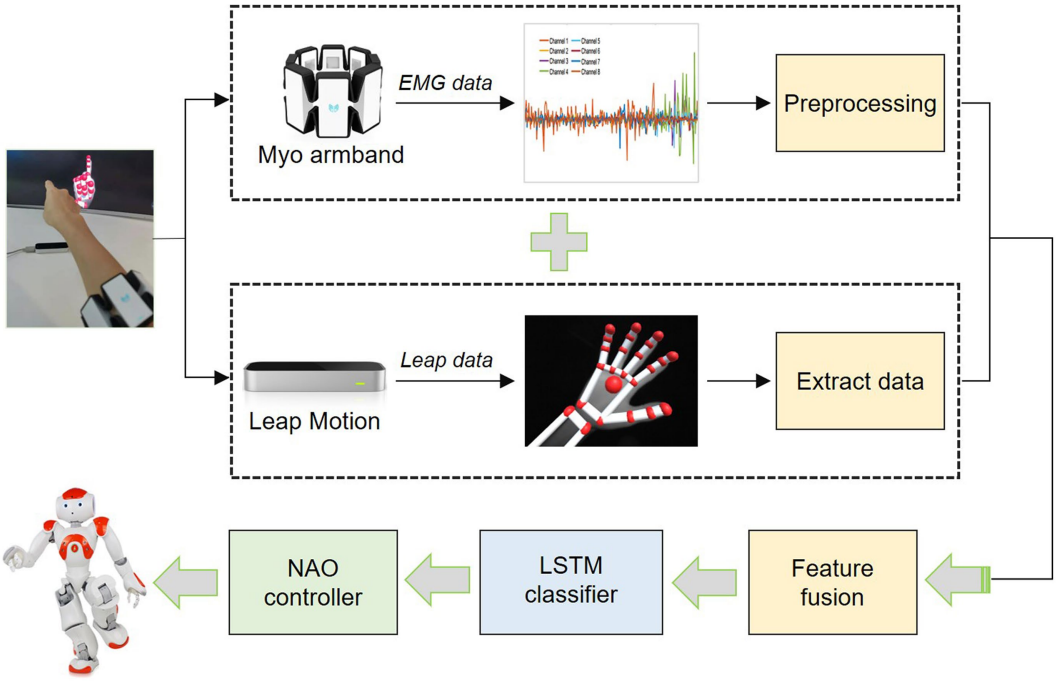


FIGURE 2
An overall diagram of the HRI system.

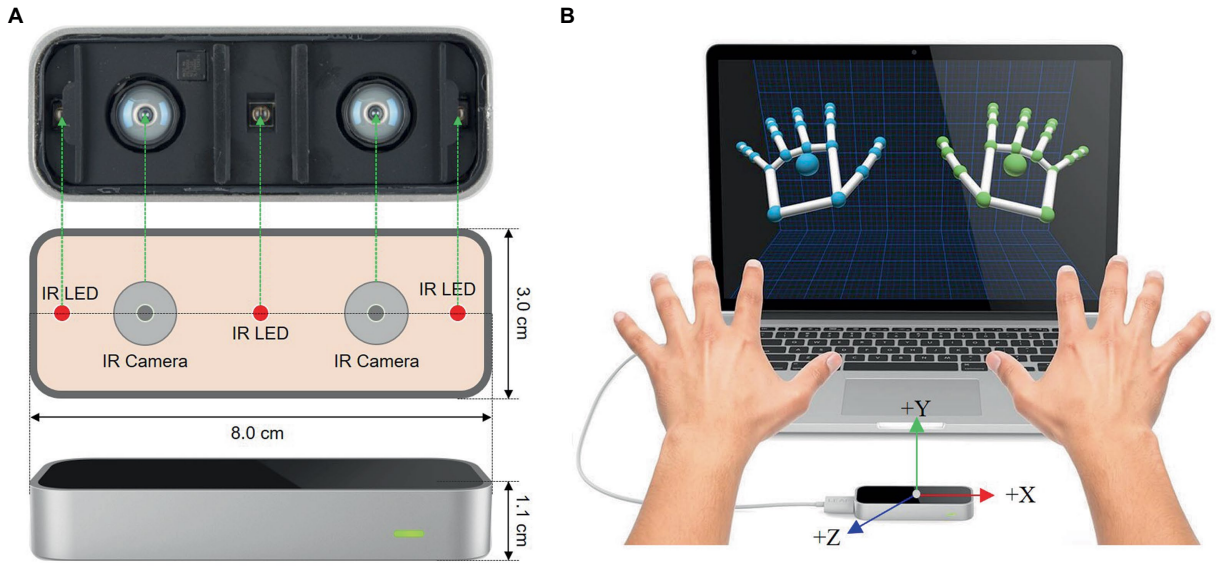


FIGURE 3
The view of Leap Motion Controller (LMC). **(A)** Schematic view of LMC. **(B)** 3D view of human hand from LMC (Weichert et al., 2013).

Figure 4 demonstrates the fingertips, wrist, and palm position. For each performed gesture, we record all the 3D coordinates of human hand. Then, the start palm position, the difference between the start palm positions, changes of palm positions, palm direction, and velocity of the palm are extracted from these 3D coordinates. As shown in Figure 4, we also extracted the yaw, pitch, and roll of the palms. It is noted that yaw is the angle between the negative z-axis and the projection of the vector onto the x-z *plane. Similarly, pitch and roll are the angles between the corresponding negative coordinate axes and the projection of corresponding vectors. In other words, pitch, yaw, and roll represent the rotations around the* x, y, *and* z *axes, respectively. The angle is calculated through two 3D vectors* (Bird et al., 2020). Assuming that the angle θ is constructed by the vectors of $\vec{a}$ and $\vec{b}$, then it can be computed as follows

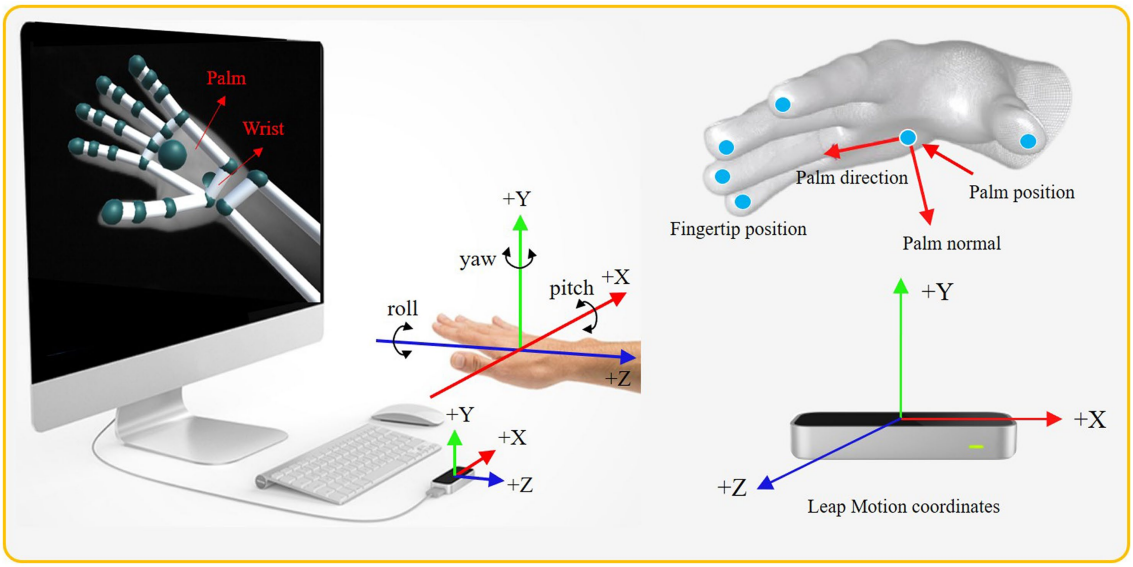


FIGURE 4
The coordinate system of the Leap Motion sensor and diagram of the bone data detected by it.

$$\theta = \arccos\left(\frac{\vec{a}\vec{b}}{|\vec{a}||\vec{b}|}\right) \tag{1}$$

where a and b are vectors made up of two points in space following the LMC coordinate system. The LMC sensor adopts a Cartesian coordinate system based on right-hand. The origin is at the top center of LMC. $|\vec{\mathbf{a}}|$ and $|\vec{\mathbf{b}}|$ are the magnitudes of the corresponding vectors, which can be computed as follows

$$\begin{aligned} |\vec{\mathbf{a}}| &= \sqrt{(a_x)^2 + (a_y)^2 + (a_z)^2} \\ |\vec{\mathbf{b}}| &= \sqrt{(b_x)^2 + (b_y)^2 + (b_z)^2} \end{aligned} \tag{2}$$

where the subscripts of a and b correspond to the x, y, and z coordinates of each vector in space, respectively.

In this work, nine features (six 3D coordinates and three one-dimensional angle) are chosen to distinguish four CSL gestures. Each 3D coordinate is three-dimensional. Hence, the total dimensions of the nine features are 21, as shown in Table 1. It is known that the features of dynamic gestures are time-varying. The change of palm position can reflect that change well. Hence, the palm position (as shown in the second feature in Table 1) is extracted as one of the features to distinguish dynamic and static gestures effectively. Thus, the proposed framework can recognize both static and dynamic gestures without another special classifier. Noting that the wrist position is extracted to reflect the change of arm.

TABLE 1 Descriptions of CSL collected from leap motion sensor.

Features	Descriptions
Palm position	3D coordinates (X , Y , and Z)
Change of palm position	3D coordinates (X , Y , and Z) The difference between the start and the end position of palm.
Palm normal	3D coordinates (X , Y , and Z)
Palm direction	3D coordinates (X , Y , and Z)
Palm velocity	3D coordinates (X , Y , and Z)
Yaw of the palm	Angle (one dimension)
Pitch of the palm	Angle (one dimension)
Roll of the palm	Angle (one dimension)
Wrist position	3D coordinates (X , Y , and Z)

2.2.2. Human arm sEMG signals captured by Myo armband sensor

Figure 5 shows the sketch of Myo armband. It is a wearable and lightweight elastic armband. Myo armband is produced by the Thalmic Labs which consists of several metal contacts. These metal contacts can measure the electrical activity of the user's forearm muscles. Thus, the Myo armband can recognize their hand gestures and detect their arm motion by reading the electrical activity of human muscles. The Myo armband has eight detection channels. Correspondingly, eight-channel sEMG signals of the human forearm arm are captured to classify sign language gestures together with LMC data. Since gestures are collected synchronously from the Myo armband and Leap Motion sensor, the sampling frequencies for both sensors are the same. The raw sEMG signals are noisy. Therefore, it is necessary to process the signals captured by the Myo armband to train the gesture classifier effectively (Zardoshti-Kermani et al., 1995; Phinyomark et al., 2013; Camargo and Young, 2019).

2.2.3. Data preprocessing for two data subjects

Based on the above-mentioned, four CSL gestures recorded from two sensors are depicted in Figure 6. These four gestures are chosen because they are commonly used by Chinese people. The useful right-hand gestures for general conversation include "you," "me," "everyone," and "good." For the four gestures, only "everyone" is the dynamic gesture.

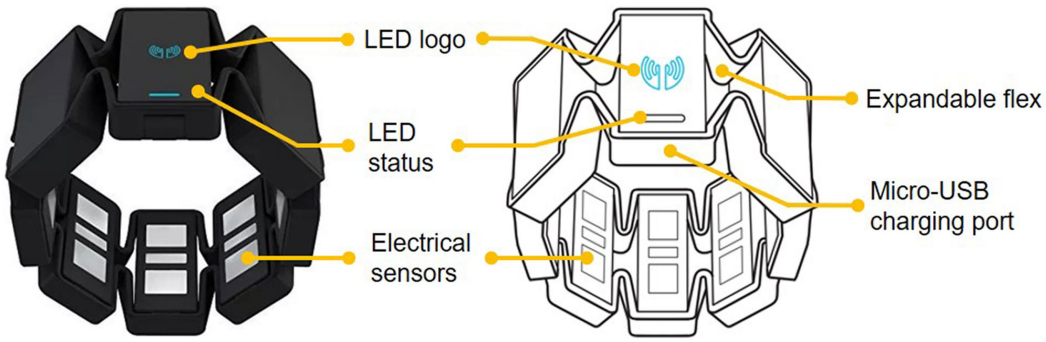


FIGURE 5
The view of Myo armband.

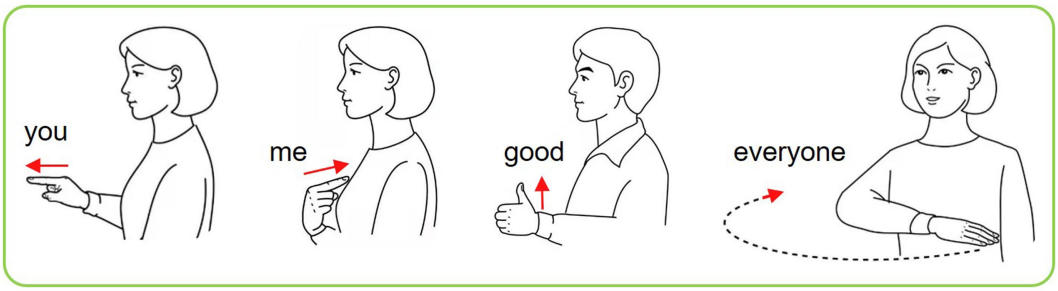


FIGURE 6
Four kinds of CSL gestures.

Before the datasets are fed into the classifier, we must preprocess them to obtain a better recognition result. For the LMC data, each feature is normalized to a value between 0 and 1. The purpose of normalization is to make the preprocessed data limited to a certain range, so as to eliminate the adverse effects (such as causing the training time to increase, which also may lead to the failure of convergence) caused by the singular samples.

As for the Myo data, preprocessing and feature extracting are necessary before training a classifier. Since the sEMG signals are noisy and different features influence the recognition performance, the preprocessing technique is an efficient way to reduce the impact on recognition results caused by the above factors. Low-pass filtering and band-pass filtering are used to preprocess the sEMG signals first. The low-pass filtering is aimed at obtaining signals with a frequency of 5–200 Hz, and band-pass filtering is used to obtain the envelope of sEMG signals. Then, the root mean square (RMS; Kundu et al., 2018; Le Sant et al., 2019) is extracted as a feature of sEMG signals. Compared with other features, such as waveform length (Phinyomark et al., 2009; Arief et al., 2015), and autoregressive model features (Subasi, 2012; Krishnan et al., 2019), it has been verified that the RMS feature obtain the best result under different lengths of sampling moving window (Luo et al., 2020).

2.2.4. Data fusion of two modalities data

After preprocessing, we can obtain two datasets from LMC and Myo armband sensors. Recent studies have shown that sensor fusion can promote richness, completeness, and accuracy of information with less uncertainty to enhance the performance of training (Chavez-Garcia and Aycard, 2015; Li et al., 2020). Here, feature-level fusion is applied to fuse information of two sensors. Two preprocessed sequences are merged into a longer sequence with 29 dimensions as input of the LSTM network. In other words, each gesture has 29 features. For each gesture, the data collected from both sensors have a history of 50 frames. Thus, the size of each gesture is 50*29.

2.3. Deep learning classification approaches

Recurrent neural network (RNN) is a commonly used approach in training and classifying time-series data. However, it is easy to occur gradients explosion or vanish when RNN handles long-term dependence. LSTM is designed to solve this problem. Compared with general RNN, LSTM performs better in learning longer time-series data. In this work, the LSTM network is used to classify the multimodal CSL gesture sequences.

The key to the effectiveness of LSTM in dealing with sequence problems lies in memory blocks and gates (Hochreiter and Schmidhuber, 1997). As shown in Figure 7A, each memory block consists of an input gate, a memory cell, an output gate, and a forget gate. The memory cell retains information relying on different time intervals. The input gate, forget gate, and output gate determine whether the information flow can enter or exit the memory cell. Three independent gates work together to ensure that the cell retains information for a long time. Figure 7B shows the actual structure of the LSTM memory cell.

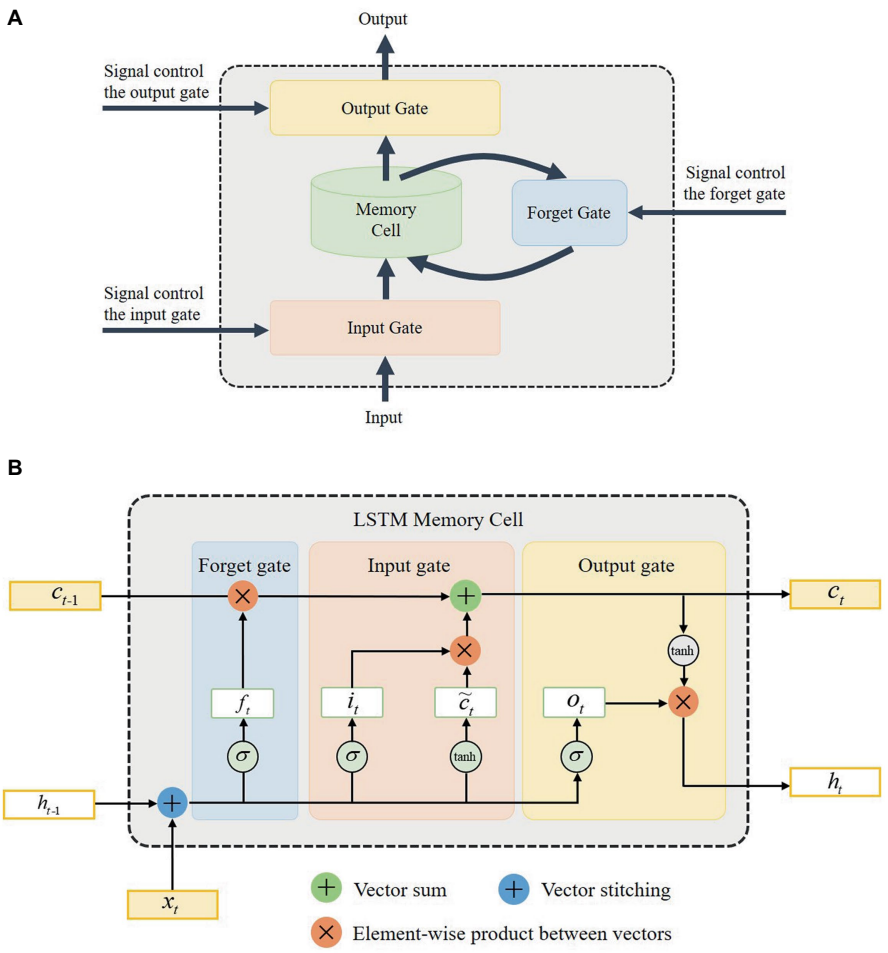


FIGURE 7
The architecture of the long short-term memory (LSTM) network. **(A)** The composition of LSTM memory blocks. **(B)** The structure of LSTM memory cell.

As shown in Figure 7B, x_t is the input of the LSTM network, and h_t is the output of the network. f_t, i_t, and o_t respectively denote the forget gate, input gate, and output gate variables of the LSTM network. The subscripts t and $t-1$ represent the current time and previous time. c_t is the memory cell state. The notation of σ and tanh denote sigmoid and hyperbolic activation functions, respectively. With the memory gates, the input, output, and key parameters of the LSTM network can be computed (Graves, 2013)

$$i_t = \sigma\left(W_{ix}x_t + W_{ih}h_{t-1} + W_{ic}c_{t-1} + b_i\right) \tag{3}$$

$$f_t = \sigma\left(W_{fx}x_t + W_{fh}h_{t-1} + W_{fc}c_{t-1} + b_f\right) \tag{4}$$

$$c_t = f_t c_{t-1} + i_t \tanh\left(W_{cx}x_t + W_{ch}h_{t-1} + b_c\right) \tag{5}$$

$$o_t = \sigma\left(W_{ox}x_t + W_{oh}h_{t-1} + W_{oc}c_t + b_o\right) \tag{6}$$

$$h_t = o_t \tanh\left(c_t\right) \tag{7}$$

where subscripts i, o, f, and c respectively represent the parameters related to the input gate, output gate, forget gate, and memory cell. The subscripts of the weight matrix are similar. For instance, W_{ih} denotes the input-hidden matrix, W_{ic} denotes the input-memory cell matrix, etc. Similarly, b_f, b_i, b_c, and b_o present the biases of corresponding subscripts for the LSTM network. tanh is the hyperbolic activation function, while Ã is the sigmoid activation function.

The special structure of the memory cell endows the LSTM network with powerful capability in modeling time-based sequences with long-range dependencies. Therefore, the applications of this network have covered a great many fields successfully. In this work, the LSTM network is used to classify the time-series CSL gestures. With this network, the CSL gestures can be classified well. Then, the classification results will be sent to a social robot for interaction and reaction.

This section first outlined the proposed framework and briefly introduced each module of this framework. Then, the collection of the

sign language datasets, and the preprocessing and fusion of two different sensor data were elaborated in detail. Lastly, the relevant classification algorithm makes the above datasets suitable to our framework was presented.

3. Experiments and results

Two experiments were performed to verify the proposed HRI framework. First, we compare the recognition performance of sensor fusion-based multimodal gesture datasets with individual sensor datasets. Then, we test the proposed framework according to several gesture recognition results and reactions with the NAO robot using LMC and Myo armbands.

3.1. Experimental setup

The experimental platform is introduced below:

3.1.1. Hardware platform

The experimental devices mainly include two gesture collection sensors and a social robot. As aforementioned, the Myo armband and LMC sensors are used to collect eight-channel sEMG signals and hand 3D information, respectively. The social robot applied in the experiment is a NAO robot. As a bipedal humanoid robot, NAO is produced by the French Aldebaran Robotics Company. It is currently the most influential social robot research platform (Bartneck et al., 2019). Because the robot is low cost, easy to program, small in size, portable, and able to conduct research outside the laboratory (Su et al., 2007; Cohen et al., 2011; Garimort et al., 2011). Therefore, it has become a widely used robotic platform for HRI research by academic institutions around the world. Here, it is used to communicate with a person by gestures.

3.1.2. Software environment

The LSTM classifier was run on an Intel i7-4600M CPU with 2.9GHZ which has 8 GB of GDDR5 memory. The LSTM model was built using the Python 3.6 library of Keras and trained using fusion data. Control software of NAO robot Choregraphe is employed to interact *via* gestures with a specified person. Both software runs on Windows 10 operating system.

3.2. Multimodal gestures comparison experiments and results

The demonstration data from the Kinect sensor and Myo armband will be preprocessed before it is fed into the incremental learning method. Firstly, the data fusion method based on the KF is used to fuse the joint angles and joint angular velocities to obtain a more accurate and smooth dataset. Since the demonstration data are not matched in the timeline, then the dynamic time warping (DTW) algorithm is applied to align it. Here, the two preprocessing methods will be introduced briefly.

3.2.1. Settings

The first experiment is performed to test the recognition performance for multimodal gestures. To compare with single modality data, three different sensor datasets are fed into the LSTM classifier. The corresponding conditions are considered as follows.

Condition 1: Single modality data from LMC sensor. The input data of the LSTM network are the 21-dimension (as listed in Table 1) 3D hand vectors collected from the LMC sensor.

Condition 2: Single modality data from Myo armband. In other word, the input data of the LSTM network are the eight-channel sEMG signals of the human forearm arm with eight dimensions.

Condition 3: Two modality sensors data from two sensors (Leap Motion sensor and Myo armband). In this condition, the input data of the LSTM network is the combination of the 21-dimension 3D hand vectors and the eight-dimension sEMG signals of the human forearm arm. Before the data are fed into the network, the two sensors datasets are preprocessed and normalized, respectively. Then, the normalized datasets are fused as a new input vector of the LSTM.

In the conditions 1 and 2, the steps are the same except that the input data is different. The LSTM model is trained by feeding each of the time-series training data in batches of 10. And this is performed over 100 epochs of training. There are 400 sequences for four CSL gestures in total. The data set is randomly divided into training data and cross-validation data at a ratio of 90–10, respectively. It means that the number of training and testing sets is 360 and 40, respectively.

There are two important parameters for the LSTM network that can improve the classification results. One is the number of hidden neurons, and the other is the epoch. To obtain the optimal performance, the number of hidden neurons and epochs for the LSTM network under the above conditions are successively valued from 1 to 150. Table 2 shows the parameters setup of the LSTM network under three conditions. Figure 8 shows the model of the LSTM network. The superscript n of $x^n_{0:49}$ denotes the dimensions of gesture features. The values of n are different under the above three conditions. The subscript of $x^n_{0:49}$ is the length of each gesture sample. The subscript m of S_m denotes the number of gesture samples.

For each epoch, the training and test accuracy are computed and echoed. The computation of accuracy is as follows

$$\text{Accuracy} = \frac{Ges_{\text{correct}}}{Ges_{\text{total}}} \tag{8}$$

where Ges_{correct} denotes the number of gestures classified correctly. Ges_{total} represents the total number of gesture samples collected.

TABLE 2 Parameters setup for the first three experiments.

Parameters	Condition 1	Condition 2	Condition 3
Size of input	50 * 21	50*8	50*29
Size of output	4	4	4
Number of hidden neurons	20	20	15
Epoch	50	50	30
Batch size	10	10	10

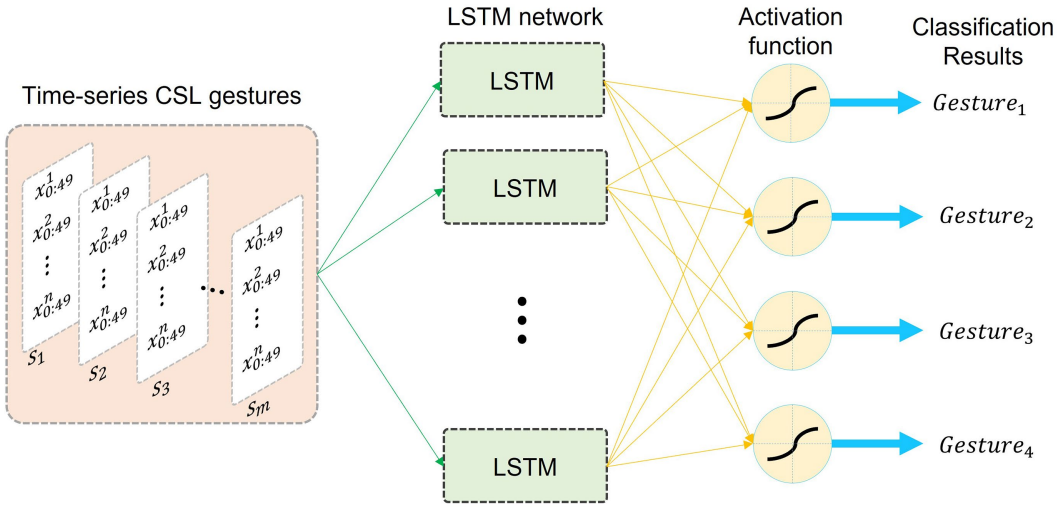


FIGURE 8
The LSTM model used in the experiment.

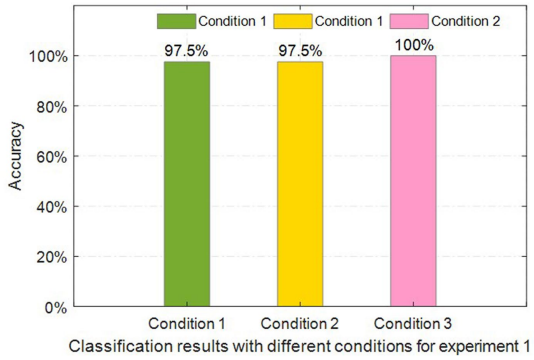


FIGURE 9
Classification accuracies under three conditions for the first experiment.

3.2.2. Results and analysis

For all conditions, the training processes were performed and repeated several times to obtain a better model of the LSTM classifier. At the end of all the epochs of training, the model is made to test with the cross-validation data and its accuracy is also echoed. To prevent overfitting, the model is trained over 100 times. At each time, the loss and accuracy are noted. At the end of each training, the model is saved. The model with the least loss and highest cross-validation accuracy is chosen for use in the second experiment. The classification results of CSL gestures under three conditions are shown in Figure 9.

Obviously, the classification accuracy under condition 3 achieves 100%, while the accuracy could not achieve that under conditions 1 and 2. In other words, the multimodal sensor fusion-based input data obtains a better performance in comparison with that of single-modality sensor data. The recognition accuracies under conditions 2 and 3 are the same when the single modality sensor datasets are used.

Figure 10 shows the classification results corresponding to three conditions of the above-mentioned scenarios. With the increase of training epochs, loss gradually converges to three different values corresponding to three conditions. It means that the multimodal fusion data achieves the highest recognition accuracy with convergent loss values.

As shown in Figure 11, the confusion matrices under three conditions are presented to explore the impact on classification results based on misclassified gestures and different modality data. The recognition accuracy under condition 3 is 100%, which means that all testing gestures are correctly recognized. Hence, we will not discuss the confusion matrix under condition 3. From Figure 11, we can find that only one CSL gesture is classified incorrectly under conditions 1 and 2 in the test samples. This is because both of the conditions have the same recognition accuracy. But the misclassified gestures are not the same. The misclassified gesture type is "you" under condition 1, and that is "me" under condition 2. That is probably because the two gestures have the same postures except for directions.

3.3. HRI experiments and results

3.3.1. Settings

The second experiments were conducted on an NAO robot based on the first experiments. Firstly, two different modalities of testing CSL gestures were sent to the LSTM classifier. Then, the recognition results were transported to the NAO robot for understanding and reaction. Based on the recognition results, Choregraphe APP converts the corresponding gestures into executable commands so that the robot can perform and respond. In other words, the output of the classifiers being coded into commands for the robot's response. Hence, the recognition of human hand motion for the robot is from the system. Choregraphe connects robots via Ethernet. The experimental platform and experiment steps are shown in Figure 12. Once these gestures are classified and sent to Choregraphe, the corresponding responses of the NAO robot will be performed.

Noting that the classifier model under the third condition in the first experiment is saved to recognize the testing gestures in

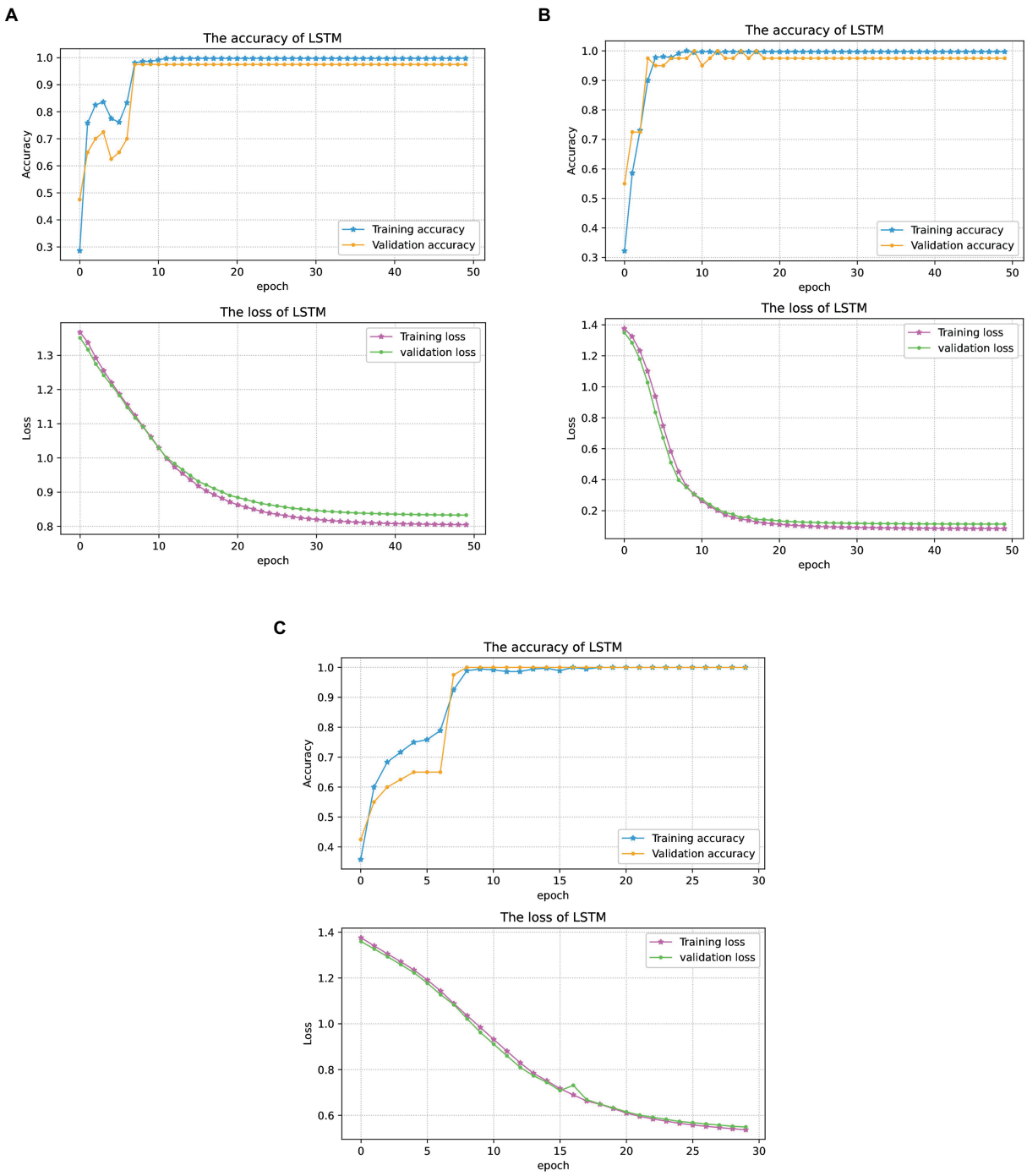


FIGURE 10
The first experiment results under three conditions. **(A)** CSL gestures classification results under condition 1. **(B)** CSL gestures classification results under condition 2. **(C)** CSL gestures classification results under condition 3. In panels **(A–C)**, the blue curves with star markers denote the recognition accuracy of the training set, and the yellow curves with circle markers denote the recognition accuracy of the testing set. The magenta and green curves are the loss values of the training sets and testing sets, respectively.

this experiment. Testing data include two types of gestures: four kinds of captured singular gestures and two combination gestures consisting of them. The combined CSL gesture is composed according to Chinese grammar which can express a complete meaning. The testing gestures are shown in Table 3. In Chinese, "hello" is a combination of the two words "you" and "good," and "hello, everyone" is a combination of the three words "you," "us," and "good." As shown in Table 3, six gestures are tested in total for the second experiment.

3.3.2. Results and analysis

The experiment was performed more than 10 times for each gesture. Figure 13 shows the response results of the NAO robot corresponding to the six gestures. In Figure 13A, the words in the

10.3389/fnins.2023.1168888

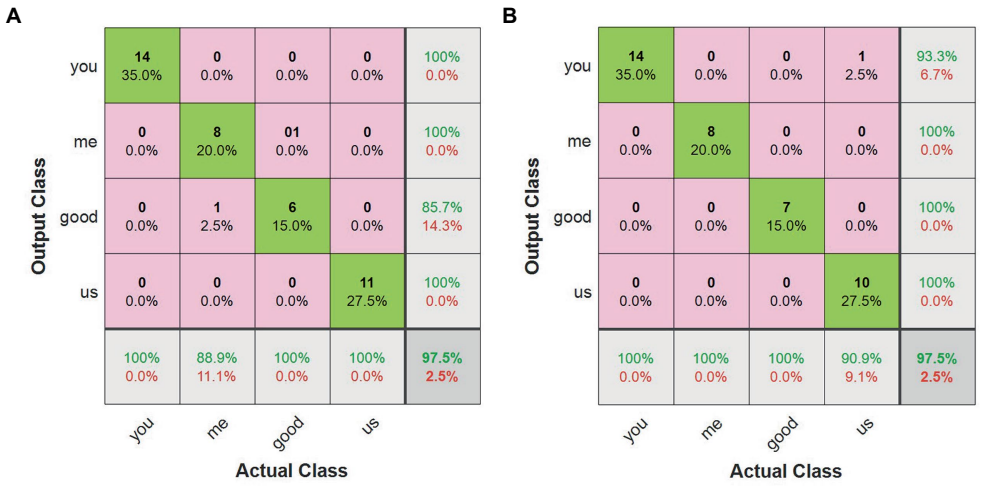


FIGURE 11
The confusion matrices of the first experiment under conditions 1 and 2. **(A)** The confusion matrix under condition 1. **(B)** The confusion matrix under condition 2. In panels **(A,B)**, *x*-axis denotes the real sample labels, and the *y*-axis denotes the predicted sample labels. The top and bottom elements on the main diagonal filled with green color, respectively, represent the number and percentage of the samples that are correctly predicted. The top and bottom elements inside of each pink square, respectively, represent the number and percentage of wrong predicted samples. The top and bottom elements inside of lower and right light gray squares represent the prediction accuracy and error rate of corresponding samples.

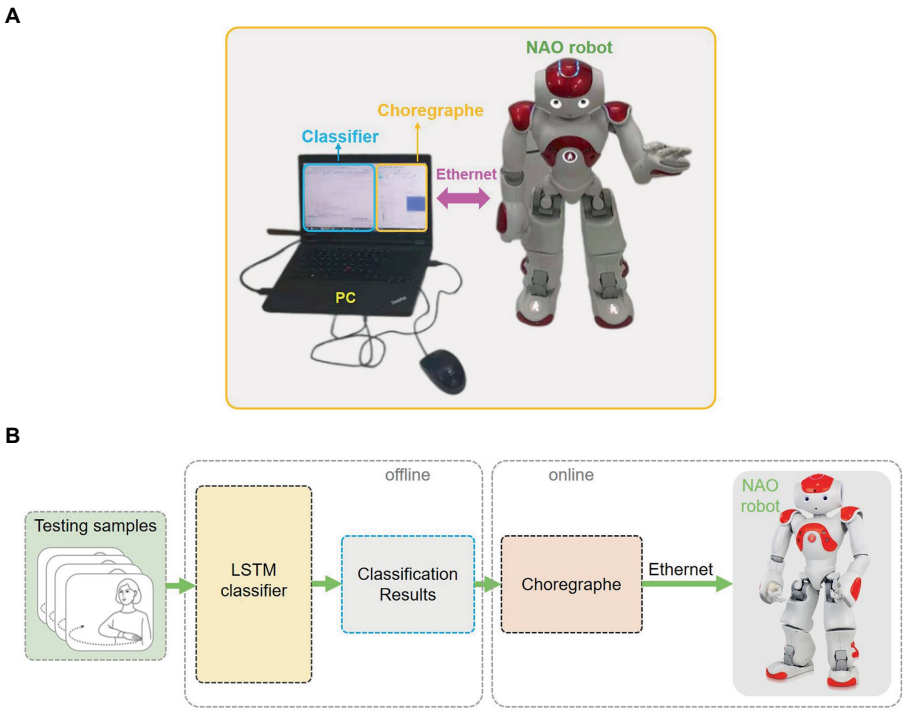


FIGURE 12
The experimental system of the second experiment. **(A)** The experimental platform of the second experiment. **(B)** The experimental steps of the second experiment.

upper right corner are four kinds of gesture results recognized by the NAO robot and the gestures of NAO are the corresponding response results. In Figure 13B, the response results of the NAO robot gesture are divided into two steps for each combination gesture. Obviously, the robot's responses to the six gestures are different. For the single gestures, the robot's response is only one step. However, the response according to the combination gestures is two steps. This implies that the proposed framework can interact with people through CSL gestures and react with reasonable responses. It also indicated that the proposed system can not only interact with the robot based on a single gesture but also interact through a combination of gestures.

10.3389/fnins.2023.1168888

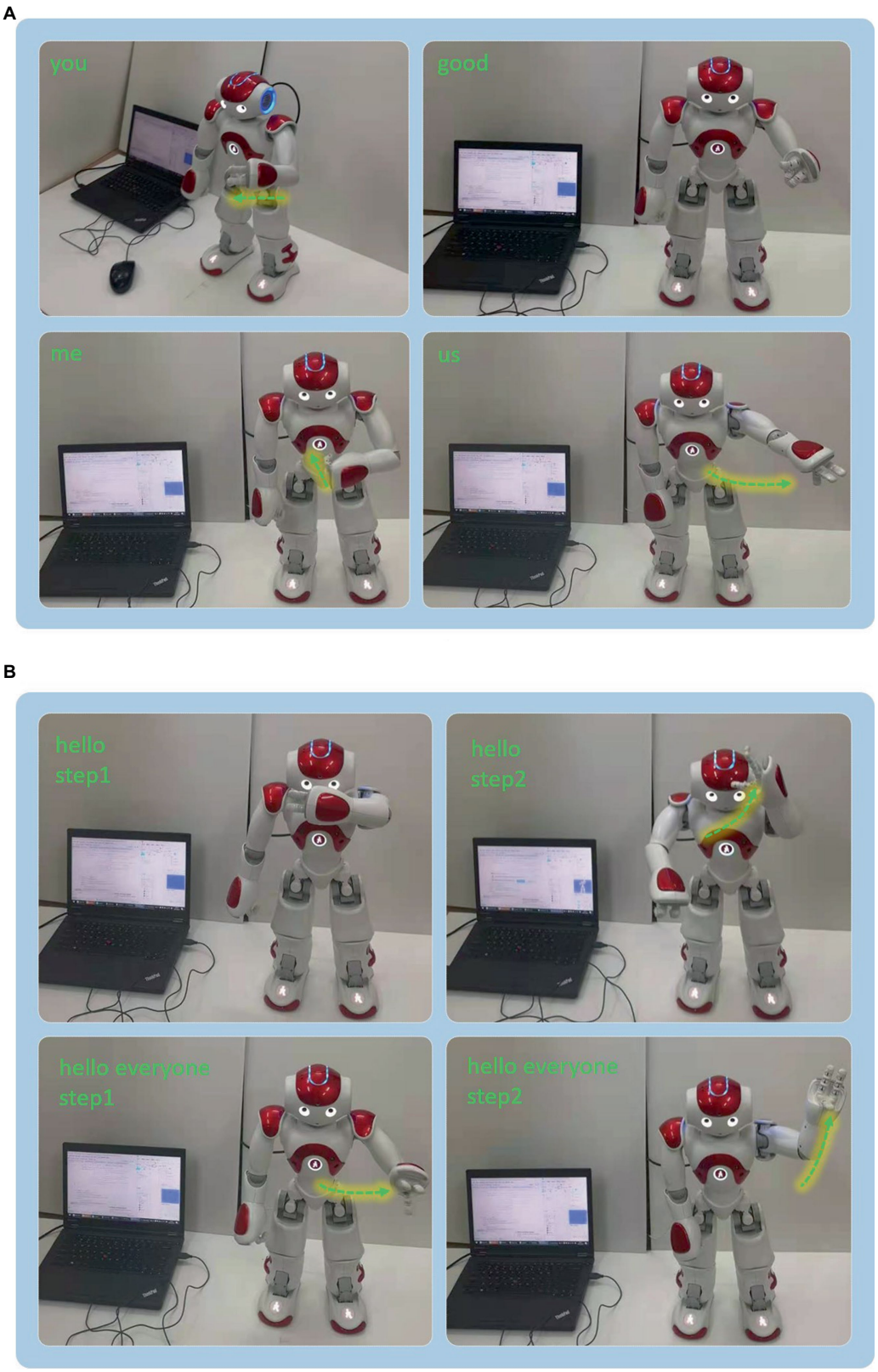


FIGURE 13
The NAO robot interaction results of experiment 2. **(A)** Robot response results of the four singular hand gestures. **(B)** Robot response results of the two combination gestures.

TABLE 3 All testing gestures in the second experiment.

Type	Gestures
Four singular hand gestures	Good
	You
	Us
	Me
Two combination gestures	Hello (combination of you and Good gestures)
	Hello, everyone (combination of you, us, and Good gestures)

4. Discussion

Two experiments are conducted to verify the effectiveness of the proposed framework. According to the first experimental results, we can conclude that the multimodal sensor data can effectively improve recognition accuracy, similar to the experimental findings of Zeng et al. (2019) and Zeng et al. (2020). The confusion matrices of experiment 1 under conditions 1 and 2 imply that different single-modal sensor data can classify different kinds of gestures. Leap Motion sensor data can achieve a good result in human hand posture by capturing a 3D skeletal hand model. The Myo armband sensor can obtain better results in gestures with significant differences in sEMG signals. This also demonstrates that different modal sensor data provides complementary information. Hence, the fused multimodal data achieves the best results in the first experiment.

To investigate the application of our proposed framework in HRI and its advantages in CSL gesture classification, we performed another experiment. In general, most of the conventional gesture classification frameworks can only classify singular static or dynamic sign language gestures. However, our SLR framework can classify both singular and combination gestures well. This combination is not only in terms of gestures but also in terms of the special framework. As the input of the LSTM network, the dynamic and static gestures samples are mixed in one dataset. We can distinguish them effectively by the specific features captured from the LMC. The second experimental results have proved that. In addition, our proposed SLR framework also can be applied in other HRI scenarios. It provides a novel way for the SLR application in social robots and provides a compatible SLR framework.

5. Conclusion and future work

This paper presented a multimodal CSL recognition framework applied in HRI between deaf-mutes and social robots. The multimodal framework considers multiple sensor information for the human hand and arm, including human 3D vector and arm sEMG signals. The Leap Motion sensor and Myo armband are used to capture corresponding signals. Then, the preprocessing techniques are carried out aimed at reducing the training process to improve recognition accuracy to some extent. For LMC data, the normalization method is to limit data to a certain range to eliminate the adverse effects of singular samples. Since the sEMG signals are noisy, low-pass filtering and band-pass filtering are used to preprocess the signals. After that, the RMS feature is extracted from sEMG signals and fused with Leap Motion data as the input data of the classifier. Our method fuses the sensor data from a wearable and vision-based devices at the feature level. Comparative experiments have validated the method. The proposed multimodal framework can facilitate deaf and speech-impaired people to learn sign language through a social robot with the ability of SLR. Our future work will concentrate on developing a framework with a stronger generalization capability to recognize various sign languages without the limitation of country and language restrictions.

Data availability statement

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

Author contributions

JL: human-robot interaction system design, methodology, and experiments. JL and NW: results analysis. JL: manuscript writing and original draft. JL, JZ, and NW: review and editing. JL and JZ: funding acquisition. Each author has read and edited the manuscript, and agrees with its content. All authors contributed to the article and approved the submitted version.

Funding

This work was supported by the Startup Foundation of Chongqing Technology and Business University under Grant No. 950321049 and No. 2056019, and partially supported by the Germany/Hong Kong Joint Research Scheme sponsored by the Research Grants Council of Hong Kong and the German Academic Exchange Service of Germany (Ref. No. G-PolyU505/22), PolyU Start-up Grant: ZVUY-P0035417, CD5E-P0043422 and WZ09-P0043123.

Acknowledgments

The authors thank the participants for their valuable time in data collecting.

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.2023.1168888/full#supplementary-material

References

AI Farid, F., Hashim, N., Abdullah, J., Bhuiyan, M. R., Shahida Mohd Isa, W. N., Uddin, J., et al. (2022). A structured and methodological review on vision-based hand gesture recognition system. *J. Imag.* 8:153. doi: 10.3390/jimaging8060153

Arief, Z., Sulistijono, I. A., and Ardiansyah, R. A. (2015). "Comparison of five time series EMG features extractions using Myo armband." in *The 2015 international electronics symposium (Surabaya)*. 11–14.

Bartneck, C., Belpaeme, T., Eyssel, F., Kanda, T., Keijsers, M., and Sabanovic, S. (2019). Human robot interaction. *Hum. Robot Interact Introduct.*, 6–17. Available at: https://sc.panda321.com/#v=onepage&q=Human%20robot%20interaction&f=false

Bird, J. J., Ekárt, A., and Faria, D. R. (2020). British sign language recognition via late fusion of computer vision and leap motion with transfer learning to american sign language. *Sensors* 20:5151. doi: 10.3390/s20185151

Breazeal, C., Dautenhahn, K., and Kanda, T. (2016). "Social robotics" in *Springer Handbook of Robotics*, 1935–1972. Available at: https://link.springer.com/chapter/10.1007/978-3-319-32552-1_72

Camargo, J., and Young, A. (2019). Feature selection and non-linear classifiers: effects on simultaneous motion recognition in upper limb. *IEEE Trans. Neural Syst. Rehabil. Eng.* 27, 743–750. doi: 10.1109/tnsre.2019.2903986

Camgoz, N. C., Hadfield, S., Koller, O., Ney, H., and Bowden, R. (2018). "Neural sign language translation" in the *IEEE Conference on Computer Vision and Pattern Recognition (Utah)*. 7784–7793.

Cao, H., Chen, G., Li, Z., Feng, Q., Lin, J., and Knoll, A. (2022a). Efficient Grasp Detection Network With Gaussian-Based Grasp Representation for Robotic Manipulation. *IEEE/ASME Transactions on Mechatronics*. doi: 10.1109/TMECH.2022.3224314

Cao, H., Chen, G., Li, Z., Hu, Y., and Knoll, A. (2022b). NeuroGrasp: multimodal neural network with Euler region regression for neuromorphic vision-based grasp pose estimation. *IEEE Transactions on Instrumentation and Measurement* 71, 1–11. doi: 10.1109/TIM.2022.3179469

Chavez-Garcia, R. O., and Aycard, O. (2015). Multiple sensor fusion and classification for moving object detection and tracking. *IEEE Trans. Intell. Transp. Syst.* 17, 525–534. doi: 10.1109/tits.2015.2479925

Chen, G., Cao, H., Conradt, J., Tang, H., Rohrbein, F., and Knoll, A. (2020). Event-based neuromorphic vision for autonomous driving: A paradigm shift for bio-inspired visual sensing and perception. *IEEE Signal Processing Magazine*. 37, 34–49. doi: 10.1109/MSP.2020.2985815

Cheok, M. J., Omar, Z., and Jaward, M. H. (2019). A review of hand gesture and sign language recognition techniques. *Int. J. Mach. Learn. Cybern.* 10, 131–153. doi: 10.1007/s13042-017-0705-5

Chong, T. W., and Lee, B. G. (2018). American sign language recognition using leap motion controller with machine learning approach. *Sensors* 18:3554. doi: 10.3390/s18103554

Cohen, I., Looije, R., and Neerincx, M. A. (2011). "Child's recognition of emotions in robot's face and body" in the *6th International Conference on Human-robot Interaction*. 123–124.

Cui, R., Liu, H., and Zhang, C. (2019). A deep neural framework for continuous sign language recognition by iterative training. *IEEE Trans. Multimedia* 21, 1880–1891. doi: 10.1109/TMM.2018.2889563

Garimort, J., Hornung, A., and Bennewitz, M. (2011). "Humanoid navigation with dynamic footstep plans" in the *2011 IEEE International Conference on Robotics and Automation (Shanghai)*. 3982–3987.

Gonzalez-Aguirre, J. A., Osorio-Oliveros, R., Rodríguez-Hernández, K. L., Lizárraga-Iturralde, J., Morales Menendez, R., Ramírez-Mendoza, R. A., et al. (2021). Service robots: trends and technology. *Appl. Sci.* 11:10702. doi: 10.3390/app112210702

Graves, A. (2013). Generating sequences with recurrent neural networks. arXiv [Preprint]. doi: 10.48550/arXiv.1308.0850

Hochreiter, S., and Schmidhuber, J. (1997). Long short-term memory. *Neural Comput.* 9, 1735–1780. doi: 10.1162/neco.1997.9.8.1735

Hong, P., Turk, M., and Huang, T. S. (2000). "Gesture modeling and recognition using finite state machines." in *The 4th IEEE International Conference on Automatic Face and Gesture Recognition (Grenoble)*. 410–415

Karami, A., Zanj, B., and Sarkaleh, A. K. (2011). Persian sign language (PSL) recognition using wavelet transform and neural networks. *Expert Syst. Appl.* 38, 2661–2667. doi: 10.1016/j.eswa.2010.08.056

Krishnan, S., Akash, R., Kumar, D., Jain, R., Rathai, K. M. M., and Patil, S. (2019). "Finger movement pattern recognition from surface EMG signals using machine learning algorithms." in the 2017 International Conference on Translational Medicine and Imaging (Vellore). 2017, 75–89.

Kumar, P., Gauba, H., Roy, P. P., and Dogra, D. P. (2017a). A multimodal framework for sensor based sign language recognition. *Neurocomputing* 259, 21–38. doi: 10.1016/j.neucom.2016.08.132

Kumar, P., Gauba, H., Roy, P. P., and Dogra, D. P. (2017b). Coupled HMM-based multi-sensor data fusion for sign language recognition. *Pattern Recogn. Lett.* 86, 1–8. doi: 10.1016/j.patrec.2016.12.004

Kundu, A. S., Mazumder, O., Lenka, P. K., and Bhaumik, S. (2018). Hand gesture recognition based omnidirectional wheelchair control using IMU and EMG sensors. *J. Intell. Robot. Syst.* 91, 529–541. doi: 10.1007/s10846-017-0725-0

Kurdyumov, R., Ho, P., and Ng, J. (2011). Sign language classification using webcam images. *Comput. Therm. Sci.* 10:9029. Available at: http://cs229.stanford.edu/proj2011/KurdyumovHoNg-SignLanguageClassificationUsingWebcamImages.pdf

Le Sant, G., Gross, R., Hug, F., and Nordez, A. (2019). Influence of low muscle activation levels on the ankle torque and muscle shear modulus during plantar flexor stretching. *J. Biomech.* 93, 111–117. doi: 10.1016/j.jbiomech.2019.06.018

Li, J., Zhong, J., Chen, F., and Yang, C. (2019). "An incremental learning framework for skeletal-based hand gesture recognition with leap motion." In The IEEE 9th Annual International Conference on CYBER Technology in Automation. Control, and Intelligent Systems (Suzhou). 13–18.

Li, J., Zhong, J., Yang, J., and Yang, C. (2020). An incremental learning framework to enhance teaching by demonstration based on multimodal sensor fusion. *Front. Neurorobot.* 14:55. doi: 10.3389/fnbot.2020.00055

Lichtenauer, J. F., Hendriks, E. A., and Reinders, M. J. (2008). Sign language recognition by combining statistical DTW and independent classification. *IEEE Trans. Pattern Anal. Mach. Intell.* 30, 2040–2046. doi: 10.1109/TPAMI.2008.123

Luo, J., Liu, C., Feng, Y., and Yang, C. (2020). A method of motion recognition based on electromyographic signals. *Adv. Robot.* 34, 976–984. doi: 10.1080/01691864.2020.1750480

Mitra, S., and Acharya, T. (2007). Gesture recognition: a survey. *IEEE Trans. Syst. Man Cyber. C.* 37, 311–324. doi: 10.1109/TSMCC.2007.893280

Naglot, D., and Kulkarni, M. (2016). "Real time sign language recognition using the leap motion controller" in the *2016 International Conference on Inventive Computation Technologies* (Karnataka). 3, 1–5.

Oudah, M., Al-Naji, A., and Chahl, J. (2020). Hand gesture recognition based on computer vision: a review of techniques. *J. Imag.* 6:73. doi: 10.3390/jimaging6080073

Phinyomark, A., Limsakul, C., and Phukpattaranont, P. (2009). A novel feature extraction for robust EMG pattern recognition. arXiv [Preprint]. doi: 10.48550/arXiv.0912.3973

Phinyomark, A., Quaine, F., Charbonnier, S., Serviere, C., Tarpin-Bernard, F., and Laurillau, Y. (2013). EMG feature evaluation for improving myoelectric pattern recognition robustness. *Expert Syst. Appl.* 40, 4832–4840. doi: 10.1016/j.eswa.2013.02.023

Pu, J., Zhou, W., and Li, H. (2018). "Dilated convolutional network with iterative optimization for continuous sign language recognition." in the *27th International Joint Conference on Artificial Intelligence (Stockholm)*. 885–891.

Qi, W., Ovur, S. E., Li, Z., Marzullo, A., and Song, R. (2021). Multi-sensor guided hand gesture recognition for a teleoperated robot using a recurrent neural network. *IEEE Robot. Automat. Lett.* 6, 6039–6045. doi: 10.1109/LRA.2021.3089999

Rastgoo, R., Kiani, K., and Escalera, S. (2020). Video-based isolated hand sign language recognition using a deep cascaded model. *Multimed. Tools Appl.* 79, 22965–22987. doi: 10.1007/s11042-020-09048-5

Roda-Sanchez, L., Garrido-Hidalgo, C., García, A. S., Olivares, T., and Fernández-Caballero, A. (2023). Comparison of RGB-D and IMU-based gesture recognition for human-robot interaction in remanufacturing. *Int. J. Adv. Manuf. Technol.* 124, 3099–3111. doi: 10.1007/s00170-021-08125-9

Si, Y., Chen, S., Li, M., Li, S., Pei, Y., and Guo, X. (2022). Flexible strain sensors for wearable hand gesture recognition: from devices to systems. *Adv. Intellig. Syst.* 4:2100046. doi: 10.1002/aisy.202100046

Siciliano, B., and Khatib, O. (2016). *Springer Handbook of Robotics* Springer, 1–6. Available at: https://link.springer.com/book/10.1007/978-3-319-32552-1?source=shoppingads&locale=en-jp&gclid=Cj0KCQiAwJWdBhCYARIsAJc4idCcF2us102UDVGHpi9py3j3kDIRfTV8W-cT0Jx8dgDKWGwDZj2053EaAqIdEALw_wcB

Su, Z. W., Huang, C. Q., and Pan, W. (2007). A study on obstacle avoidance for Nao robot based on Webots platform.

Subasi, A. (2012). Classification of EMG signals using combined features and soft computing techniques. *Appl. Soft Comput.* 12, 2188–2198. doi: 10.1016/j.asoc.2012.03.035

Tharwat, A., Gaber, T., Hassanien, A. E., Shahin, M. K., and Refaat, B. (2015). Sift-based arabic sign language recognition system. *Adv. Intellig. Syst. Comput.* 334, 359–370. doi: 10.1007/978-3-319-13572-4_30

Wang, L., Hu, W., and Tan, T. (2003). Recent developments in human motion analysis. *Pattern Recogn.* 36, 585–601. doi: 10.1016/S0031-3203(02)00100-0

Wei, C., Zhou, W., Pu, J., and Li, H. (2019). "Deep grammatical multi-classifier for continuous sign language recognition" in *The 5th International Conference on Multimedia Big Data (Singapore)*. 435–442.

Weichert, F., Bachmann, D., Rudak, B., and Fisseler, D. (2013). Analysis of the accuracy and robustness of the leap motion controller. *Sensors* 13, 6380–6393. doi: 10.3390/s130506380

Wong, S. F., and Cipolla, R. (2005). "Real-time adaptive hand motion recognition using a sparse bayesian classifier." in the *International Workshop on Human-Computer Interaction*. 170–179.

Wu, Y., and Huang, T. S. (1999). "Vision-based gesture recognition: a review." in International Gesture Workshop (France). 103–115.

Wu, D., Pigou, L., Kindermans, P. J., Le, N. D. H., Shao, L., Dambre, J., et al. (2016). Deep dynamic neural networks for multimodal gesture segmentation and recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* 38, 1583–1597. doi: 10.1109/tpami.2016.2537340

Xue, Y., Ju, Z., Xiang, K., Chen, J., and Liu, H. (2018). Multimodal human hand motion sensing and analysis—a review. *IEEE Trans. Cogn. Dev. Syst.* 11, 162–175. doi: 10.1109/tcds.2018.2800167

Yang, C., Chen, C., He, W., Cui, R., and Li, Z. (2018a). Robot learning system based on adaptive neural control and dynamic movement primitives. *IEEE Trans. Neural Net. Learn. Syst.* 30, 777–787. doi: 10.1109/TNNLS.2018.2852711

Yang, C., Chen, C., Wang, N., Ju, Z., Fu, J., and Wang, M. (2018b). Biologically inspired motion modeling and neural control for robot learning from demonstrations. *IEEE Trans. Cogn. Dev. Syst.* 11, 281–291. doi: 10.1109/TCDS.2018.2866477

Ye, Y., Tian, Y., Huenerfauth, M., and Liu, J. (2018). "Recognizing american sign language gestures from within continuous videos" in *The IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 2064–2073.

Zardoshti-Kermani, M., Wheeler, B. C., Badie, K., and Hashemi, R. M. (1995). EMG feature evaluation for movement control of upper extremity prostheses. *IEEE Trans. Rehabil. Eng.* 3, 324–333. doi: 10.1109/86.481972

Zeng, C., Yang, C., Zhong, J., and Zhang, J. (2019). Encoding multiple sensor data for robotic learning skills from multimodal demonstration. *IEEE Access* 7, 145604–145613. doi: 10.1109/access.2019.2945484

Zeng, C., Yang, C., Cheng, H., Li, Y., and Dai, S. L. (2020). Simultaneously encoding movement and sEMG-based stiffness for robotic skill learning. *IEEE Transactions on Industrial Informatics* 17, 1244–1252. doi: 10.1109/TII.2020.2984482

frontiers | Frontiers in Neuroscience

TYPE Original Research
PUBLISHED 20 April 2023
DOI 10.3389/fnins.2023.1158246

Check for updates

OPEN ACCESS

EDITED BY
Manning Wang,
Fudan University, China

REVIEWED BY
Ankit Bhurane,
Visvesvaraya National Institute of Technology, India
Shifeng Wang,
Naval Medical Center, China

*CORRESPONDENCE
Xianghong Lin
✉ linxh@nwnu.edu.cn

RECEIVED 03 February 2023
ACCEPTED 03 April 2023
PUBLISHED 20 April 2023

CITATION
Zhao Y, Lin X, Zhang Z, Wang X, He X and Yang L (2023) STDP-based adaptive graph convolutional networks for automatic sleep staging. *Front. Neurosci.* 17:1158246. doi: 10.3389/fnins.2023.1158246

COPYRIGHT
© 2023 Zhao, Lin, Zhang, Wang, He and Yang. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

STDP-based adaptive graph convolutional networks for automatic sleep staging

Yuan Zhao, Xianghong Lin*, Zequn Zhang, Xiangwen Wang, Xianrun He and Liu Yang

College of Computer Science and Engineering, Northwest Normal University, Lanzhou, China

Automatic sleep staging is important for improving diagnosis and treatment, and machine learning with neuroscience explainability of sleep staging is shown to be a suitable method to solve this problem. In this paper, an explainable model for automatic sleep staging is proposed. Inspired by the Spike-Timing-Dependent Plasticity (STDP), an adaptive Graph Convolutional Network (GCN) is established to extract features from the Polysomnography (PSG) signal, named STDP-GCN. In detail, the channel of the PSG signal can be regarded as a neuron, the synapse strength between neurons can be constructed by the STDP mechanism, and the connection between different channels of the PSG signal constitutes a graph structure. After utilizing GCN to extract spatial features, temporal convolution is used to extract transition rules between sleep stages, and a fully connected neural network is used for classification. To enhance the strength of the model and minimize the effect of individual physiological signal discrepancies on classification accuracy, STDP-GCN utilizes domain adversarial training. Experiments demonstrate that the performance of STDP-GCN is comparable to the current state-of-the-art models.

KEYWORDS

sleep stage classification, graph convolutional network (GCN), spike-timing-dependent plasticity (STDP), domain adaptation, Polysomnography (PSG)

1. Introduction

A proper sleep cycle plays a vital role in maintaining one's mental and physical wellbeing. However, with the increasing mental stress of modern life, sleep disorders have become an issue that cannot be overlooked. Sleep quality and sleep disturbances are usually assessed by dividing the sleep state according to the patient's Polysomnography (PSG) throughout the night, PSG records various human physiological signals such as Electroencephalography (EEG), Electromyogram (EMG), Electrooculogram (EOG) and Electrocardiogram (ECG). The Rechtschaffen and Kales standard (Wolpert, 1969) and American Academy of Sleep Medicine (AASM) standard (Berry et al., 2012) are commonly used to classify PSG signals as a standard code for classifying sleep states. One person's overnight PSG recording is a very large amount of data, manually labeling such a large number of PSG signals is a very single and tedious task, and it is prone to errors, which is unbearable for clinically diagnosed sleep disorders. Therefore, it is crucial to identify and categorize sleep state staging in order to properly diagnose sleep-related disorders. Automatic sleep staging can greatly improve the efficiency and accuracy of sleep state classification, and greatly liberate human resources so that experts can focus more on diagnosing and treating diseases.

There has been a lot of valuable work on automatic sleep state classification in recent years, automatic sleep staging mainly uses traditional machine learning methods in the early stage, such as Support Vector Machine (SVM) (Alickovic and Subasi, 2018) or Random Forest (RF) (Memar and Faradji, 2018), which have high requirements for handcrafted features. Since traditional machine learning methods require complex feature engineering, researchers began to use deep learning for automatic sleep staging and achieved high accuracy (Supratak et al., 2017; Phan et al., 2019; Bakker et al., 2022; Li et al., 2022; Martín-Montero et al., 2023; Zhang et al., 2023). Although deep learning methods have achieved high accuracy, they have not fully exploited the topology of functional connections in different brain regions. The brain is a complex network of structurally and functionally interconnected regions, localized dysfunction often propagates and affects other regions leading to large-scale network changes. The recent development of graph neural networks (GNN) (Kipf and Welling, 2016) has led researchers to explore the use of GNN to extract spatial features of PSG signals, the GraphSleepNet (Jia et al., 2020) uses GCN to extract EEG signals by using EEG signals as nodes of GNN, which achieves state-of-the-art performance compared to previous methods.

In addition, the automatic sleep staging task also face a challenge, which is the trained model often performs well on the training data set, but the performance of the model is often unsatisfactory due to individual differences or measurement equipment errors in actual application. The physiological signals of different subjects vary greatly, so it is necessary to consider improving the adaptability of the model to different data distributions. Some efforts have tried to use domain adaptation to improve the adaptability of the model (Tzeng et al., 2014; Ganin et al., 2016; Jia et al., 2021a) and have achieved good results. The basic idea of Domain Adaptation is to map the source domain and target domain data into a feature space. By finding a unified metric in the same feature space, the feature distribution of the source domain and target domain data is as close as possible, which can improve the performance of the model based on source domain data feature training on target domain data.

The current method has achieved high accuracy in automatic sleep staging tasks, but the following challenges still need to be solved: (1) The feature extraction ability of the model needs to be improved. In particular, the current model does not make full use of the functional connection between brain regions and the interdependence between different modes of data in PSG data. (2) The graph-building algorithm of the GNN model is often based on back-propagation while ignoring the interpretability of the graph-building algorithm. (3) It is necessary to effectively improve the adaptive ability of the model to the data. Due to the huge differences in physiological signals between individuals, models with good performance in training data sets often perform poorly in actual deployment.

The establishment method of graph structure is the core to solving the first two challenges, which due to a graph of different brain regions can be seen as an explainable result, as brain region connections with abnormal patterns can help explain the causes of sleep disorders (Griffa et al., 2013). Building an explainable graph structure is difficult due to: (1) Pre-defined graphs cannot adapt to functional connectivity of brain regions in different sleep stages; (2) Graph generation algorithms trained by end-to-end may learn unsuitable parameters with small amounts of train data, and this approach is less explainable.

To address the difficulty of building graphs for GNN, we adaptively compute graph structures through a neuroscience mechanism. When using GNN for automatic sleep staging, we assume that each PSG channel corresponds to a node in the graph, and the connections between channels correspond to connections between different brain regions. The connections between brain regions are made up of connections between neurons, and neurons are connected through synapses, so it is reasonable to build connections between brains through the strength of synapses. The synapses adjustment rule between neurons has made a lot of progress in neuroscience (Fornito et al., 2015), such as the Hebbian theory proposed by Hebbian (Hebb, 1949), which shows that the weight between two neurons increases if the two neurons activate simultaneously, and reduces if they activate separately, which is often summarized as “Cells that fire together wire together". But Hebbian theory doesn’t make predictions about the firing of presynaptic neurons after postsynaptic neurons, which is solved by spike-timing-dependent plasticity (STDP). The concept of STDP was first proposed by Taylor (1973), Bi and Poo (1998) discovered that postsynaptic synapses that were activated within 5–20 ms before the spike were strengthened, whereas synapses that were activated within a similar time window after the spike was weakened, STDP core idea is to calculate the weight of the direct connection of two neurons according to the sequence of the two connected neurons firing pulses (Dan and Poo, 1992). These rules about weight adjustment between neurons motivate us to apply weight adjustment rules between neurons to build graph structures.

In this paper, we propose an adaptive GCN based on Spike-Timing-Dependent Plasticity, named STDP-GCN. The connection between various PSG signal channels forms a graph structure, and the channel of the PSG signal may be thought of as a neuron, the STDP process used to build the strength of the synapses between neurons, which builds the graph. The transition rules between sleep stages are extracted using temporal convolution after the GCN, and classification is performed using a fully connected neural network. In particular, domain adaptation is applied in the classification network to improve the adaptive ability of the STDP-GCN. We summarize the main contributions of this paper:

- An explainable STDP adaptive graph learning algorithm is proposed. The STDP adaptive graph learning algorithm employs the STDP mechanism from neuroscience to dynamically establish inter-channel dependencies without any labeling and exhibits exceptional performance.
- The proposed STDP-GCN can capture both temporal and spatial features of PSG separately through spatio-temporal graph convolution. Furthermore, it can reduce discrepancies between individual physiological signals and enhance performance through domain adaption.
- Through comparative experiments on the ISRUC-S3 dataset and SLEEP-EDF-153 dataset, the proposed STDP-GCN demonstrated the highest accuracy compared to existing models.

2. Related works

2.1. Sleep stage classification problem

The human sleep process can be divided into three main parts: Wake, Rapid Eye Movement (REM), and Non-rapid Eye Movement (NREM) according to AASM standard (Berry et al., 2012). The main features of REM are rapid eye movements and relaxation of body muscles, while NREM is characterized by shallower, slower, and more uniform breathing, slower heart rate, lower blood pressure, and no obvious eyeballs. NREM can be divided into three stages: N1, N2, and N3 to assess the depth of sleep. This article divides sleep states into five categories (Wake, N1, N2, N3, and REM) according to the AASM standard.

The PSG signal is divided into epochs of the 30s, and each epoch is labeled as a sleep state. According to the AASM standard, experts use the features of the PSG data of the current epoch and the previous and previous epochs to mark the sleep state of the current epoch, because sleep state transition patterns are very valuable, for example, it usually enters the N1 stage after wake stage.

In this paper, the sleep stage classification problem can be defined as input multiple epochs, which is defined as $X = (x_{i-c}, ..., x_i, ..., x_{i+c}) \in R^{M\times N\times L}$, output a sleep state of the current epoch $\hat{y}$, where c indicates the temporal context, and $M = 2c + 1$ is the number of temporal contexts, N is the number of nodes in the PSG, L is the number of features per channel.

2.2. Automatic sleep staging methods

Recent years have seen a significant amount of research in the academic field surrounding automatic sleep staging, due to its crucial role in the diagnosis of sleep disorders. Designing features for PSG signals manually through traditional methods is a challenging task due to the complexity of the signal features, which makes deep learning particularly effective in the task of automatic sleep staging.

With the rapid development of deep learning, Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) are widely used in automatic sleep staging. Zhang and Wu (2017) propose a new model called Fast Discriminant Complex-valued Convolutional Neural Network (FDCCNN) for extracting features from raw EEG data and classifying sleep stages. Chambon et al. (2018) introduced a deep neural network to perform temporal sleep stage classification from multimodal and multivariate time series, which can be learned end-to-end without computing spectrograms or extracting manual features. Phan et al. (2019) propose a hierarchical recurrent neural network named SeqSleepNet, which is designed to run on multi-channel time-frequency image inputs to solve the automatic sleep staging problem. Perslev et al. (2019) propose U-time to analyze physiological time series segmentation of sleep data. Cai et al. (2021) propose a novel graph-time fusion dual-input convolutional neural network approach to detect sleep stage. Perslev et al. (2021) introduce A deep learning-based automated sleep staging system (U-SLEEP) that provides accurate segmentation of A wide range of patient cohorts and PSG protocols that were not considered when building the system. Jia et al. (2021b) propose the SalientSleepNet, which is a multimodal significant wave detection network for sleep staging.

Although deep learning achieves high performance, it ignores the interdependencies between PSG signal channels. Jia et al. (2020) propose a new deep graph neural network GraphSleepNet for automatic sleep stage classification, which can adaptively learn the internal connections between different EEG channels. Thus, it can better serve the spatio-temporal graph convolution network (ST-GCN) for sleep stage classification. The lack of interpretability in the above methods highlights the need for a model with explainable features, as interpretability is crucial for understanding the underlying cause of sleep disorders in neuroscience.

3. Methodology

The overall architecture of STDP-GCN is shown in Figure 1. The main ideas of STDP-GCN are as follows: (1) Build the graph structure using an adaptive STDP graph learning algorithm; (2) After a spatio-temporal GCN aggregates the signal, and a fully connected network is used for classification. Models are carefully designed to get the best results in this paper.

3.1. STDP Graph Learning

The process of STDP graph learning algorithm is: (1) encode PSG signals into pulse sequences; (2) calculate the connection weights between pulse sequences according to STDP algorithm, so as to obtain the interdependence between PSG channels. This section first introduces the encoding algorithm and STDP algorithm, and then introduces the STDP graph learning algorithm.

Encoding: STDP learning needs the spike train as input, so the raw PSG signal needs to be converted into a spike train at first. Encoding continuous signals is typically accomplished using analog-to-spike encoding algorithms, including Threshold Based Representation (TBR), Ben's Spiking Algorithm (BSA) (Schrauwen and Van Campenhout, 2003), and Moving Window (MW) (Petro et al., 2020). Typically, BSA algorithms are employed to transform audio data into a spike train. However, as PSG signals are also distributed across the frequency spectrum, some studies (Nuntalid et al., 2011; Medini et al., 2015). have utilized BSA algorithms to encode PSG signals. So the BSA algorithm is well suited to encode PSG signals. In this paper, the BSA algorithm is used to convert the PSG signal into a spike train. The BSA algorithm is based on encoding a signal using an FIR filter. The Finite Impulse Response (FIR) filter is widely used in digital signal processing, the main function is to leave a useful signal, we set the cutoff frequency of FIR to 0.8 and the length to 20 according to the BSA algorithm. By computing two error values Eq.(1) and Eq.(2) at each time instant τ, which can be defined as

$$error1 = \sum_{k=0}^{M} abs\left(s\left(k+\tau\right) - h\left(k\right)\right), \tag{1}$$

$$error2 = \sum_{k=0}^{M} abs\left(s\left(k+\tau\right)\right), \tag{2}$$

here s is the original signal and h is an FIR filter of length M. If Eq.(1) is less than Eq.(2) minus the threshold, then encode a spiking

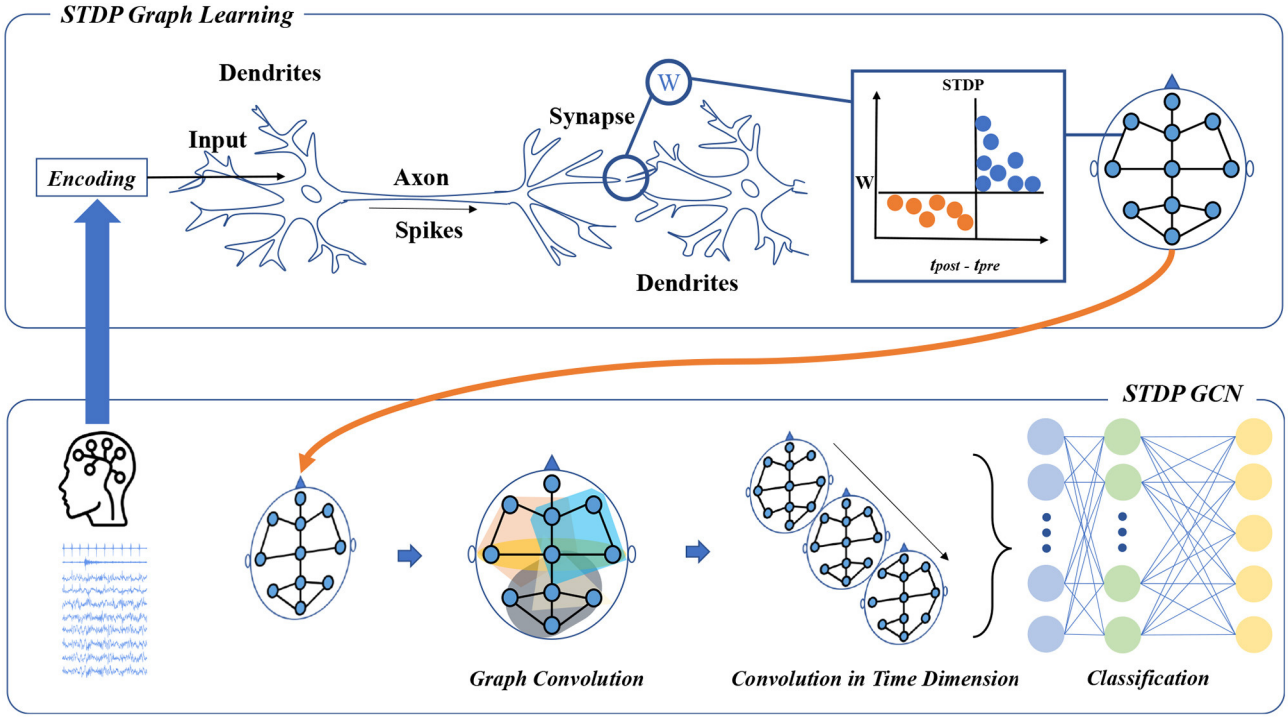


FIGURE 1
The framework of STDP-GCN. The STDP graph learning approach views each channel in the PSG as a neuron, with the STDP mechanism determining the strength of connections between neurons to form a graph structure across the PSG channels. The graph structure is then used for graph convolution, followed by temporal convolution to learn the sleep stage transition rules, and a fully connected neural network is applied for classification.

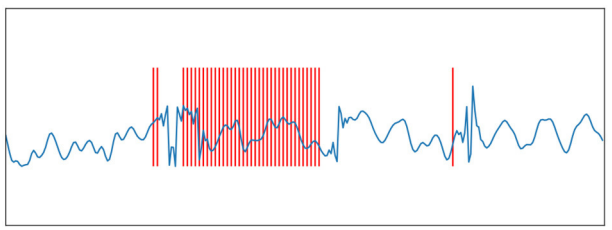


FIGURE 2
Ben's Spiking Algorithm. The blue curve is an EEG signal, and the red vertical line is the spike sequence encoded by the EEG signal according to the BSA encoding algorithm.

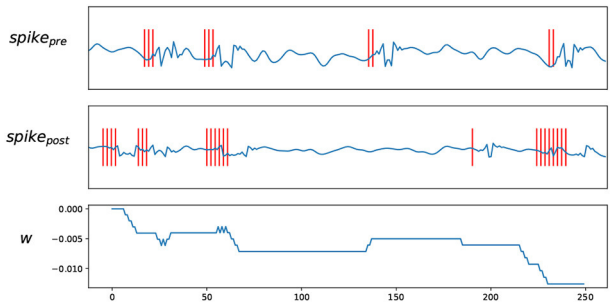


FIGURE 3
Trace STDP rules. $spike_{pre}$ and $spike_{post}$ are the EEG signals and their spike trains encoded by BSA, W shows how the weights change according to the spike trains.

and subtract the filter from the input. The signal can be recovered from the spike train by a convolution between the spike train and the FIR filter. The origin signal and its spike train encoded using BSA are shown in Figures 2, 3.

Spike-timing-dependent plasticity: After encoding the PSG signal into a spike train, the STDP algorithm is used to learn the correlation between the pulse sequences. STDP learning rules are a synaptic plasticity mechanism discovered in biological experiments (Bi and Poo, 1998). A typical neuron consists of a cell body (soma), dendrites, and a single axon. Dendrites receive action potentials from other neurons and transmit them to the body of the cell. Axon's function is to transmit information to different neurons.

Under the STDP process, the synapse will strengthen if the firing spike of the pre-neuron tends to occur on average before the output spike of the post-neuron. If the spiking of the pre-neuron tends to occur immediately after the output spiking of the post-neuron, the synapse weights of the two neurons are slightly weaker (Bi and Poo, 1998). In general, the STDP process can be defined as

$$\Delta\omega = \begin{cases} Ae^{\frac{t_{pre}-t_{post}}{\tau}}, t_{pre} - t_{post} < 0, \\ Be^{-\frac{t_{pre}-t_{post}}{\tau}}, t_{pre} - t_{post} > 0, \end{cases} \quad (3)$$

here, $\Delta\omega$ represents the amount of change in synaptic strength, $t_{pre} - t_{post}$ represents the time difference between the presynaptic pulse and the postsynaptic pulse, $A > 0$ and $B < 0$ are the learning rates that control the $\Delta\omega$. However, the implementation of Eq. (3) is not feasible as it requires separate recording of the firing times of neurons before and after. It is easier to implement STDP using the double-pulse trace-based approach (Morrison et al., 2008) provided by Eq.(4), Eq.(5). The core idea of the double-pulse trace-based approach is that synaptic weights decrease when the pre-neuron fired spike and the synaptic weight increases when the post-neuron is fired.

$$\Delta\omega_{ij}^{-}\left(t_j^f\right) = -F_{-}\left(\omega_{ij}\right) y_i\left(t_j^f\right), \tag{4}$$

$$\Delta\omega_{ij}^{+}\left(t_i^f\right) = -F_{+}\left(\omega_{ij}\right) y_i\left(t_i^f\right), \tag{5}$$

here, Eq.(4) depicts a decrease in synaptic weight when a spike t_j^f from pre-neuron j arrives; Eq.(5) expresses an increase in synaptic weight when a spike t_i^f from post-neuron i arrives, $-F_{+}\left(\omega_{ij}\right)$ and $-F_{-}\left(\omega_{ij}\right)$ are functions that control the increment of weights.

$$\frac{dx_j}{dt} = -\frac{x_j}{\tau_x} + \sum_{t_j^f} \delta\left(t - t_j^f\right), \tag{6}$$

$$\frac{dy_i}{dt} = -\frac{y_i}{\tau_y} + \sum_{t_i^f} \delta\left(t - t_i^f\right). \tag{7}$$

The double-pulse trace-based approach uses trace to describe pre-neuron membrane potential x_j and post-neuron membrane potential y_i. The membrane potential rises immediately when the neuron receives a spike, and then slowly decreases to the resting potential over time, which can be expressed by differential equation Eq.(4), Eq.(5). t_j^f and t_i^f is the spike firing time of post-neuron i after pre-neuron j, δ is the pulse function, which is 1 at $t = 0$, and 0 at other times.

Adaptive graph learning: In this paper, the PSG signal input of an epoch is defined as a graph $G(V, E, A)$, where V is the set of nodes in the graph, each node corresponds to a channel in the PSG, and E represents the edge between nodes, A is the adjacency matrix of the graph. The adjacency graph is an important input of the graph neural network. In this paper, the graph learning algorithm can be defined as inputting PSG data of an epoch and outputting a graph structure of the epoch.

The main purpose of adaptive STDP graph learning is to learn graph structures using STDP. As shown in the upper part of Figure 1, when the PSG is input to the STDP adaptive graph learning module, the signals of each channel in the PSG are first encoded into a spike train by the BSA algorithm. The spike train of a channel is regarded as the spike train emitted by a neuron, the connection between the channel and the channel can be regarded as a synapse, and the synapse strength can be obtained by the STDP. If the pre-neuron emits a spike before the post-neuron emits a spike, it can be seen that there is a connection between the pre-neuron and the post-neuron. The STDP graph learning algorithm used in this paper is distinct from other graph structure construction algorithms in that it relies on the STDP algorithm to establish interdependencies between different channels. However, this algorithm requires time steps for simulation, resulting in increased computational overhead. To save time, we employ an improved STDP algorithm and GPU parallel computing. After the STDP graph learning module, the relationship between channels and channels can be obtained, which is represented as an adjacency matrix. The topology of multivariate data can be used as input to spatio-temporal graph convolution to extract feature representations in the spatial dimension.

The input of STDP-GCN is a sequence of multiple epochs, each epoch will use the STDP graph algorithm to adaptively learn a graph structure, which can be defined as input $X = (x_{t-c}, ..., x_t, ..., x_{t+c})$, output $A = (a_{t-c}, ..., a_t, ..., a_{t+c})$. The PSG signal and graph structure at time step t are represented by x_t and a_t, respectively. The weight calculation between channel j and channel i in the STDP graph structure algorithm can be expressed as

$$a_{ji} = \sum_{t=1}^{L} i(\Delta\omega_{ij}^{-}\left(t_j^f\right) + \Delta\omega_{ij}^{+}\left(t_j^f\right)), \tag{8}$$

here, a_{ji} is the synapse weight between pre-neuron and post-neuron, $\Delta\omega_{ij}^{-}\left(t_j^f\right)$ is the amount of change in the synaptic weight when the presynaptic spike is fired, and $\Delta\omega_{ij}^{+}\left(t_i^f\right)$ is the amount of change in the synaptic weight when the post-synaptic fired spike. After constructing the graph, the preprocessed original signal and the adjacency graph enter the STGCN layer together.

The cross-entropy is used as a loss function to tune the parameters of the spatio-temporal graph convolution, which is defined as

$$L = -\frac{1}{L}\sum_{i=1}^{L}\sum_{n=1}^{N} y_{i,n}\log(\hat{y}), \tag{9}$$

here, L is the number of samples, while N is the number of categories of sleep stages, and y is the ground truth label.

3.2. Spatial-temporal graph convolution

Graph convolution: The main purpose of graph convolution is to aggregate and extract the spatial dimension features of signals. EEG of different channels can measure the electrical signals of corresponding brain regions, and the relationship between signals between brain regions can be aggregated by graph convolution. We use spectral graph convolution theory to build graph convolution layers and to speed up training, we use a simplified GCN. The signal propagation between layers is shown in Eq.(10), where $D^{-\frac{1}{2}}AD^{-\frac{1}{2}}$ is the constructed Laplacian matrix, A is the adjacency graph matrix constructed by STDP, H is the result of the previous layer, and W is the learnable parameter matrix, σ is the activation function.

$$H^{(l)} = \sigma\left(D^{-\frac{1}{2}}AD^{-\frac{1}{2}}H^{(l-1)}W^{(l-1)}\right). \tag{10}$$

Convolution in time dimension: According to the AASM standard, the sleep transition rule, that is, the sleep staging of

the preceding and following periods is an important reference condition for judging the current sleep state. Therefore, taking transition rules into account can improve the accuracy of the classification. STDP-GCN utilizes an adaptive STDP graph learning algorithm for graph construction and feature extraction through GCN at distinct time steps. It subsequently employs time-wise convolution to learn the transition rules. After the data passes through the graph convolution layer, the information of the data has been fully aggregated, and then convolution in the time dimension will better extract the sleep transition rules. The convolution in the time dimension in this paper can be described as follows:

$$H^{(l+1)} = softmax\left(\Phi * \left(softmax\left(H^{(l)}\right)\right)\right), \tag{11}$$

here *softmax* is the activation function, Φ denotes the convolution kernel, $*$ denotes the standard convolution operation.

Domain adaptation:Machine learning models rely heavily on data distribution and the data distribution of PSG may vary significantly due to individual differences. Therefore, we hope that STDP-GCN can effectively learn how to extract common core features. By treating an individual's physiological signal as a domain, we can use the domain adaptation to learn the common features between domains and effectively improve the robustness of the model.

The idea of domain adversarial training (Ganin et al., 2016) originates from Generative Adversarial Network (GAN) (Goodfellow et al., 2014), which consists of a generator and a discriminator. Generators are used to generate false data, and discriminators are used to determine whether the input data is generated false data or real data. The core idea of GAN is to hope that the false data generated by the generator can deceive the discriminator, which is also improving the discriminant ability to prevent being deceived. The two play against each other until the whole system reaches a stable state. Similarly, domain adversarial training is when the model extracts features from the source domain and the target domain, respectively, and then trains the discriminator, hoping that the discriminator cannot distinguish the extracted features from the source domain from the target domain. This allows the target domain's data to be generated with a feature distribution as close to the source image as possible, thereby reducing the domain shift.

As depicted in Figure 4, there are two main tasks to be completed in the domain adversarial training of STDP-GCN: (1) Accurate classification of source domain datasets to minimize the error of automatic sleep staging; (2) To confuse the source domain dataset with the target domain dataset, maximize the domain classification error. Feature extractor G_f maps input x_i to feature space to get domain-invariant feature X_f, and then X_f input domain discriminator D and sleep stage classifier. Feature extractor G_f is defined as

$$X_f^i = G_f\left(x_i; \theta_f\right), \tag{12}$$

where X_f^i denotes the transferred features of x_i, θ_f is the trainable parameter of G_f.

Sleep stage classifier G_f and its loss can be defined as

$$\hat{y}_c^i = G_y\left(X_f^i; \theta_y\right) \tag{13}$$

$$L_c(\hat{y}_c^i, y_i) = \log \frac{1}{\hat{y}_c^i} y_i \tag{14}$$

where $\hat{y}_i^c$ predicted label, θ_y is the trainable parameter of G_y. Domain discriminator G_d and its loss can be defined as

$$\hat{y}_d^i = G_d\left(X_f^i; \theta_d\right), \tag{15}$$

$$L_d(\hat{y}_d^i, d_i) = d_i \log \frac{1}{\hat{y}_d^i} + (1 - d_i) \log \frac{1}{\hat{y}_d^i} \tag{16}$$

where $\hat{y}_i^d$ is the predicted result of the domain discriminator, d_i represents the binary label of the i-th sample and is used to indicate whether the sample belongs to the source or target domain, θ_y is the trainable parameter of G_y. The overall loss of training can be defined as

$$E(\theta_f, \theta_c, \theta_d) = \frac{1}{n} \sum_{i=1}^{n} L_c^i - \lambda \left(\frac{1}{n} \sum_{i=1}^{n} L_d^i + \frac{1}{n_i'} \sum_{i=n+1}^{n} L_d^i \right) \tag{17}$$

Through Gradient Reversal Layer (GRL), domain adaptation can be naturally integrated into the back-propagation algorithm of the network to unify the training process. The network optimization process is defined as

$$(\hat{\theta}_f, \hat{\theta}_c) = \arg\min_{\hat{\theta}_f, \hat{\theta}_c} L(\theta_f, \theta_c, \hat{\theta}_d) \tag{18}$$

$$\hat{\theta}_d = \arg\min_{\theta_d} L(\hat{\theta}_f, \hat{\theta}_y, \theta_d) \tag{19}$$

The parameters of the sleep stage classifier are updated by minimizing the objective function, and the parameters of the domain discriminator are updated by maximizing the objective function.

4. Experiments

4.1. Datasets and experimental settings

In this paper, the ISRUC-S3 dataset and SLEEP-EDF-153 dataset are used to verify the validity of the STDP-GCN model. There are PSG recordings of 10 healthy subjects in the ISRUC-S3 dataset, with 6 EEG channels, 2 EOG channels, 3 EMG channels, and 1 ECG channel. Each epoch is divided into 5 sleep states by AASM standard. The SLEEP-EDF-153 dataset recorded the PSG signals of 78 healthy subjects, and the EEG was obtained by sampling from the Fpz-Cz and Pz-Oz electrode positions at 100 HZ. The SLEEP-EDF-153 dataset classifies labels into eight modes (wake-up, S1, S2, S3, S4, REM, motion, and unknown) according to the Rechtschaffen and Kales standard (Wolpert, 1969). To simplify the process of setting experimental parameters, we combine S3 and S4 into S3 according to AASM standards (Berry et al., 2012).

We use subject-independent cross-validation to test the effect of the STDP-GCN. Due to the different number of individuals

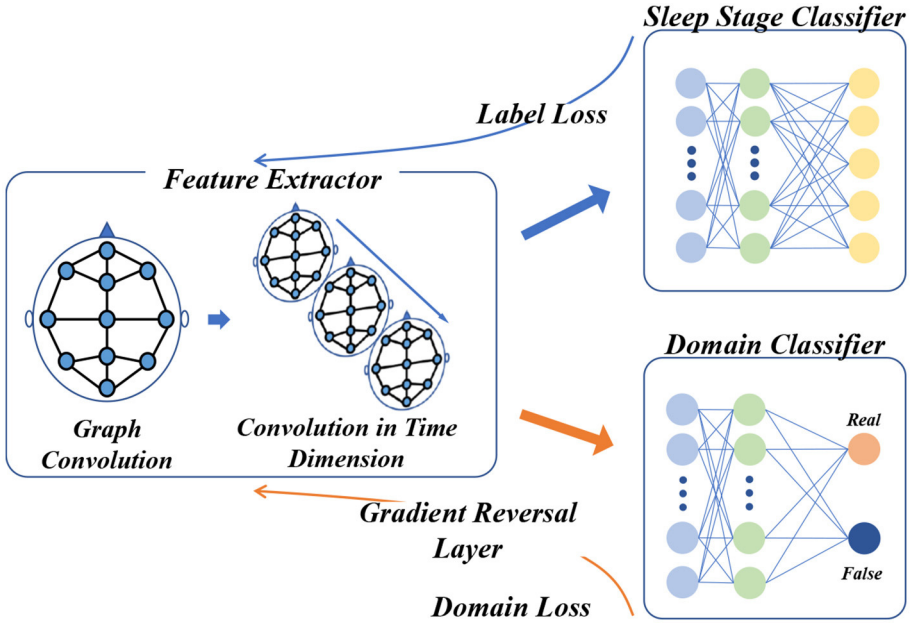


FIGURE 4
Domain adversarial training framework for STDP-GCN. Following the blue arrow, the PSG signal is processed through the feature extractor to acquire the domain-invariant feature, which then proceeds to the sleep stage classifier for classification and computation of the label loss, before undergoing back-propagation. Along the yellow arrow, the domain-invariant feature is fed into the domain discriminator for domain classification and calculation of the domain loss, which then undergoes backpropagation through the Gradient Reverse Layer.

TABLE 1 Experiment hyperparameter setting.

Hyperparameter description	Value
Optimizer	Adam
Learning rate	1e-4
Number of training epochs	500
Batch size	256
Dropout probability	0.5
Weight decay	1e-3
Layer number of GCN	1
The number of temporal contexts M	5
τ values of neurons	100.0
Threshold for spiking	1.0
Learning rate for STDP	1e-2
Domain classifier architecture	450-512-100-2
Initial λ value of Gradient Reversal Layer	1.0

contained in the data set, we apply 10-fold cross-validation on the ISRUC-S3 data set and 20-fold cross-validation on the SLEEP-EDF-153 dataset. The hyperparameters of STDP-GCN are listed in Table 1, and we apply the same experimental settings to all baselines to pursue comparative fairness. This article uses PyTorch to implement the model and training, and the code has been released at: https://github.com/thegoist/STDP-GCN.

4.2. Experimental results and comparison

This section uses STDP-GCN to compare with the other baselines, showing the superiority of the current STDP-GCN. As evident from Tables 2, 3, STDP-GCN outperforms prior methods in multiple metrics. By utilizing the STDP mechanism in constructing its graph structure, STDP-GCN aligns with the principles of neuroscience and effectively leverages the inter-channel dependencies to enhance the extraction of spatial features, resulting in better performance across various indicators. It can be observed from the table that the traditional machine learning algorithm SVM and RF is less accurate than other methods because it cannot learn temporal transition rules, while CNN and RNN can rely on learning transition rules in the time dimension and learning features in the spatial dimension to achieve higher accuracy. The channels in the PSG signal are not separated by Euclidean distances, so using Euclidean distance for convolution may overlook the non-Euclidean distance information between channels. The experimental data demonstrates that Wake and N1 indicators are always mutually exclusive. An increase in the Wake indicator leads to a decrease in the N1 indicator. The reason behind this is that the N1 stage is prone to misclassification as Wake due to the shared characteristics between them. Based on the AASM standard (Berry et al., 2012), both fully awake and drowsiness are included in the Wake stage, and the electrophysiological signals and psychological characteristics of drowsiness even continue to the N1 stage, which could be the main reason for misclassification. In addition, we also explored the effect of different folds on cross-validation, as shown in Table 4. We also applied 5-fold cross-validation on ISRUC-S3, where the performance of 5-fold

TABLE 2 Overall results comparison on ISRUC-S3.

Methods	Overall results		F1-score for each class				
	Accuracy	F1-score	Wake	N1	N2	N3	REM
SVM (Alickovic and Subasi, 2018)	73.3%	72.1%	86.8%	52.3%	69.9%	78.6%	73.1%
RF (Memar and Faradji, 2018)	72.9%	70.8%	85.8%	47.3%	70.4%	80.9%	69.9%
MLP+LSTM (Dong et al., 2018)	77.9%	75.8%	86.0%	46.9%	76.0%	87.5%	82.8%
CNN+BiLSTM (Supratak et al., 2017)	78.8%	77.9%	88.7%	60.2%	74.6%	85.8%	80.2%
CNN (Chambon et al., 2018)	78.1%	76.8%	87.0%	55.0%	76.0%	85.1%	80.9%
ARNN+RNN (Phan et al., 2019)	78.9%	76.3%	83.6%	43.9%	79.3%	87.9%	86.7%
STGCN (Jia et al., 2020)	79.9%	78.7%	87.8%	57.4%	77.6%	86.4%	84.1%
MSTGCN (Jia et al., 2021a)	82.1%	80.8%	**89.4%**	59.6%	80.6%	**89.0%**	85.6%
STDP-GCN	**82.6%**	**81.0** %	83.5%	**62.9%**	**83.1%**	86.0%	**90.6%**

The bold result is the best result.

TABLE 3 Overall results comparison on SLEEP-EDF-153.

Methods	Overall results		F1-score for each class				
	Accuracy	F1-score	Wake	N1	N2	N3	REM
SVM (Alickovic and Subasi, 2018)	71.2%	57.8%	80.3%	13.5%	79.5%	57.1%	58.7%
RF (Memar and Faradji, 2018)	72.7%	62.4%	81.6%	23.2%	80.6%	65.8%	60.8%
CNN+BiLSTM (Supratak et al., 2017)	78.5%	75.3%	91.0%	47.0%	81.0%	69.0%	79.0%
MSTGCN (Jia et al., 2021a)	86.4%	**84.1%**	85.5%	**75.3%**	**89.8%**	80.4%	89.3%
STDP-GCN	**87.4%**	83.2 %	**91.1%**	60.1%	89.1%	**84.6%**	**88.8%**

The bold result is the best result.

TABLE 4 Cross-validation of different fold numbers on the ISRUC-S3 dataset.

	Overall results		F1-score for each class				
	Accuracy	F1-score	Wake	N1	N2	N3	REM
5-folds	80.3%	78.5%	83.6%	58.8%	82.0%	82.0%	84.6%
10-folds	82.6%	81.0%	83.5%	62.9%	83.1%	86.0%	90.6%

cross-validation decreased relative to 10-fold, probably due to the increase in the adversarial sample and the decrease in the test sample.

4.3. Experiments and analysis

To visualize the graph structure generated by the STDP graph learning algorithm, we applied the algorithm to generate the adjacency graph structure of all data in the ISRUC-S3 dataset. By summing up all the adjacency graph matrices adaptive learned through the STDP graph learning algorithm in each state, the brain functional connectivity in each sleep state is shown in Figure 5. The explainability of STDP-GCN can be explored by observing the graph structure generated by the STDP graph learning algorithm. There are numerous functional connections because the brain is more active during the wake period (Larson-Prior et al., 2011). During the NREM stage, the brain gradually enters a deep sleep state and exhibits limited connectivity, typically represented by one or two channels. Conversely, in the REM stage, the functional connections between brain regions are relatively weak (Spoormaker et al., 2010).

In order to verify the effectiveness of the STDP graph learning algorithm, this paper uses different graph construction methods to compare the graph structures, which is shown in Figure 6. The graph structures used for comparison mainly include (1) Fully connected adjacency matrix. Fully connected adjacency matrix means that each brain area has functional connections with the same weight, which is not conducive to extracting the spatial features of the graph. (2) Random matrix, each brain area is randomly connected. (3) Graph learning algorithm, which builds the loss by establishing the feature difference between different channels, and learns through backpropagation. Through the experimental results, it can be seen that the fully connected adjacency matrix has the worst effect, and the graph learning

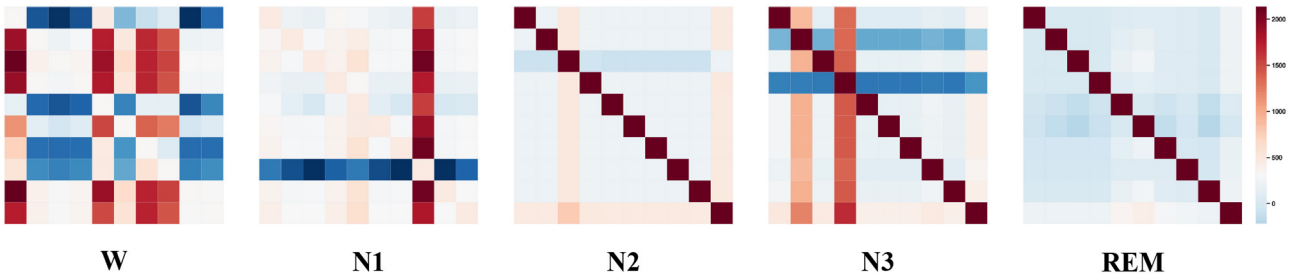


FIGURE 5
Visualization of the adjacency matrix learned by the STDP graph learning algorithm(Wake Stage, N1 Stage, N2 Stage, N3 Stage, and REM Stage).

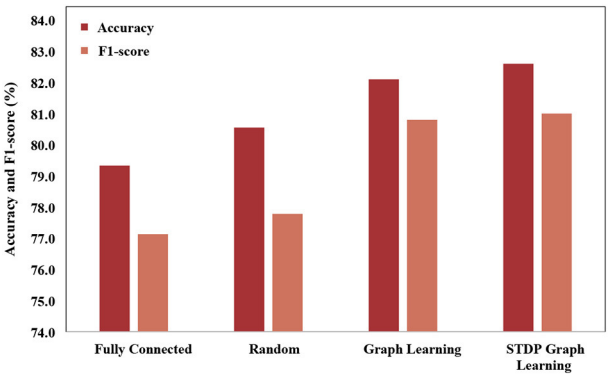


FIGURE 6
Performance comparison of different graphs.

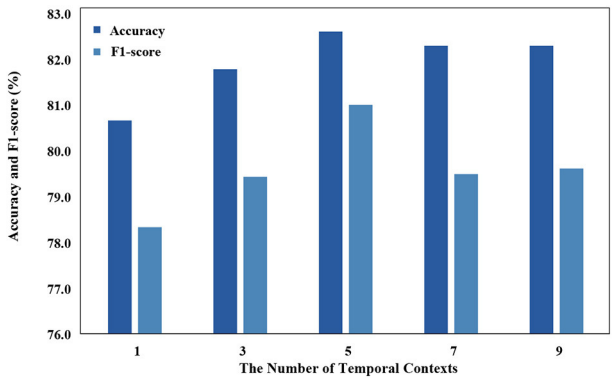


FIGURE 7
Performance comparison of different time contexts.

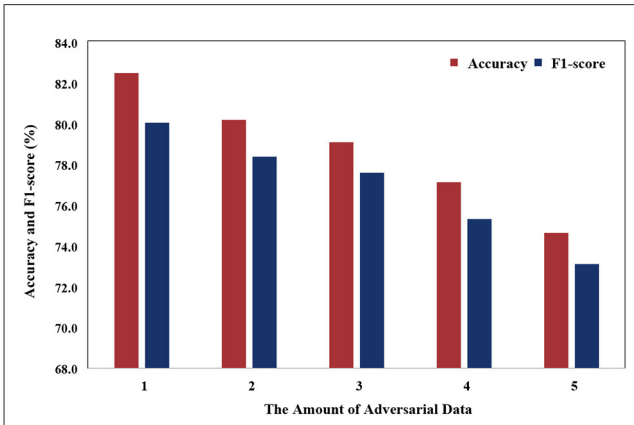


FIGURE 8
Comparing performance with varying numbers of adversarial data.

algorithm has the best effect, while the STDP graph learning algorithm is close to the graph learning algorithm, and is better than the random matrix algorithm, which it is shown that the graph learned by the STDP graph learning algorithm is effectiveness, and it also shows that the relationship between brain regions can be constructed through synaptic plasticity. The reason why the STDP graph learning algorithm is slightly lower than the graph learning algorithm may be that the STDP algorithm only pays attention to the changes in the synaptic strength caused by the impulse signal between neurons, and the connections between other brain areas are not fully utilized, such as adjacent brain areas. There should also be some connection between the zones.

Temporal context is used as an input that has a significant impact on the model, and we use different temporal contexts to test their impact on performance. As demonstrated in Figure 7, the classification performance of STDP-GCN on the ISRUC-S3 dataset varies with the number of input contexts M. With insufficient input contexts, the model will struggle to learn the temporal transition rules, while an excessive number of contexts will make it challenging for the model to accurately comprehend the temporal transition rules. Optimal performance has been observed when the number of input contexts $M = 5$.

Figure 8 illustrates that the model's performance gradually decreases as the number of adversarial data increases. This phenomenon can be attributed to the variance in data distribution of PSG, which is influenced by individual differences. As the number of adversarial samples increases, the number of non-adversarial samples decreases, causing the model to face challenges in learning common features.

We plot the training loss curves for subject-independent cross-validation of the ISRUC-S3 dataset. As shown in Figure 9, the loss of the domain classifier decreases and converges as

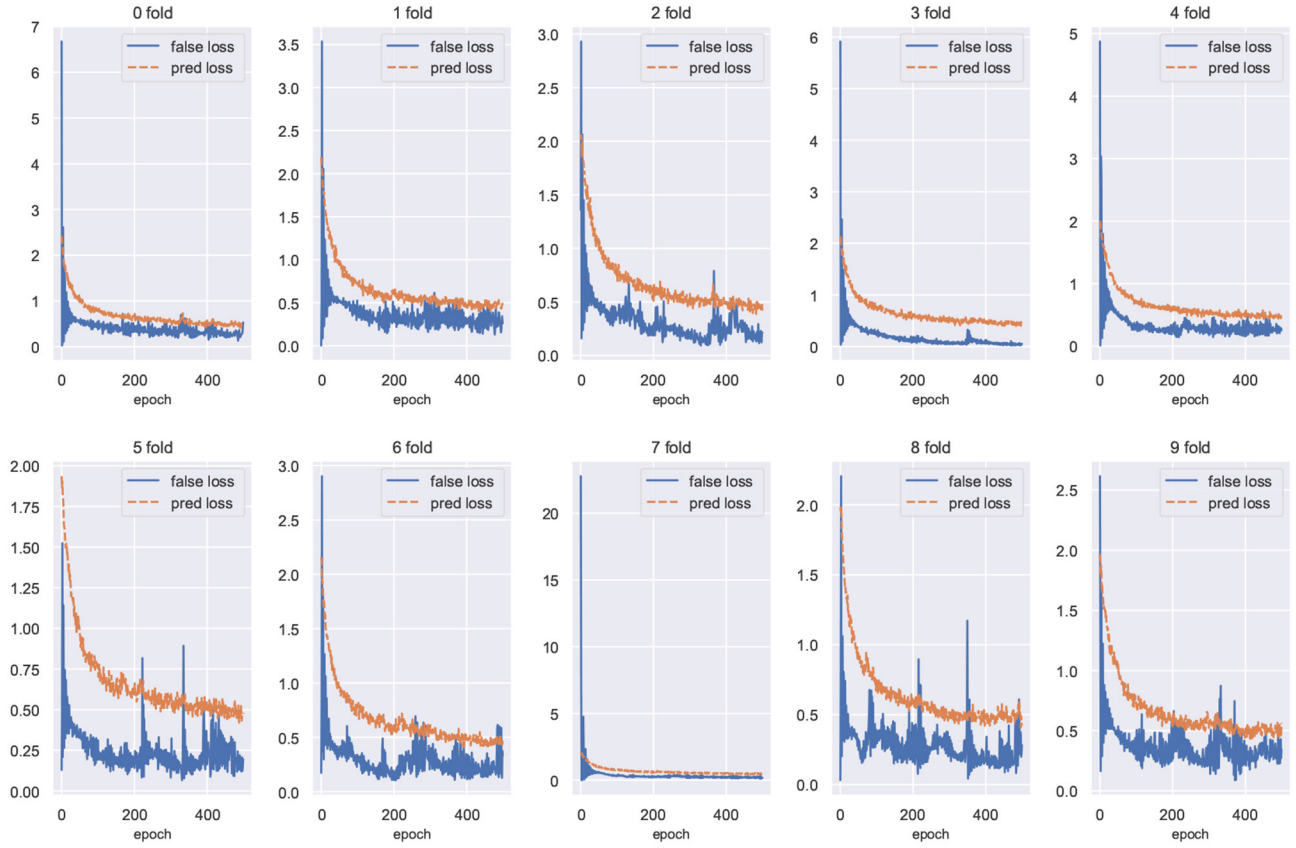


FIGURE 9
Training loss curves for cross-validation of the ISRUC-S3 dataset. The loss value of the domain discriminator is represented by the blue curve, and the loss of the domain classifier is represented by the red curve.

the epoch increases, while the loss of the domain discriminator oscillates but eventually decreases and converges, suggesting that adversarial training is helping the model learn invariant features between domains.

5. Discussion

In this paper, we propose STDP-GCN for automatic sleep staging. The main advantage of STDP-GCN is to compute the interdependencies between nodes using the STDP algorithm with neuroscience mechanism, and then construct the graph structure between nodes, so STDP-GCN makes full use of the interdependencies between nodes through GCN to extract features. The STDP graph learning algorithm does not require backpropagation and labeling, it only needs to encode the PSG signal as a pulse sequence to calculate the graph structure of the PSG channel, which not only has a neuroscience mechanism but also has a good performance. As shown in Figure 6, when compared with other graph structure construction algorithms, the STDP graph learning algorithm had the highest accuracy metrics on both the ISRUC-S3 dataset and the SLEEP-EDF-153 dataset, and most of the remaining evaluation metrics outperformed existing methods. In automated sleep staging, individual differences in physiological signals often result in models that perform well in training and poorly in testing. This problem can be effectively addressed by using adversarial training. Figure 9 shows the loss curves of the domain classifier and the domain discriminator during adversarial training on the ISRUC-S3 dataset, from which it can be seen that the loss curve of the domain discriminator decreases in oscillation. This phenomenon indicates that the domain discriminator acts as an adversarial training operation, and in addition, the performance metrics of the model training also prove the effectiveness of adversarial training.

STDP-GCN also comes with some disadvantages. Firstly, the STDP algorithm requires time steps for simulation, and even with the modified STDP algorithm and GPU parallel computing, the STDP graph learning algorithm is still slower than the rest of the graph structure algorithms. Second, STDP-GCN sometimes misclassifies Wake and N1 because both Wake and N1 have similar features. This still indicates that STDP-GCN needs to strengthen its feature learning capability.

6. Conclusion

Inspired by Spike-Timing-Dependent Plasticity, this paper proposes an adaptive graph convolution network (GCN) for

automatic sleep staging, named STDP-GCN. The key advantage of STDP-GCN is its ability to establish connections between brain regions through the synaptic weight adjustment mechanism among neurons. This algorithm dynamically establishes inter-channel dependencies without any labeling and exhibits exceptional performance. Comparative experiments show that the performance of STDP-GCN is comparable to the leading models in the field.

Data availability statement

Publicly available datasets were analyzed in this study. This data can be found here: https://sleeptight.isr.uc.pt/; https://physionet.org/content/sleep-edfx/1.0.0.

Author contributions

YZ wrote the paper and performed the experiment. XL guided the experiment design and reviewed the manuscript. ZZ contributed significantly to the experiment. XW discussed about the results and analysis. XH performed the analysis. LY helped perform the analysis with constructive discussions. All authors helped with developing the concepts and writing the paper. All authors contributed to the article and approved the submitted version.

Funding

This research was supported by the National Natural Science Foundation of China (Grant no. 62266040), the Key Research and Development Project of Gansu Province (Grant no. 20YF8GA049), the Industrial Support Plan Project for Colleges and Universities in Gansu Province (Grant no. 2022CYZC-13), and the Lanzhou Municipal Science and Technology Project (Grant no. 2019-1-34).

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

Alickovic, E., and Subasi, A. (2018). Ensemble svm method for automatic sleep stage classification. *IEEE Trans. Instrum. Meas.* 67, 1258–1265. doi: 10.1109/TIM.2018.2799059

Bakker, J. P., Ross, M., Cerny, A., Vasko, R., Shaw, E., Kuna, S., et al. (2022). Scoring sleep with artificial intelligence enables quantification of sleep stage ambiguity: hypnodensity based on multiple expert scorers and auto-scoring. *Sleep*. 46, zsac154. doi: 10.1093/sleep/zsac154

Berry, R. B., Budhiraja, R., Gottlieb, D. J., Gozal, D., Iber, C., Kapur, V. K., et al. (2012). Rules for scoring respiratory events in sleep: update of the 2007 aasm manual for the scoring of sleep and associated events: deliberations of the sleep apnea definitions task force of the american academy of sleep medicine. *J. Clinical Sleep Med.* 8, 597–619. doi: 10.5664/jcsm.2172

Bi, G., and Poo, 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

Cai, Q., Gao, Z., An, J., Gao, S., and Grebogi, C. (2021). A graph-temporal fused dual-input convolutional neural network for detecting sleep stages from EEG signals. *IEEE Trans. Circuits Syst. II Express Briefs*. 68, 777–781. doi: 10.1109/TCSII.2020.3014514

Chambon, S., Galtier, M. N., Arnal, P. J., Wainrib, G., and Gramfort, A. (2018). A deep learning architecture for temporal sleep stage classification using multivariate and multimodal time series. *IEEE Trans. Neural Syst. Rehabilitation Eng.* 26, 758–769. doi: 10.1109/TNSRE.2018.2813138

Dan, Y., and Poo, M. M. (1992). Hebbian depression of isolated neuromuscular synapses in vitro. *Science*. 256, 1570–1573. doi: 10.1126/science.1317971

Dong, H., Supratak, A., Pan, W., Wu, C., Matthews, P. M., and Guo, Y. (2018). Mixed neural network approach for temporal sleep stage classification. *IEEE Trans. Neural Syst. Rehabilitation Eng.* 26, 324–333. doi: 10.1109/TNSRE.2017.2733220

Fornito, A., Zalesky, A., and Breakspear, M. (2015). The connectomics of brain disorders. *Nat. Rev. Neurosci.* 16, 159–172. doi: 10.1038/nrn3901

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., et al. (2016). Domain-adversarial training of neural networks. in *Domain Adaptation in Computer Vision Applications. Advances in Computer Vision and Pattern Recognition,* eds G. Csurka (Springer, Cham). doi: 10.1007/978-3-319-58347-1_10

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., et al. (2014). "Generative adversarial nets," in *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2, NIPS'14.* p. 2672–2680, Cambridge, MA, USA: MIT Press.

Griffa, A., Baumann, P. S., Thiran, J.-,p., and Hagmann, P. (2013). Structural connectomics in brain diseases. *Neuroimage*. 80, 515–526. doi: 10.1016/j.neuroimage.2013.04.056

Hebb, D. (1949). Organization of behavior. *J. Clin. Psychol.* 6, 335–307. doi: 10.1002/1097-4679(195007)6:3<307::AID-JCLP2270060338>3.0.CO;2-K

Jia, Z., Lin, Y., Wang, J., Ning, X., He, Y., Zhou, R., et al. (2021a). Multi-view spatial-temporal graph convolutional networks with domain generalization for sleep stage classification. *IEEE Trans. Neural Syst. Rehabilitation Eng.* 29, 1977–1986. doi: 10.1109/TNSRE.2021.3110665

Jia, Z., Lin, Y., Wang, J., Wang, X., Xie, P., and Zhang, Y. (2021b). "SalientSleepNet: Multimodal salient wave detection network for sleep staging," in *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*. ed Z.-H. Zhou (International Joint Conferences on Artificial Intelligence Organization), 2614–2620. doi: 10.24963/ijcai.2021/360

Jia, Z., Lin, Y., Wang, J., Zhou, R., Ning, X., He, Y., et al. (2020). GraphSleepNet: Adaptive spatial-temporal graph convolutional networks for sleep stage classification. *IJCAI*. 2021, 1324–1330. doi: 10.24963/ijcai.2020/184

Kipf, T. N. and Welling, M., (2016). "Semi-supervised classification with graph convolutional networks," in *International Conference on Learning Representations (ICLR 2017)*.

Larson-Prior, L. J., Power, J. D., Vincent, J. L., Nolan, T. S., Coalson, R. S., Zempel, J., et al. (2011). Modulation of the brain's functional network architecture in the transition from wake to sleep. *Prog. Brain Res.* 193, 277–294. doi: 10.1016/B978-0-444-53839-0.00018-1

Li, M., Chen, H., and Cheng, Z. (2022). An attention-guided spatiotemporal graph convolutional network for sleep stage classification. *Life*. 12, 5. doi: 10.3390/life12050622

Martín-Montero, A., Armañac-Julián, P., Gil, E., Kheirandish-Gozal, L., Álvarez, D., Lázaro, J., et al. (2023). Pediatric sleep apnea: characterization of apneic events and sleep stages using heart rate

variability. *Comput. Biol. Med.* 154, 106549. doi: 10.1016/j.compbiomed.2023.106549

Medini, C., Zacharia, R. M., Nair, B., Vijayan, A., Rajagopal, L. P., and Diwakar, S. (2015). "Spike encoding for pattern recognition: Comparing cerebellum granular layer encoding and bsa algorithms," in *2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI)* (Kochi: IEEE), 1619–1625. doi: 10.1109/ICACCI.2015.7275845

Memar, P., and Faradji, F. (2018). A novel multi-class EEG-based sleep stage classification system. *IEEE Trans. Neural Syst. Rehabilitation Eng.* 26, 84–95. doi: 10.1109/TNSRE.2017.2776149

Morrison, A., Diesmann, M., and Gerstner, W. (2008). Phenomenological models of synaptic plasticity based on spike timing. *Biol. Cybern.* 98, 459–478. doi: 10.1007/s00422-008-0233-1

Nuntalid, N., Dhoble, K., and Kasabov, N. (2011). "Eeg classification with bsa spike encoding algorithm and evolving probabilistic spiking neural network," in *Neural Information Processing*, Lu, B.-L., Zhang, L., and Kwok, J., (eds). Berlin, Heidelberg:. Springer Berlin Heidelberg. p. 451–460. doi: 10.1007/978-3-642-24955-6_54

Perslev, M., Darkner, S., Kempfner, L., Nikolic, M., Jennum, P. J., and Igel, C. (2021). U-Sleep: resilient high-frequency sleep staging. *NPJ Digital Med.* 4, 1–12. doi: 10.1038/s41746-021-00440-5

Perslev, M., Jensen, M. H., Darkner, S., Jennum, P. J., and Igel, C. (2019). U-Time: A fully convolutional network for time series segmentation applied to sleep staging. *Adv. Neural Infor. Proc. Syst.* 4415–4426.

Petro, B., Kasabov, N., and Kiss, R. M. (2020). Selection and optimization of temporal spike encoding methods for spiking neural networks. *IEEE Trans. Neural Netw. Learn Syst.* 31, 358–370. doi: 10.1109/TNNLS.2019.2906158

Phan, H., Andreotti, F., Cooray, N., Chen, O. Y., and De Vos, M. (2019). SeqSleepNet: end-to-end hierarchical recurrent neural network for sequence-to-sequence automatic sleep staging. *IEEE Trans. Neural Syst. Rehabilitation Eng.* 27, 400–410. doi: 10.1109/TNSRE.2019.2896659

Schrauwen, B., and Van Campenhout, I. (2003). "BSA, a fast and accurate spike train encoding scheme," in *Proceedings of the International Joint Conference on Neural Networks*. Portland, Oregon USA: IEEE. p. 2825–2830. doi: 10.1109/IJCNN.2003.1224019

Spoormaker, V. I., Schroter, M. S., Gleiser, P. M., Andrade, K. C., Dresler, M., Wehrle, R., et al. (2010). Development of a large-scale functional brain network during human non-rapid eye movement sleep. *J. Neurosci.* 30, 11379–11387. doi: 10.1523/JNEUROSCI.2015-10.2010

Supratak, A., Dong, H., Wu, C., and Guo, Y. (2017). DeepSleepNet: a model for automatic sleep stage scoring based on raw single-channel EEG. *IEEE Trans. Neural Syst. Rehabilitation Eng.* 25, 1998–2008. doi: 10.1109/TNSRE.2017.2721116

Taylor, M. M. (1973). The problem of stimulus structure in the behavioural theory of perception. *South African J. Psychol.* 3:23–45.

Tzeng, E., Hoffman, J., Zhang, N., Saenko, K., and Darrell, T. (2014). Deep domain confusion: Maximizing for domain invariance. *arXiv [Preprint]*. arXiv:1412.3474.

Wolpert, E. A. (1969). A manual of standardized terminology, techniques and scoring system for sleep stages of human subjects. *Arch. Gen. Psychiatr.* 20, 246–247.

Zhang, J., and Wu, Y. (2017). A new method for automatic sleep stage classification. *IEEE.* 11, 1097–1110. doi: 10.1109/TBCAS.2017.2719631

Zhang, Y., Cao, W., Feng, L., Wang, M., Geng, T., Zhou, J., et al. (2023). Shnn: a single-channel eeg sleep staging model based on semi-supervised learning. *Expert Syst. Appl.* 213, 119288. doi: 10.1016/j.eswa.2022.119288

frontiers | Frontiers in Neuroscience

TYPE Original Research
PUBLISHED 05 June 2023
DOI 10.3389/fnins.2023.1203823

Check for updates

OPEN ACCESS

EDITED BY
Alois C. Knoll,
Technical University of Munich, Germany

REVIEWED BY
Yakang Dai,
Chinese Academy of Sciences (CAS), China
Anita Sebasthiyar,
St Anne's College of Engineering and Technology, India
Zhong Xue,
United Imaging Intelligence, China

*CORRESPONDENCE
Shiyao Chen
✉ chen.shiyao@zs-hospital.sh.cn
Yonghong Shi
✉ yonghong.shi@fudan.edu.cn

†These authors have contributed equally to this work

RECEIVED 11 April 2023
ACCEPTED 12 May 2023
PUBLISHED 05 June 2023

CITATION
Song G, Zhou J, Wang K, Yao D, Chen S and Shi Y (2023) Segmentation of multi-regional skeletal muscle in abdominal CT image for cirrhotic sarcopenia diagnosis.
Front. Neurosci. 17:1203823.
doi: 10.3389/fnins.2023.1203823

COPYRIGHT
© 2023 Song, Zhou, Wang, Yao, Chen and Shi. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

Segmentation of multi-regional skeletal muscle in abdominal CT image for cirrhotic sarcopenia diagnosis

Genshen Song[1,2†], Ji Zhou[3†], Kang Wang[1,2], Demin Yao[1,2], Shiyao Chen[3*] and Yonghong Shi[1,2,4*]

[1]Digital Medical Research Center, School of Basic Medical Sciences, Fudan University, Shanghai, China, [2]Shanghai Key Laboratory of Medical Imaging Computing and Computer Assisted Intervention, Shanghai, China, [3]Department of Gastroenterology and Hepatology, Zhongshan Hospital, Fudan University, Shanghai, China, [4]Academy for Engineering & Technology, Fudan University, Shanghai, China

Background: Sarcopenia is generally diagnosed by the total area of skeletal muscle in the CT axial slice located in the third lumbar (L3) vertebra. However, patients with severe liver cirrhosis cannot accurately obtain the corresponding total skeletal muscle because their abdominal muscles are squeezed, which affects the diagnosis of sarcopenia.

Purpose: This study proposes a novel lumbar skeletal muscle network to automatically segment multi-regional skeletal muscle from CT images, and explores the relationship between cirrhotic sarcopenia and each skeletal muscle region.

Methods: This study utilizes the skeletal muscle characteristics of different spatial regions to improve the 2.5D U-Net enhanced by residual structure. Specifically, a 3D texture attention enhancement block is proposed to tackle the issue of blurred edges with similar intensities and poor segmentation between different skeletal muscle regions, which contains skeletal muscle shape and muscle fibre texture to spatially constrain the integrity of skeletal muscle region and alleviate the difficulty of identifying muscle boundaries in axial slices. Subsequentially, a 3D encoding branch is constructed in conjunction with a 2.5D U-Net, which segments the lumbar skeletal muscle in multiple L3-related axial CT slices into four regions. Furthermore, the diagnostic cut-off values of the L3 skeletal muscle index (L3SMI) are investigated for identifying cirrhotic sarcopenia in four muscle regions segmented from CT images of 98 patients with liver cirrhosis.

Results: Our method is evaluated on 317 CT images using the five-fold cross-validation method. For the four skeletal muscle regions segmented in the images from the independent test set, the avg. DSC is 0.937 and the avg. surface distance is 0.558 mm. For sarcopenia diagnosis in 98 patients with liver cirrhosis, the cut-off values of Rectus Abdominis, Right Psoas, Left Psoas, and Paravertebral are 16.67, 4.14, 3.76, and 13.20 cm^2/m^2 in females, and 22.51, 5.84, 6.10, and 17.28 cm^2/m^2 in males, respectively.

Conclusion: The proposed method can segment four skeletal muscle regions related to the L3 vertebra with high accuracy. Furthermore, the analysis shows that the Rectus Abdominis region can be used to assist in the diagnosis of sarcopenia when the total muscle is not available.

KEYWORDS

cirrhotic sarcopenia, skeletal muscle segmentation, rectus abdominis, texture attention enhancement block, skeletal muscle index

1. Introduction

Sarcopenia is a pathological decrease in skeletal muscle, including primary sarcopenia and secondary sarcopenia. Primary sarcopenia is the aging and atrophy of skeletal muscle with age, which is related to the aging process of humans. And secondary sarcopenia is caused by poor dietary intake, malnutrition and chronic diseases such as cirrhosis of the liver (Bauer et al., 2019). Sarcopenia is a common complication in patients with liver cirrhosis, characterized by the loss of muscle strength and mass. According to statistics (Xiao et al., 2019), as many as 7 million people in China suffer from cirrhosis, accounting for 0.5% of the total population. The prevalence of sarcopenia in cirrhotic patients is between 40% and 70% due to metabolic abnormalities resulting from decreased liver function (Cao et al., 2017). Study (Tantai et al., 2022) shows that cirrhotic sarcopenia increases the risk of falls, fractures, decreased quality of life, or acute-on-chronic liver failure in patients with cirrhosis. Sarcopenia is significantly associated with morbidity and mortality in cirrhotic patients (Hanai et al., 2015) and is an independent predictor of survival in patients with cirrhosis (Kim et al., 2017). Therefore, early and accurate diagnosis of sarcopenia is helpful for the clinical treatment and management of liver cirrhosis patients.

Sarcopenia is generally diagnosed by the third lumbar skeletal muscle index (L3SMI). L3SMI is defined by measuring the skeletal muscle area in the axial CT slice of the third lumbar (L3) vertebra, and then calculating the ratio of cross-sectional muscle area to the square of body height. For diagnosing patients with cirrhotic sarcopenia, the L3SMI's cut-off values are 50 cm^2/m^2 in males and 39 cm^2/m^2 in females (Carey et al., 2017). However, in some diseases, it would not be enough to only measure these muscles. For example, parts of the abdominal muscles of patients with severe ascites may be severely squeezed; or the progression of myosteatosis varies in different muscle regions in nonalcoholic fatty liver disease. A recent study also explored the sarcopenia defined by different muscle groups such as total skeletal muscle, psoas major muscle, and rectus abdominis muscle as a prognostic factor for patients with advanced hepatocellular carcinoma (Wu et al., 2021). This shows that in the diagnosis of cirrhotic sarcopenia, considering the effect of disease on muscle in different regions, partitioning skeletal muscle regions and analyzing each muscle region separately may be a useful supplement to the analysis of the total skeletal muscle.

Therefore, this paper will study the multi-regional skeletal muscles from multiple L3-related CT slices. As shown in Figure 1, red, yellow, green, and blue represent the labels of Rectus Abdominis (the rectus abdominis, external oblique abdominis, internal oblique abdominis, and transversus abdominis at the anterior periphery of L3), Paravertebral (the paravertebral muscle groups such as the erector spine at the posterior part of L3), Right Psoas and Left Psoas (the psoas major, psoas minor, and psoas square on the right and left sides of L3) respectively. Once these skeletal muscle regions are segmented from the L3-related axial CT slices, they can efficiently assist in the diagnosis of sarcopenia.

However, there are various challenges in segmenting multiple skeletal muscle regions in abdominal or abdominopelvic CT images. As shown in Figure 1, there are obvious differences in the shape and size of different skeletal muscles; the boundaries between different skeletal muscle regions or between skeletal muscle and surrounding tissue are unclear or rough, such as the edges of the Right Psoas and Left Psoas in Figure 1; morphological differences of the same skeletal muscle region between different individuals affect segmentation; physiopathological conditions such as muscle fatty degeneration and muscle-reducing obesity affect muscle morphology and signal intensity in CT images; artifacts in CT images increase the difficulty of segmentation.

Deep Convolution Neural Network (CNN) (LeCun et al., 1998) is an effective model for muscle region segmentation in abdominal CT images, including Fully Convolutional Network (FCN) (Long et al., 2015) architecture and encoder–decoder-based models such as 2D U-Net (Ronneberger et al., 2015), 3D U-Net (Çiçek et al., 2016), and Swin-unet (Cao et al., 2023). For example, Dabiri et al. (2019) used FCN and 2D U-Net to segment skeletal muscles in L3- or L4-related CT slices for body composition analysis. Castiglione et al. (2021) and Dabiri et al. (2020) firstly automatically located the axial slice at the L3 centroid from a whole-body or partial-body CT image, and then used 2D U-Net–based models to segment body components, such as skeletal muscle. Park et al. (2020) developed and validated an FCN-based system to analyze skeletal muscles in the axial CT images at the inferior endplate of the L3. Blanc-Durand et al. (2020) used CNN to predict the muscle surface from the axial CT slices related to L3. And Weston et al. (2020) used U-Net variant architecture to segment muscles and other tissues in the abdominopelvic CT images. However, these methods only considered the total skeletal muscle segmentation but did not pay attention to different muscle region segmentation. The relationship between the total skeletal muscle and the diagnosis of sarcopenia can be obtained, but the diagnostic effectiveness of muscles in each region cannot be analyzed.

Recent studies have gradually focused on the segmentation of multiple skeletal muscle regions. Burns et al. (2020) used 2D U-Net-based model to automatically segment multiple muscle groups in the L3- and L4-related axial CT slices to detect central sarcopenia. Huang et al. (2020) used BS-ESNet to automatically segment paravertebral muscles in axial MRI slices at different spine levels. Barnard et al. (2019) used 2D U-Net based model to automatically segment the left paraspinal muscle in the axial CT slice at the twelfth thoracic vertebra. Although these methods focused on muscle segmentation in different regions, they did not pay attention to the multi-regional analysis in multiple axial CT slices related to L3. And they did not explore the relationship between cirrhotic sarcopenia and each skeletal muscle region.

Therefore, the study presents the method to accurately segment multiple skeletal muscle regions in the axial slices associated with the L3 vertebra, and then calculate the clinical indices and use them for the diagnosis of sarcopenia. L3SMI can usually be calculated from muscle

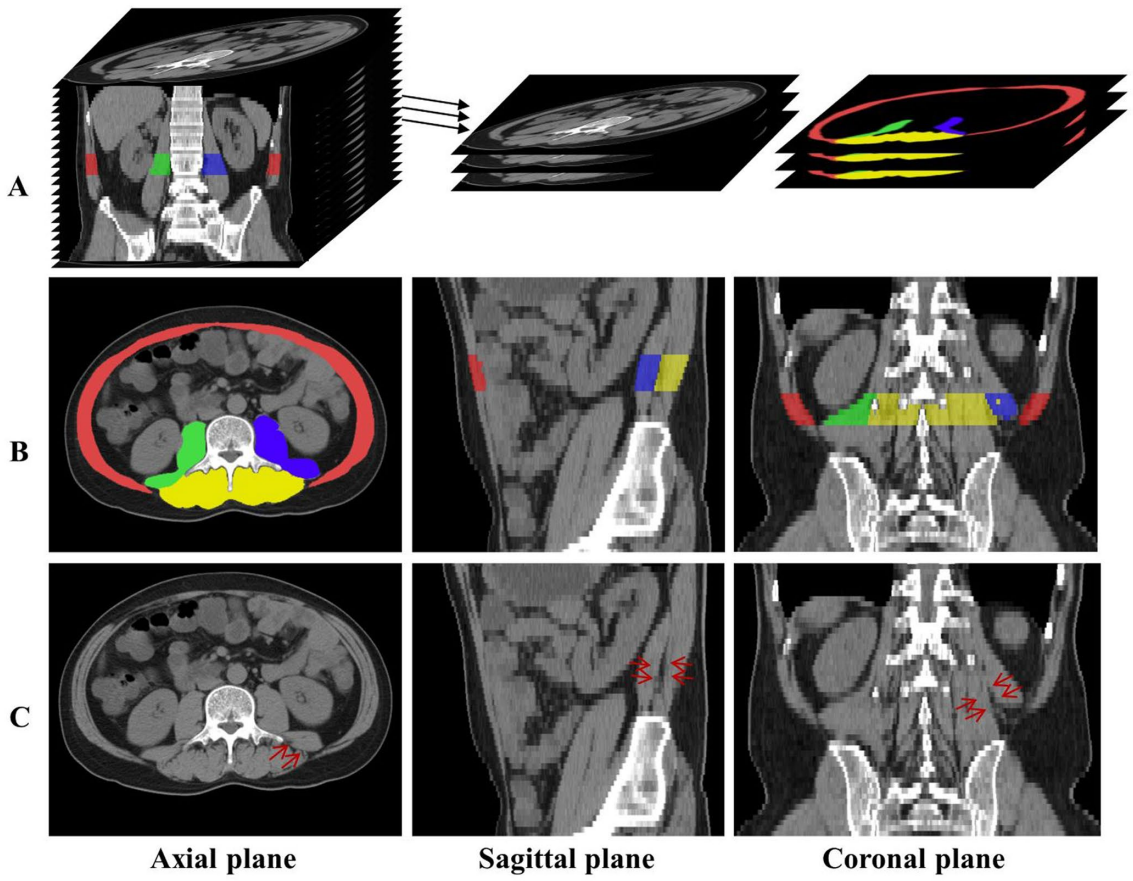


FIGURE 1
(A) The axial CT slices related to L3 are labeled and extracted. **(B)** The distribution of the four skeletal muscle regions is displayed in axial, sagittal, and coronal planes, and red, green, blue, and yellow represent the labels of Rectus Abdominis, Right Psoas, Left Psoas, and Paravertebral, respectively. **(C)** The red arrows indicate the skeletal muscle in the same location, and the skeletal muscles indistinguishable in the axial plane have distinct distinguishing features in the sagittal and coronal planes.

regions segmented in two consecutive axial slices associated with the L3 vertebra, i.e., L3 middle and its adjacent lower slices (Wang et al., 2020), or one axial slice, i.e., L3 upper (Carey et al., 2017) or end slice (Li et al., 2020). However, recent studies demonstrated that the average difference of the skeletal muscle volume measurement was significantly lower than that of the corresponding region in a single CT slice by segmenting the entire abdominopelvic skeletal muscle (Borrelli et al., 2021). Inspired by this, the study uses the average cross-sectional area of the total skeletal muscle volume corresponding to the L3 vertebra to calculate a more reasonable skeletal muscle index. Furthermore, the relationship between each regional skeletal muscle and L3SMI is also investigated for sarcopenia diagnosis.

2. Materials and methods

2.1. Data description

This study used abdominal or abdominopelvic CT images of 317 patients from Zhongshan Hospital affiliated to Fudan University in Shanghai, China, including 216 cirrhotic patients and 101 non-cirrhotic patients. And height and gender of 98 patients in the cirrhosis group were also collected to analyze the relationship between sarcopenia and each skeletal muscle region. According to the diagnostic criteria of cirrhotic sarcopenia (Carey et al., 2017), there were 54 patients with sarcopenia, 43 of which were male and 11 females, and 44 patients with non-sarcopenia, 21 of which were male and 23 females. The mean age of the patients was 57 years old.

The imaging parameters for abdominal or abdominopelvic CT scans are as follows: the in-plane spacing is between 0.562 mm × 0.562 mm and 0.888 mm × 0.888 mm; the slice thickness is 5.0 mm; the image acquisition matrix is 512 × 512; and the number of L3 related axial slices are between 4 and 8.

Experienced clinicians manually labeled the skeletal muscle regions in all L3-related axial CT slices. According to muscle type and distribution, four skeletal muscle regions in the axial, sagittal, and coronal planes are obtained and shown in rows A and B of Figure 1. Here, red, green, blue, and yellow represent the label of Rectus Abdominis, Right Psoas, Left Psoas, and Paravertebral, respectively.

2.2. L3 localization and image preprocessing

Abdominal or abdominopelvic CT images contain many abdominal and lumbar regions, so it is necessary to accurately locate

the L3 vertebra. This can be achieved by our developed method of automatic localization and identification of vertebra in spine CT images (Qin et al., 2021), which is further checked and confirmed by the clinician. Once the L3 was successfully detected, all axial slices related to L3 can be extracted, totaling about 4 to 8 slices, as shown in row A of Figure 1.

For all L3-related axial slices extracted from each CT image, if the number of the slices was less than 8, zero-padding was performed along the axial direction, so that the number of L3-related axial slices of all CT images was equal. Finally, the image block composed of L3-related slices was represented by a tensor with size 8 × 512 × 512 (depth × height × width), which was convenient for inputting the network and extracting the axial space feature. The extracted slices were processed by intensity normalization. Considering the fact that the minimum and maximum Hounsfield Unit (HU) values are varied among all CT images, in order to obtain better image contrast, the full range of HU values of each image was mapped to [0, 1].

2.3. Skeletal muscle segmentation network

Figure 2 depicts the lumbar skeletal muscle segmentation network (LSMU-Net for short). The input of the network is the multiple L3-related axial slices of the abdominal CT images, and the outputs are the labels of the four skeletal muscle regions. The network mainly consists of two hybrid architectures, i.e., a 2.5D encoding–decoding network improved by residual structure, and a 3D encoding branch that enhances the spatial texture information. Specifically, the dedicated texture attention enhancement block is utilized to discern the blurred skeletal muscle boundaries in the 2D axial image shown in row C of Figure 1 from the 3D image space. The details are described as following.

2.3.1. 2.5D encoding-decoding network

The 2.5D network, which is composed of the encoding and decoding branches connected by skip channel connections, is used to implement segmentation in the axial CT slice image. Here, although the 2.5D branch uses 2D convolution kernel, the input of the network consists of CT volumes with multiple slices. In particularly, to adapt to the 2.5D network, slices of the input volume are squeezed into a batch, so that a volume represented by a tensor with size 1 × 1 × 8 × 512 × 512 (batch × channel × depth × height × width) is squeezed and processed by a dimensional permutation to the size of 8 × 1 × 512 × 512 (batch × channel × height × width). Here, 8 originally denotes the depth dimension of the volume and then the number of image batch. The batch of the permutated slices is fed into the 2.5D encoding branch containing 5 successive Block2D modules with 4 stages of 2× max-pooling layer, and then goes through the decoding branch with 4 stages of 2× Upsample and Block2D module to obtain the hierarchical feature map at each stage. The features of the corresponding layer are concatenated in the channel dimension. In the last layer of the decoding branch, the feature map is restored to the same size as the input image and fused with the output features of both the 3D encoding branch and the channel connection in the channel direction, then the feature map represented by a tensor with size 8 × 152 × 512 × 512 is obtained. Finally, a 1 × 1 convolution layer is deployed to obtain the prediction maps of 5 categories represented by a tensor with size 8 × 5 × 512 × 512 as the final outputs (4 regions of skeletal muscles and background).

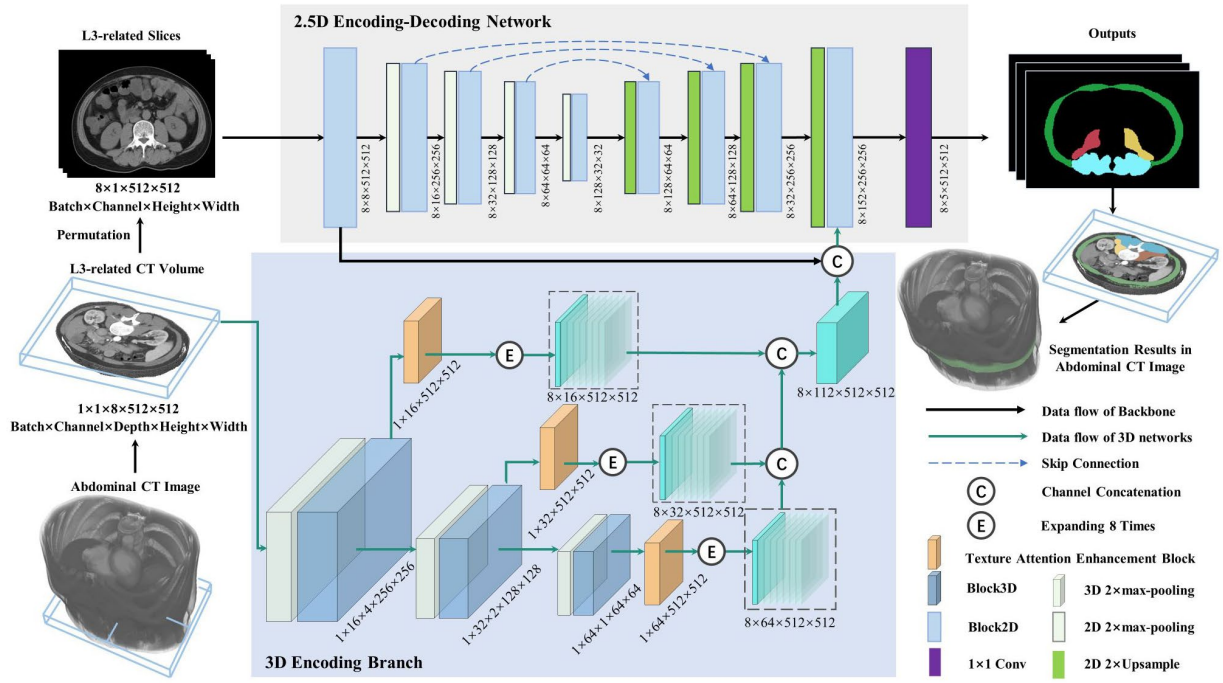


FIGURE 2
Illustration of the lumbar skeletal muscle segmentation network (LSMU-Net). LSMU-Net containing two parts, i.e., 2.5D Encoding-Decoding Network and 3D Encoding Branch. The dimension representation pattern is batch × channel × height × width in the 2.5D network, whereas batch × channel × depth × height × width in the 3D branch.

2.3.2. 3D encoding branch

This is the contextual feature extraction network of the volumetric region composed of multiple L3-related axial slices. The network consists of 3 layers. Each layer of the 3D encoding branch is composed of the max-pooling layer, a Block3D module, and a texture attention enhancement block. The output feature map of the Block3D is halved by down-sampling using a max-pooling operation. The obtained feature maps are transferred to the next layer, simultaneously enhanced by the texture attention enhancement block, then restored to the original image size, and finally expanded 8 times by duplication operation for connecting with the output feature of the 2D decoder branch in the channel dimension. In the study, the input is an L3-related volumetric image represented by a tensor with size $1 \times 1 \times 8 \times 512 \times 512$. After multiple layers of extracted features are concatenated to form 3D hierarchical features, the tensor size is $8 \times 112 \times 512 \times 512$ (batch × channel × height × width). Furthermore, the channel connection operation is performed with the feature of the last layer of the 2.5D network.

Block2D and Block3D: as the basic structure of the LSMU-Net, Block2D and Block3D take the residual structure of 2D ResU-Net (He et al., 2016) as a reference, but they also have differences. First, the convolutional layers are cascaded with InstanceNorm and LeakyReLU to form the basic blocks; subsequently, three groups of basic blocks are cascaded and jump-connected to form Block2D or Block3D with residual structures, respectively, as shown in Figure 3. Block2D has a 3×3 convolutional kernel and Block3D has a $3 \times 3 \times 3$ kernel. These two structural blocks do not change the number of channels, but can effectively deepen the model, facilitating finer edge feature extraction and providing better correction for skeletal muscle refinement. The down-sampling process of the 3D branch contains more trainable features, which requires more convolutional layers to extract spatial information. Therefore, the designs of residual connections in Block2D and Block3D are different, with more convolutional layers in Block3D so that spatial information and 3D structural characteristics can be sufficiently propagated and utilized in the whole network.

Here, it should be noted that the study takes the 2.5D network as the backbone structure. The initial reason is that the number of L3 axial CT slices is small, which limits multiple down-sampling of the 3D network. And the studies (Liu et al., 2017; Isensee et al., 2021) shows that conventional 3D segmentation methods may deteriorate the performance in the anisotropic medical image, and anisotropic convolution on specific planes with better resolution and more appearance features may also improve the accuracy (Jia et al., 2022). As shown in row C of Figure 1, the red arrows indicate the same position of the axial, sagittal, and coronal planes, and the skeletal muscles that cannot be distinguished in the axial plane have distinct characteristics in the sagittal and coronal planes. The 3D encoding branch precisely extracts the spatial context information (shown by the red arrows) lost in the 2.5D network, and the fusion of these features enables the 2.5D network to refine the edges of the skeletal muscle region from the shape of the muscle fiber bundle, improving the segmentation performance. And the studies (Meyer et al., 2021) also shows that the ensemble of 2.5D and 3D network does improve the accuracy in 3D medical image segmentation. Finally, the training time is reduced because the number of parameters in the 2.5D network is less than that of the 3D network.

2.3.3. Texture attention enhancement block

To better integrate 3D features and 2.5D features, Zhou et al. (2019) simultaneously selected and trained four adjacent 2D slice

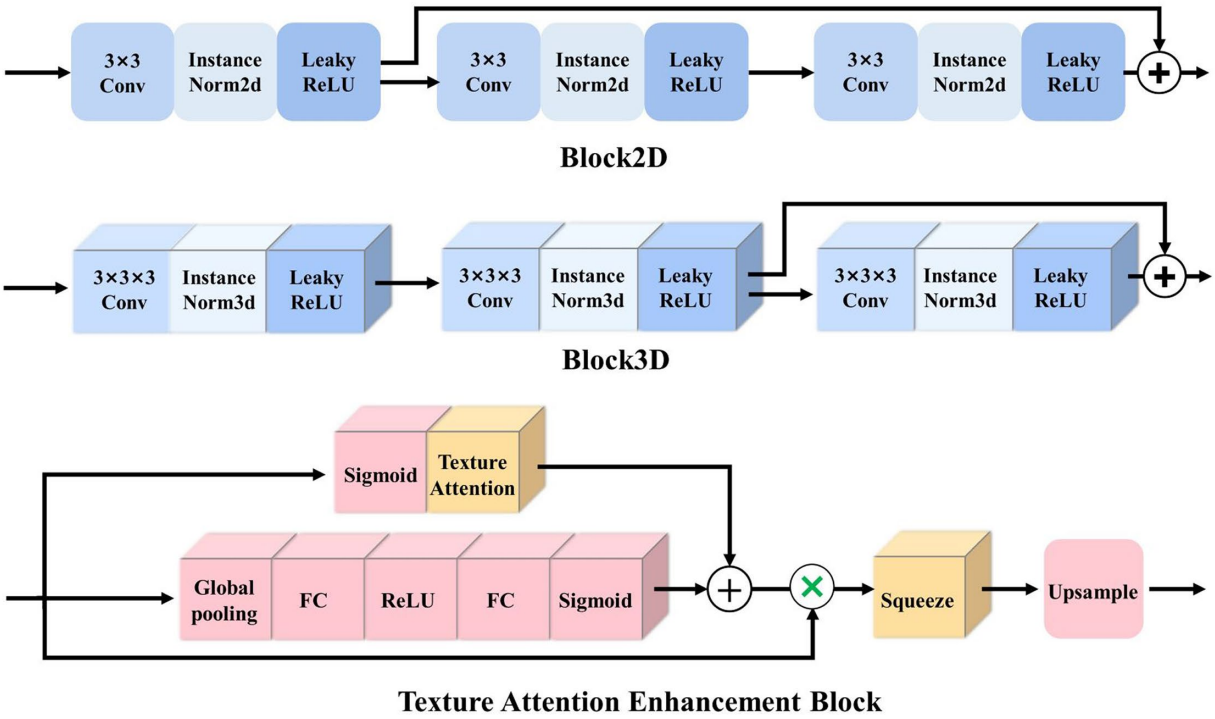


FIGURE 3
Illustration of the basic residual structures of 2.5D and 3D networks, i.e., Block2D and Block3D, as well as Texture Attention Enhancement Block.

images to complement the 3D features, then extracted the features from 2.5D and 3D branches, and fused them after attention enhancement. In this study, based on the standard Squeeze-and-excitation (SE) Block (Hu et al., 2020), a texture attention enhancement block is constructed to compress the channel feature in the 3D network and enhance the blurred edge regions, as shown in Figure 3. The features extracted by Block3D represented by a tensor with size channel × depth × height × width are fed into the attention block. Firstly, the global average pooling is carried out to obtain the feature map represented by a tensor with size channel × 1 × 1 × 1. Then it passes through two layers of a fully connected layer, in which the number of neurons in the first fully connected layer is channel/16 (following SE Block), and the second fully connected layer restores the original number of neurons. This operation increases the nonlinear processing and can fit the complex correlation between channels. Then the probability map is generated through the Sigmoid function. Secondly, the features extracted by Block3D are input into the Texture Attention Block after passing the Sigmoid function, and the pixel-level attention information is obtained through this operation. The proposed Texture Attention Block can increase the range of attention, as shown in Eq. (1).

$$\mathrm{TextureAttention}(x) = x(1-x) \tag{1}$$

where x represents the input probability map. This formula assigns a higher weight to the edge region whose probability is close to 0.5 and a lower weight to the area whose probability is far away from 0.5.

By adding the output feature of Eq. (1), the network no longer only pays attention to the middle part of the skeletal muscle but also enhances the edge refinement based on the shape constraints of the skeletal muscle fibre bundles. The texture attention enhancement block applied in the 3D branch is aim to calculate the weight of the corresponding pixel level and the weight of the channel at the same time, and combine the two. The utility of the texture attention enhancement block is based on the local information of the image, and more attention is paid to the skeletal muscle edge. The part of the skeletal muscle edge is given a high weight value through the pixel-level weight, and the background and the internal area of the skeletal muscle are set a small weight value. Finally, the channels are compressed by the Squeeze and Upsample block to restore the feature map represented by a tensor of original size 512 × 512 in the height and width directions for easy fusion with the 2.5D network.

2.3.4. Loss function

For an input abdominal CT image, four skeletal muscle regions are segmented by combining the multi-class cross-entropy loss function, $\mathrm{Loss_{ce}}$, and the dice loss function, $\mathrm{Loss_{dice}}$. The calculation of these loss function is shown in Eqs. (2)–(4).

$$\mathrm{Loss} = \mathrm{Loss_{ce}} + \mathrm{Loss_{dice}} \tag{2}$$

$$\mathrm{Loss_{ce}} = -\sum_{c=1}^{C}\sum_{i=1}^{H\times W} \omega_c y_i^c \log\left(\frac{e^{\hat{y}_i^c}}{\sum_{j=1}^{C} e^{j}}\right) \tag{3}$$

$$\mathrm{Loss_{dice}} = \sum_{c=1}^{C}\omega_c\left(1-\frac{2\sum_{i=1}^{H\times W} y_i^c \hat{y}_i^c}{\sum_{i=1}^{H\times W}\left(y_i^c\right)^2+\sum_{i=1}^{H\times W}\left(\hat{y}_i^c\right)^2}\right) \tag{4}$$

where $C=5$ denotes the four skeletal muscle regions and the background. ω_c denotes the weight of region c. y_i^c indicates the ground truth value of the i^{th} pixel which belongs to the c^{th} label. $\hat{y}_i^c$ denotes the predicted value of the i^{th} pixel which is predicted as the c^{th} label. H and W denote the height and width of the 2D axial CT image, respectively.

The sizes of the four skeletal muscle regions vary greatly, which means there is a class imbalance problem that may lead to the instability of the segmentation network. Therefore, during the training stage, it is necessary to punish the low confidence (such as Right Psoas and Left Psoas) prediction by setting the weight in the loss function. Specifically, the pixel proportions of the four skeletal muscle regions and background in the training images are counted, and then the regions with smaller proportions are set with large weights, and the regions with large proportions are set with small weights, as shown in Eq. (5).

$$\omega_c = \begin{cases} 1-\dfrac{N_c}{H\times W\times D}-0.2, & \text{if } c=1 \text{ or } c=4 \\ 1-\dfrac{N_c}{H\times W\times D}, & \text{if } c=2 \text{ or } c=3 \end{cases} \tag{5}$$

where H, W, and D denote the height, width, and depth of the training image and $N_c$denotes the number of pixels counted in the c^{th} label. As a result, the prior statistics of ω_c ensure the class equilibrium optimization of the loss function.

2.4. Training and testing parameter settings

The experiments were conducted on Ubuntu 18.04 operating system and PyTorch framework, configured with Intel® Core™ i5-9600K (3.70 GHz × 6 CPUs), 64 GB RAM and RTX 3090 GPUs. The study was evaluated on the abdominal CT images of 317 patients, including 216 cirrhotic patients and 101 non-cirrhotic patients. Firstly, we randomly divided these data into training group ($n=252$) and independent test group ($n=65$). Cirrhotic images and non-cirrhotic images were evenly distributed in each group. Secondly, on the train group, we used the five-fold cross-validation method to evaluate the proposed algorithm. That is, we randomly divided all the sampled into five groups and used four groups for training and the left-out group for testing in each fold. Cirrhotic images and non-cirrhotic images were evenly distributed in each fold. And the Adam optimizer with a learning rate of 0.001 was used to execute for 30 epochs in each fold, and five models were obtained. Thirdly, the model with the best performance of five models was selected to run on the independent test group for the final inference.

2.5. Evaluation indicators

The evaluation metrics of our segmentation results are based on standard measures calculated from pixel-level confusion matrix,

including Dice similarity coefficient (DSC) (Zou et al., 2004) and Sensitivity calculated from Eqs. (6) and (7), respectively.

$$\mathrm{DSC} = \frac{2TP_c}{2TP_c + FP_c + FN_c} \tag{6}$$

$$\mathrm{Sensitivity} = \frac{TP_c}{TP_c + FN_c} \tag{7}$$

where c denotes a category label. $\mathrm{TP_c}$ and $\mathrm{TN_c}$ denote the numbers of the true positive and the true negative pixels in the c^{th} skeletal muscle region, while $\mathrm{FP_c}$ and $\mathrm{FN_c}$ are the numbers of the false positive and the false negative pixels in that category, respectively.

Average symmetrical surface distance (ASSD) is the average Hausdorff distance between the outer surfaces of the segmentation result and the ground truth, calculated from Eqs. (8) and (9).

$$d\left(v_S, G_c\right) = \min_{v_g \in G_c} \left\| v_S - v_g \right\|_2 \tag{8}$$

$$\mathrm{ASSD} = \frac{1}{N_{G_c} + N_{S_c}} \left(\sum_{S_c} \left(\sum_{v_s=1}^{N_{S_c}} d\left(v_S, G_c\right) + \sum_{v_g=1}^{N_{G_c}} d\left(v_g, S_c\right) \right) \right) \tag{9}$$

where $\mathrm{S_c}$ and $\mathrm{G_c}$ denote the surfaces of the segmentation and the ground truth of class c, respectively. The shortest distance from any voxel $\mathrm{v_s}$ belonging to $\mathrm{S_c}$ to $\mathrm{G_c}$ is calculated in Eq. (8). $||\cdot||_2$ represents the Euclidean Distance. $\mathrm{N_{S_c}}$ and $\mathrm{N_{G_c}}$ represent the number of voxels in the surfaces of the segmentation and the ground truth of class c, respectively. The unit for ASSD is a millimeter.

3. Experimental result

3.1. Ablation comparison experiments

To illustrate the overall structural validity of the proposed network, we reproduced 2D U-Net, 3D U-Net, 2D ResU-Net, and 3D ResU-Net (Lee et al., 2017) for comparison experiments. The normalization function and the activation function were InstanceNorm and LeakyReLU. In addition, since nnU-Net (Isensee et al., 2021) is an out-of-the-box representative of 3D U-Net, and has achieved the excellent results in several medical image segmentation tasks, in order to evaluate the performance of LSMU-Net, we used the latest code (nnU-Net V2, including 2D nnU-Net and 3D nnU-Net) from the official nnU-Net website[1] to segment the same data set following the same cross-validation method. The cirrhosis dataset has previously been used for slice-based segmentation in literature (Liu et al., 2019), but that study only described the segmentation model and lacked a detailed description of the training set, the validation set, cross-validation, and DSC calculation of each dataset, so no comparison was made with it. It is worth noting that, in order to make a fair comparison, all the experiments in the study did not enhance the data, which indicates that the results may have the risk of overfitting.

Table 1 shows the DSC results of different methods in segmenting the four skeletal muscle regions of Rectus Abdominis, Right Psoas, Left Psoas and Paravertebral regions. It can be noted from rows 1–4 of Table 1 that the DSC values of the methods combining with the residual structure, namely 2D and 3D ResU-Net, are generally better than those of the corresponding 2D and 3D U-Net, respectively. Therefore, these structures were used in our network. The ablation experiments in rows 7 to 10 of Table 1 also show that the combination of residual structure in our method did improve the DSC values of all skeletal muscles. Rows 5 and 6 shows that the DSC values of the four skeletal muscle regions segmented by 2D and 3D nnU-Net. In particular, the DSC values by 3D nnU-Net are 0.948, 0.929, 0.922, and 0.954, respectively, and the corresponding values by our LSMU-Net are 0.943, 0.928, 0.922, and 0.957 in row 12. The segmentation performance of 3D nnU-Net is slightly higher than that of LSMU-Net in regions of Rectus Abdominis and Right Psoas. The DSC value of LSMU-Net is higher than that of 3D nnU-Net in Paravertebral. The DSC values of 3D nnU-Net and LSMU-Net are the same in the Left Psoas.

To evaluate the utility of each module, the ablation experiments of LSMU-Net were conducted from different perspectives while keeping the parameter settings unchanged. The DSC results of the four skeletal muscle regions are shown in rows 7 to 11 of Table 1. First, the residual module, 3D encoding branch, and texture attention enhancement block were removed from LSMU-Net, respectively. In row 7, there are no 3D branch nor attention block to enhance spatial information. Although the remaining 2.5D backbone network has the same residual structure as 2D ResU-Net, their convolutional layers are arranged differently because our Block2D has an additional layer of convolutional operation before the residual structure. In row 8, DSC increase in all four skeletal muscle regions by comparing to row 7 with the addition of the 3D branch without attention. As the 3D branch with attention block in the study mainly focuses on edge refinement to obtain more accurate skeletal muscle edges, which is more effective in improving the edges of skeletal muscles with small areas like the Right Psoas and Left Psoas. Thus, DSC could also be improved with the addition of only one 3D branch. In row 9, the network including Block2D and Block3D removes all residual structures compared to LSMU-Net. It can be seen that the decrease of DSC indicates that the residual structures in the network is useful for the segmentation of skeletal muscle. To illustrate the validity of the weights in the loss function, we removed the weights from LSMU-Net in row 10 and found that the DSCs of Rectus Abdominis, Right Psoas, and Left Psoas decrease compared with LSMU-Net. Secondly, row 11 shows the results of LSMU-Net using the SE block, which is slightly worse than the results of LSMU-Net using the texture attention enhancement block in row 12. This indicates that our texture attention enhancement block can improve the segmentation of fuzzy regions and optimize the performance of our method.

Table 2 shows the LSMU-Net ablation comparison experiment on the average index of four skeletal muscle regions in the independent test dataset. The prediction results of 3D nnU-Net are 0.938, 0.942, and 0.578 mm in terms of DSC, Sensitivity, and ASSD (mm), respectively. The corresponding results of LSMU-Net are 0.937, 0.944,

1 https://github.com/MIC-DKFZ/nnUNet

TABLE 1 LSMU-Net ablation comparison experiment shown on DSC in the independent test dataset.

	#	3D	AB	RS	W	Rectus abdominis	Right psoas	Left psoas	Paravertebral
3D U-Net	1					0.924 ± 0.001	0.914 ± 0.002	0.902 ± 0.004	0.946 ± 0.001
2D U-Net	2					0.936 ± 0.002	0.925 ± 0.002	0.915 ± 0.003	0.955 ± 0.000
3D ResU-Net	3			✓		0.925 ± 0.002	0.916 ± 0.003	0.909 ± 0.003	0.952 ± 0.001
2D ResU-Net	4			✓		0.940 ± 0.001	0.926 ± 0.002	0.918 ± 0.003	**0.957 ± 0.001**
3D nnU-Net	5					**0.948 ± 0.001**	**0.929 ± 0.002**	**0.922 ± 0.002**	0.954 ± 0.001
2D nnU-Net	6					0.946 ± 0.001	0.926 ± 0.003	0.916 ± 0.003	0.956 ± 0.000
LSMU-Net based	7			✓	✓	0.940 ± 0.001	0.925 ± 0.002	0.914 ± 0.002	0.954 ± 0.001
	8	✓		✓	✓	0.942 ± 0.001	0.926 ± 0.002	0.920 ± 0.003	**0.957 ± 0.001**
	9	✓	✓		✓	0.931 ± 0.001	0.916 ± 0.001	0.906 ± 0.003	0.950 ± 0.001
	10	✓	✓	✓		0.941 ± 0.001	0.927 ± 0.002	0.918 ± 0.003	**0.957 ± 0.001**
LSMU-Net + SE	11	✓		✓	✓	0.942 ± 0.001	0.926 ± 0.002	0.915 ± 0.003	0.956 ± 0.001
LSMU-Net	12	✓	✓	✓	✓	0.943 ± 0.001	0.928 ± 0.002	**0.922 ± 0.002**	**0.957 ± 0.001**

#, method number; 3D, 3D encoding branch; AB, attention block; RS, residual structure; W, weights. The bold value indicates that the method in row has achieved the best performance.

TABLE 2 LSMU-Net ablation comparison experiment shown on the average index of four skeletal muscle regions in the independent test dataset.

	#	3D	AB	RS	W	DSC	Sensitivity	ASSD (mm)
3D U-Net	1					0.922 ± 0.002	0.918 ± 0.003	1.263 ± 19.103
2D U-Net	2					0.934 ± 0.002	0.941 ± 0.001	0.695 ± 2.261
3D ResU-Net	3			✓		0.925 ± 0.002	0.923 ± 0.003	0.691 ± 2.045
2D ResU-Net	4			✓		0.935 ± 0.002	0.942 ± 0.001	0.641 ± 0.845
3D nnU-Net	5					**0.938 ± 0.001**	0.942 ± 0.002	0.578 ± 1.187
2D nnU-Net	6					0.936 ± 0.002	0.941 ± 0.002	0.814 ± 6.055
LSMU-Net based	7			✓	✓	0.933 ± 0.002	0.937 ± 0.001	0.631 ± 0.785
	8	✓		✓	✓	0.936 ± 0.002	0.942 ± 0.001	0.695 ± 3.711
	9	✓	✓		✓	0.926 ± 0.002	0.922 ± 0.003	1.279 ± 12.220
	10	✓	✓	✓		0.936 ± 0.002	**0.947 ± 0.001**	0.677 ± 1.381
LSMU-Net + SE	11	✓		✓	✓	0.935 ± 0.002	0.939 ± 0.001	0.623 ± 0.881
LSMU-Net	12	✓	✓	✓	✓	0.937 ± 0.002	0.944 ± 0.001	**0.558 ± 0.715**

#, method number; 3D, 3D encoding branch; AB, attention block; RS, residual structure; W, weights. The bold value indicates that the method in row has achieved the best performance.

and 0.558 mm, respectively. The ASSD of LSMU-Net is slightly lower than those of other networks.

3.2. Quantitative segmentation results

Our LSMU-Net method was used to segment the four skeletal muscle regions in the CT image. Table 3 shows the accuracy of the segmentation results for the independent test dataset. For the four skeletal muscle regions, DSC reached above 0.92, and Sensitivity exceeded 0.93. Our method achieved the best segmentation results for the Paravertebral muscles, which were easy to segment because of their large area and concentration near the L3 vertebra. However, the skeletal muscles represented by Right Psoas and Left Psoas are very small, so the corresponding metrics are low, which makes the average values of the corresponding muscles less than those of the Paravertebral muscles.

Table 3 also shows the average surface distance error of the four skeletal muscle regions. The Paravertebral muscle had the smallest ASSD of 0.410 mm; while the Left Psoas muscle had the largest ASSD at 0.689 mm. The average ASSD for all skeletal muscles reached 0.558 mm.

3.3. Qualitative segmentation results

To observe whether our method achieved effective segmentation of skeletal muscle edges, Figure 4 shows the comparison of the segmented contours and the target contours in a CT image from the independent test dataset. The green line shows the contour of the target, and the red line denotes the contour of the segmentation result.

As seen in Figure 4, 2D U-Net, 3D U-Net, 3D ResU-Net, and LSMU-Net-RS have a poor effect on segmenting skeletal muscle in this data. Compared to LSMU-Net-3D, LSMU-Net-AB, and

TABLE 3 DSC, sensitivity, and ASSD (mm) of four skeletal muscle regions of the CT image in the independent test dataset segmented by our LSMU-Net.

	DSC	Sensitivity	ASSD (mm)
Rectus abdominis	0.943 ± 0.001	0.943 ± 0.002	0.431 ± 0.045
Right psoas	0.928 ± 0.002	0.941 ± 0.001	0.689 ± 1.299
Left psoas	0.922 ± 0.002	0.938 ± 0.001	0.701 ± 1.409
Paravertebral	0.957 ± 0.001	0.953 ± 0.001	0.410 ± 0.030
Average	0.937 ± 0.002	0.944 ± 0.001	0.558 ± 0.715

LSMU-Net-W, the LSMU-Net can reduce the wrong pixels at the edges of the segmentation results. In LSMU-Net, the green line contours overlap more with the red line contours, especially in uneven areas. While using the 2D ResU-Net, most of the muscles are well segmented, but in the magnified edge part, it is still non-fine edge segmentation compared to LSMU-Net. While compared with the network using SE attention (i.e., LSMU-Net + SE), LSMU-Net shows smoother boundary segmentation. This shows that the effect of edge refinement of the network proposed in the study is obvious. The visualization results of nnU-Net is similar to that of LSMU-Net.

To illustrate the performance of the 3D encoding branch, Figure 5 visualizes the results of any three CT images segmented by LSMU-Net-3D and LSMU-Net from the sagittal or coronal views, respectively. Red, green, blue, and yellow represent the segmentation of Rectus Abdominis, Right Psoas, Left Psoas, and Paravertebral, respectively. For the muscle boundary region between the Right Psoas and Paravertebral muscles that is difficult to distinguish, the segmentation result by LSMU-Net is closer to the Ground Truth label than that by LSMU-Net-3D by observing the magnified corresponding area. The reason is that the texture attention block of 3D encoding branch enhances the spatial integrity of the skeletal muscle bundle, thereby solving the challenge of identifying the boundaries of skeletal muscle bundles.

3.4. Auxiliary diagnostic information

As mentioned previously, the existing diagnostic index for 'sarcopenia' is the assessment of overall skeletal muscle (e.g., L3SMI). However, the larger the skeletal muscle volume involved in the calculation, the more reasonable the calculated value for diagnosing the presence or absence of sarcopenia. In the study, the average cross-sectional area of skeletal muscle volume corresponding to the L3 vertebra is used. Furthermore, since our LSMU-Net can segment four skeletal muscle regions in all L3-related axial slices, this makes it possible to quantitatively investigate the symptoms of cirrhotic sarcopenia in multiple muscle regions around L3. Therefore, this study will take the L3SMI, the diagnostic index of sarcopenia, as criterion to explore its relationship with the muscle indices of the four skeletal muscle regions.

The relationship was explained by the correlation analysis in the CT images of 98 patients in the cirrhosis group. Firstly, the average cross-sectional areas of the total skeletal muscle volume, as well as the four skeletal muscle regions, were calculated, respectively; secondly, referring to L3SMI's formula, that is, the ratio of the skeletal muscle area to the square of the body height, four potential diagnostic indices were obtained, i.e., rectus abdominal index (RAI) based on Rectus Abdominis region, right psoas index (RPI) based on Right Psoas muscle region, left psoas index (LPI) based on Left Psoas muscle region and paravertebral index (PI) based on Paravertebral muscle region. Here, the total psoas index (TPI) was calculated by summing the Left Psoas and Right Psoas muscle region; finally, according to gender and whether it is sarcopenia, the correlations between the new index and L3SMI were calculated and listed in Table 4.

Figure 6 also visualizes the correlation analysis between the new indicators and L3SMI depending on the gender of the patients in the cirrhosis group. As seen in Table 4 and Figure 6, the correlation between the five new indicators and L3SMI is higher in Non-sarcopenia patients than in Sarcopenia patients. Compared to the Male patients, the RAI and PI of the Female patients have a higher correlation with L3SMI, while their RPI, LPI, and TPI have a lower correlation with L3SMI. The correlation between RAI and L3SMI is the highest regardless of gender and whether the patient suffered from sarcopenia. From the overall data, the correlation between all indices and L3SMI is greater than 0.80.

Furthermore, according to the diagnostic cut-off value of L3SMI, Table 5 lists the cut-off value, corresponding Accuracy and AUC of the five new indicators in the diagnosis of cirrhotic sarcopenia in female and male, respectively. Due to the highest correlation between RAI and L3SMI, the diagnostic accuracy of 0.941 can be achieved by selecting the appropriate cut-off value such as 16.67 cm^2/m^2 in female. Therefore, the Rectus Abdominis can achieve the alternative diagnostic effect in cases where the overall skeletal muscle is not available. As seen in Table 5, the diagnostic effect of skeletal muscle region index is RAI > PI > LPI = RPI = TPI for female and RAI > LPI = TPI > RPI > PI for male.

The receiver operating characteristic (ROC) curve provides a simple way to observe the diagnostic performance of a clinical indicator. The performance of the ROC curve is usually expressed by the area under curve (AUC), the value of which is the size of the area under the ROC curve. The closer the AUC is to 1.0, the higher the performance of the diagnostic index. When the AUC is equal to 0.5, the performance of the diagnostic index is the lowest. Table 5 shows the AUC of RAI, RPI, LPI, PI and TPI in females and males, respectively. It can be seen that the RAI index performed best in identifying cirrhotic sarcopenia in females and males. The diagnostic cut-off values for skeletal muscle regional indicators selected from the AUC results are ordered as RAI > PI > RPI > LPI > TPI for female and RAI > LPI > TPI > RPI > PI for male.

4. Conclusion

This study presented an automatic segmentation method of multi-region skeletal muscle in abdominal or abdominopelvic CT images. Our method achieved good performance by combining the appearance of skeletal muscle regions in CT images into advanced U-Net architecture. Specifically, our method includes enhancement of the existing U-Net models; texture attention enhancement block for augmenting the blurred edges of skeletal muscles; 3D encoding branch for extracting feature of muscle fiber bundles; and loss functions using the prior knowledge to reduce the class imbalance. Therefore, our method accurately segmented the multiple skeletal muscle regions from all L3-related axial slices in more than 300 abdominal or

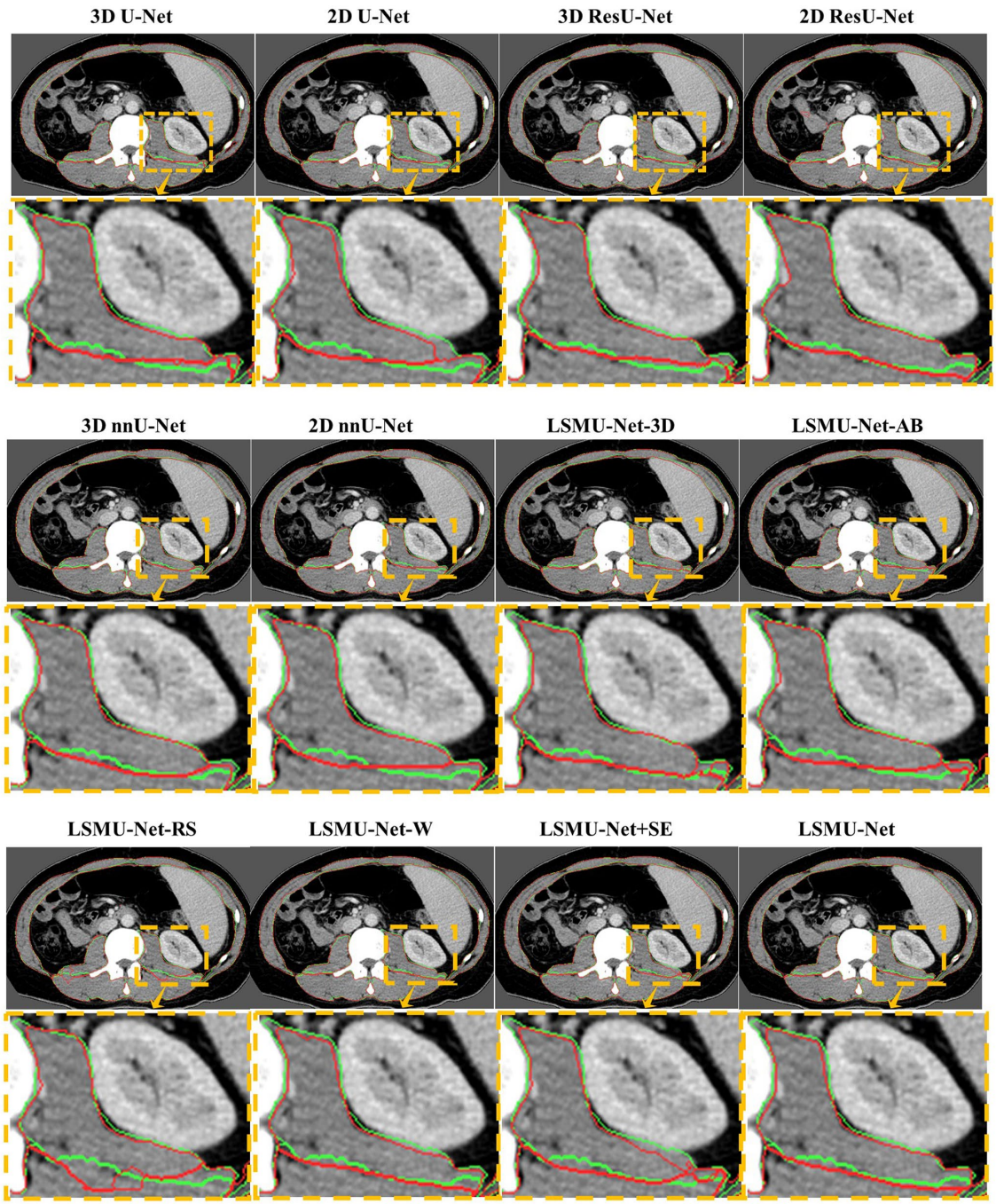


FIGURE 4
Comparison of the segmented contours and the target contours in a CT image from the generalized dataset. The green line denotes the target contour, and the red line denotes the contour of the segmentation result.

abdominopelvic CT images, and the segmentation prediction time meets the clinical real-time requirement.

Based on the segmentation results of four skeletal muscle regions, the five skeletal muscle region indices were calculated, and their correlation with L3SMI was quantitatively analyzed in the diagnosis of sarcopenia. The five skeletal muscle region indices, especially RAI, could be used to assist in the diagnosis of sarcopenia in cases where the total muscle was not available.

5. Discussion

Clinically, sarcopenia is usually diagnosed by L3SMI calculated on the skeletal muscle region. Existing deep learning methods have greatly improved the performance of skeletal muscle segmentation, however, for patients with cirrhosis, skeletal muscle may be squeezed and deformed by pathological changes, resulting in errors in the calculation of L3SMI. This study proposed the lumbar skeletal muscle

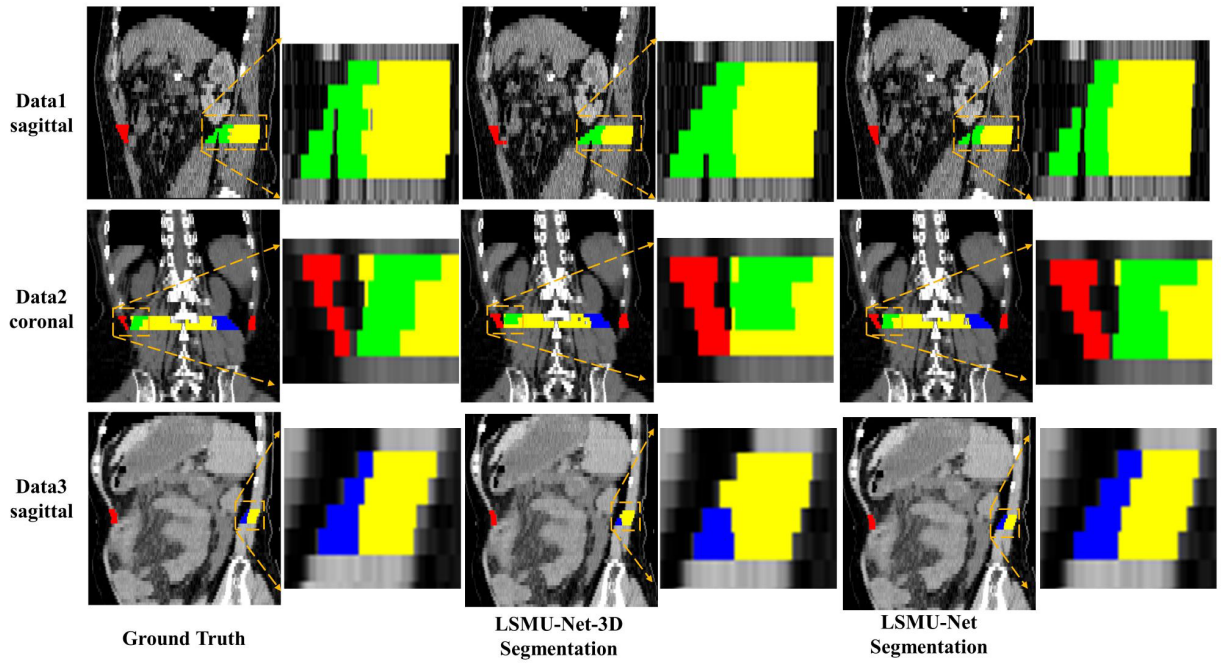


FIGURE 5
Illustration of the performance of 3D encoding branch by qualitative comparison of the segmentation results of any three CT images by LSMU-Net-3D and LSMU-Net from the sagittal or coronal views, respectively. The magnified region shows that the result output by LSMU-Net is closer to the ground truth label for the area between the right psoas (green) and the paravertebral muscles (yellow) that is difficult to distinguish.

TABLE 4 Correlation analysis between new indices (RAI, RPI, LPI, PI, and TPI) and L3SMI in the CT images of 98 patients in the cirrhosis group according to gender and whether it is sarcopenia.

	RAI	RPI	LPI	PI	TPI
Sarcopenia	0.879	0.767	0.693	0.697	0.767
Non-sarcopenia	0.902	0.842	0.799	0.772	0.848
Female	0.935	0.712	0.692	0.862	0.766
Male	0.926	0.842	0.780	0.796	0.836
All Data	0.932	0.839	0.801	0.831	0.847

segmentation network based on the U-Net enhanced by residual structure to segment four skeletal muscle regions in all axial CT slices related to L3 (i.e., LSMU-Net). The average cross-sectional area of four skeletal muscle regions can be used to calculate the diagnostic indexes of sarcopenia.

Comparative ablation experiments showed that the LSMU-Net method proposed in the study has good performance in terms of DSC, Sensitivity, and ASSD, which indicates the feasibility of LSMU-Net. The experimental results also showed that 2D nnU-Net and 3D nnU-Net perform well in the segmentation tasks. LSMU-Net is slightly superior to 2D nnU-Net in DSC, Sensitivity, and ASSD. LSMU-Net is slightly superior to 3D nnU-Net in ASSD and Sensitivity, while DSC is lower than the corresponding values of 3D nnU-Net. The performance of our method is achieved by combining the advanced 2D U-Net with residual structure, texture attention enhancement blocks, 3D encoding branches and the priori knowledge. Different from our LSMU-Net, nnU-Net still uses the original U-Net structure, but achieves good performance with the help of many advanced techniques, such as image preprocessing, dynamic adaptation of network topology, training strategy, inference post-processing and so on. Therefore, in addition to the improvement of network structure, the optimization and integration of data processing and training methods are also extremely important in future segmentation work.

Among the four skeletal muscle regions, Rectus Abdominis and Paravetebral muscle are larger, while Right Psoas and Left Psoas are smaller. From the perspective of segmentation evaluation index, the index of the first two regions is higher, while that of the latter two regions is slighter lower. This shows that the proposed network still has difficulties in segmenting small targets such as the Right Psoas and the Left Psoas, and the performance of the segmentation method needs to be improved.

In addition to L3SMI, sarcopenia is also diagnosed by psoas muscle index (PMI). PMI is often calculated based on the psoas major muscle, defined as the ratio of the cross-sectional area of bilateral psoas major muscles to the square of body height. The PMI's cut-off values are 5.24 cm^2/m^2 in males and 3.85 cm^2/m^2 in females (Dolan et al., 2019). In this study, TPI was calculated based on the custom Left Psoas and Right Psoas regions, which includes the psoas major muscle and the psoas square muscle. TPI's cut-off values are 12.51 cm^2/m^2 in males and 7.27 cm^2/m^2 in females. Obviously, TPI is defined in a larger skeletal muscle region than PMI, which may be a useful complement to PMI.

According to the results of AUC, in females, the comprehensive performance of RPI and LPI is higher than that of TPI; while in

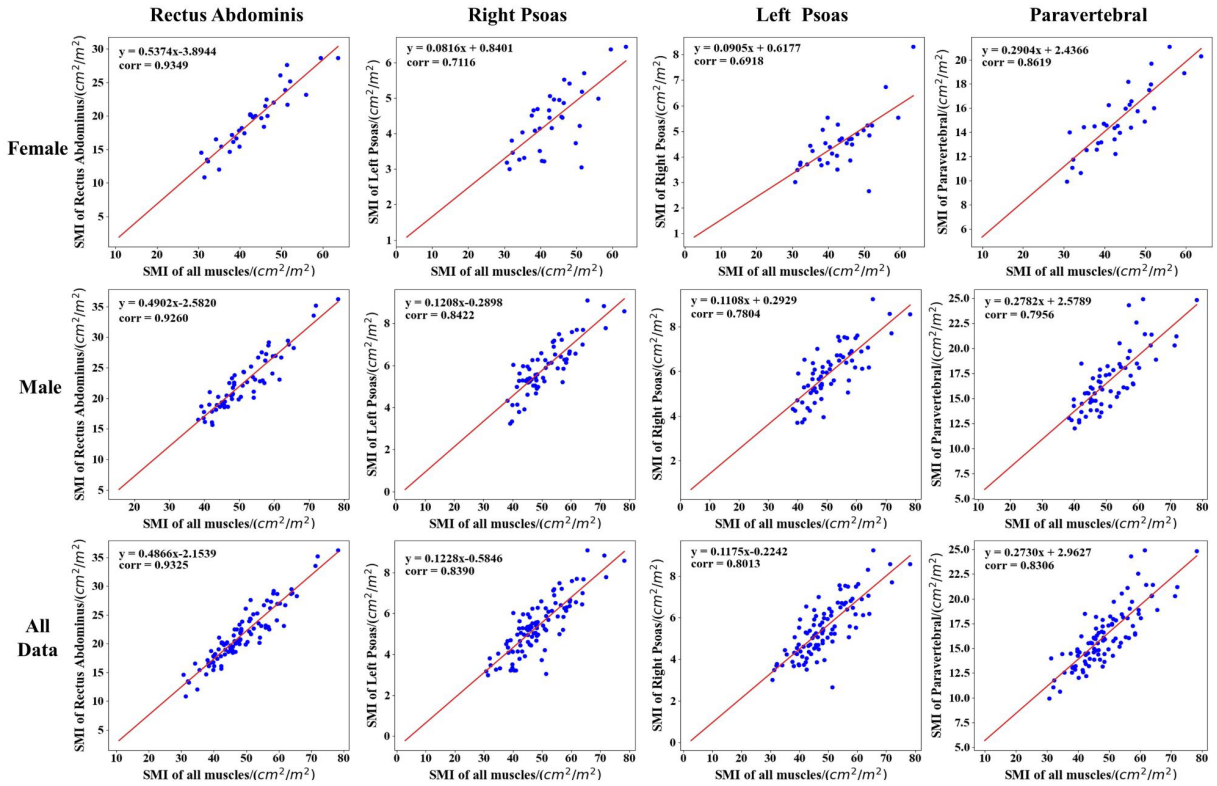


FIGURE 6
Correlation analysis between the new indicator and L3SMI according to the gender of the patients in the cirrhosis group.

TABLE 5 Cut-off value and corresponding accuracy of the five new indicators in the diagnosis of cirrhotic sarcopenia in female and male.

		RAI	RPI	LPI	PI	TPI
Female	Cut-off value	16.67	4.14	3.76	13.20	7.27
	Accuracy	0.941	0.794	0.794	0.882	0.794
	AUC	0.929	0.796	0.708	0.829	0.679
Male	Cut-off value	22.51	5.84	6.10	17.28	12.51
	Accuracy	0.891	0.859	0.875	0.828	0.875
	AUC	0.889	0.862	0.875	0.823	0.871

males, the diagnostic value of LPI (AUC = 0.875) is similar to that of TPI (AUC = 0.871), but different from that of RPI (AUC = 0.862). The comprehensive performance of RPI and LPI could not be compared to that of TPI in males. This issue may be related to the small number of samples of existing data sets, which need to be explored and analyzed in more cases of cirrhotic sarcopenia.

The segmented network and the five new metrics of skeletal muscle regions could better assist physicians. The results of this study may play a very important auxiliary role in the diagnosis of cirrhotic sarcopenia, especially in cases where intact skeletal muscle is not available in axial CT slices. However, this study also has some shortcomings, such as the data set is only from one institution, and the number of cases is only 317. In addition, the study only considered the effects of muscle and did not address other parameters, such as intra-abdominal fat, organ fat and subcutaneous fat. In the future, we will combine these parameters for further study to improve the automatic diagnosis of sarcopenia.

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 authors.

Ethics statement

Written informed consent was obtained from the individual(s) for the publication of any potentially identifiable images or data included in this article.

Author contributions

GS did the experiment and wrote the article. JZ scanned and labeled the clinical images. KW and DY reviewed the experimental

process and results. SC determined the clinical patient collection plan and designed the whole plan. YS designed the whole plan, guided the experiment, and revised the article. GS and JZ are co-first authors. All authors contributed to the article and approved the submitted version.

Acknowledgments

This work was supported by the medical-industrial integration project of Fudan University under grant XM03211181. This work was also supported by the National Natural Science Foundation of China under grant 82072021.

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.2023.1203823/full#supplementary-material

References

Barnard, R., Tan, J., Roller, B., Chiles, C., Weaver, A. A., Boutin, R. D., et al. (2019). Machine learning for automatic paraspinous muscle area and attenuation measures on low-dose chest CT scans. *Acad. Radiol.* 26, 1686–1694. doi: 10.1016/j.acra.2019.06.017

Bauer, J., Morley, J. E., Schols, A. M., Ferrucci, L., Cruz-Jentoft, A. J., Dent, E., et al. (2019). Sarcopenia: a time for action. An scwd position paper. *J. Cachexia. Sarcopenia Muscle* 10, 956–961. doi: 10.1002/jcsm.12483

Blanc-Durand, P., Schiratti, J. B., Schutte, K., Jehanno, P., Herent, P., Pigneur, F., et al. (2020). Abdominal musculature segmentation and surface prediction from CT using deep learning for sarcopenia assessment. *Diagn. Interv. Imaging* 101, 789–794. doi: 10.1016/j.diii.2020.04.011

Borrelli, P., Kaboteh, R., Enqvist, O., Ulén, J., Trägårdh, E., Kjölhede, H., et al. (2021). Artificial intelligence-aided CT segmentation for body composition analysis: a validation study. *Eur. Radiol. Exp.* 5:11. doi: 10.1186/s41747-021-00210-8

Burns, J. E., Yao, J., Chalhoub, D., Chen, J. J., and Summers, R. M. (2020). A machine learning algorithm to estimate sarcopenia on abdominal ct. *Acad. Radiol.* 27, 311–320. doi: 10.1016/j.acra.2019.03.011

Cao, P., Yao, J., Zhu, N., Chang, L., and Yuan, L. (2017). Sarcopenia as an assessment of nutritional status and its risk factors in patients with hepatic cirrhosis. *Chinese Remed. Clin.* 17, 1737–1739. doi: 10.11655/zgywylc2017.12.007

Cao, H., Wang, Y., Chen, J., Jiang, D., Zhang, X., Tian, Q., et al. (2023). Swin-unet: Unet-like pure transformer for medical image segmentation. *Computer Vision--ECCV 2022 Workshops: Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part III*. 24, 205–218.

Carey, E. J., Lai, J. C., Wang, C. W., Dasarathy, S., Lobach, I., Montano-Loza, A. J., et al. (2017). A multicenter study to define sarcopenia in patients with end-stage liver disease. *Liver Transpl.* 23, 625–633. doi: 10.1002/lt.24750

Castiglione, J., Somasundaram, E., Gilligan, L. A., Trout, A. T., and Brady, S. (2021). Automated segmentation of abdominal skeletal muscle on pediatric CT scans using deep learning. *Radiology: Artif. Intell.* 3:e200130. doi: 10.1148/ryai.2021200130

Çiçek, Ö, Abdulkadir, A, Lienkamp, S. S, and Brox, T., and Ronneberger, O. (2016). 3D U-Net: learning dense volumetric segmentation from sparse annotation, International conference on medical image computing and computer-assisted intervention, Springer, 424–432

Dabiri, S., Popuri, K., Feliciano, E. M. C., Caan, B. J., Baracos, V. E., and Beg, M. F. (2019). Muscle segmentation in axial computed tomography (ct) images at the lumbar (l3) and thoracic (t4) levels for body composition analysis. *Comput. Med. Imaging Graph.* 75, 47–55. doi: 10.1016/j.compmedimag.2019.04.007

Dabiri, S., Popuri, K., Ma, C., Chow, V., Feliciano, E. M. C., Caan, B. J., et al. (2020). Deep learning method for localization and segmentation of abdominal CT. *Comput. Med. Imaging Graph.* 85:101776. doi: 10.1016/j.compmedimag.2020.101776

Dolan, D., Knight, K., Maguire, S., and Moug, S. (2019). The relationship between sarcopenia and survival at 1 year in patients having elective colorectal cancer surgery. *Tech. Coloproctol.* 23, 877–885. doi: 10.1007/s10151-019-02072-0

Hanai, T., Shiraki, M., Nishimura, K., Ohnishi, S., Imai, K., Suetsugu, A., et al. (2015). Sarcopenia impairs prognosis of patients with liver cirrhosis. *Nutrition* 31, 193–199. doi: 10.1016/j.nut.2014.07.005

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.* 2016, 770–778. doi: 10.1109/CVPR.2016.90

Hu, J., Shen, L., and Sun, G. (2020). Squeeze-and-excitation networks. *IEEE Trans. Pattern Anal. Mach. Intell.* 42, 2011–2023. doi: 10.1109/TPAMI.2019.2913372

Huang, J., Shen, H., Chen, B., Wang, Y., and Li, S. (2020). Segmentation of paraspinal muscles at varied lumbar spinal levels by explicit saliency-aware learning, International conference on medical image computing and computer-assisted intervention, Springer, 652–661

Isensee, F., Jaeger, P. F., Kohl, S. A., Petersen, J., and Maier-Hein, K. H. (2021). nnU-net: a self-configuring method for deep learning-based biomedical image segmentation[J]. *Nat. Methods* 18, 203–211. doi: 10.1038/s41592-020-01008-z

Jia, H., Cai, W., Huang, H., and Xia, Y. (2022). Learning multi-scale synergic discriminative features for prostate image segmentation. *Pattern Recogn.* 126:108556. doi: 10.1016/j.patcog.2022.108556

Kim, G., Kang, S. H., Kim, M. Y., and Baik, S. K. (2017). Prognostic value of sarcopenia in patients with liver cirrhosis: a systematic review and meta-analysis. *PLoS One* 12:e0186990. doi: 10.1371/journal.pone.0186990

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, K., Zung, J., Li, P., Jain, V., and Seung, H. S. (2017). Superhuman accuracy on the snemi3d connectomics challenge. *arXiv*. Available at: https://arxiv.org/abs/1706.00120

Li, T., Kong, M., Song, W., Xu, M., and Chen, Y. (2020). Relationship between skeletal muscle index of third lumbar vertebrae and clinical characteristics in patients with acute on chronic liver failure. *J. Practic. Hepatol.* 23, 467–470. doi: 10.3969/j.issn.1672-5069.2020.04.004

Liu, S., Xu, D., Zhou, S. K., Mertelmeier, T., Wicklein, J., Jerebko, A. K., et al. (2017). 3D anisotropic hybrid network: transferring convolutional features from 2D images to 3D anisotropic volumes. *arXiv*. Available at: https://arxiv.org/abs/1711.08580

Liu, Y., Zhou, J., Chen, S., and Liu, L. (2019). Muscle segmentation of l3 slice in abdomen CT images based on fully convolutional networks, 2019 ninth international conference on image processing theory, Tools and Applications (IPTA), IEEE, 1–5. doi: 10.1109/IPTA.2019.8936106

Long, J., Shelhamer, E., and Darrell, T. (2015). Fully convolutional networks for semantic segmentation. *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.* 2015, 3431–3440. doi: 10.1109/CVPR.2015.7298965

Meyer, A., Chlebus, G., Rak, M., Schindele, D., Schostak, M., van Ginneken, B., et al. (2021). Anisotropic 3D multi-stream cnn for accurate prostate segmentation from multi-planar mri. *Comput. Methods Prog. Biomed.* 200:105821. doi: 10.1016/J.CMPB.2020.105821

Park, H. J., Shin, Y., Park, J., Kim, H., Lee, I. S., Seo, D. W., et al. (2020). Development and validation of a deep learning system for segmentation of abdominal muscle and fat on computed tomography. *Korean J. Radiol.* 21, 88–100. doi: 10.3348/kjr.2019.0470

Qin, C., Zhou, J., Yao, D., Zhuang, H., Wang, H., Chen, S., et al. (2021). Vertebrae labeling via end-to-end integral regression localization and multi-label classification

network[J]. *IEEE Trans. Neural Netw. Learn. Syst.* 33, 2726–2736. doi: 10.1109/TNNLS.2020.3045601

Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: convolutional networks for biomedical image segmentation, International conference on medical image computing and computer-assisted intervention, Springer, 16591–16603, 9

Tantai, X., Liu, Y., Yeo, Y. H., Praktiknjo, M., Mauro, E., Hamaguchi, Y., et al. (2022). Effect of sarcopenia on survival in patients with cirrhosis: a meta-analysis. *J. Hepatol.* 76, 588–599. doi: 10.1016/j.jhep.2021.11.006

Wang, S., Xie, H., and Gong, Y. (2020). The value of L3 skeletal muscle index in evaluating preoperative nutritional risk and long-term prognosis in colorectal cancer patients. *Sci. Rep.* 10:8153. doi: 10.1038/s41598-020-65091-0

Weston, A. D., Korfiatis, P., Philbrick, K. A., Conte, G. M., Kostandy, P., Sakinis, T., et al. (2020). Complete abdomen and pelvis segmentation using u-net variant architecture. *Med. Phys.* 47, 5609–5618. doi: 10.1002/mp.14422

Wu, C. H., Liang, P. C., Hsu, C. H., Chang, F. T., Shao, Y. Y., and Shih, T. T. F. (2021). Total skeletal, psoas and rectus abdominis muscle mass as prognostic factors for patients with advanced hepatocellular carcinoma. *J. Formos. Med. Assoc.* 120, 559–566. doi: 10.1016/j.jfma.2020.07.005

Xiao, J., Wang, F., Wong, N. K., He, J., Zhang, R., Sun, R., et al. (2019). Global liver disease burdens and research trends: analysis from a Chinese perspective. *J. Hepatol.* 71, 212–221. doi: 10.1016/j.jhep.2019.03.004

Zhou, Y., Huang, W., Dong, P., Xia, Y., and Wang, S. (2019). D-unet: a dimension-fusion u shape network for chronic stroke lesion segmentation. *IEEE/ACM Trans. Comput. Biol. Bioinform.* 18, 940–950. doi: 10.1109/TCBB.2019.2939522

Zou, K. H., Warfield, S. K., Bharatha, A., Tempany, C. M., Kaus, M. R., Haker, S. J., et al. (2004). Statistical validation of image segmentation quality based on a spatial overlap index1. *Acad. Radiol.* 11, 178–189. doi: 10.1016/s1076-6332(03)00671-8

frontiers | Frontiers in Neuroscience

TYPE Original Research
PUBLISHED 05 June 2023
DOI 10.3389/fnins.2023.1200576

Check for updates

OPEN ACCESS

EDITED BY
Alois C. Knoll,
Technical University of Munich, Germany

REVIEWED BY
Jianming Wei,
Chinese Academy of Sciences (CAS), China
Huiqun Wu,
Nantong University, China

*CORRESPONDENCE
Xinrong Chen
✉ chenxinrong@fudan.edu.cn
Zhijian Song
✉ zjsong@fudan.edu.cn

RECEIVED 05 April 2023
ACCEPTED 15 May 2023
PUBLISHED 05 June 2023

CITATION
Wang J, Zhang X, Chen X and Song Z (2023) A touch-free human-robot collaborative surgical navigation robotic system based on hand gesture recognition.
Front. Neurosci. 17:1200576.
doi: 10.3389/fnins.2023.1200576

COPYRIGHT
© 2023 Wang, Zhang, Chen and Song. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

A touch-free human-robot collaborative surgical navigation robotic system based on hand gesture recognition

Jie Wang[1,2], Xinkang Zhang[1,2], Xinrong Chen[1,2]* and Zhijian Song[2,3]*

[1]Academy for Engineering and Technology, Fudan University, Shanghai, China, [2]Shanghai Key Laboratory of Medical Image Computing and Computer Assisted Intervention, Shanghai, China, [3]Digital Medical Research Center, School of Basic Medical Science, Fudan University, Shanghai, China

Robot-assisted minimally invasive surgery (RAMIS) has gained significant traction in clinical practice in recent years. However, most surgical robots rely on touch-based human-robot interaction (HRI), which increases the risk of bacterial diffusion. This risk is particularly concerning when surgeons must operate various equipment with their bare hands, necessitating repeated sterilization. Thus, achieving touch-free and precise manipulation with a surgical robot is challenging. To address this challenge, we propose a novel HRI interface based on gesture recognition, leveraging hand-keypoint regression and hand-shape reconstruction methods. By encoding the 21 keypoints from the recognized hand gesture, the robot can successfully perform the corresponding action according to predefined rules, which enables the robot to perform fine-tuning of surgical instruments without the need for physical contact with the surgeon. We evaluated the surgical applicability of the proposed system through both phantom and cadaver studies. In the phantom experiment, the average needle tip location error was 0.51 mm, and the mean angle error was 0.34 degrees. In the simulated nasopharyngeal carcinoma biopsy experiment, the needle insertion error was 0.16 mm, and the angle error was 0.10 degrees. These results indicate that the proposed system achieves clinically acceptable accuracy and can assist surgeons in performing contactless surgery with hand gesture interaction.

KEYWORDS

robot-assisted minimally invasive surgery, surgical robot, human-robot interaction, gesture recognition, contactless surgery

1. Introduction

Robot-assisted minimally invasive surgery (RAMIS) is now well established in clinical practice due to its high precision and minimal invasiveness (Nagyné Elek and Haidegger, 2019; Haidegger et al., 2022). In RAMIS, preoperative medical image data is utilized to plan the surgical path, while the robot performs the approach during the surgery as per the plan. Surgeons must manipulate various software to control the navigational surgical robot throughout the procedure, especially to fine-tune surgical instruments with their perspective in complex surgeries. However, at present, adapting the surgical robot by manual means increases the risk of bacterial diffusion, rendering the surgeon unable to control the robot during surgery while complying with the high sterile requirements. To address this issue, various types of study have been proposed. Some

studies have attempted to solve this problem by using other devices such as joysticks and pedals to transform the surgeon's command into actions (Díaz et al., 2014; Ohmura et al., 2018). Nevertheless, in the case of joysticks, human-robot interaction (HRI) tasks applied to surgical robots are performed through master–slave operations, with which has not been effectively resolved on the movement difference between the master and slave console and the problem of over-operation. On the other hand, the pedal-based solutions are still limited by behavioral consistency, which impedes their use for every surgeon in RAMIS, particularly those who are unskilled. Recently, several studies have attempted to address this issue through contactless HRI using touch-free solutions (Nestorov et al., 2016; Cho et al., 2018; Despinoy et al., 2018), with a particular focus on hand gesture recognition-based HRI. The researchers have made significant progress in modeling and analyzing hand gesture recognition. These studies have adopted various frameworks to predict users' intentions in HRI tasks and enable robots to perform corresponding actions, including probabilistic graphical models of temporal processes, deep learning techniques with supervised learning, and other methods including unsupervised learning algorithms, among others (Van Amsterdam et al., 2021; Cao et al., 2022).

Probabilistic graphical models of temporal processes, which have been widely utilized in speech recognition for time series analysis, have also served as a source of inspiration for gesture recognition in HRI tasks (Ahmidi et al., 2017). Chen et al. (2015) introduces a novel hand gesture recognition model based on hidden Markov models (HMM), which could identify a worker's gesture patterns and intentions with reliable accuracy. Mavroudi et al. (2018) proposes a framework for fine-grained gesture segmentation and recognition, which employs a Conditional Random Field (CRF) model and a frame-level representation based on discriminative sparse coding. Reiley et al. (2008) utilizes Linear Discriminant Analysis (LDA) and HMM to build models for gesture recognition, which improved the recognition rate by promoting discrimination between sub-gestures instead of the entire gesture, thus enabling them to capture the internal variability of each segment. The aforementioned models have been implemented effectively to analyze the kinematic signals for the da Vinci surgical robot. Deep learning techniques, specifically the implementation of deep convolutional neural networks (CNN), have been employed for the purpose of recognizing gestures. In the study by Oyedotun and Khashman (2017), the image is first preprocessed using binarization, followed by setting a threshold to locate the gesture, and finally, a CNN is utilized to recognize the gestures. Similarly, ElBadawy et al. (2017) uses a 3D CNN-based gesture recognition system to analyze normalized images, achieving a recognition rate averaging 90%. Huynhnguyen and Buy (2021) introduces a 2-stage surgical gesture recognition approach, where one stage detects the transition between consecutive gestures using a 3D CNN, and the other stage classifies video clips into corresponding gesture classes based on a long short-term memory (LSTM) neural network. Experimental results using JIGSAWS's suturing video dataset show that the proposed method achieves an accuracy of over 70% for both tasks. Moreover, Fang et al. (2019) presents a gesture recognition system that combines generative adversarial network (GAN) and CNN, achieving better results with fewer samples.

Furthermore, there are alternative approaches for gesture recognition in HRI tasks. Huang et al. (2011) presents a gesture recognition approach that relies on Gabor filters and a support vector machine (SVM) classifier. Their proposed method is highly resistant to variations in illumination, leading to recognition rates that improve from 72.8 to 93.7%. Tarvekar (2018) introduces a skin threshold segmentation approach for recognizing and categorizing gestures by segmenting hand regions in images and extracting color and edge features through an SVM classifier. Shi et al. (2021) proposes a novel domain adaptive framework for robotic gesture recognition that aligns unsupervised kinematic visual data, enabling the real robot to acquire multi-modality knowledge from a simulator. The empirical evidence indicates that the model has the potential to significantly enhance the operational efficiency of the real robot, resulting in a noteworthy 12.91% increase in precision. Moreover, there exist cases in which the recognition of hand gestures is facilitated through the utilization of Leap Motion™ and Kinect™ devices (Ahmad et al., 2016; Jin et al., 2016).

In current RAMIS procedures, limited interactions between the surgeon and the robot restrict surgical efficiency. And it is apparent that the majority of present-day studies employ relatively intricate techniques to achieve specific HRI tasks using various devices. However, little study has presented a comprehensive framework for gesture recognition that exhibits strong generalization capabilities and high efficiency, which can be applied effectively to address the problem of the contactless HRI in RAMIS. To this end, we propose a concise and effective framework for navigational surgical robots to perform actions in response to the surgeon's gestures in this paper, utilizing touch-free solutions based on hand gesture recognition technology. This framework facilitates the robot to execute surgical interventions under the guidance of an expert surgeon and a surgical navigation system, resulting in enhanced medical treatment efficacy and conserved healthcare resources, while also ensuring aseptic conditions that impede bacterial dissemination.

2. Materials and methods

2.1. System composition

As depicted in Figure 1, the collaborative surgical navigation robot system is primarily composed of a computer workstation, an optical positioning-based surgical navigator, auxiliary accessories such as surgical probes and locators, and a robot module that includes a 7-DoF robotic arm and its controller. Both the operating table and surgical navigator are mobile devices that can be adjusted to fit the patient and surgeon's positions. The workstation and its internal software connect the surgical navigator and the robot into a closed-loop structure. The surgical navigator tracks the patient and surgical instruments in real-time by positioning reflective balls mounted on the operating table and the robot manipulator. The collaborative surgical robot, with its terminal surgical instruments, can be positioned flexibly around the operation table and controlled by the surgeon's hand gestures. The navigator constructs an enhanced surgical field by integrating preoperative medical information of patients (e.g., target organs, vessels and planned surgical paths) with the location and target points of intraoperative instruments attached to the actual patient body to provide surgeons with augmented visual information. With the direct navigation interface, hand gesture guidance can be used as a direct and natural method to interact with the surgical robot.

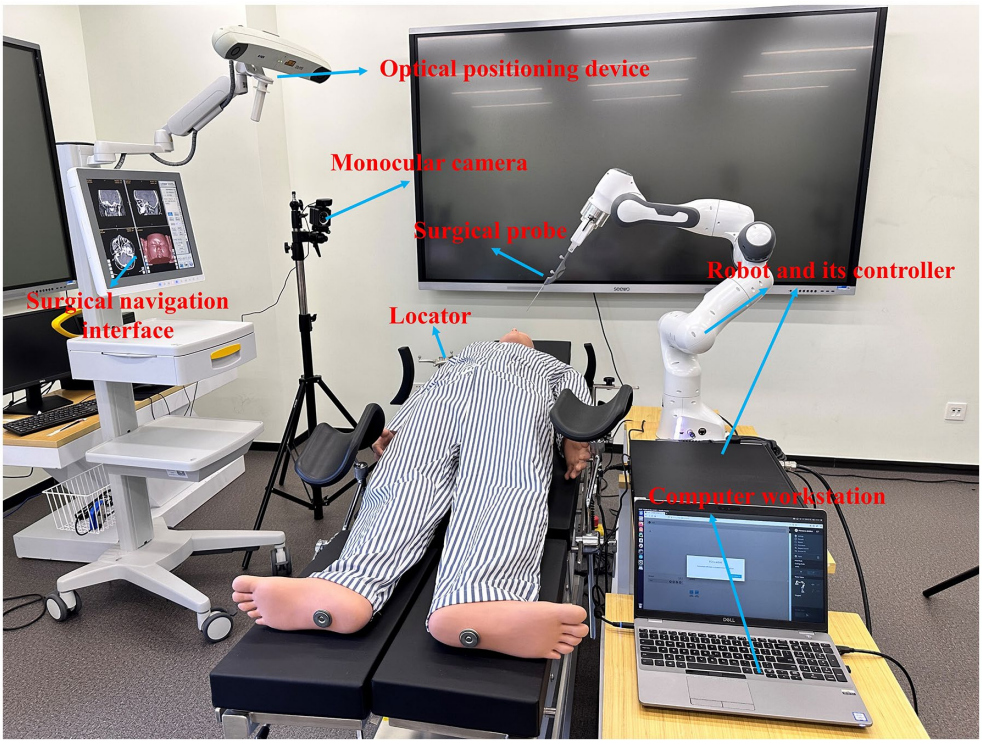


FIGURE 1
Overview of the non-contact collaborative surgical navigation robot system.

2.2. System workflow

The figure displayed in Figure 2 outlines the workflow of the proposed surgical system. The computer workstation serves as the main control and computation center, enabling robot control, generating enhanced surgical visual information, and supporting human-robot interaction. The hardware layer of the collaborative surgical robot system is denoted with blue dotted lines, while the human subjects involved in the touch-free surgical procedure, such as the surgeon, surgical navigation interface, surgical robot motion/execution, patient, and HRI interface, are located above this blue layer. The data flow is indicated by black arrow lines, including control and feedback flows between the subjects and hardware modules. The surgeon-centered interaction flow is shown with red dotted lines, highlighting the data flow among the subjects. For this contactless robot-assisted puncture treatment with a surgical navigation interface, a semi-automatic mode of surgical procedure is proposed. Surgeons are required to select surgical targets and needle insertion sites through patient image guidance before surgery and plan corresponding surgical paths. During surgery, surgeons can fine-tune the needle's posture directly through hand gesture interaction. The generation of the surgical navigation interface is based on our previous research (Liu et al., 2017; Chen et al., 2021). This paper centers on the attainment of contactless HRI objectives, specifically, the detection of gestures and the subsequent control of surgical robots.

2.3. Gesture recognition model

In this section, our attention is directed towards the gesture recognition module of our approach. The specific architecture of the model is illustrated in Figure 3.

The main function of the hand gesture recognition network is to process monocular images captured by the camera to acquire the desired pose and shape of the hand. The 3D pose of the hand is denoted by the 3D position of keypoints, while the shape of the hand is represented in the form of a mesh. We have identified a total of 21 keypoints on the hand as regression targets, which include the position of the wrist, finger joints, and fingertips. The 3D position of each key point is denoted by the (x, y, z) coordinates. The hand shape is represented by a mesh consisting of 778 nodes, with associated connection information between them. We represent the mesh in the network as a graph G(V, E), where V represents the 778 nodes and E denotes the connection information between them. Our gesture recognition module employs a Unet architecture and utilizes a multi-layer convolutional network for feature extraction, resulting in a feature map of varying sizes. This is followed by convolution and upsampling to extend the feature map and combine it with the previously extracted features at each layer. Our approach is divided into several distinct parts.

2.3.1. Keypoint regression branch

In order to simultaneously regress the pose of both hands, a regression approach is utilized to predict the keypoint locations. The

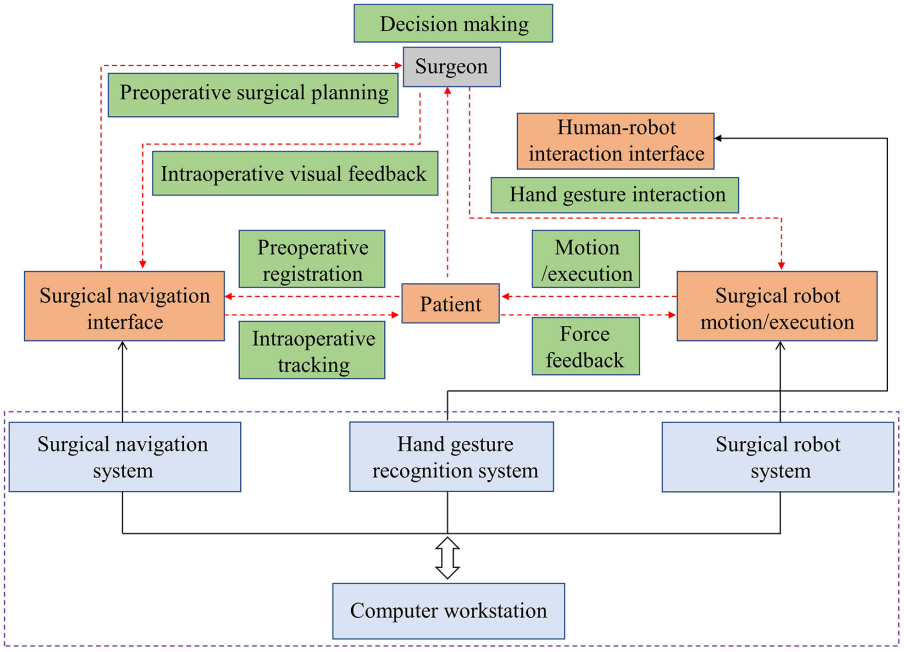


FIGURE 2
Workflow of the hand-gesture based surgeon-robot cooperation.

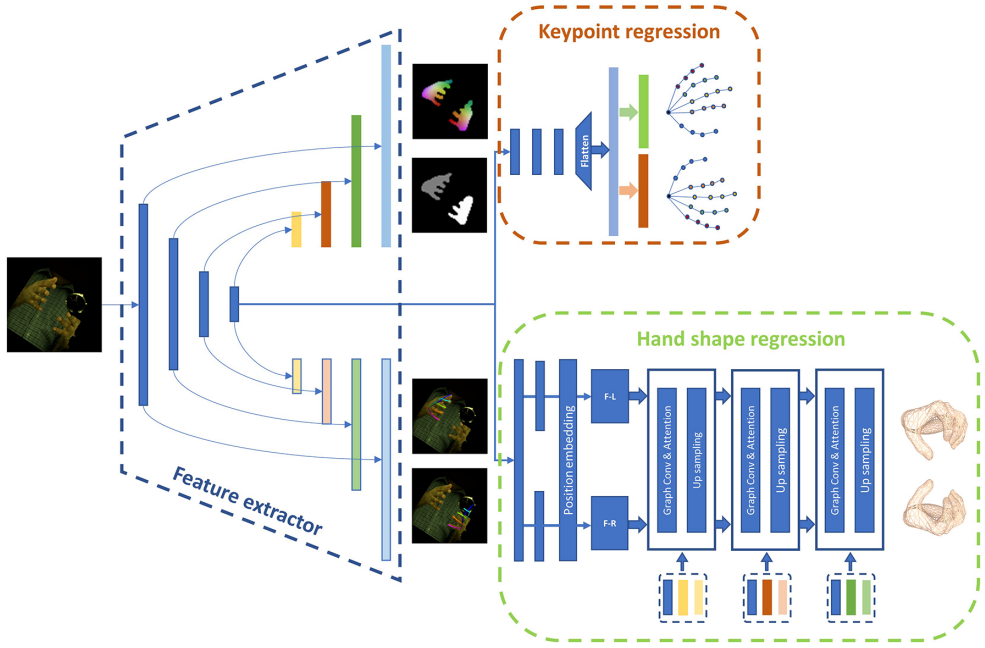


FIGURE 3
The concrete architecture of the hand gesture recognition model.

position information of the hand keypoints, with a shape of (2, 21, 3), is unfolded into a 126-dimensional vector, which is then paired with dataset labels to calculate the L2 loss values. This enables the network to learn how to regress the keypoints. Eq. (1) depicts the generation of the keypoint positions. The regression process employs the last layer of the encoder output, which passes through multiple homogeneity networks and is subsequently expanded into a 126-dimensional vector that represents the 3D positions of joints.

$$P = f_{flat}\left(f_{de}\left(f_{en}\left(X_{img}\right)\right)\right) \quad (1)$$

where $P \in \mathbb{R}^{2\times21\times3}$ indicates the vector of 126 dimensions, X_{img} is the RGB image, f_{en} and f_{de} represent encoder and decoder of the regression branch respectively, and f_{flat} denotes a flatten function that can convert a three-dimensional matrix to a 126-dimensional vector.

2.3.2. Auxiliary prediction branch

Three distinct auxiliary methods are employed to aid the model in making accurate predictions. These methods consist of hand segmentation, density mapping, and 2D pose. The 2D pose branch transforms the ultimate feature map of the Unet architecture into 21 heatmaps, which denote the 2D positions of both hands. Meanwhile, the hand segmentation branch restores the feature map to its original resolution, producing a mask with distinct pixel values for the left hand, right hand, and background. In addition, dense matching produces a dense mapping map with the same dimensions as the original map by establishing correspondences between images in a manner analogous to positional coding. We utilize dense matching to define the correspondence between vertices and image pixels, employing various hues to represent individual vertices.

The three categories of auxiliary information are labeled independently in the dataset and are utilized to compute the loss values, allowing the network to more effectively extract hand features.

2.3.3. Hand shape regression branch

Convolutional mesh regression is utilized in this branch to generate precise and dense 3D shapes for the hands. This classical method produces a 3D mesh aligned with the image, enabling the generation of intricate and fine 3D shapes. The hand shape regression branch comprises a network with graph convolution. As demonstrated in Eq. 2, the feature maps are spanned and propagated into two fully connected layers with a position embedding module to derive the left-hand and right-hand graph structures, respectively. The process for the former graph structure is illustrated in Eq. 3, where it is first subjected to graph convolution and subsequently, the outputs are passed through a multi-head attention mechanism module to establish attention between nodes within itself and merge with the features extracted from different layers by the feature extractor. Finally, it is transmitted to the interaction attention module across the left and right hands to determine the interaction relationship between the hands and assist in modifying their shape information. The hand mesh is generated in a coarse-to-fine approach, where a coarse mesh is initially generated, and then, according to the nearest neighbor mode, it is up-sampled following the rules of graph coarsening to acquire a finer mesh, with the features of the coarse mesh assigned to its children vertices. With the final layer of the graph processed, a mesh consisting of 778 vertices is obtained.

$$V_L^0, V_R^0 = f_g\left(F_{img}\right) \quad (2)$$

where F_{img} represents the feature map from Resnet50 encoder, f_g indicates the function that converts feature map into graph structure with fully connected layers and position embedding module.

$$V_L^{i+1}, V_R^{i+1} = G_i\left(\left[V_L^i, V_R^i, I_i\right]\right), i = 0,1,2 \quad (3)$$

where V_L^i and V_R^i denote the hand vertices of the i-th layer of the left and right hand shape regression branch, respectively. I_i is the feature map from the encoder and decoder, G_i indicates the function with Graph convolution, interaction attention, and up-sampling.

2.4. Hand gesture mapping to robot

Upon identifying hand gestures, it becomes imperative to regulate the surgical robot's motion, ensuring the meticulous adjustment of surgical instruments, culminating in the seamless execution of a non-invasive surgical procedure. To maneuver the robot with precision, it is essential to encode the hand's posture and correspondingly map it to appropriate commands.

2.4.1. Encoding

To quantify the position information of keypoints, we initially need to extract appropriate features. We opt to use the Euclidean distance between joints to calculate the distance between each pair of keypoints. By using features with high differentiation, we can accurately represent and distinguish various commands, thereby enhancing the system's reliability. For this purpose, we employ the distance between the fingertip and the root point as features for binary encoding, where a distance greater than a predefined threshold is encoded as 1 and vice versa. The binary encoding principle is illustrated in Eq. 4.

$$B_i = \begin{cases} 1 \text{ if } d_{i-0} \geq threshold \\ 0 \text{ otherwise} \end{cases} \quad (4)$$

where d_{i-0} indicates the distance between the i-th finger's tip and root point, and B_i represents the code for the corresponding finger.

2.4.2. Gesture mapping

To ensure optimal system stability and minimize the risk of accidental touches, a dual-hand posture control method is employed. A set of eight distinctive hand gestures has been carefully selected to showcase this control scheme, as illustrated in Figure 4. Gesture A involves clenching the left hand while extending the forefinger of the right hand. In Gesture B, the right hand is clenched while the left hand extends the forefinger. Gesture C is characterized by an open left hand with the right hand extending the forefinger, while in Gesture D, the right hand is open with the left hand extending the forefinger. Gesture E entails an open left hand while the right hand is clenched, and Gesture F involves an open right hand while the left hand is clenched. In Gesture G, both hands extend their forefingers, whereas in Gesture H, both hands are clenched.

2.4.3. Safety strategies for HRI

Our methodology involves employing a continuous and uninterrupted stream of video frames that are captured by the camera. Relying on a single frame for recognition and command transmission would give rise to ambiguity and instability within the system. To circumvent this, we formulated a simple state machine to

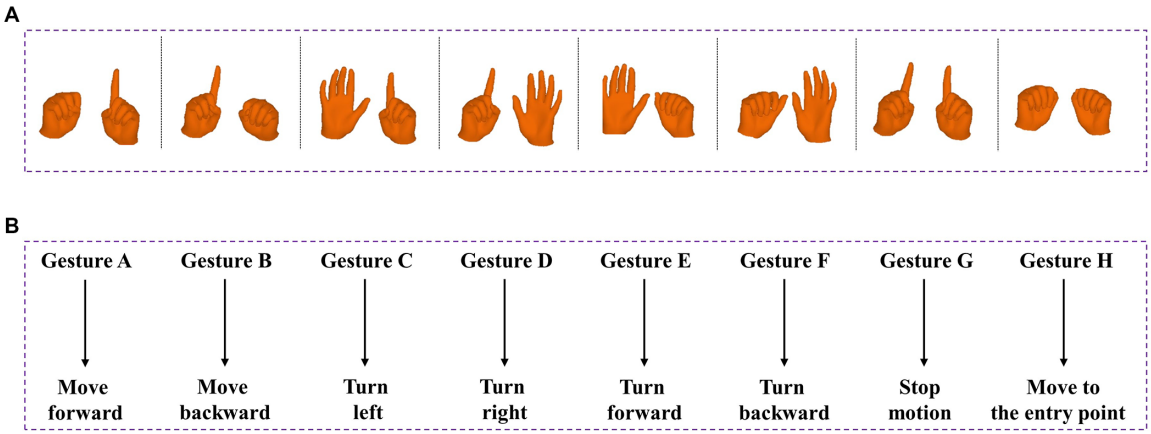


FIGURE 4
Predefined 8 gestures and its corresponding commands in robot. **(A)** From left to right are: Gesture A, Gesture B, Gesture C, Gesture D, Gesture E, Gesture F, Gesture G and Gesture H. **(B)** Gesture-motion corresponding rules.

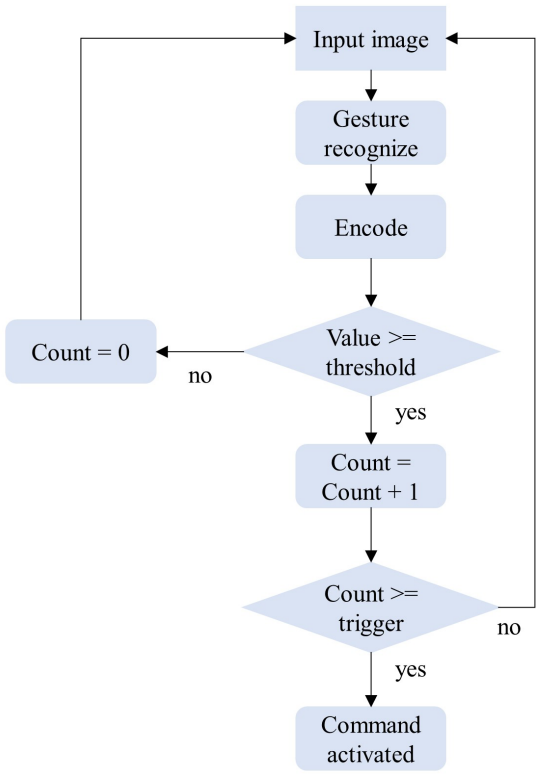


FIGURE 5
State machine design for HRI tasks.

manage and filter the triggers of gestures. As shown in Figure 5, we assigned a distinct counter for each gesture, which increments each time the recognition outcomes correspond to the code of that specific gesture and resets to zero if another gesture appears. To curtail erroneous touches and bolster the robustness of the system, we programmed the counter to activate the corresponding control command when it reaches a specific threshold. Following the triggering of a gesture, the counter does not immediately clear but rather remains at a value above the threshold until a subsequent action clears it.

3. Results

The hand gesture interaction model is implemented on the collaborative robotic arm, Franka Emikia, and its effectiveness is verified through experiments on phantom and cadaver research, following the process outlined in Figure 2.

3.1. Gesture recognition accuracy

In order to apply gesture recognition model to surgical robots, it is necessary to first test the accuracy of recognizing predefined gestures. We evaluated the recognition accuracy and corresponding robot operation effects of eight gestures through two experiments involving 10 volunteers. In the first experiment, each volunteer performed the predefined gestures at different locations, and each gesture was tested five times on the same volunteer. The average recognition accuracy for each gesture is shown in Table 1. It is worth noting that there was one recognition failure in the third and fifth categories, which was due to the fingers being obstructed. In the second experiment, volunteers manipulated the robot through gestures to complete a specified task, aimed at verifying the learning difficulty and efficiency of the gesture interaction. The completion time for the task, which involved touching a specified object, ranged from 1 min 27 s to 2 min 32 s among the 10 volunteers, with an average of 1 min 49 s. These results demonstrate that the gestures we designed for interaction are straightforward and easily learned, and that the corresponding actions of the robot are reasonable.

TABLE 1 Accuracy of eight gesture recognition with predefined categories.

Category	1	2	3	4	5	6	7	8
Accuracy	100%	100%	98%	100%	98%	100%	100%	100%

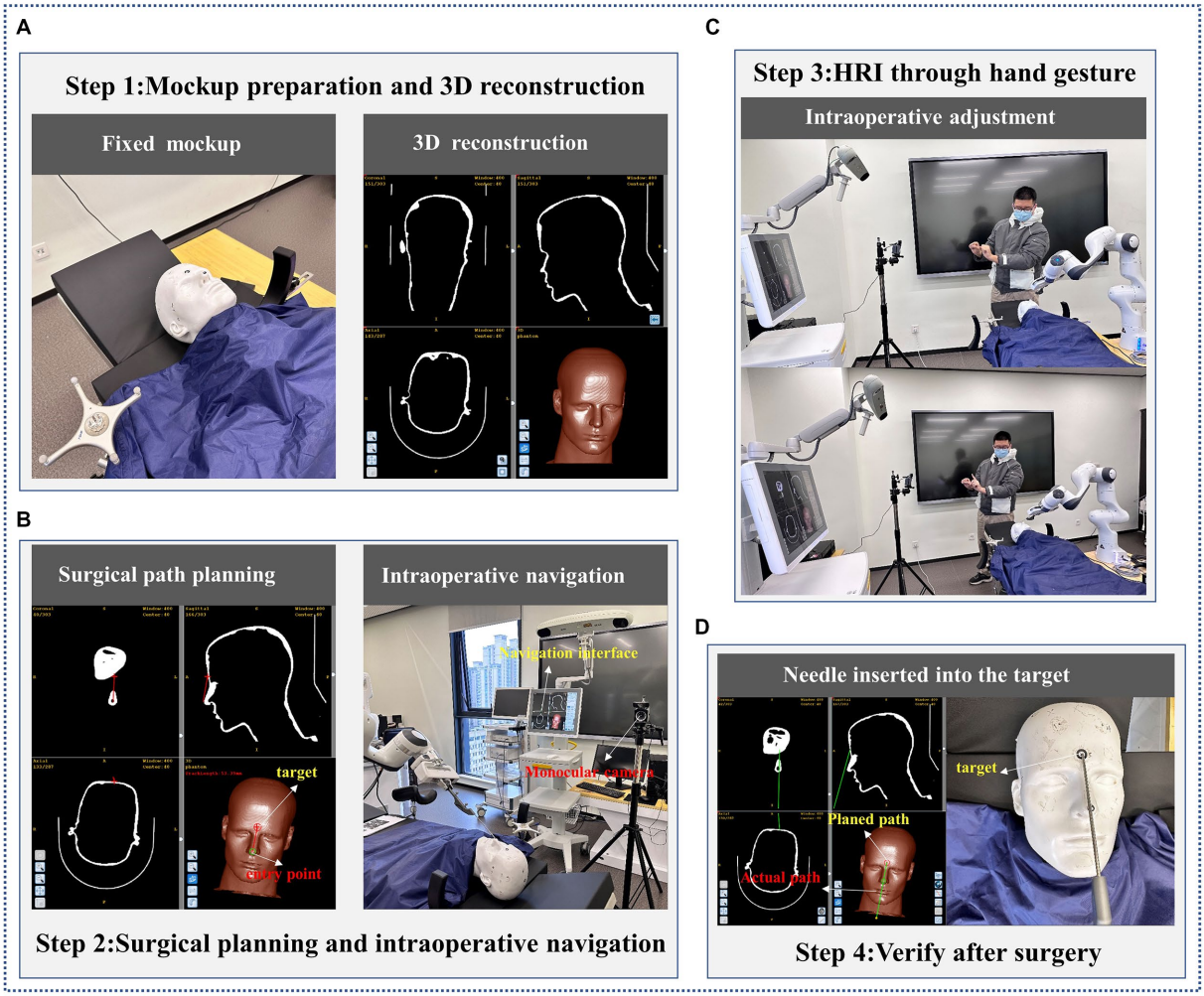


FIGURE 6
The flow of phantom experiment. **(A)** Mockup preparation and 3D reconstruction; **(B)** Surgical planning and intraoperative navigation; **(C)** HRI through hand gesture; **(D)** Verify after surgery.

3.2. Phantom experiment

3.2.1. Experimental settings

The phantom experiment involved the use of a surgical robot to perform expected actions based on human hand gestures on a skull model. A target tumor composed of a metal nail with dimensions of 2×2×2 mm was prepared in the eyebrow center of a phantom to simulate the location of the puncture target. Simultaneously, we located a metal block, measuring 4 mm^3, onto the model's nasal tip to imitate the surgical entry point. The skull model was then subjected to CT scanning and introduced into a surgical navigator to simulate surgical path planning, with the entry point being the tip of the nose and the target location being the eyebrow center.

A surgeon from a hospital participated in the phantom experiment. With the guidance of the surgical navigation system, the needle held by the surgical robot was gradually inserted through touch-free hand gesture interaction with the surgeon. The target and actual path of the needle were derived after insertion into the phantom, and the position error of the needle tip in the preoperative plan was estimated with the help of the surgical system. The workflow of the phantom experiment is illustrated in Figure 6.

3.2.2. Experimental result

Table 2 showcases the mean positioning error of the needle tip and the rotation angle error of the needle. The experimental data reveals that the needle tip's average positioning error is 0.51 mm, and the average angle error is 0.34 degrees. The results obtained from the surgeon's five experiments are outlined in Table 2.

3.3. Cadaver experiment

3.3.1. Experimental settings

In this section, a simulated experiment was conducted to perform a biopsy for nasopharyngeal carcinoma on a cadaver. To create the lesion, we placed a small metal block with a volume of 8 mm^3 at the nasopharyngeal apex of the cadaver head. CT scanning was then performed to obtain medical information with marked points of the cadaver, as illustrated in Figure 7. Using these CT data, we performed 3D reconstruction to create an image-guided space where surgical planning could take place. The needle entry point was located at the top of the anterior nostril, and the target point was at the top of the nasopharynx where the metal block was located.

Similar to the phantom experiment, the surgical robot adjusted its position gradually until the needle tip reached the insertion point of the surgical path, following the hand gesture interactive instructions of the surgeon. With the guidance of surgical navigation and hand gesture interaction, the needle was positioned to align with the planned path and maintained in that posture until it reached the lesion. In the same way, we estimated the errors in the location and angle of the needle tip.

3.3.2. Experimental result

Figure 8 illustrates the outcomes of the simulated biopsy for nasopharyngeal carcinoma. The red and green lines displayed in the figure indicate the intended surgical path and the real needle location, respectively. In Figure 8A, the path of the simulated biopsy is demonstrated. Figure 8B exhibits that the needle approached the entry point with a posture that is in agreement with the planned path. Ultimately, Figure 8C portrays the outcome of the needle insertion into the simulated lesion. As can be seen from Figure 8, the actual needle position nearly matches the planned surgical path. By fitting the needle information in the surgical navigator, we obtained an actual location error of 0.16 mm and an angle error of 0.10 degrees.

TABLE 2 Error of the needle insertion in phantom experiment.

Number of experiments	Needle	
	Position error (mm)	Angle error (°)
1	0.40	0.14
2	0.54	0.22
3	0.67	0.57
4	0.63	0.41
5	0.32	0.36

4. Discussion

In this study, we presented a novel framework for recognizing gestures using monocular color images, achieving an accuracy rate of over 98% in recognizing all predefined gestures. Compared with traditional manual procedures, the proposed framework for gesture recognition facilitates efficient contactless interaction between surgeons and surgical robot in RAMIS, thereby mitigating the risk of bacterial transmission and enhancing surgical efficacy by enabling precise fine-tuning of related instruments.

Both phantom experiments and cadaver studies were successfully conducted to provide proof of concept for the contactless HRI to assist in RAMIS, and it is evident that sub-millimeter precision was achieved after implementing the trials with hand gesture interaction. We suggest two potential rationales for the positive results observed in our experimentation. The first evidence of the enhanced precision in hand gesture recognition is derived from the auxiliary prediction branch, which significantly contributes to the extraction of both 3D and 2D hand features. Another possible explanation could be that each adjustment of the robot we designed is considerably subtle, especially in terms of its ability to make adjustment for orientation. This, in turn, increases the possibility of accurate movement of the surgical robot according to the intended surgical plan. Moreover, the surgical robot can be configured with high efficiency, and the HRI interface exhibits a shallow learning curve (the average learning time of only 1 min and 49 s) in a simulated task, thereby results in no significant increase in surgery preparation time. It was proved that with the aid of hand

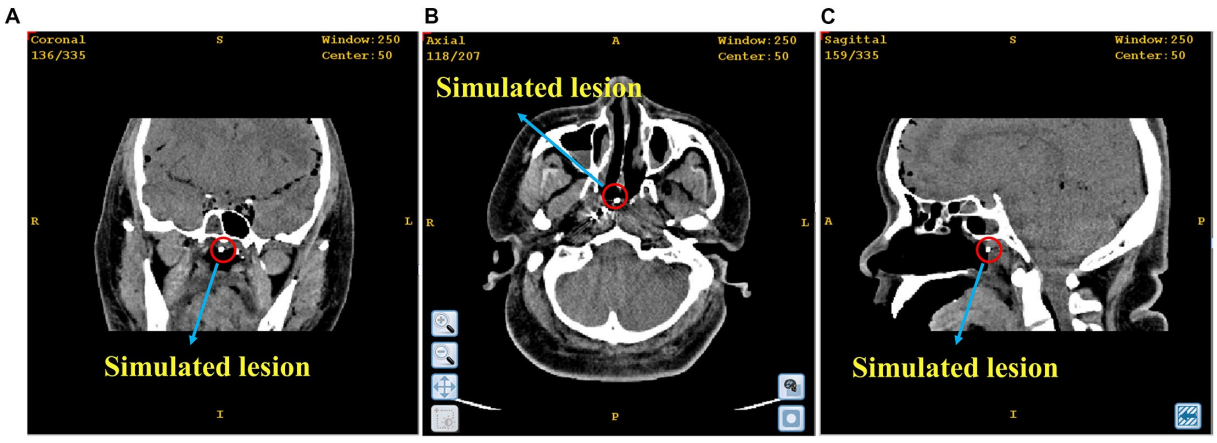


FIGURE 7
The position of the simulated lesion in nasopharyngeal apex. **(A)** Coronal plane; **(B)** Cross section; **(C)** Sagittal plane.

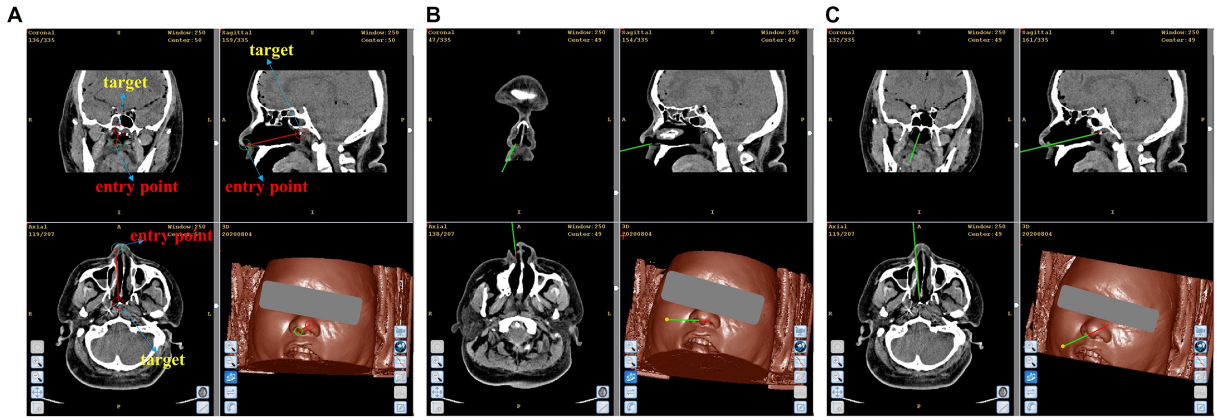


FIGURE 8
Simulated biopsy results. **(A)** Simulated surgical path. **(B)** The needle reaches the top of the anterior nostril. **(C)** The needle reaches the simulated lesion.

gesture interaction, the surgeon can effortlessly fine-tune surgical instruments without physical contact for a majority of the time.

Although the current version of the non-contact collaborative surgical navigation robot system has showed favorable outcomes, it is not without its limitations. One example of this is the limitation faced by the surgeons in adjusting the surgical instruments, as they can merely adjust them through one gesture at a time. This results in an increase of the motion steps and a decrease in task efficiency. We think this can be effectively solved by designing more gestures. Additionally, the complex environment of the operating room with a multitude of medical instruments and limited space may lead to restricted image acquisition and occasional hand occlusion, resulting in recognition failures. Hand occlusion is also the reason for the two cases of recognition failure in Table 1. Furthermore, the proposed state machine may cause discomfort as it necessitates maintaining a gesture for a period of time, while the designed model lacks recognition of dynamic gestures, limiting the surgeon's control over the surgical robots through dynamic gestures.

At present, the available data from phantom and cadaver cases is sufficient to establish the feasibility of the touchless HRI interface for RAMIS. We believe that our work will be regarded as the fundamental basis of touchless surgical robot HRI, and it has been preliminary substantiated by both phantom and cadaveric investigations. The findings have indicated the efficacy of the design of the collaborative system in aiding other surgical procedures involved RAMIS and demand stringent sterility standards. We believe that the framework we have established will form a practical system and be applied in clinic.

5. Conclusion

The feasibility and validity of the framework we proposed in this paper are verified through the experiments on both phantoms and cadavers. The experimental findings evince the surgical robot's ability of fine-tuning instruments through augmented visual feedback from the navigation surgical system and contactless hand gesture recognition, thus by minimizing bacterial, the surgical safety can be enhanced. At the same time, the framework is easily integrated into a real surgical robot. The future works should endeavor the study of surgical robot application utilizing mixed reality technology that integrates touch-free solutions and the development of more dynamic hand gestures to augment the integration flexibility.

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 authors.

Author contributions

JW organized and performed experiments, wrote and revised the manuscript. XZ proposed and did the research and designed the model. XC and ZS conceived the initial idea, supervised the work, and revised the manuscript. All authors contributed to the article and approved the submitted version.

Funding

This work was supported by the National Natural Science Foundation of China (Grant No. 62076070) and the Local Cooperation Project of artificial Intelligence Medical Hospital in Xuhui District, Shanghai (Grant No. 2021–008).

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

Ahmad, A.-M., Al Majzoub, R., and Charanek, O. (2016). "3D gesture-based control system using processing open source software." in: 2016 2nd International Conference on Open Source Software Computing (OSSCOM): IEEE, 1–6.

Ahmidi, N., Tao, L., Sefati, S., Gao, Y., Lea, C., Haro, B. B., et al. (2017). A dataset and benchmarks for segmentation and recognition of gestures in robotic surgery. *IEEE Trans. Biomed. Eng.* 64, 2025–2041. doi: 10.1109/TBME.2016.2647680

Cao, H., Chen, G., Li, Z., Feng, O., Lin, J., and Knoll, A. (2022). Efficient Grasp Detection Network With Gaussian-Based Grasp Representation for Robotic Manipulation. *IEEE/ASME Trans. Mechatron.: IEEE*, 1–11. doi: 10.1109/TMECH.2022.3224314

Chen, X., Yang, F., Zhang, Z., Bai, B., and Guo, L. (2021). Robust surface-matching registration based on the structure information for image-guided neurosurgery system. *J. Mech. Med. Biol.* 21:2140009. doi: 10.1142/S0219519421400091

Chen, F., Zhong, Q., Cannella, F., Sekiyama, K., and Fukuda, T. (2015). Hand gesture modeling and recognition for human and robot interactive assembly using hidden markov models. *Int. J. Adv. Robot. Syst.* 12:48. doi: 10.5772/60044

Cho, Y., Lee, A., Park, J., Ko, B., and Kim, N. (2018). Enhancement of gesture recognition for contactless interface using a personalized classifier in the operating room. *Comput. Methods Prog. Biomed.* 161, 39–44. doi: 10.1016/j.cmpb.2018.04.003

Despinoy, F., Zemiti, N., Forestier, G., Sánchez, A., Jannin, P., and Poignet, P. (2018). Evaluation of contactless human–machine interface for robotic surgical training. *Int. J. Comput. Assist. Radiol. Surg.* 13, 13–24. doi: 10.1007/s11548-017-1666-6

Díaz, I., Gil, J. J., and Louredo, M. (2014). A haptic pedal for surgery assistance. *Comput. Methods Prog. Biomed.* 116, 97–104. doi: 10.1016/j.cmpb.2013.10.010

Elbadawy, M., Elons, A., Shedeed, H. A., and Tolba, M. (2017). "Arabic sign language recognition with 3D convolutional neural networks." in: 2017 Eighth International Conference on Intelligent Computing and Information Systems (ICICIS): IEEE, 66–71.

Fang, W., Ding, Y., Zhang, F., and Sheng, J. (2019). Gesture recognition based on CNN and DCGAN for calculation and text output. *IEEE Access* 7, 28230–28237. doi: 10.1109/ACCESS.2019.2901930

Haidegger, T., Speidel, S., Stoyanov, D., and Satava, R. M. (2022). Robot-assisted minimally invasive surgery—surgical robotics in the data age. *Proc. IEEE* 110, 835–846. doi: 10.1109/JPROC.2022.3180350

Huang, D.-Y., Hu, W.-C., and Chang, S.-H. (2011). Gabor filter-based hand-pose angle estimation for hand gesture recognition under varying illumination. *Expert Syst. Appl.* 38, 6031–6042. doi: 10.1016/j.eswa.2010.11.016

Huynhnguyen, H., and Buy, U. A. (2021). "Toward gesture recognition in robot-assisted surgical procedures." in: 2020 2nd International Conference on Societal Automation (SA): IEEE, 1–4.

Jin, H., Chen, Q., Chen, Z., Hu, Y., and Zhang, J. (2016). Multi-LeapMotion sensor based demonstration for robotic refine tabletop object manipulation task. *CAAI Trans. Intell. Technol.* 1, 104–113. doi: 10.1016/j.trit.2016.03.010

Liu, Y., Song, Z., and Wang, M. (2017). A new robust markerless method for automatic image-to-patient registration in image-guided neurosurgery system. *Comput. Assist. Surg.* 22, 319–325. doi: 10.1080/24699322.2017.1389411

Mavroudi, E., Bhaskara, D., Sefati, S., Ali, H., and Vidal, R. (2018). "End-to-end fine-grained action segmentation and recognition using conditional random field models and discriminative sparse coding." in: 2018 IEEE Winter conference on applications of computer vision (WACV): IEEE, 1558–1567.

Nagyné Elek, R., and Haidegger, T. (2019). Robot-assisted minimally invasive surgical skill assessment—manual and automated platforms. *Acta Polytech. Hungarica* 16, 141–169. doi: 10.12700/APH.16.8.2019.8.9

Nestorov, N., Hughes, P., Healy, N., Sheehy, N., and O'hare, N. (2016). "Application of natural user interface devices for touch-free control of radiological images during surgery." in: 2016 IEEE 29th international symposium on computer-based medical systems (CBMS): IEEE, 229–234.

Ohmura, Y., Nakagawa, M., Suzuki, H., Kotani, K., and Teramoto, A. (2018). Feasibility and usefulness of a joystick-guided robotic scope holder (Soloassist) in laparoscopic surgery. *Visc. Med.* 34, 37–44. doi: 10.1159/000485524

Oyedotun, O. K., and Khashman, A. (2017). Deep learning in vision-based static hand gesture recognition. *Neural Comput. Applic.* 28, 3941–3951. doi: 10.1007/s00521-016-2294-8

Reiley, C. E., Lin, H. C., Varadarajan, B., Vagvolgyi, B., Khudanpur, S., Yuh, D. D., et al. (2008). Automatic recognition of surgical motions using statistical modeling for capturing variability. *Stud. Health Technol. Inform.* 132, 396–401.

Shi, X., Jin, Y., Dou, Q., Qin, J., and Heng, P.-A. (2021). "Domain adaptive robotic gesture recognition with unsupervised kinematic-visual data alignment." in: 2021 IEEE/RSJ international conference on intelligent robots and systems (IROS): IEEE, 9453–9460.

Tarvekar, M. P. (2018). "Hand gesture recognition system for touch-less car interface using multiclass support vector machine." in: 2018 second international conference on intelligent computing and control systems (ICICCS): IEEE, 1929–1932.

Van Amsterdam, B., Clarkson, M. J., and Stoyanov, D. (2021). Gesture recognition in robotic surgery: a review. *IEEE Trans. Biomed. Eng.* 68, 2021–2035. doi: 10.1109/TBME.2021.3054828

frontiers | Frontiers in Neuroscience

TYPE Original Research
PUBLISHED 29 June 2023
DOI 10.3389/fnins.2023.1213176

Check for updates

OPEN ACCESS

EDITED BY
Alois C. Knoll,
Technical University of Munich, Germany

REVIEWED BY
Longbin Zhang,
Royal Institute of Technology, Sweden
Jie Li,
Chongqing Technology and Business University, China
Xiaoyu Wu,
National University of Singapore, Singapore

*CORRESPONDENCE
Juan Zhang
✉ 17904127@qq.com

RECEIVED 27 April 2023
ACCEPTED 08 June 2023
PUBLISHED 29 June 2023

CITATION
Jiang H, Ma L, Wang X, Zhang J, Liu Y, Wang D, Wu P and Han W (2023) Focus prediction of medical microscopic images based on Lightweight Densely Connected with Squeeze-and-Excitation Network.
Front. Neurosci. 17:1213176.
doi: 10.3389/fnins.2023.1213176

COPYRIGHT
© 2023 Jiang, Ma, Wang, Zhang, Liu, Wang, Wu and Han. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

Focus prediction of medical microscopic images based on Lightweight Densely Connected with Squeeze-and-Excitation Network

Hesong Jiang[1], Li Ma[1], Xueyuan Wang[1], Juan Zhang[1]*, Yueyue Liu[2], Dan Wang[1], Peihong Wu[1] and Wanfen Han[1]

[1]School of Information Engineering, Southwest University of Science and Technology, Mianyang, China, [2]The School of Internet of Things Engineering, Institute of Automation, Jiangnan University, Wuxi, China

Due to the demand for sample observation, optical microscopy has become an essential tool in the fields of biology and medicine. In addition, it is impossible to maintain the living sample in focus over long-time observation. Rapid focus prediction which involves moving a microscope stage along a vertical axis to find an optimal focus position, is a critical step for high-quality microscopic imaging of specimens. Current focus prediction algorithms, which are time-consuming, cannot support high frame rate imaging of dynamic living samples, and may introduce phototoxicity and photobleaching on the samples. In this paper, we propose Lightweight Densely Connected with Squeeze-and-Excitation Network (LDSE-NET). The results of the focusing algorithm are demonstrated on a public dataset and a self-built dataset. A complete evaluation system was constructed to compare and analyze the effectiveness of LDSE-NET, BotNet, and ResNet50 models in multi-region and multi-multiplier prediction. Experimental results show that LDSE-NET is reduced to 1E-05 of the root mean square error. The accuracy of the predicted focal length of the image is increased by 1~2 times. Training time is reduced by 33.3%. Moreover, the volume of the model only reaches the KB level, which has the characteristics of being lightweight.

KEYWORDS

focus prediction, deep learning, medical microscopy, DenseNet, squeeze and excitation

1. Introduction

Nowadays, microscopy is still the most frequently used microscopic detection technology for examining thin sections and stained tissue sections on slides, playing an irreplaceable role in biomedicine, materials chemistry, industrial inspection, and other aspects (Ikeda et al., 2009; Zhang et al., 2014; Carrera et al., 2017). When the microscope is used for imaging living cells, defocus blur may occur due to thermal fluctuation of the microscope body and the movement of the microscope sample (Kreft et al., 2005). In addition, motion blur will also occur due to the uneven morphology of samples (Xu and Jia, 2010). Defocus blur and motion blur, as two of the most common microscopic imaging artifacts, can seriously degrade the imaging quality of digital pathology instruments (Redondo et al., 2012). Thus, maintaining the internal focal position of the microscope is a challenge. And when faced with a large number of samples, a large sample area, and a long observation time, manual focusing is impractical (Wang et al.,

2018; Pinkard et al., 2019a). Therefore, autofocusing is crucial for high-precision microscope imaging.

The earliest research on autofocusing technique can be traced back to 1898 (Haosheng and Yu, 2021), but it was not until the 1960s that autofocusing technique was first used in the photographic system (Qiumei, 2006). Traditional autofocusing techniques almost use active focusing methods based on range finding, through the sensor to measure the distance to achieve (Lichang, 2015; Chen et al., 2020). With the gradual development of precision instruments toward intelligence and automation, higher requirements have been put forward for microscopes (Meng et al., 2022). Hence, a micro autofocusing technique based on digital image processing has gradually gained the attention of researchers (Kui, 2018). Image processing-based autofocusing methods are mainly divided into depth from defocus and depth from focus (Yunhao, 2019).

Depth from defocus was first proposed by Pentland in 1987, to obtain depth information from the defocused images and use optical principles to calculate the focal distance, to achieve the purpose of autofocusing (Pentland, 1987). Although depth from defocus processes fewer images and has a faster-focusing speed (Gao and Ge, 2014), the focusing accuracy depends on the establishment of a correct focusing mathematical model (Rui, 2021), which can only be estimated theoretically at present, it is not completely accurate and just approach to idealization infinitely, which result in larger error effect (Meng, 2005). Depth from focus does not need to establish the mathematical model of the imaging system in advance, it is a method of focusing search process (Shiyun, 2022), whose core is focusing search algorithm and definition evaluation function (Yuhu et al., 2013). However, it still does not equip with good adaptability, and cannot get accurate definition evaluation on some collected images with multi-noise (Yu and Lu, 2022). Meanwhile, depth from focus algorithm needs to acquire and process a series of data that image from clear to fuzzy, which takes much time (Yipeng et al., 2005) and cannot satisfy both focusing accuracy and real-time at the same time, unable to coordinate the two to a favorable standard (Fan, 2021).

In recent years, with the rapid development of computer technology, deep learning has also ushered in explosive growth (Wang et al., 2016; Cao et al., 2021; Hu et al., 2022), and has achieved a good application prospect in computer vision tasks (Cao et al., 2022a) such as image classification (Cao et al., 2020; Cheng et al., 2021; Hussain et al., 2021; Safari et al., 2021; Hang et al., 2022), and object detection (Ranjan et al., 2018; Hassaballah et al., 2021; Cao et al., 2022b). By extracting the image deep feature information, and predicting information within a very short period, can greatly improve the validity and accuracy of the detection results. Therefore, the use of deep learning techniques for microscopic imaging autofocusing has become a focused research of biomedical microscopic images in recent years.

In 2018, Jiang et al. (2018) explored the application of deep convolution neural networks (CNNs) for microscope autofocusing. They used the trained model to predict the focal position of the acquired image without axial scanning, which significantly improved the autofocusing speed of the microscope and avoided the defects associated with autofocusing algorithm. In the following year, Pinkard et al. (2019b) designed a fully connected Fourier neural network based on coherent illumination, which uses an additional non-axial illumination source to predict the single image focus and emphasizes the generalization Capability between sample types. Dastidar (2019) First improved on input dataset by no longer acquiring multiple images in the vertical direction and maximizing the image sharpness to achieve autofocusing, instead, the difference image of two defocus images with a fixed spacing of 2 μm as inputs for deep convolution networks (CNN) to predict the optimal distance to be moved, to achieve the best focus relative for current position. In 2021, Luo proposed an autofocusing method (deep-R) based on deep learning. The network blindly and automatically outputs the focused image by training the sample microscopic image obtained at any defocus plane (Luo et al., 2020).

In the same year, Li et al. (2021) proposed a deep learning-based autofocusing framework that estimates the position of the focal plane of the objective lens relative to the plate by receiving two defocus images acquired by the fluorescence microscope of the plate, providing a deterministic measure in the prediction. Therefore, image blocks that may contain background or low-contrast objects can be excluded, improving accuracy. However, organisms have unique forms and characteristics, which may make microscopic images too different. The method proposed by Li needs to rely on a relatively large dataset to fit the ideal model, otherwise, the predictive performance of unseen samples will be reduced, the network generalization capability is weak, and the efficient prediction of multi-domain, multi-rate microscopic defocus images cannot be realized.

To accurately predict the focal length of defocus images, this paper proposes a deep learning network architecture with lightweight, faster computing speed, wider prediction area, and stronger generalization ability, while considering both efficiency and accuracy. The implementation of the method is described in detail from the construction of the dataset, model construction, and training method. A complete evaluation system is constructed, comparing and analyzing the performance gap of this network and other network models such as ResNet50 and BotNet. Finally, summarized and analyzed the important results of the experiment.

2. Construction of test dataset

2.1. Test facility and data acquisition

The dataset for this experiment consists of two parts, one using the open source dataset, and the other part was observed using the ML-31-M biomicroscope equipped with a 10X/22 large field of view eyepiece as standard (Provided by Guangzhou Mingmei Technology). Under the lighting conditions of the LED coherent illumination that comes with the device, an MD50-T microscope digital camera with 2.2 μm × 2.2 μm image element size and 5 megapixels was used to acquire an effective pixel high-resolution image of 2,592 × 1,944 size.

Figure 1 shows the schematic diagram of the ML-31-M biological microscope, where part A is the MD50-T microscope camera with a resolution of 5 megapixels, which can provide a frame rate of 14fps in full pixel mode. Part B is an adjustable large field of view WF10X/22 mm double-headed eyepiece. C is a four-hole converter equipped with four infinity distance flat-field achromatic objectives of 10X/0.10, 20X/0.25, 40X/0.65, and 100X/1.25. The ML-31-M used is a binocular microscope with two fluoroscopic systems. The imaging principle is based on binocular stereo vision, where different parts of the objective are observed through different eyepieces and the images are subsequently combined through brain vision processes. As shown in Figure 2A, the sample slide forms an inverted real image by the

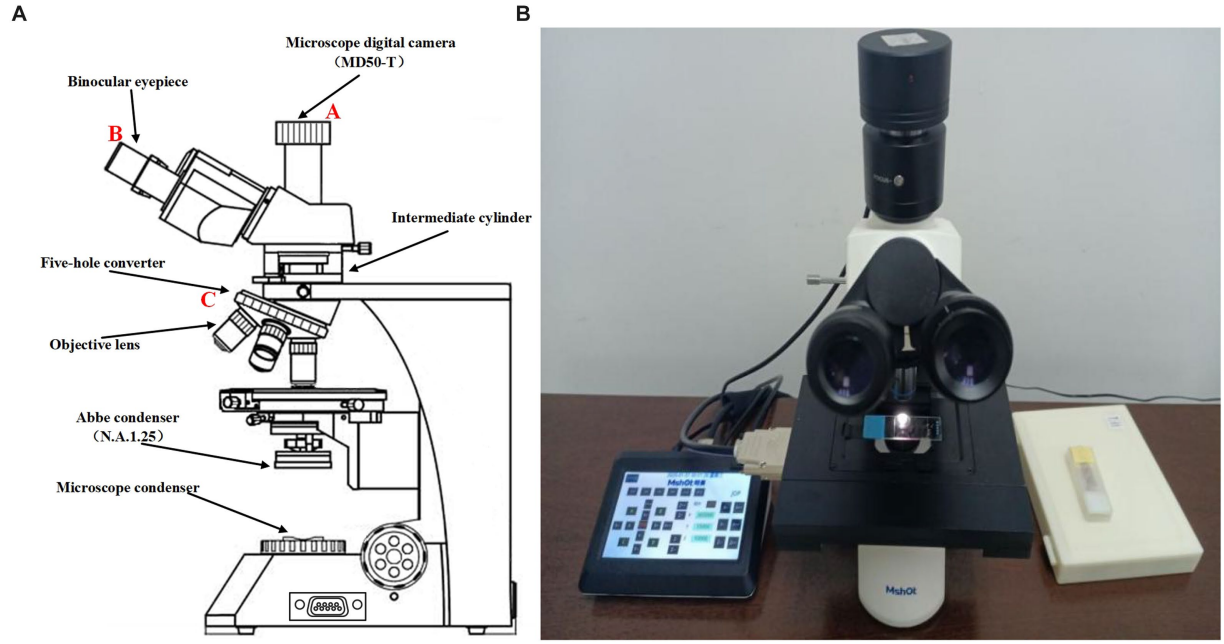


FIGURE 1
(A) Sketch of ML-31-M biological microscope. **(B)** Real picture of ML-31-M biological microscope.

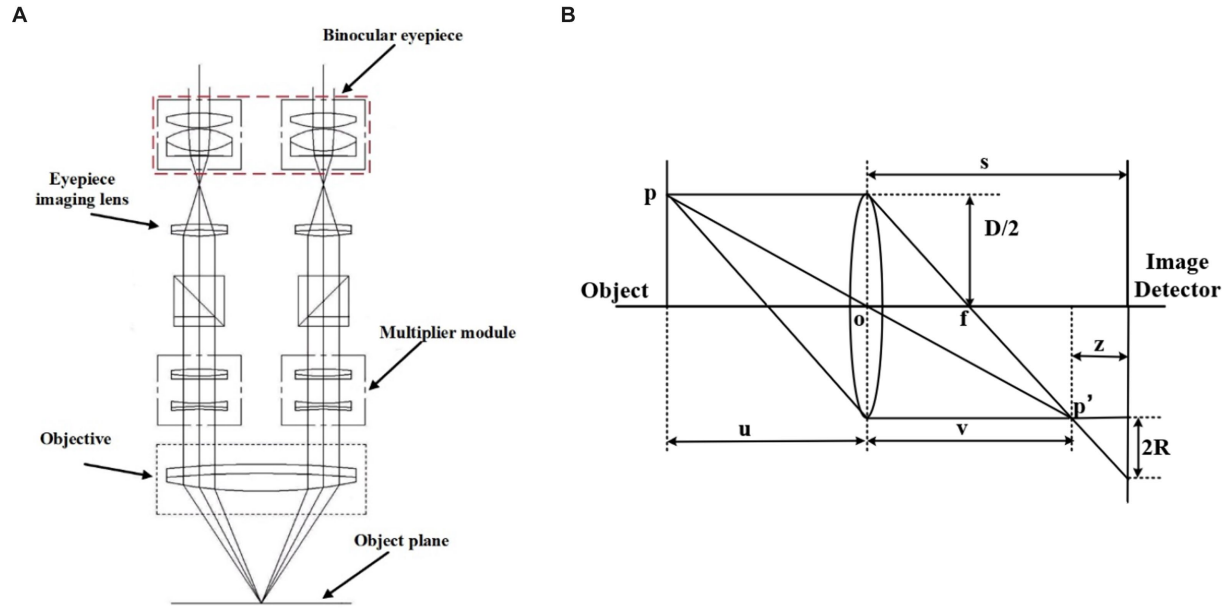


FIGURE 2
(A,B) are the principle diagram. **(A)** Binocular microscope imaging principle diagram (Wu et al., 2020). **(B)** Microscope imaging principle diagram.

magnification of the objective lens, and the light rays are secondarily magnified by the multiplier module, and then the light rays are cast down to the eyepiece imaging lens for convergence, and finally enter the eyepiece to form a magnified orthogonal virtual image to be observed.

The microscope imaging process is based on the imaging principle of the convex lens, and the schematic diagram is shown in Figure 2B. P is the observation point of the sample, s is the distance from the center of the convex lens to the image detector, D is the diameter of the lens aperture, and R is the radius of the blurred image point where p' falls on the image detector. The relationship between focal length f, object distance u, and image distance v satisfies Gauss's formula:

$$\frac{1}{u}+\frac{1}{v}=\frac{1}{f} \quad (1)$$

When a clear flat image is observed, the viewing surface at this point is the focal plane of the system. But in defocus plane, will form a fuzzy image point on the observation surface, the radius of the image point R can characterize the degree of focus of the image, that the value of z in the figure is greater, the image is more away from the focal plane, the image point fuzzy circle is larger, the relationship holds:

$$\frac{R}{D/2}=\frac{z}{v}=s\left(\frac{1}{v}-\frac{1}{s}\right) \tag{2}$$

From the above two equations, we can obtain:

$$R=s\frac{D}{2}\left(\frac{1}{f}-\frac{1}{u}-\frac{1}{s}\right) \tag{3}$$

Clear imaging by changing the value of u\s\f so that the image plane is located at the focal plane.

Figure 3 shows the microscopic imaging of the same centroid in the tumor cells depicting the imaging situation at different focal planes. When z = 0um, microscopic imaging is in the plane of focus when the image clarity is the highest. Subsequently, the defocus plane image is acquired by moving up and down a certain step, and the *z*-value is the distance the objective lens is moved with respect to the plane of focus. It is clear from the microscopic images that the further away from the defocus plane to the focal plane, the lower the sharpness of the image.

2.2. Building of the dataset

High-resolution microscopic images located in the focus can clearly observe the morphology and structure of the sample. However, it is impractical to achieve manual focusing in the face of a large number of data samples, so there is an urgent need to develop a method that enables accurate prediction of the focal length to achieve high-resolution autofocusing. To achieve the accuracy of the model, it is not enough that only use the public dataset. For this reason, the experiments in this paper use a two-part dataset.

One part is a self-built dataset, that using the ML-31-M biomicroscope to collect. The process starts with the initial focusing of the sample, the next fine-tuning of the focus to achieve optimal definition, and move the sample to different defocus positions ranging from −10 μm to +10 μm in steps of 0.5 μm to obtain defocus images. As shown in Figure 4. The above steps were repeated for the entire sample in 1 mm lateral steps, and a total of 20 sets of data were collected, each containing approximately 40 images. Finally, the images and the corresponding focus position information were saved, and the defocus image under two magnifications of 20X and 40X were acquired by same method.

Another part is the public dataset (Jiang et al., 2018), micrographs were observed with an Eclipse electron microscope (provided by Nikon Eclipse) at 0.75 NA, 20X lens, which was obtained from a 5-megapixel color camera (Pointgrey BFS-U3-51S5C-C) with a 3.45 μm pixel size. Keeping the defocus distance range from +10 μm to −10 μm, 40 defocus stacks with 0.5 μm step spacing were captured in the same field of view, totaling 40 × 40 images in the same field of view, and the obtained images were segmented into approximately 130,000 images of size 224 × 224 for network training. This is shown in Figure 4.

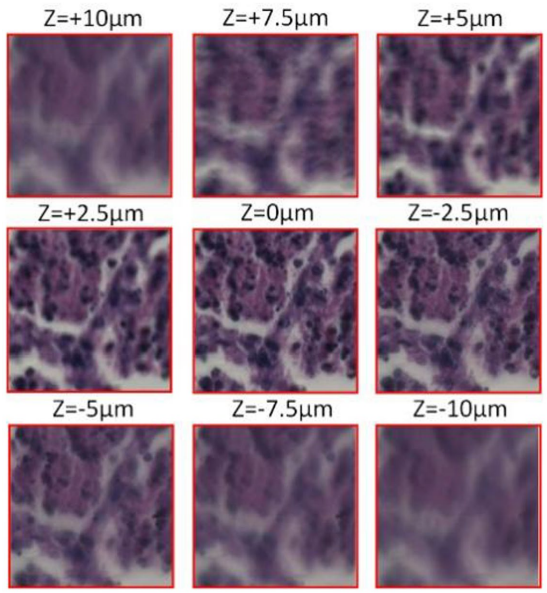


FIGURE 3
Cell micrograph (observed with a 20X Eclipse motorized microscope, acquired by Pointgrey BFS-U3-51S5C-C camera).

3. Model structure

3.1. Method overview

Deep convolutional neural networks have been widely used in image classification and processing in recent years. In this paper, we use the collected defocus image data combined with convolutional neural networks to construct an end-to-end model to predict the focal length of an image, and maximize the requirements for high-accuracy prediction under multiple regions. The model is as follows:

$$D^p=F\left(S_k,\delta\right) \tag{4}$$

S_k denotes a 224 × 224 size defocus cell image, D^p is the predicted focal length obtained after training the network model, and F is the regression function obtained after training, and δ is a set of network learning parameters including learning rate, number of iterations, etc. In the training process, by feeding a large number of dataset consisting of defocus images into the network, the training is continuously iterated to obtain the optimal parameters δ of the model, the gap between the predicted focal length D^p and the real focal length D^t is minimized, which makes the problem transformed into:

$$\delta_m=\arg\min L_\delta\left(D^t,D^p\right) \tag{5}$$

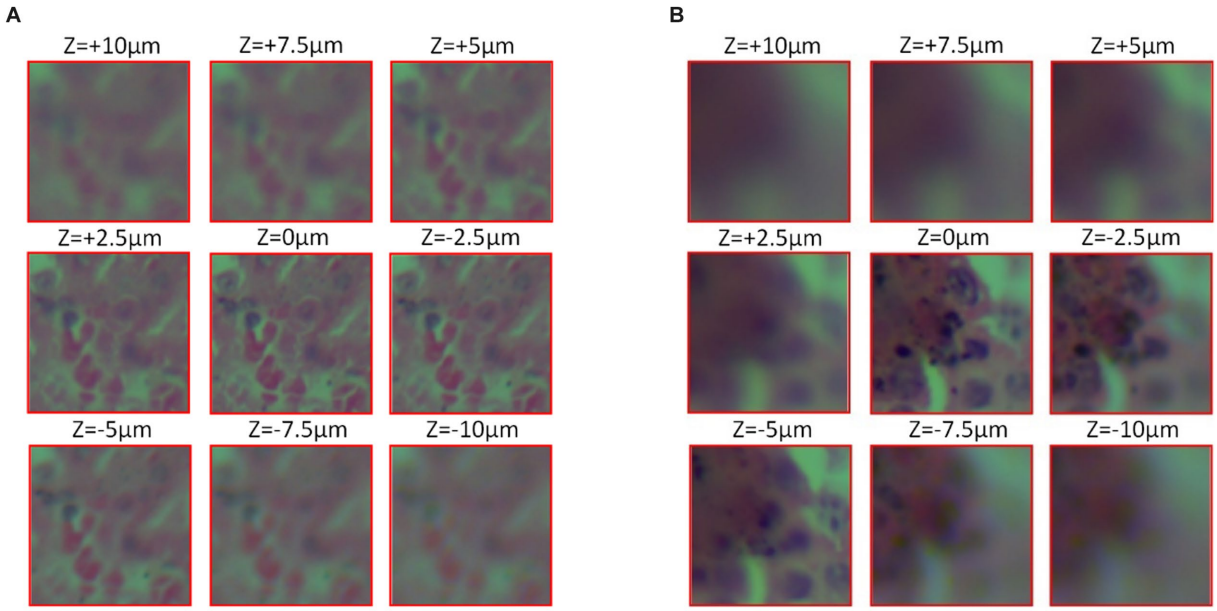


FIGURE 4
Cell micrograph (observed with the ML-31-M biological microscope, acquired by the MD50-T microscope camera). **(A)** 20X objective observation **(B)** 40x objective observation.

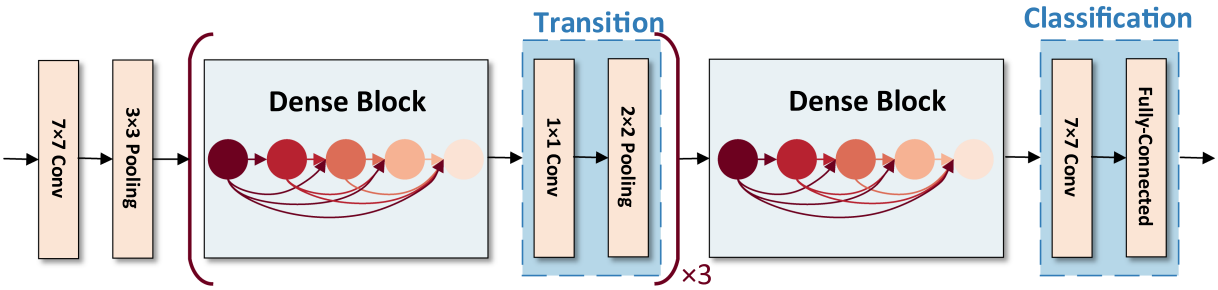


FIGURE 5
Network structure of DenseNet.

For some wide-field high-resolution images, in order to get the focal length more accurately and quickly, it needs to be partitioned into small images of 224×224 in height and width for prediction respectively, and the results will be averaged so that the original model will be transformed into:

$$D^p = avg(\sigma(F(s^1{}_k, \delta) + F(s^2{}_k, \delta) + .. + F(s^h{}_k, \delta)) \quad (6)$$

D^p is the focal length of the predicted wide-field and high-resolution image, $avg(\ )$ is the averaging function, $s^1{}_k, s^2{}_k \supset$ is the same wide-field high-resolution image split into different 224×224 small images, and σ is the discriminant function, Because in the process of segmenting the wide-field high-resolution image into small images, a part of the image will include most of the blank area, resulting in unreliable prediction results obtained from this part of area, which needs to be discarded.

3.2. The proposed network structure

For the autofocusing of wide-field and high-resolution microscopic images, this paper proposes an LDSE-NET automatic focal distance prediction deep learning framework, using DenseNet as the main framework of the model in the network. Since 2015 He et al. (2016) proposed ResNet for the problems of vanishing gradient, explosion gradient and performance degeneracy that occur with deeper network layer structures, and the performance of deeper networks can be further improved by jumping connections between shallow and deep networks, weakening the strong connections between each layer. However, due to the large number of layers built by ResNet, more computational resources and time are required. So Huang et al. (2016) further improved the feature reuse capability based on ResNet and proposed DenseNet with dense connection operation.

Figure 5 the input of each layer of the network and the output of all previous layers of the network, which mainly focuses on improving the network performance from the perspective of feature reuse, enhancing

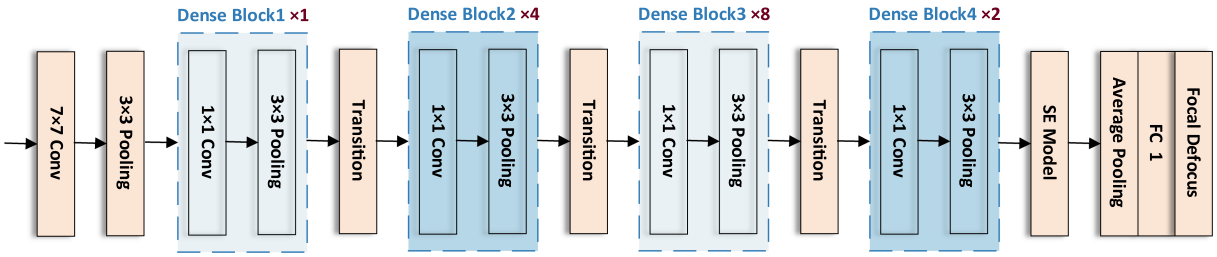


FIGURE 6
Network structure of LDSE-NET.

the feature propagation, and improving the efficiency of information and gradient transmission in the network. The network contains three layers structure of Dense block layer, Transition layer, and Classification layer. Where Dense block layer consists of a composite with BN, ReLU, and Conv nonlinear mapping functions, designed with a pre-activation strategy to make network training easier and generalization performance better; Conv represents the convolution layer in the deep neural network, which undertakes convolution calculation in the process of model reasoning. Transition layer is used for the connection between dense blocks and contains 1 × 1Conv, 2 × 2 Average pooling; Classification layer consists of Global average pooling and Fully connected layer, the input of each layer of the network includes the output of all previous layers of the network. Compared with ResNet50, this network mainly focuses on improving the network performance from the perspective of feature reuse, enhancing feature propagation, and improving the efficiency of information and gradient transmission in the network.

In addition, to emphasize the information feature channel, better adapt to the dataset, and further improve the prediction performance, the network architecture is adjusted and optimized in this paper. The number of Convolutional layers in the Dense Block is reduced, and some of the activation function in it are replaced with Tanh, making the network structure simpler and more efficient. On this basis, the SE module is connected after the last Dense Block to improve the accuracy of the image focal length prediction task to a certain extent. In this paper, the model is completed by sequentially superimposing the dense block, transition block, and squeeze excitation module. As shown in Figure 6.

The input of this CNN network structure is an unfocused blurred image captured by a microscope. The image is first passed through a Convolutional layer of a 7 × 7 matrix with a step size of 2 and a padding of 3 and then passed through a 3 × 3 maximum pooling layer with a step size of 2. The output is passed through the constructed Dense block layer in turn, compressing the Dense block layer input and all the extracted feature information with the help of Transition blocks, changing the size of the channels' number so that the number of channels between adjacent dense blocks can correspond to each other, further enhancing the feature propagation between each layer, and the output is passed through the SE module to extract more feature information. Finally, the output is sent to the 7 × 7 Global average pooling layer and the Fully connected layer. The output of the network is a Regression layer, and the result is the predicted sample focal length.

3.2.1. Dense block layer

Dense block layer is an important part of LDSE-NET, which is used to further improve the effectiveness of information transfer between each layer, and the specific propagation formula is as follows:

$$X_L = H_L\left(X_0, X_1, \ldots, X_{L-1}\right) \tag{7}$$

$\left[X_0, X_1, \ldots, X_{L-1}\right]$ refers to the concatenation of the feature-maps produced in layer $0, 1, \ldots, L-1$, and $\left[H_L\right]$ is defined as a composite function of three consecutive operations consisting of normalization function, activation function, and convolution function, and the input of each layer is the output of the mapping results of all previous layers, and also the feature mapping result of the current layer is used as the input of the later layers, and the structure is shown in Figure 7.

3.2.2. Transition layer

The above-mentioned Dense block layer equation only works if the feature map is the same size, so the Transition layer needs to be used to do pooling and convolution to change the size of the feature map. So that the size of the feature map output from the Dense block layer is consistent with the shape size of the input of the next layer. The structure of the Transition layer used in this network is shown below, consisting of the BN layer of normalization function, Tanh of activation function, Conv of 1 × 1, and Average pooling of 2 × 2, and the structure is shown in Figure 8.

3.2.3. Squeeze-excitation module

The SE module improves the representativeness of the network by enabling it to perform dynamic channel-wise feature recalibration (Jie et al., 2018). This structure consists of Global average pooling layer, Fully connected layer, and linear activation function. The feature outputs of LDSE-NET are used as input to the SE module to increase the sensitivity to useful feature information. It learns the global information by fusing the convolutional features of each channel and filters out the less useful feature information to improve the expressiveness of the model. This is shown in Figure 9.

3.2.4. Squeeze

Squeeze works by compressing the global spatial information into a single channel using Global average pooling. In principle, the channel statistics Z is achieved by reducing the spatial U dimension height and width, which can be summarized by the equation:

$$Z_m = F_{sq}\left(u_m\right) = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} u_m\left(i, j\right) \tag{8}$$

3.2.5. Excitation

This module is designed to take advantage of the global information obtained by compression and aims to fully capture channel-wise

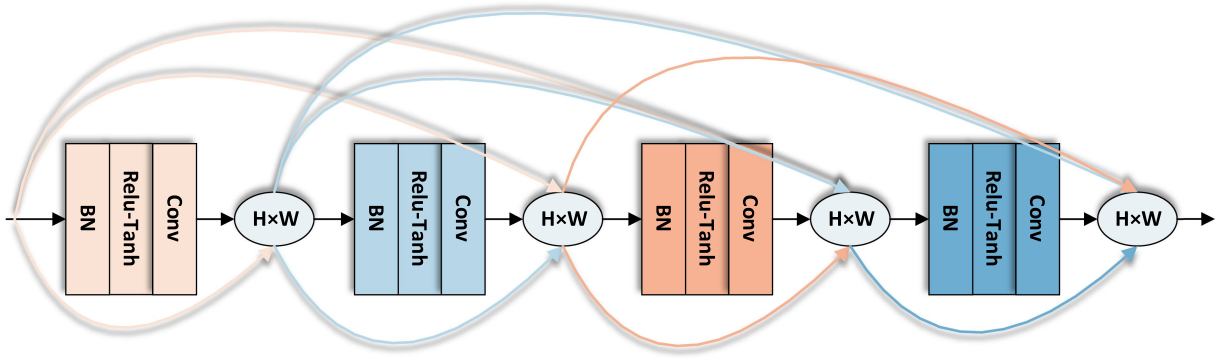


FIGURE 7
Dense block layer of LDSE-NET.

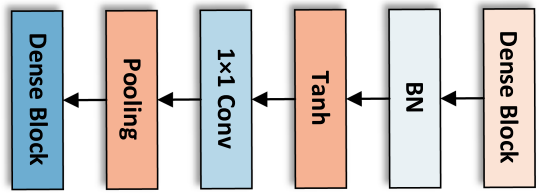


FIGURE 8
The transition layer of LDSE-NET.

dependencies (Jie et al., 2018), and consists of two Fully connected layers and an activation function, as shown in Figure 9A. In order to better adapt to the data set of this experiment, the sigmoid activation function is changed to the Tanh activation function, so the excitation operation S can be summarized by the formula:

$$S = F_{es}(z,W) = \sigma\left(g(z,W)\right) = \sigma\left(W_2\delta\left(W_1 z\right)\right) \tag{9}$$

where $W_1 \in R^{\frac{C}{r}\times C}$, $W_2 \in R^{C\times\frac{C}{r}}$, $\delta(x) = \max(0,x)$ representing the ReLU activation function alleviates the vanishing gradient problem, and compared with Sigmoid activation function, $\sigma(x) = \frac{1-e^{-2x}}{1+e^{-2x}}$ improves the convergence speed, z is the channel information collected by the above Squeeze operation. The final output $\tilde{X}_m \in R^{H\times W}$ is obtained by multiplying the channels between the scalar s_m and the feature map u_m. This can be written as

$$\tilde{X}_m = F_{scale}\left(u_m, s_m\right) = s_m \bullet u_m \tag{10}$$

4. Experiment results and analysis

4.1. Model training

The experimental training process in this paper was run on a desktop computer with an NVIDIA GeForce RTX 3080 graphics card, an Inter Core i5-12600KF CPU and 32 GB of RAM. After some small sample tests, the parameters of the LDSE-NET were determined. Mean square error (MSE) was used as the model loss function, defined as:

$$Loss = MSE\left(D^t, D^p\right) = \frac{1}{n}\sum_{i=1}^{n}\left(D_i^t - D_i^p\right)^2 \tag{11}$$

In the above equation, $MSE(\ )$ represents the root mean square error function, D^t represents the true focal length in the dataset, D^p represents the result predicted by the network, and n is the number of samples. The training optimizer uses Adam deep learning optimization algorithm, sets the network learning rate to 0.001, and uses the lr_scheduler mechanism to adjust the learning rate at certain epoch intervals to achieve a better training effect. The batch size is set to 50 images, and the training is stopped when the loss values of the test set and training set tend to stabilize and do not decline. Using RGB channels images from the public dataset, dividing the dataset RGB Channels images into training set and test set in a 9:1 ratio, and there is no intersection between them. To verify the performance of the network, this experiment compares the network structures of ResNet50 and BotNet (Bottleneck Transformer Network) and obtains the experimental results of each network structure separately.

4.2. Prediction results and analysis

According to the above indexes for training, the results of the prediction accuracy changes are shown in Figures 10A–C, the LDSE-NET has a small oscillation range of the loss values of the training set and the test set throughout the training process, and after about 50 epochs, the loss values of both the training set and the test set fluctuated within 0.005, and there was a significant decline in the training process, with the final model loss value stabilizing around 1E-05. On the contrary, the other two networks showed larger fluctuations in the loss values during the training process. The BotNet test set loss value fluctuated sharply between 0.01 and 0.02 and could not decrease; when ResNet50 had a sharp increase in error after training to a certain epoch, followed by a dramatically decrease, and the loss value could not be stable. The final model loss values of both networks can only drop to around 1E-04, and the training effect is poor compared to the LDSE-NET.

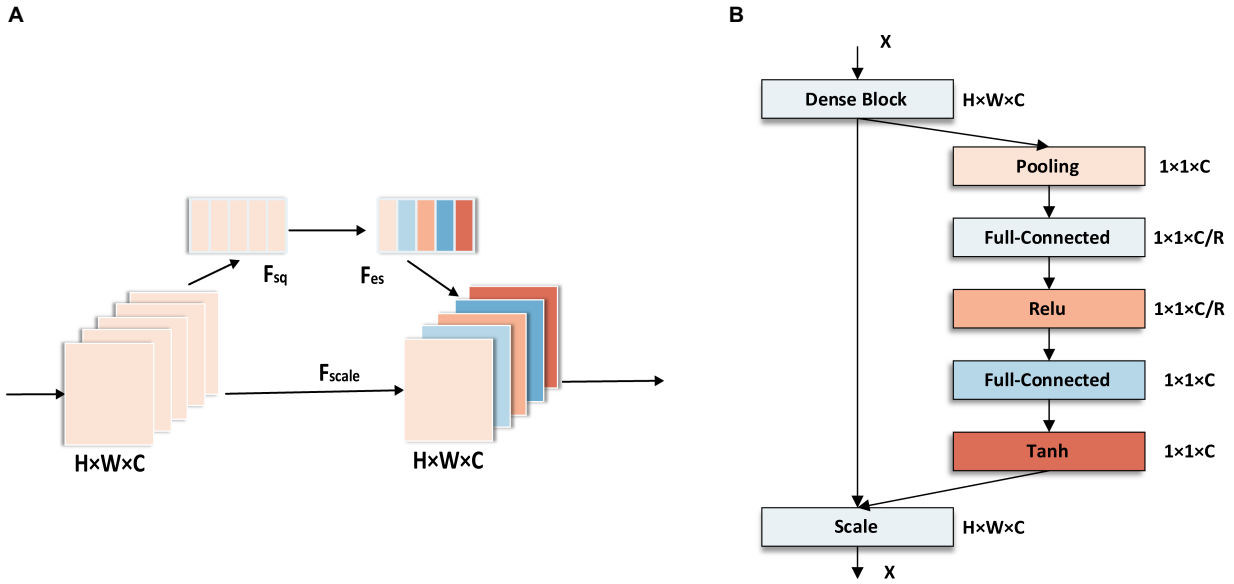


FIGURE 9
(A) Squeeze-Excitation module overall structure diagram. **(B)** Detail connection diagram.

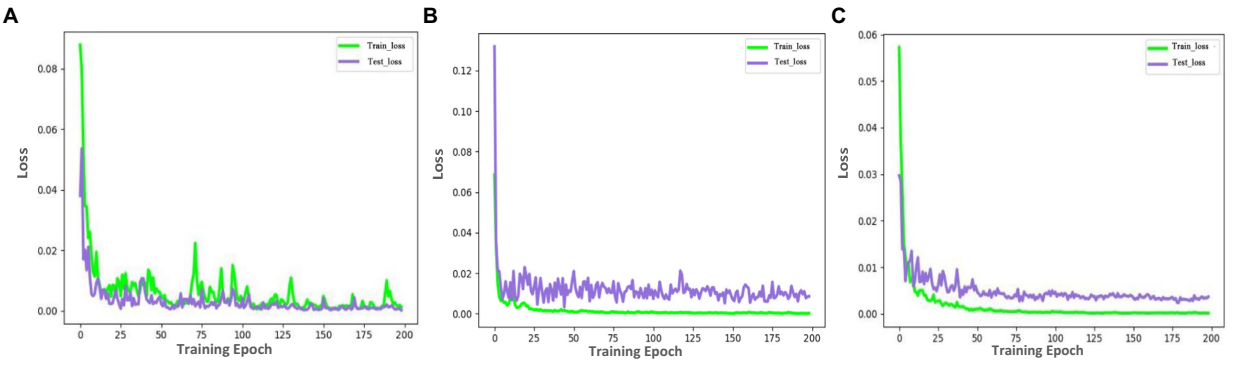


FIGURE 10
(A–C) represents train_loss and test_loss for ResNet50, BotNet, LDSE-NET three different networks.

Besides, this paper compares three models in multiple sets of renal sample images of the focal length prediction results, and selects the error result distribution of three sets of data as shown in Figures 11A–C. From the figure, the three networks have little difference in the prediction effect in the range of −10 μm to −5 μm, because defocus images blur to a large degree and contain fewer image features, making each network have the same effect. In the interval of −5 μm ~ +5 μm, by evaluating the prediction error distribution, it can be found that most of the error distribution of LDSE-NET model is within +250 nm ~ −250 nm. In contrast, the prediction error of the ResNet50 model is mostly distributed beyond 500 nm. Overall, compared with BotNet and ResNet50 networks, the prediction accuracy of LDSE-NET network is improved by 1 and 2.5 times, respectively.

In addition to the improvement in prediction accuracy, the purpose of this paper is to increase the computational speed of the model as well as to make it more lightweight. The specific comparison results are shown in Table 1. All three networks were trained by the same hardware device, and when the model training was completed, the training time of LDSE-NET was about 6 h, and the speed was improved by 20% compared to BotNet and 33.3% compared to Resnet50. The model size is simpler and lighter than the other two networks, with only about 12% of the size of the two networks.

4.3. Comparison of the predicted effect of variable magnification, variable area

To further evaluate the performance of the network. In this paper, we also use the 20 sets of data collected above containing a total of about 110,000 images of size 224 × 224 for training and testing, which are also divided into training set and test set in the ratio of 9:1, with no intersection between the two sets of data. The network models were trained according to the above-mentioned network parameter metrics to obtain the network models under 20X lens and 40X lens, respectively, and used to predict the focal length of defocus images under different magnifications.

As shown in Figure 12, the predicted images were first divided into nine regions of 3 × 3, which do not have overlapping parts, and

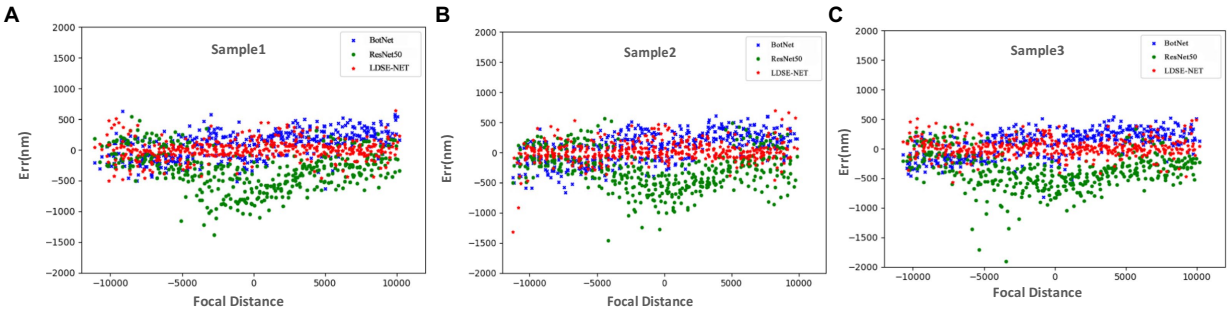


FIGURE 11
(A–C) Prediction error distribution of the three networks for sample 1, sample 2, and sample 3.

TABLE 1 Performance comparison of ResNet50, BotNet, and LDSE-NET.

NET	Focusing error (nm)	Model size	Final loss	Train time
ResNet50	0.5 ± 0.32	4.002 MB	3.89E-04	9 h
BotNet	0.38 ± 0.29	3.701 MB	2.04E-04	7.5 h
LDSE-NET	0.15 ± 0.17	480 KB	7.65E-05	6 h

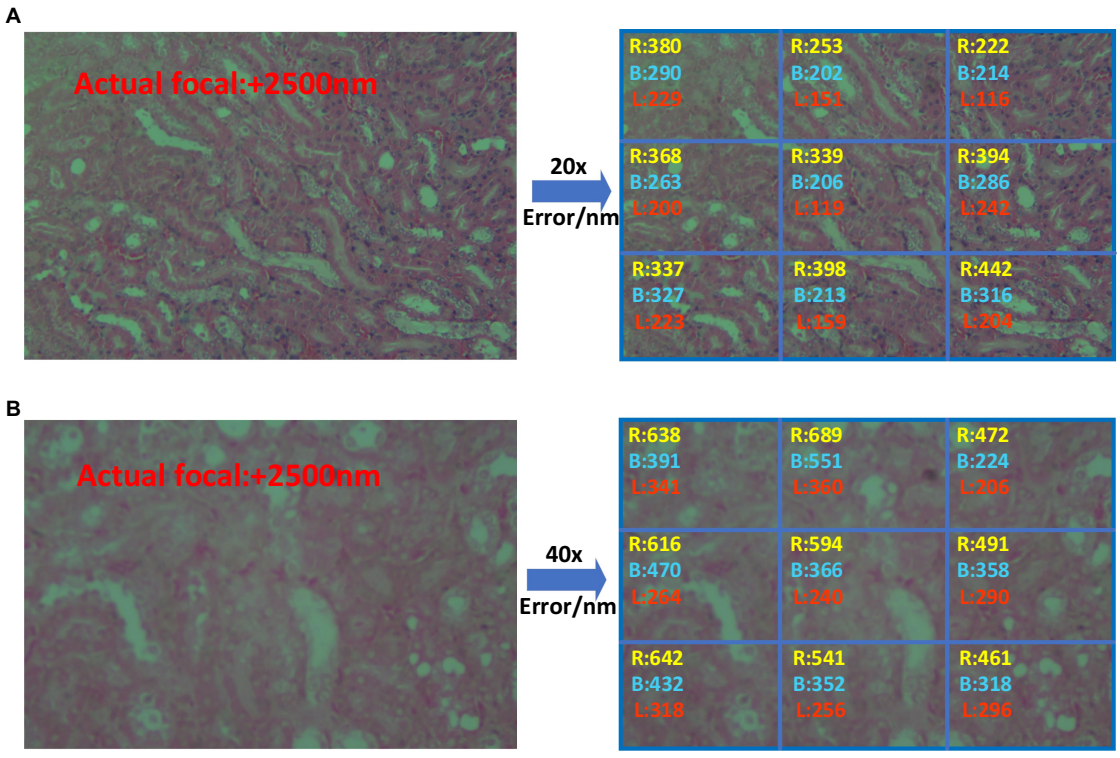


FIGURE 12
R, B, and L represent ResNet50, BotNet, and LDSE-NET three different networks. **(A)** Prediction error of 20X lens **(B)** Prediction error of 40X lens.

the focal length prediction was performed for these regions separately. Comparing the prediction results under the two magnifications, it can be seen that the prediction effect of 20X is better than 40X, and the error is reduced by about 100 nm ~ 200 nm. This is due to the fact that the field of view under the 40X lens is narrower and contains fewer cells, and the edge position becomes more blurred compared to the 20X lens, which makes each image may contain many blank areas after cutting into small images, resulting in its feature information is more blurred and sparse, which makes the prediction focal error increase. In addition, the prediction results of the network for the middle of the image are better than the edge locations, which is most likely because the entire field of view is too large for the microscope head and camera

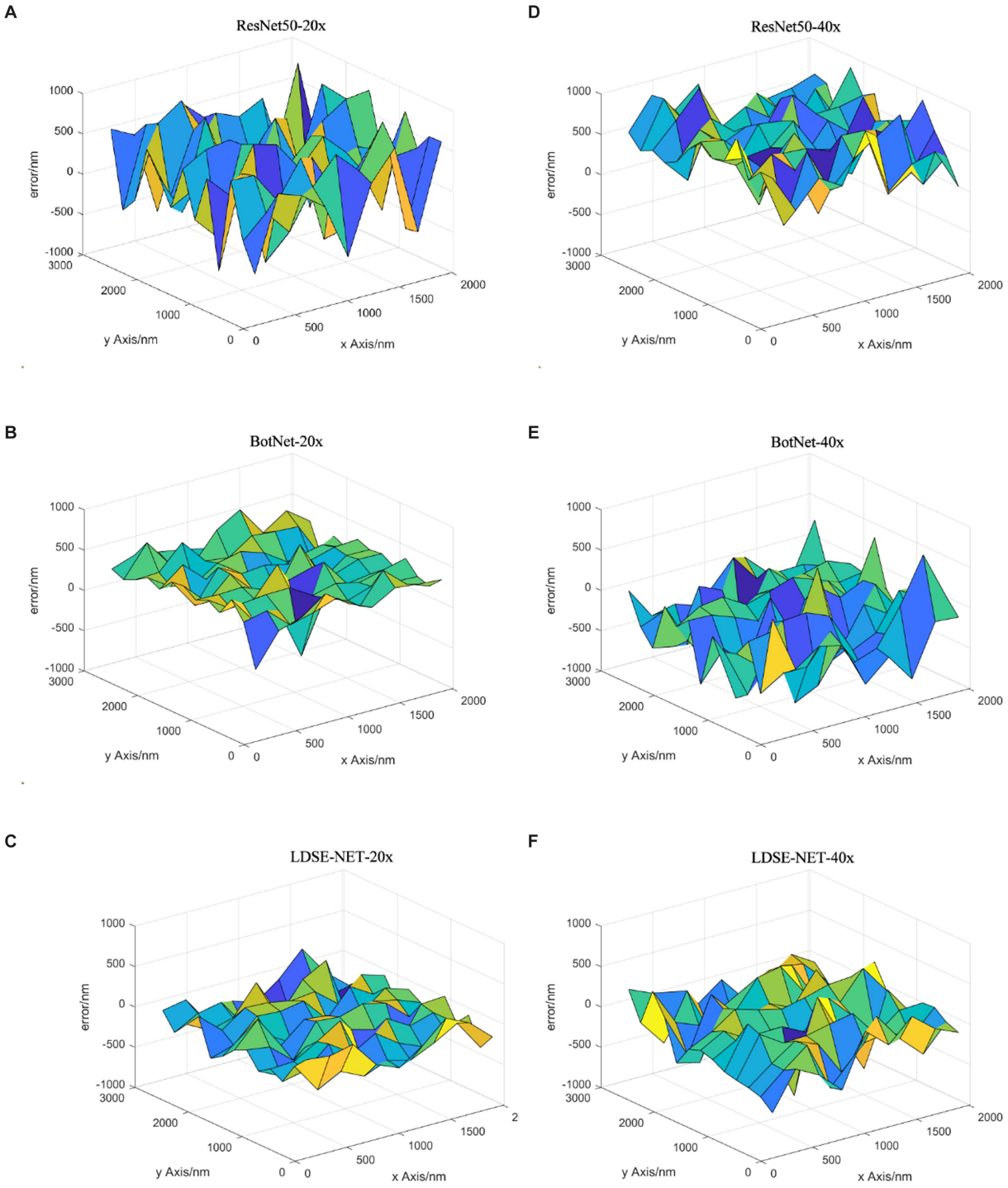


FIGURE 13
Focal length error of 20X, 40X images predicted by different networks.

to focus over the entire field of view, resulting in the possible existence of more blurred locations on the edges. Therefore, during the training and prediction process, the image in the center of the field of view can be selected, and more accurate results will be obtained.

Secondly, Figure 13 shows the prediction focal error plots of each of the three networks for the same high-resolution defocus image with a large field of view at different magnifications. Combining the results of this experiment, it can be seen that for 20X magnification images, the prediction error of ResNet50 and BotNet are mostly above 300 nm. On the contrary, most of the prediction error of LDSE-NET remain below 300 nm. Similarly, from the 40X magnification error map distribution, it can be seen that more than 60% of regions of ResNet50 and BotNet have error over 500 nm, while the average error of LDSE-NET is controlled

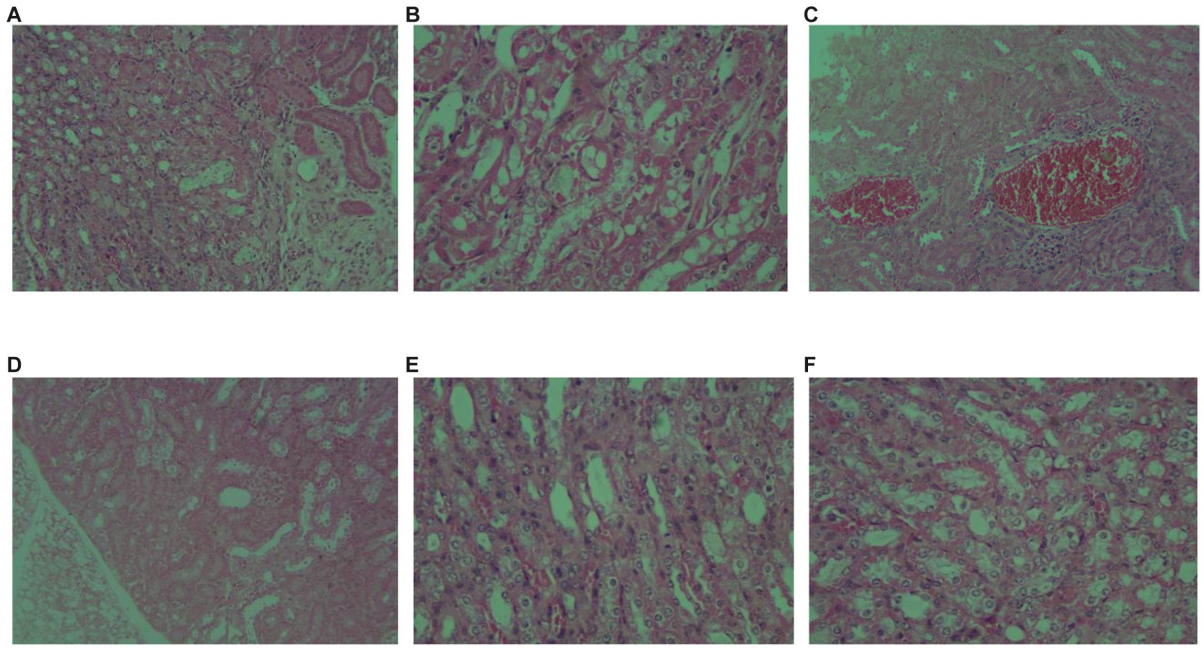


FIGURE 14
(A–F) represent picture1~picture6 test sample images respective.

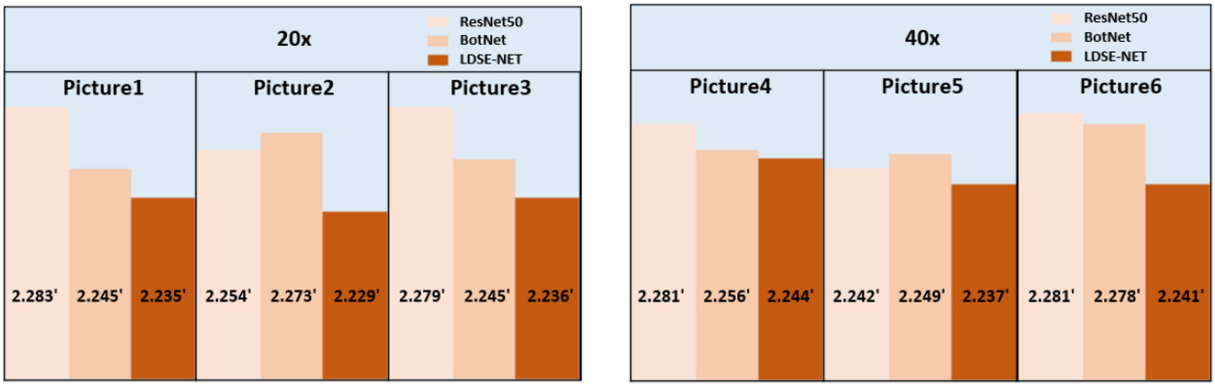


FIGURE 15
Comparison of the time required to predict a single image by different network predictions.

around 300 nm. Therefore, the error of LDSE-NET is significantly smaller than the other two networks for both 20X and 40X magnification data, and the accuracy of some areas is improved by 1 ~ 2 times compared to BotNet and ResNet50.

In addition, to satisfy the requirement for error reduction, the computational speedup is also an important purpose. Here, three models are utilized to directly predict the single image focal length, as shown in Figure 14, large-field high-resolution images of different regions of the same sample are selected, all experiments are conducted on the same computational platform and obtain the running time. The specific comparison effect is shown in Figure 15, in terms of time efficiency comparison, the computation time of LDSE-NET network is improved by 0.02 s ~ 0.04 s. Combined with the above experimental contents, this shows that this network is better than ResNet50 and BotNet in terms of accuracy and time.

5. Conclusion

In this paper, we present a dense model LDSE-NET with squeeze excitation for predicting the focal length of the defocus images under the medical microscope. Its effectiveness in focal length prediction is verified by using multi-region and multi-magnification image data. Through the evaluation of the prediction results in the test set, compared with the other two networks of BotNet and

ResNet50, the accuracy of the image focal length prediction of LDSE-NET is improved and the model proposed is lighter. This network reduces the information loss, improves the transmission efficiency of information in the network, and further proves the feasibility and practicability of deep learning in the prediction of focal length of microscopic imaging on the basis of previous studies, and provides ideas for future research.

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

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

Funding

This study was supported by Natural Science Foundation of Sichuan Province, Grant/Award Number: 23NSFSC1257.

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

Cao, H., Chen, G., Li, Z., Feng, Q., Lin, J., and Knoll, A. (2022a). Efficient grasp detection network with Gaussian-based grasp representation for robotic manipulation. *IEEE/ASME Trans. Mechatron.*, 1–11. doi: 10.1109/TMECH.2022.3224314

Cao, H., Chen, G., Li, Z., Hu, Y., and Knoll, A. (2022b). NeuroGrasp: multimodal neural network with Euler region regression for neuromorphic vision-based grasp pose estimation. *IEEE Trans. Instrum. Meas.* 71, 1–11. doi: 10.1109/TIM.2022.3179469

Cao, R., Fang, L., Lu, T., and He, N. (2020). Self-attention-based deep feature fusion for remote sensing scene classification. *IEEE Geosci. Remote Sens. Lett.* 18, 43–47. doi: 10.1109/LGRS.2020.2968550

Cao, H., Wang, Y., Chen, J., Jiang, D., Zhang, X., Tian, Q., et al. (2021). Swin-Unet: Unet-like pure transformer for medical image segmentation. *arXiv*. doi: 10.48550/arXiv.2105.05537

Carrera, D., Manganin, F., Boracchi, G., and Lanzarone, E. (2017). Defect detection in SEM images of Nanofibrous materials. *IEEE Trans. Indust. Inform.* 13, 551–561. doi: 10.1109/tii.2016.2641472

Chen, G., Cao, H., Conradt, J., Tang, H., Rohrbein, F., and Knoll, A. (2020). Event-based neuromorphic vision for autonomous driving: a paradigm shift for bio-inspired visual sensing and perception. *IEEE Signal Process. Mag.* 37, 34–49. doi: 10.1109/MSP.2020.2985815

Cheng, G., Si, Y., Hong, H., Yao, X., and Guo, L. (2021). Cross-scale feature fusion for object detection in optical remote sensing images. *IEEE Geosci. Remote Sens. Lett.* 18, 431–435. doi: 10.1109/LGRS.2020.2975541

Dastidar, T. R. (2019). Automated focus distance estimation for digital microscopy using deep convolutional neural networks[C]// computer vision and pattern recognition. *IEEE*, 1049–1056. doi: 10.1109/CVPRW.2019.00137

Fan, H. E. (2021) *Research on fast focusing algorithm based on image blur prediction*. Chongqing: Chongqing University of Posts and Telecommunications.

Gao, F., and Ge, F. (2014). Automatic focusing system of morphological detection microscope based on image recognition. *Enterprise Technol. Dev.* 33:2. doi: 10.3969/j.issn.1006-8937.2014.01.019

Hang, R., Li, Z., Liu, Q., Ghamisi, P., and Bhattacharyya, S. S. (2022). Hyperspectral image classification with attention-aided CNNs. *IEEE Trans. Geosci. Remote Sens.* 59, 2281–2293. doi: 10.1109/TGRS.2020.3007921

Haosheng, X., and Yu, F. (2021). Automatic focusing algorithm of digital microscope. *Laser Optoelectron. Progress* 58:8. doi: 10.3788/LOP202158.0400002

Hassaballah, M., Kenk, M. A., Muhammad, K., and Minaee, S. (2021). Vehicle detection and tracking in adverse weather using a deep learning framework. *IEEE Trans. Intell. Transp. Syst.* 22, 4230–4242. doi: 10.1109/TITS.2020.3014013

He, K., Zhang, X., Ren, S., Sun, J. (2016) *Deep residual learning for image recognition[C]//IEEE conference on computer vision and pattern recognition*. IEEE Conference on Computer Vision and Pattern Recognition: IEEE. 770–778. doi: 10.1109/CVPR.2016.90

Hu, Y. B., Chen, G., Li, Z., and Knoll, A. (2022). Robot policy improvement with natural evolution strategies for stable nonlinear dynamical system. *IEEE Trans. Cybernet.* 53, 4002–4014. doi: 10.1109/TCYB.2022.3192049

Huang, G., Liu, Z., Maaten, L. V. D., Weinberger, K. Q. (2016). Densely connected convolutional networks. *IEEE Comput. Soc.* doi: 10.1109/CVPR.2017.243

Hussain, E., Hasan, M., Rahman, M. A., Lee, I., Tamanna, T., and Parvez, M. Z. (2021). CoroDet: a deep learning based classification for COVID-19 detection using chest X-ray images. *Chaos Solitons Fractals* 142, 110495. doi: 10.1016/j.chaos.2020.110495

Ikeda, K., Muto, S., Tatsumi, K., Menjo, M., Kato, S., Bielmann, M., et al. (2009). Dehydriding reaction of AlH3:in situmicroscopic observations combined with thermal and surface analyses. *Nanotechnology* 20:204004. doi: 10.1088/0957-4484/20/20/204004

Jiang, S., Liao, J., Bian, Z., Guo, K., Zhang, Y., and Zheng, G. (2018). Transform- and multi-domain deep learning for single-frame rapid autofocusing in whole slide imaging. *Biomed. Opt. Express* 9, 1601–1612. doi: 10.1364/BOE.9.001601

Jie, H., Li, S., and Gang, S. (2018) *Squeeze-and-excitation networks[C]//IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, 2011–2023. doi: 10.1109/TPAMI.2019.2913372

Kreft, M., Stenovec, M., and Zorec, R. (2005). Focus-drift correction in time-lapse confocal imaging. *Ann. N. Y. Acad. Sci.* 1048, 321–330. doi: 10.1196/annals.1342.029

Kui, X. (2018). *Research on auto-focusing Technology of Micro Vision Measurement System*. Guangzhou: The Guangdong University of Technology.

Li, C., Moatti, A., Zhang, X., Troy Ghashghaei, H., and Greenbaum, A. (2021). Deep learning-based autofocus method enhances image quality in light-sheet fluorescence microscopy. *Biomed. Opt. Express* 12, 5214–5226. doi: 10.1364/BOE.427099

Lichang, W. (2015). *Design and implementation of automatic focusing system for surveillance camera*. Suzhou: Soochow University.

Luo, Y., Huang, L., Rivenson, Y., and Ozcan, A. (2020). Single-shot autofocusing of microscopy images using deep learning. *arXiv*, 8, 625–638. doi: 10.1021/acsphotonics.0c01774

Meng, Z. (2005). *Adjustment method and experimental research of optical projection system based on defocus depth*. Hangzhou: Zhejiang University.

Meng, J.-Y., Lu, H., Ma, S., Zhang, J.-Q., He, F.-M., Su, W.-T., et al. (2022). Progress of application of functional atomic force microscopy in study of nanodielectric material properties. *Acta Phys. Sin.* 71:240701. doi: 10.7498/aps.71.20221462

Pentland, P. A. (1987). A new sense for depth of field. *IEEE Trans. Pattern Anal. Machine Intel.* 9, 523–531. doi: 10.1109/TP.00.767940

Pinkard, H., Phillips, Z., Babakhani, A., Fletcher, D. A., and Waller, L. (2019a). Deep learning for single-shot autofocus microscopy. *Optica* 6:794. doi: 10.1364/OPTICA.6.000794

Pinkard, H., Phillips, Z., Babakhani, A., Fletcher, D. A., Waller, L. (2019b) *Single-shot autofocus microscopy using deep learning*. Cold Spring Harbor Laboratory: Optica. 6, 794–797. doi: 10.1101/587485

Qiumei, Z. (2006). *Research and implementation of automatic focusing algorithm for digital camera*. Xi'an: Northwestern Polytechnical University.

Ranjan, R., Patel, V. M., and Chellappa, R. (2018). HyperFace:a deep multi-task learning framework for face detection, landmark localization, pose Estimation, and

gender recognition. *IEEE Trans. Pattern Anal. Machine Intel*:1. doi: 10.1109/TPAMI.2017.2781233

Redondo, R., Bueno, G., Valdiviezo, J. C., Nava, R., Cristóbal, G., Déniz, O., et al. (2012). Autofocus evaluation for brightfield microscopy pathology. *J. Biomed. Opt.* 17:036008. doi: 10.1117/1.JBO.17.3.036008

Rui, X. (2021). *Research on autofocus technology based on digital image processing*. Hefei: University of Science and Technology of China.

Safari, K., Prasad, S., and Labate, D. (2021). A multiscale deep learning approach for high-resolution hyperspectral image classification. *IEEE Geosci. Remote Sens. Lett.* 18:987. doi: 10.1109/LGRS.2020.2966987

Shiyun, L. (2022). *Research on some key technologies of automatic focusing system based on image processing*. Fuzhou: Fujian Institute of Technology.

Wang, C., Qizhi, T., He, H., Haibo, H., Guifen, S. (2018). Automatic focusing method of rock slice microscopic image acquisition. *Modern Comput. Profes. Edn.* 630, 45–48.

Wang, F. Y., Zhang, J. J., Zheng, X., Wang, X., Yang, L. (2016). Where does AlphaGo go: from church-turing thesis to AlphaGo thesis and beyond. *IEEE/CAAJ. Autom. Sinica* 3, 113–120. doi: 10.1109/JAS.2016.7471613

Wu, K., Biao, X., and Can, F. (2020). Autofocusing microscope. Sichuan Province: CN210605176U.

Xu, L., and Jia, J. (2010). Two-phase kernel estimation for robust motion Deblurring. *Lect. Notes Comput. Sci*, 157–170. doi: 10.1007/978-3-642-15549-9_12

Yipeng, L., Xiyu, P., Huajun, F., Qi, L. (2005). An autofocus algorithm based on DFD. *Optical Instruments* 27:6. doi: 10.3969/j.issn.1005-5630.2005.04.009

Yu, C., and Lu, R. (2022). Performance evaluation method of focus evaluation operator in a large depth of field imaging. *Laser Optoelectron. Progress* 59:1415027. doi: 10.3788/LOP202259.1415027

Yuhu, Y., Tong, L., and Jiawen, L. (2013). Review of autofocus technology based on image processing. *Laser Infrared* 43:5. doi: 10.3969/jissn.1001-5078.2013.02.003

Yunhao, G. (2019). *Study on autofocus and global precision imaging of pathological microscope based on convolutional neural network*. Shanghai: Shanghai Jiao Tong University.

Zhang, C., Wang, H., Huang, J., and Gao, Q. (2014). The visible to the near infrared narrow band acousto-optic tunable filter and the hyperspectral microscopic imaging on biomedicine study. *J. Opt.* 16:125303. doi: 10.1088/2040-8978/16/12/125303

frontiers | Frontiers in Neuroscience

TYPE Methods
PUBLISHED 04 July 2023
DOI 10.3389/fnins.2023.1219363

Check for updates

OPEN ACCESS

EDITED BY
Alois C. Knoll,
Technical University of Munich, Germany

REVIEWED BY
Zhenzhong Wang,
Hong Kong Polytechnic University,
Hong Kong SAR, China
Zhenyu Wen,
Zhejiang University of Technology, China
Changting Lin,
Zhejiang University, China

*CORRESPONDENCE
Yongquan Chen
✉ yqchen@cuhk.edu.cn

†These authors have contributed equally to this work

RECEIVED 09 May 2023
ACCEPTED 13 June 2023
PUBLISHED 04 July 2023

CITATION
Yu J, Zheng W, Chen Y, Zhang Y and Huang R (2023) Surrounding-aware representation prediction in Birds-Eye-View using transformers. *Front. Neurosci.* 17:1219363. doi: 10.3389/fnins.2023.1219363

COPYRIGHT
© 2023 Yu, Zheng, Chen, Zhang and Huang. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

Surrounding-aware representation prediction in Birds-Eye-View using transformers

Jiahui Yu[1†], Wenli Zheng[2†], Yongquan Chen[1*], Yutong Zhang[1] and Rui Huang[1]

[1]Shenzhen Institute of Artificial Intelligence and Robotics for Society, and the SSE/IRIM, The Chinese University of Hong Kong, Shenzhen, Guangdong, China, [2]The Shenzhen Academy of Inspection Quarantine, Shenzhen, Guangdong, China

Birds-Eye-View (BEV) maps provide an accurate representation of sensory cues present in the surroundings, including dynamic and static elements. Generating a semantic representation of BEV maps can be a challenging task since it relies on object detection and image segmentation. Recent studies have developed Convolutional Neural networks (CNNs) to tackle the underlying challenge. However, current CNN-based models encounter a bottleneck in perceiving subtle nuances of information due to their limited capacity, which constrains the efficiency and accuracy of representation prediction, especially for multi-scale and multi-class elements. To address this issue, we propose novel neural networks for BEV semantic representation prediction that are built upon Transformers without convolution layers in a significantly different way from existing pure CNNs and hybrid architectures that merge CNNs and Transformers. Given a sequence of image frames as input, the proposed neural networks can directly output the BEV maps with per-class probabilities in end-to-end forecasting. The core innovations of the current study contain (1) a new pixel generation method powered by Transformers, (2) a novel algorithm for image-to-BEV transformation, and (3) a novel network for image feature extraction using attention mechanisms. We evaluate the proposed Models performance on two challenging benchmarks, the NuScenes dataset and the Argoverse 3D dataset, and compare it with state-of-the-art methods. Results show that the proposed model outperforms CNNs, achieving a relative improvement of 2.4 and 5.2% on the NuScenes and Argoverse 3D datasets, respectively.

KEYWORDS

BEV maps, deep learning, attention, transformers, autonomous driving

1. Introduction

The advancement in deep learning has facilitated a better understanding of semantic representation and contributed to more accurate prediction of object locations. This line of research has a wide range of applications in autonomous driving (Ohn-Bar et al., 2020; Yi et al., 2021; Cao et al., 2022a; Wang et al., 2022).

Recent studies have made significant strides in mapping multiple side-view images to Birds-Eye-View (BEV) semantic maps, aiming to predict the positional probability of each element. These BEV maps have proven to be potent tools for environment perception, fundamental to autonomous navigation and driver assistance systems. As illustrated

in Figure 1, cameras strategically positioned around the vehicle capture RGB images from all directions. Surrounding-aware systems then model these images to generate comprehensive 360-degree BEV maps, offering a panoramic understanding of the vehicle's environment. However, creating BEV maps is challenging; it represents a complex, multi-stage processing flow encompassing ground plane estimation, road segmentation, lane detection, and object detection, as described in Chen et al. (2020), Pan et al. (2020), and Roddick and Cipolla (2020). It's a laborious process with challenges, yet its importance for safe and efficient autonomous navigation cannot be overstated. The ideal scenario is to design an end-to-end framework powered by deep learning. This approach would directly predict the desired map representation from sensor observations, providing a comprehensive understanding of the environment in a single step. In this context, semantic segmentation emerges as an indispensable tool, particularly in autonomous driving. Semantic segmentation helps distinguish various environmental elements, like roads, pedestrians, vehicles, etc., enabling the system to interpret and interact safely with its surroundings. By integrating this with our proposed end-to-end BEV map generation, we aim to facilitate a more robust, efficient, and safer autonomous driving system.

Several studies, such as Hendy et al. (2020), Mani et al. (2020), Wu et al. (2021), Cao et al. (2022b), and Han et al. (2022), have shown that CNNs are capable of capturing a large receptive field; however, this comes with a trade-off involving deepening the neural network structure. Despite being highly discriminative, semantic features extracted from deeper convolution layers are not suitable for representing small-sized/multi-class elements, which limits the accuracy of predicting multi-element BEV representations. Recent studies, including Yi et al. (2021) and Yu et al. (2021), have indicated that shallow feature maps are more effective for small-scale object detection as they provide rich spatial information. As a result, balancing the need for capturing large receptive field and extracting highly discriminative features can be challenging for CNNs. Current studies have shown that Transformers are able to achieve feature extraction with a large receptive field in a shallow structure.

Exploring various strategies for developing high-quality Bird's Eye View (BEV) maps has become increasingly essential in technology and science, particularly with the rise of autonomous navigation and robotics. While several methodologies have been presented, they tend to rely on large training samples and display less resilience when faced with varying circumstances. Furthermore, these previous studies primarily utilized Transformers for tasks involving classification, which output a set of per-class probabilities as exhibited by Han et al. (2022), Hu et al. (2022), and Li et al. (2022). This leaves a significant area within the transformative potential of Transformers untapped—generating BEV semantic representation. We venture into relatively uncharted territory, exploring the potential of using Transformers exclusively to generate a BEV semantic representation, thereby bypassing the necessity for convolution layers. Unlike the conventional approaches, which focus on "classifying" image-based data, our approach looks at both input and output as images—a procedure we refer to as "image generation." This shift from classification tasks to generation tasks, utilizing Transformers, might pave the way to more efficient, scalable, and diverse applications, ultimately expanding the possibilities of BEV mapping technologies. Comprehensive semantic feature extraction is the bedrock for constructing high-quality BEV maps. To improve this process, the research community must be willing to test novel approaches. Our proposed use of Transformers as a sole agent for generating BEV semantic representation stands as a pioneering endeavor in this domain, challenging the conventional paradigms that have been established. As such, it will contribute to the broader discourse on effective BEV mapping and extend the functional capabilities of Transformers. The outcomes of this study could potentially guide further developments and improvements in autonomous navigation systems, robotics, and other related areas where BEV maps are paramount. This research is not merely a theoretical experiment but a concrete step forward in the practical application of Transformers in the real world.

To achieve this goal, the paper addresses two main challenges: (1) how to extract global-local discriminative features using Transformers, and (2) how to generate pixels from image features without the use of convolution layers. A main challenge for Transformers is their high dependence on data availability for training the model to achieve promising performance. Additionally, the way attention-a core component of Transformers-is applied to image generation is still under-explored. In the paper, we propose a Transformer-based framework that generates BEV semantic representations in an end-to-end process. To fully capture features, a new attention mechanism is employed for spatial relationships. Traditional neural networks always overlook a large number of semantic features due to the projections of features in different planes. To overcome this limitation, we design a new plane transformation algorithm. Previous methods have relied mostly on convolution layers for multi-class representation generation. However, since Transformers and convolution would restrict the performance improvement of each other, a pure Transformer-powered generator is proposed to address the issue. Alongside this, a stable training scheme is also developed to specifically target the new framework.

The contributions of this study can be summarized as follows.

- We propose a new framework purely powered by Transformers for predicting BEV semantic representations. The framework achieves this objective in an end-to-end manner without convolution layers. This approach differs dramatically from the existing CNN-based methods, greatly simplifying the pipeline and improving perceptual details.
- We propose a feature extractor based on a Transformer to memorize global clues and further mine local clues. The extractor contains an overlapping patch generation method and multi-head attention-based blocks.
- We design a new generator powered by Transformers to generate pixels based on per-class prediction probabilities. The carefully-designed pipeline is essential for generating successful BEV semantic representations. In addition, a simple trick is proposed that can transform image features into BEV features.
- We achieve competitive results on two challenging datasets, namely 19.9% Mean IoU on the NuScenes dataset and 19.1%

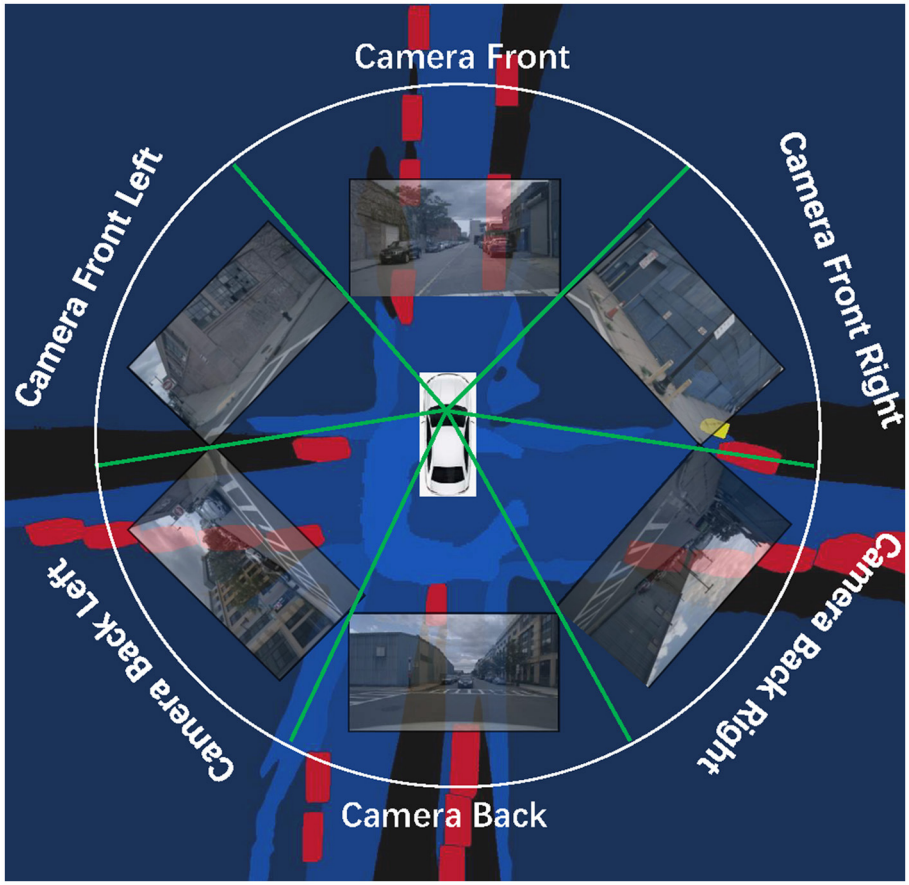


FIGURE 1
Schematic illustrating the process of BEV semantic representation prediction in environment perception. A set of surround-view monocular cameras are used to capture the surrounding environment (RGB images). These images are passed through a road-aware system to generate BEV maps containing the location and shape information of various elements.

Mean IoU on the Argoverse 3D dataset. Compared with leading CNN-based methods, our model demonstrate an improvement of 2%–6 % in Mean IoU, and about 1% IoU improvement is achieved on challenging prediction tasks.

The remainder of the paper is structured as follows. Section 2 reviews deep learning-powered studies. Section 3 provides details on the proposed frameworks and technologies. Section 4 presents the experimental results and discussion. Section 5 concludes the works and shows the future research direction.

2. Related work and basics

2.1. Surrounding-aware system in autonomous driving

The surrounding-aware system plays a crucial role in connecting road conditions and driving assistance systems. However, current technologies, such as sensing, detection, and segmentation, are limiting the development of these systems Predicting BEV representation based on monocular images is a challenging problem for several reasons including multitasking, complex 3D estimation, and multi-class prediction. Traditional studies have proposed neural networks based on semantic segmentation for BEV representation prediction (Pan et al., 2020; Lu et al., 2021). However, these 2D representations lack spatial relationships and are not effective in complex 3D spatial scenarios. Recent studies have reported two categories of solving this challenge: camera geometry and transformation implicitly, as reported by Lu et al. (2021) and Yao et al. (2021). The former has achieved significant performance by using on multi-type data input, while the latter is more suitable for building a simple learning framework in an end-to-end manner.

Although current methods are effective based on single data input, they are not able to fully mine the spatial dependency, which severely hinders the performance of multi-scale/multi-class element prediction. In the paper, the proposed neural networks are specifically developed to exploit rich spatial clues by considering the global spatial relationship in a shallow framework, instead of focusing on particular regions in a deep structure. The purely Transformer-based neural networks are proposed in the paper, and several related or derived techniques are developed.

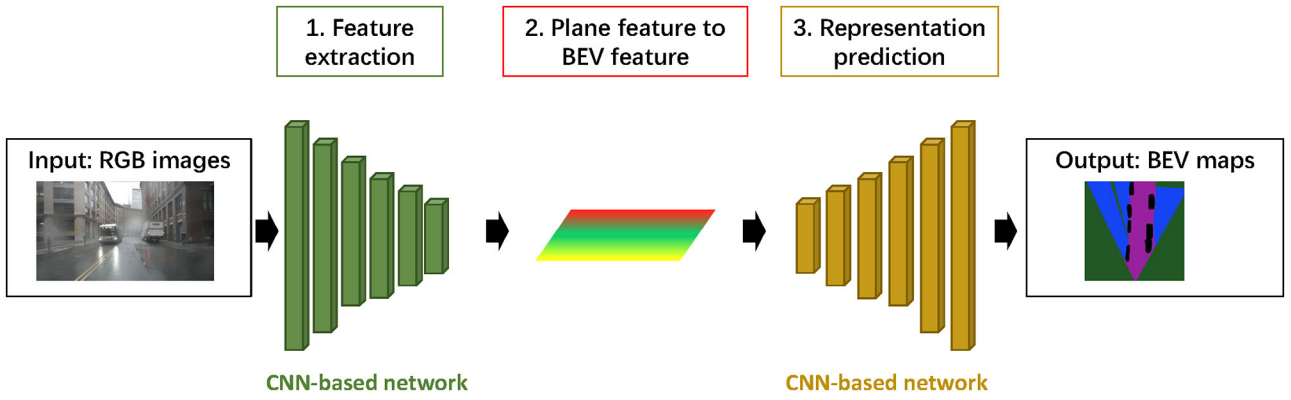


FIGURE 2
Architecture of CNN-based networks for BEV representation prediction.

2.2. CNN-based studies

In previous studies, Convolutional Neural networks (CNN) have been widely used for image processing. As shown in Figure 2, the process typically contains three steps: (1) extracting features from input images, (2) projecting planar image features to BEV features, and (3) generating pixels under the BEV map. Current studies have utilized leading CNN-based backbones for feature extraction, such as ResNet (He et al., 2016), Feature Pyramid Network (FPN) (Lin et al., 2017), and DeepLab network (Yang et al., 2018). In addition, recent works by Hendy et al. (2020) and Mani et al. (2020), have incorporated BEV view transformation based on FPN and employed adversarial loss to optimize BEV representation. Inspired by the Generative Adversarial Network (GAN), some studies have proposed Top-down networks, with the Inverse Perspective Mapping (IMP). The front view image is mapped to the ground plane by homography (Zhu et al., 2018; Hu et al., 2023).

However, a significant issue has arisen in these CNN-based studies. While the accurate prediction of large-size objects has reached a saturation point, the forecast of small-size things still remains an unresolved challenge. The majority of CNN-based networks excel in local semantic segmentation. Still, they need to catch up when predicting global BEV maps. The reason for this lies in the inherent trade-off between network depth and the range of the receptive field. In CNN-based networks, the receptive field can access the depth of the network. While this sounds beneficial in theory, allowing the network to capture more complex patterns with more layers, it also escalates the challenges associated with model training. Deeper networks tend to suffer from difficulties in training due to issues such as vanishing and exploding gradients. Moreover, as the web grows in depth, it becomes increasingly computationally intensive, which might not be sustainable in real-world, resource-limited applications. In light of this, there is a pressing need for novel solutions that can accurately predict large and small objects in BEV maps while also addressing the computational and training challenges associated with deep CNN-based networks. We can only unlock the full potential of BEV mapping for autonomous navigation and related applications by overcoming these hurdles.

2.3. Transformer-based studies

In 2020, Google AI introduced Vision Transformer (ViT) for image classification without convolution layers (Dosovitskiy et al., 2020). ViT divides the input image into square patches of equal sizes, followed by the pure Transformer architecture processing directly on the patch sequence to mine global-local features and output per-class probabilities. Originally, Transformers were proposed for Natural Language Processing (NLP) tasks (Vaswani et al., 2017), but ViT has achieved impressive performance on multiple image recognition benchmarks. Transformers have also been used to solve other vision-related problems, including object detection, semantic segmentation, and image processing, where they outperformed CNN-based networks including object detection, semantic segmentation, and image processing (Han et al., 2022). Typical studies by Carion et al. (2020), Chen et al. (2021a), Misra et al. (2021), and Mazzia et al. (2022), have reported that self-attention mechanism used in Transformers help model long-term features effectively. Furthermore, some studies, such as Ba et al. (2016), Liu et al. (2021), and Zheng et al. (2021) have extended Transformers to the field of semantic segmentation by designing the encoder based on Transformers, then adding other existing decoders to model the image context. However, the following limitations exist in recent Transformers-based studies: (1) the global modeling scheme leads to high computational costs and requires a large amount of data, and (2) the decoder design still relies on convolutional layers. To the best of our knowledge, no prior studies have explored a pure Transformer-based framework for predicting BEV semantic representations without convolution layers.

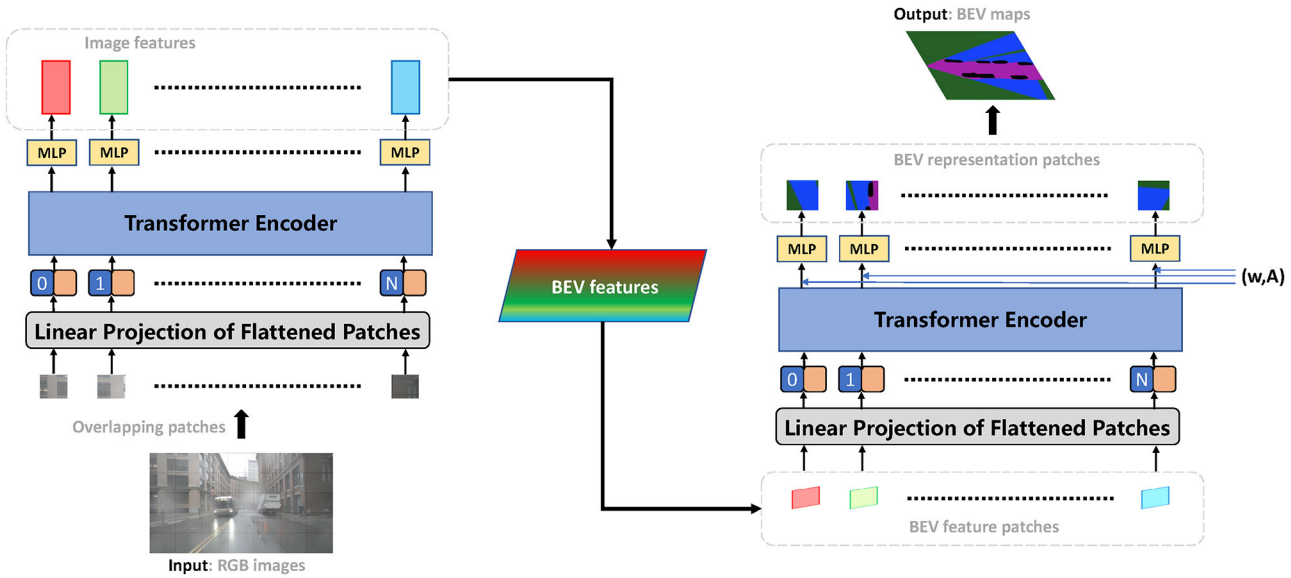


FIGURE 3
Architecture of the proposed approach. The end-to-end framework contains three modules: (1) Feature extraction from input images, (2) Image features are projected to BEV features, and (3) BEV semantic representation prediction. A Transformer-based network encodes global-local spatial features powered by multi-head attention. The image-plane features can be transformed to BEV-plane features with the smallest possible loss of feature information. A generative network further processes BEV global-local spatial features and predicts the final classification probabilities.

3. Method

We approach the prediction of BEV representation by framing it as a global-local spatial relationship mining problem. Given a set of look-around image inputs, the proposed method generates the corresponding BEV representations in order. These representations can be simply synthesized into full-space BEV maps. In this section, we present the technical details of our proposed method according to the processing order in the end-to-end framework, as shown in Figure 3. We begin by explaining a Transformer-based extractor that achieves image encoding, while balancing the global spatial attention and receptive field. Next, we describe how we transform the side views into BEV views and mine the relationship of the inter-frame through a homography-based algorithm. We also present a new Transformer-based predictor for making predictions in BEV views. To represent the state of the world, including vehicles, drivable areas, and land boundaries, we use the semantic occupancy grid, which is an extension of occupancy grid maps. We then introduce an association training scheme that ensures the stable convergence of Transformer-based neural networks.

3.1. Image encoding

In alignment with the conventional Vision Transformer (ViT) method, our proposed model uses a backbone to extract image features crucial for generating Bird's Eye View (BEV) semantic representations. The extraction process is mathematically outlined in Equation (1), with the details underpinning this method comprehensively discussed in Dosovitskiy et al. (2020).

Our feature extractor incorporates several vital components: multi-head self-attention, multilayer perceptrons (MLPs), residual connections, layer normalization, positional encoding, and meticulously structured network topology. Each component plays an instrumental role in the overall process of BEV generation. The multi-head self-attention mechanism is particularly crucial in this process. It enables the model to focus on different parts of the input image simultaneously, thus allowing it to capture complex patterns and dependencies in the input data. This capability is vital for tasks like BEV prediction, where various aspects of an image contribute to the final output. Multilayer perceptrons further enhance the model's capability to understand complex patterns in the data. At the same time, residual connections help combat the vanishing gradient problem, enabling the model to learn more effectively from the data. Layer normalization ensures that the model's training remains stable and efficient by standardizing the inputs to each network layer. Meanwhile, positional encoding is employed to provide the model with information about the relative positions of the pixels in the input image, which is crucial for tasks involving spatial data. Finally, the network topology defines the overall structure of the model and is designed in a way to optimize the information flow and processing within the model. By intertwining these components, our feature extractor presents an effective means of obtaining and interpreting image data, fulfilling the essential role in BEV generation.

$$
\begin{aligned}
&z_0 = [x_{class}; x_p^1 E; x_p^2 E; .....; x_p^N E] + E_{pos}, \\
&E \in R^{(P^2 \cdot C) \times D}, E_{pos} \in R^{(N+1) \times D} \\
&z_\ell{}' = MSA(LN(z_{\ell-1})) + z_{\ell-1}, \quad \ell = 1.....L \\
&z_\ell = MLP(LN(z_\ell{}')) + z_\ell{}', \quad \ell = 1.....L
\end{aligned} \tag{1}
$$

For image encoding, we design a Transformer-based network considering the requirements of feature transformation. First, the input of the network X takes an image with three dimensions ($C \times H \times W$) as input and converts it into a 2D vector [$x_p \in HW/P^2 \times (P^2 \cdot C)$]. To achieve the conversion, we split the image into multiple patches of fixed-size (size: $P \times P$) that is carefully designed to mine local cues more effectively. Specifically, each patch is extended by i pixels to create overlap between patches, so the size of each patch is $(P+2i) \times (P+2i)$. Our research has shown that this careful design achieves regular training and robust optimization. Moreover, the embedding matrix Ecan covert each patch to the ($N \times D$) dimensions, and the E_{pos} is the position code to prevent the patch order from being disrupted. MSA means the operation powered by the multi-head self-attention (MSA) whose technical details are similar to that of the ViT. The MLP consists of the Linear layer (LN) and the tanh function.

Due to poor inductive bias performance, Transformer-based models show high sensitivity to input perturbations. However, to enhance their generalization performance, the proposed neural networks are expected to be insensitive to input perturbations. To achieve this, a new attention ($A_{Ed}(X)$) is introduced in the transformer encoder process. This new attention model draws inspiration from the contributions in Kim et al. (2021), and recomputes the dot product similarity in attention using Euclidean distance, as illustrated in Equation (2), where P_q, P_k, and P_v respectively denote the important parameters in the projection process i.e., Query, Key, and Value; and d_n denotes the dimension of features in multi-head attention.

$$A_{Ed}(X) = Soft\max(Ed(XP_q, XP_k)/\sqrt{d_n})XP_v \tag{2}$$

Our proposed approach diverges significantly from typical Transformer-based classification models in terms of its output. In conventional models, the multilayer perceptrons (MLPs) output within the Transformer is typically a set of classification probabilities. However, in our work, the MLPs output image features. These features, rich in semantic information, are then transformed into Bird's Eye View (BEV) features, as further detailed in the following sub-section. This innovation has multiple potential benefits. Most importantly, it has a considerable impact on the computational efficiency of our model. Given that image features contain essential information in a condensed form, this method dramatically reduces the volume of data to be processed in subsequent stages. Instead of classification probabilities, the output features bring the significant advantage of lowering the model's complexity and reducing the computation load. Furthermore, the model can bypass the computationally intensive step of converting probabilities back into image features by directly working with image features instead of possibilities. This further economizes the computational resources required, making the model more efficient and quicker. In essence, our approach is designed to effectively extract and utilize image features for BEV mapping, all while maintaining computational efficiency. This streamlined process, which provides detailed BEV features without the usual computational burdens, is a key advancement over traditional Transformer-based classification models.

3.2. Image-to-BEV

The process of converting a side view captured by the vehicle camera to the BEV perspective is significantly challenging, largely due to the fundamental differences between the two coordinate systems. Unfortunately, the feature extraction network can only output image-plane features. Hence, the main objective is to reduce feature loss during the feature transformation process. An image-plane feature map that has a height H and width W is transformed into a BEV-plane that has a depth Z and width X, with channel C unchanged.

Motived by the Hough transform, we design an effective method for projecting features from image-plane to BEV-plane features ($F^{IP} \rightarrow F^{BEV}$), as shown in Equation (3). Where r_l, θ_l, and c_l denote horizontal plane, azimuth, and elevation in a feature map location, respectively; and $w_{(r_l,\theta_l)}$ denotes the weights learned by the framework. Technically, we first collapse the vertical and channel dimensions into a transition dimension and keep the horizontal dimension unchanged. We then reset the transition features to get a new tensor with the size ($C \times Z \times W$). Finally, we resample into a Cartesian coordinate system, namely, the BEV of the trapezoid, thus establishing a new camera geometry.

$$F^{BEV}_{(r_l,\theta_l,c_l)} = \sum w_{(r_l,\theta_l)} \times F^{IP}_{(r_l,\theta_l,c_l)} \tag{3}$$

Compared with recent studies, such as Pan et al. (2020) and Philion and Fidler (2020), the proposed method utilizes a cost-saving operation to address the challenge of retaining the depth features of the input.

3.3. BEV semantic representation generation

Generating semantic features entirely using Transformer-based model can be a huge challenge because the Transformers need to generate pixels in spatial regions instead of traditional predicted class labels. Inspired by studies of combining Transformers to GANs, such as Chen et al. (2021b), Jiang et al. (2021), an Lee et al. (2021), we proposed a new projector to generate BEV semantic representation without convolution and pooling layers through two stages: BEV semantic representation generation from single image and BEV representation fusion. Unlike GANs, the discriminator (a special Loss of GAN) is not required, and we design an associate training scheme to supervise the end-to-end framework.

We develop the Transformer Encoder module to generate new pixels in spatial space. We first introduce an affine transformation A to each image feature patch, followed by the use of the Fourier function for patch embedding. In technical terms, the architecture is represented by Equations (4) and (5), where (x, y) denotes the values of patch pixel obtained from the patch embedding, L is the length of the input sequence, E_{fou} is the Fourier encoding to compute the spatial position of the pixel, and M_θ is the MLP operation. The results show that the proposed module is effective in

generating BEV pixels, as shown in the details presented in Figure 6.

$$\begin{aligned} &z_0 = E_{pos}, \quad E_{pos} \in R^{(N+1)\times D} \\ &z_\ell{}' = MSA(LN(z_{\ell-1}, w)) + z_{\ell-1}, \quad \ell = 1.....L \\ &z_\ell = MLP(LN(z_\ell{}', w)) + z_\ell{}', \quad \ell = 1.....L \end{aligned} \tag{4}$$

$$\begin{aligned} &y = LN(h_L, w) \\ &x = [M_\theta(E_{fou}, y^1), ..., M_\theta(E_{fou}, y^L)] \end{aligned} \tag{5}$$

Motivated by the multiple observation information methods (Wang et al., 2019; Roddick and Cipolla, 2020), we propose a Bayesian-based information natural fusion method. The main objective is to build a wraparound BEV representation by calculating the occupancy probability of each view feature in the global coordinate system. First, we use the Log-odds operation (denoted as $l^c_{i,t}$) to equate the occupancy probability $p(m^c_i|o_t)$ that are the output of the network, where m^c_i is the i-th observation of an object of class $c$in network output. Next, The combination of observations from the 1st to the t-th is shown in Equations (6) and (7).

$$\begin{aligned} &l^c_{i,1:t} = l^c_{i,1:t-1} + l^c_{i,t} - l^c_0 \\ &l^c_0 = \frac{p(m^c_i)}{1-p(m^c_i)} \end{aligned} \tag{6}$$

$$l^c_0 = \frac{p(m^c_i)}{1-p(m^c_i)} \tag{7}$$

3.4. Training

We design an association training scheme to obtain more accurate predicted probabilities containing three Loss functions, as shown in Equation (8). First, the Binary cross-entropy (L_{bce}) is used to train semantic occupancy probabilities $P(\cdot)$, as shown in Equation (9). Second, to stimulate the framework to achieve efficient convergence on complex images, such as small object-contained images and partial occlusions, we introduce another Loss function (L_{comp}), as shown in Equation (10). Finally, we design a feature transformation loss L_{ft} to train Hough transfom-based process, as shown in Equation (11). Here, D is a discriminant that distinguishes between the ground-truth and the predictions, and g_{cls} denotes the prediction ground-truth. The proposed framework is programmed in an end-to-end manner.

$$L_{asso} = L_{bce} + L_{comp} + L_{ft} \tag{8}$$

$$L_{bce} = \sum_{i=1}^{N} \alpha m_i \cdot \log(p(m_i)) + (1-\alpha)(1-m_i) \cdot \log(1-p(m_i)) \tag{9}$$

$$L_{comp} = 1 - P(m_i)\log_2 P(m_i) \tag{10}$$

$$L_{ft} = \sum D(f_{IP\to BEV} \cdot f_{ip}(X), g_{cls}) \tag{11}$$

4. Experiment and discussion

In this subsection, we first empirically compare CNNs with Transformers and discuss the results. Next, we empirically assess the effectiveness of the proposed method on two challenging benchmarks and compare it with state-of-the-art methods. Moreover, we show the performance of neural networks on typical challenging tasks.

4.1. Experimental settings

4.1.1. Database

We choose two particularly challenging benchmarks to evaluate the proposed model, i.e., the NuScenes dataset (Caesar et al., 2020) and the Argoverse 3D dataset (Chang et al., 2019). They are large-scale datasets in the field of autonomous driving. For data selection, we follow the standard procedures used in most previous studies (Philion and Fidler, 2020; Roddick and Cipolla, 2020). From the NuScenes dataset, we select four categories of maps predicted by images, which contain 14 elements. From the Argoverse 3D dataset, we select eight out of 15 elements for map prediction. Additionally, since both datasets are designed for object detection, and the labels are provided in vectorized and 3D bounding boxes, we regenerate labels to fit the map prediction task. As for technical details, we follow the recent works Philion and Fidler (2020) and Roddick and Cipolla (2020). The main approach we apply is generating annotations for rasterized BEV images via vector label mapping and binary mask generation. The predicted elements consist of Drivable (Dri.), Pedestrian Crossing (Ped.C.), Vehicle (Veh.), Large Vehicle (L.Veh.), Walkway (Wal.), Carpark (Carp.), Car, Truck, Bus, Trailer (Tra.), Construction Vehicle (Con.V.), Pedestrian (Ped.), Motorcycle (Mot.), Bicycle (Bic.), Traffic cone (Tra.C.), and Barrier (Bar.).

4.1.2. Evaluation

To ensure fairness in comparison, we select the Intersection over Union (IoU) score as the main evaluation metric. The IoU evaluation shows the similarity between the element prediction area and the ground truth area, with higher values indicating more accurate predictions.

4.1.3. Implementation

We first pre-train the proposed network with the ImageNet dataset using SGD, with a batch size of 512. Considering that the smaller the input patch size is, the more computationally expensive it is, we choose a patch size of 64 × 64. The number of attention heads is 6, and the number of the transformer blocks is increased to 6 (typically 4). The Adam algorithm is utilized for training, with a weight decay of 0.1 and a batch size of 32.

TABLE 1 Performance of various design.

Method	Dri.	Bar.	Ped.C.	Wal.	Carp.	Car	Truck	Bus	Tra.	Con.V.	Ped.	Mot.	Bic.	Tra.C.	Mean
R + IPM	44.3	2.7	10.6	13.8	13.2	4.7	0.2	2.7	7.4	6.2	0.6	0.6	0.3	2.4	7.9
R-FPN + IPM	53.7	5.1	14.9	16.3	14.7	11.2	5.8	9.7	11.1	8.2	2.4	2.7	2.1	3.9	11.6
ViT-FE + IPM	56.4	7.1	27.3	26.1	15.8	23.6	16.1	20	15.6	11.7	7.1	5.7	7.5	5.4	17.5
R + ViT-RG	55.2	7.3	23.6	24.7	15.2	20.7	12.2	17.2	14.3	10.4	5.8	4.4	6	4.5	15.8
R-FPN + ViT-RG	56.3	10	28	28.5	17.4	23.4	15.7	19.9	15.8	11.7	6.7	5.6	6.9	5.3	17.9
Proposed	61.1	11.9	28.7	32.7	19.2	25.8	17.4	20.6	16.9	12.3	8.3	7.3	9.8	6.6	19.9

A larger value indicates better performance.

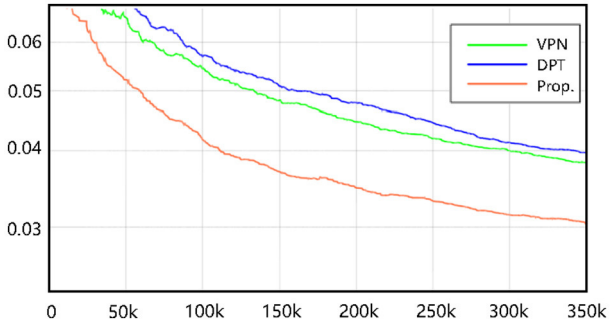


FIGURE 4
The loss function curves for three methods on the NuScenes.

4.2. Ablation study

Our first step is to conduct ablation study on the NuSences dataset (14 elements) to evaluate the effectiveness of two proposed fundamental techniques: Transformer-based feature extraction model (ViT-FE) and the Transformer-based BEV semantic representation generation model (ViT-RG). The main purpose of this study is to assess how well Transformers perform in BEV representation prediction. In most CNN-based studies, ResNet-based networks are used for feature extraction, and networks with a Top-down structure are used for BEV representation generation. Hence, we mix and match the proposed different modules and the leading CNN-based networks.

In the ablation study, we select three leading deep modules for mixing and matching, including a ResNet-50 backbone (R) (He et al., 2016), a ResNet-50 with the FPN structure (R-FPN) (Yu et al., 2022), and an IPM-based Top-down network (IPM) (Deng et al., 2019). We train the above three models using SGD with a momentum of 0.9, a batch size of 32, and a wight decay of 0.1.

Table 1 shows the results of the ablation study, and all of the Abbreviated names are listed above for reference. The results clearly indicate that the proposed ViT-FE and ViT-RG show a considerable improvement in performance as compared to CNN-based models, with an increase of around 6% higher (Mean IoU). These findings highlight the effectiveness of using Transformer-based to predict BEV semantic representation. Specifically, the Transformer-based modules can gradually improve the prediction accuracy for large-scale objects by about 2% and significantly improve it for challenging small-scale objects by about 1% IoU.

Ensuring stable training is important for neural networks, especially for Transformer-based networks. We propose a new framework based purely on Transformers. To evaluate the training effect of the proposed neural networks, we conduct empirical experiments on a challenging benchmark (NuScenes). As depicted in Figure 4, our method achieves comparable training performance to other CNN-based methods. We suppose that the proposed Transformer-based neural networks can be generalized for the BEV semantic prediction task. Stable training serves as the basis for further improving the performance of Transformer-based methods.

4.3. Main results and comparison to state-of-the-art works

In this subsection, we present a comparison of the proposed model with three recent works, including a View Parsing Network (VPN, published in IEEE RAL, 2020) (Pan et al., 2020), a Top-down network with transformer layer (DPT, published in CVPR 2020) (Roddick and Cipolla, 2020), and Lift-Splat-Shoot network (L-S-S, published in ECCV 2020) (Philion and Fidler, 2020). The CNN-based baseline [R-FPN (Yu et al., 2022) + IPM (Deng et al., 2019)] is also shown for reference. It should be noted that the sub-datasets chosen for each study are not identical due to the large scale of autonomous driving datasets. To ensure a fair comparison, we follow the part of the results reported by the DPT, and then we train and test the L-S-S using the same experimental setup.

4.3.1. Main results

The NuScenes dataset contains a greater variety of objects than the Argoverse 3D dataset, making it more challenging. As demonstrated in Table 2 our proposed network achieved a significant improvement in the Mean IoU metric, with 0.8% higher than the DPT, 1.4% higher than the L-S-S, and 2.4% higher than the VPN. Table 3 shows that the proposed network attains the leading performance on the Argoverse 3D dataset. Specifically, the proposed network exhibits further improvements in the prediction rate of large-scale objects by about 2% in the prediction rate of large-scale objects. The main result is breaking through the bottleneck of small-scale object prediction. In comparison, the prediction accuracy based on the CNN networks remains essentially unchanged. Figure 5 shows the Precision-recall curves of four interesting elements selected from the NuScenes dataset. The closer these curves are to the upper right, the better the models prediction performance for positive samples.

4.3.2. Discussion

Through an extensive examination of our experimental results, several key insights have been gleaned, reaffirming the innovative nature and potential of the method. First, Transformer-based networks substantially improved the prediction of BEV semantic features. This represents a significant stride forward compared to traditional methods, suggesting that Transformers hold great promise in advancing BEV mapping capabilities. This marked performance enhancement demonstrates the effectiveness of Transformer-based approaches. It provides a valuable reference point for future research endeavors, opening up new avenues for exploration and innovation. Second, our proposed Transformer-based feature extractor demonstrated a superior ability to extract finer features than CNN-based networks. This superiority is particularly significant in predicting small-scale objects. In this task, the extraction of intricate details is of utmost importance. This underlines the capacity of Transformer-based models to outperform their CNN counterparts in tasks that require a keen discernment of finer details, thus broadening their potential applications in related fields such as object detection and recognition. Last, our unique contribution is the introduction

TABLE 2 Main results (IoU) on the NuScenes dataset.

Method	Dri.	Bar.	Ped.C.	Wal.	Carp.	Car	Truck	Bus	Tra.	Con.V.	Ped.	Mot.	Bic.	Tra.C.	Mean
R-FPN + IPM	53.7	5.1	14.9	16.3	14.7	11.2	5.8	9.7	11.1	8.2	2.4	2.7	2.1	3.9	11.6
VPN (Pan et al., 2020)	58	10.8	27.3	29.4	12.9	25.5	17.3	20	16.6	4.9	7.1	5.6	4.4	4.6	17.5
L-S-S (Philion and Fidler, 2020)	60.2	10.9	26.6	31.1	17.6	24.2	16.5	20.3	15.4	10.8	7.5	5.8	7.2	5.9	18.5
DPT (Roddick and Cipolla, 2020)	60.4	8.1	28	31	18.4	24.7	16.8	20.8	16.6	12.3	8.2	7	9.4	5.7	19.1
Proposed	61.1	11.9	28.7	32.7	19.2	25.8	17.4	20.6	16.9	12.3	8.3	7.3	9.8	6.6	19.9

A larger value indicates better performance.

TABLE 3 Main results (IoU) on the Argoverse 3D dataset.

Method	Dri.	Bus	Tra.	L.Veh.	Ped.	Mot.	Bic.	Veh.	Mean
R-FPN + IPM	54.2	5.2	0.3	8.5	2.7	0.8	0.2	15.8	11
VPN (Pan et al., 2020)	64.9	3	0.4	9.7	6.2	1.9	0.9	23.9	13.9
L-S-S (Philion and Fidler, 2020)	65.2	13.7	1.8	11.7	8	5.7	3.4	30.8	17.5
DPT (Roddick and Cipolla, 2020)	65.4	11	0.7	11.1	7.4	5.7	3.6	31.4	17
Proposed	66	17.5	2.9	13.8	8.6	6.4	4.2	33.7	19.1

A larger value indicates better performance.

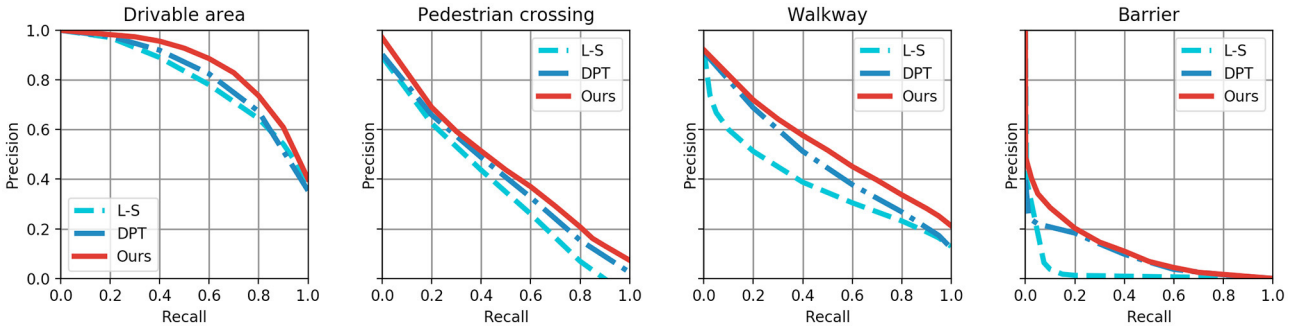


FIGURE 5
Precision-recall tradeoff on the NuScenes dataset.

of a Transformer-based feature generator capable of outputting pixel points instead of per-class probability. This novel approach has exhibited superior performance compared to traditional Top-down networks. By moving from per-class possibilities to pixel point outputs, the proposed model offers a more nuanced and detailed understanding of the input image, essential for complex image generation tasks like BEV mapping. It also presents a more versatile and granular output format that can be more readily adapted to various applications. These observations demonstrate the superiority and innovation of our proposed Transformer-based approach to BEV semantic feature extraction and generation. This research has not only bridged a significant gap in the field but also paved the way for further advancements and applications of Transformer-based models in the broader domain of image processing and analysis.

4.4. Performance on challenging scenarios

This paper aims to making contribution to the discussion of global-local spatial relationship learning, which is better at multi-class and multi-scale element prediction. To further show the performance of the proposed method on challenging tasks, we evaluate its performance using challenging samples, i.e., complex lanes, small-scale pedestrian, dark environments, two key traffic signals, and multi-class element.

Figure 6 shows the qualitative results on the NuScenes dataset. Two state-of-the-art methods are introduced for comparison. The key conclusions are as follows. (1) The proposed method predicts BEV semantic representation that closely matches ground-truth labels. (2) The proposed method can effectively perceive more detailed features, such as vehicle contours, subtle changes in lane lines, and small-sized pedestrians. For example, the VPN fails to predict small-size elements like pedestrians, and the DPT only predicts elements that are close to the camera, while the proposed method works well. (3) The proposed method can achieve state-of-the-art performance in complex field situations, including illuminant-changed scenarios and occlusion. For example, in the night driving, the prediction result of the VPN does not contain vehicles, and the DPT can only predict parts of vehicles. In comparison, the proposed method can still accurately predict the number and location of vehicles in occlusion scenarios.

Furthermore, we test our proposed method by examining its performance on multi-class element prediction, as shown in Figure 7. We deployed the technique to generate Bird's Eye View (BEV) semantic representations for each element in scenarios that pose significant challenges, such as occlusion, the presence of multi-class/multi-scale objects, and dim illumination. The results have been highly encouraging, demonstrating that our model can accurately predict the position and shape of each element. These positive results confirm our method's effectiveness and point toward its high computational efficiency and scalability, particularly in large-scale environments. Despite the complexity introduced by multi-class/multi-scale objects and conditions like occlusion and dim lighting, the model maintains an impressive performance. This attests to the model's robustness and adaptability in diverse and challenging situations. Notably, the computational efficiency of our method does not compromise its scalability. Our model can seamlessly handle an increased number of classes or a larger scale of images without requiring a proportional

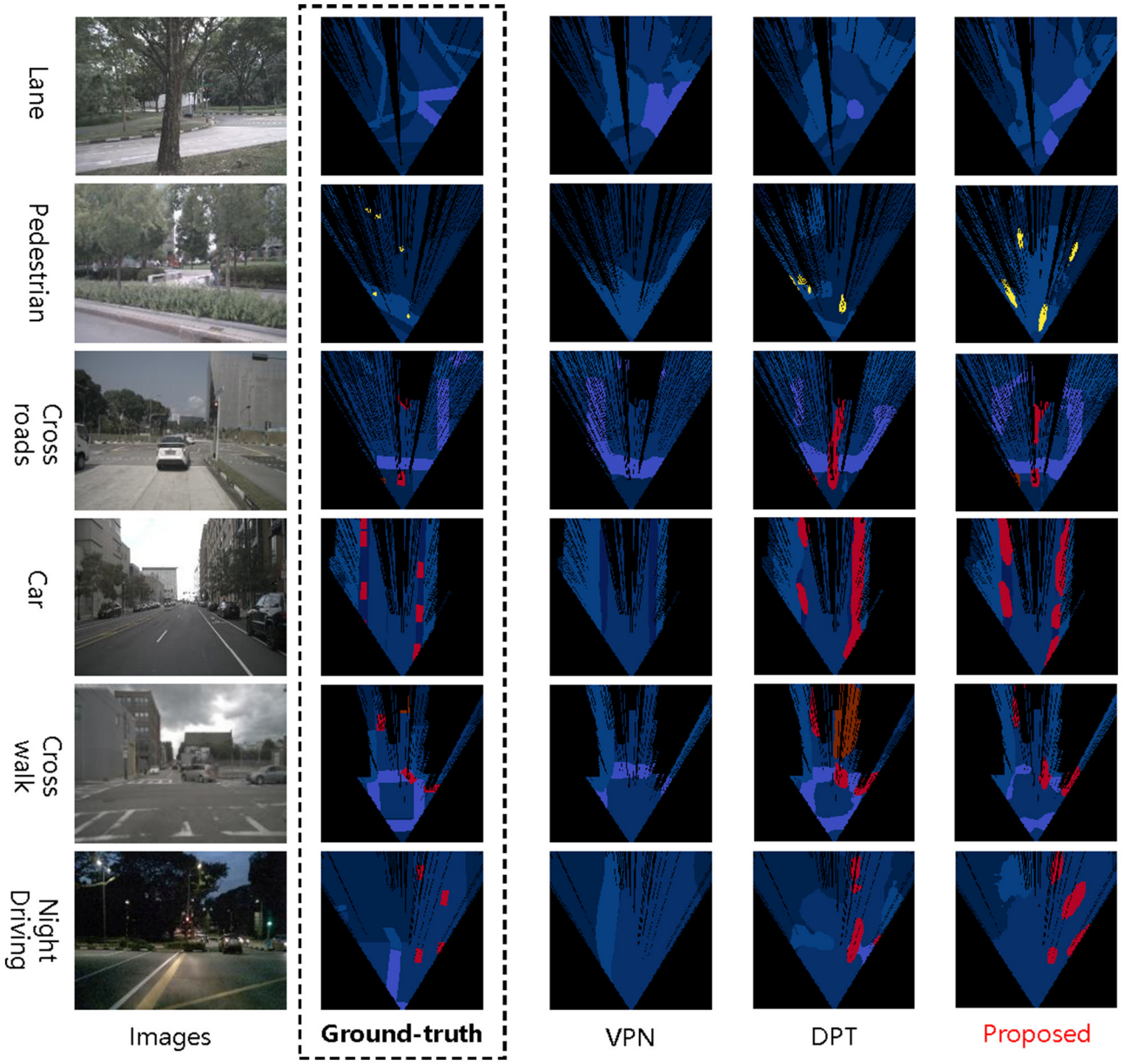


FIGURE 6
Qualitative results on the NuScenes dataset. For a fair comparison, we follow the color scheme utilized in the DPT. We evaluate methods on six challenging scenarios and compare the proposed method with two baselines.

increase in computational resources. This computational scalability is crucial for real-world applications where the model might need to operate in vast and complex environments. This capability could be highly beneficial in numerous practical applications, from autonomous navigation systems to robotics, which requires a nuanced understanding of their surroundings.

5. Conclusions

The paper presents novel neural networks powered by Transformers for BEV representation prediction, which is substantially different from CNN-based networks which are commonly reported in existing works. Our method focuses on map generation through image-plane feature extraction and transformation, without the use of convolution and pooling layers. In this way, per-class element prediction and BEV map generation are implemented through an end-to-end framework. Results demonstrate strong performance on two large-scale benchmarks, i.e., the NuScenes dataset and the Argoverse 3D dataset. The model attains greater accuracy improvement for large-size object prediction (about 2 % IoU) and a breakthrough for small-scale object prediction (about 1 % IoU). Furthermore, the proposed method shows a leading performance in challenging scenarios.

In the future, we will study (1) train Transformer-based networks with less data, (2) memorize more distant global clues,

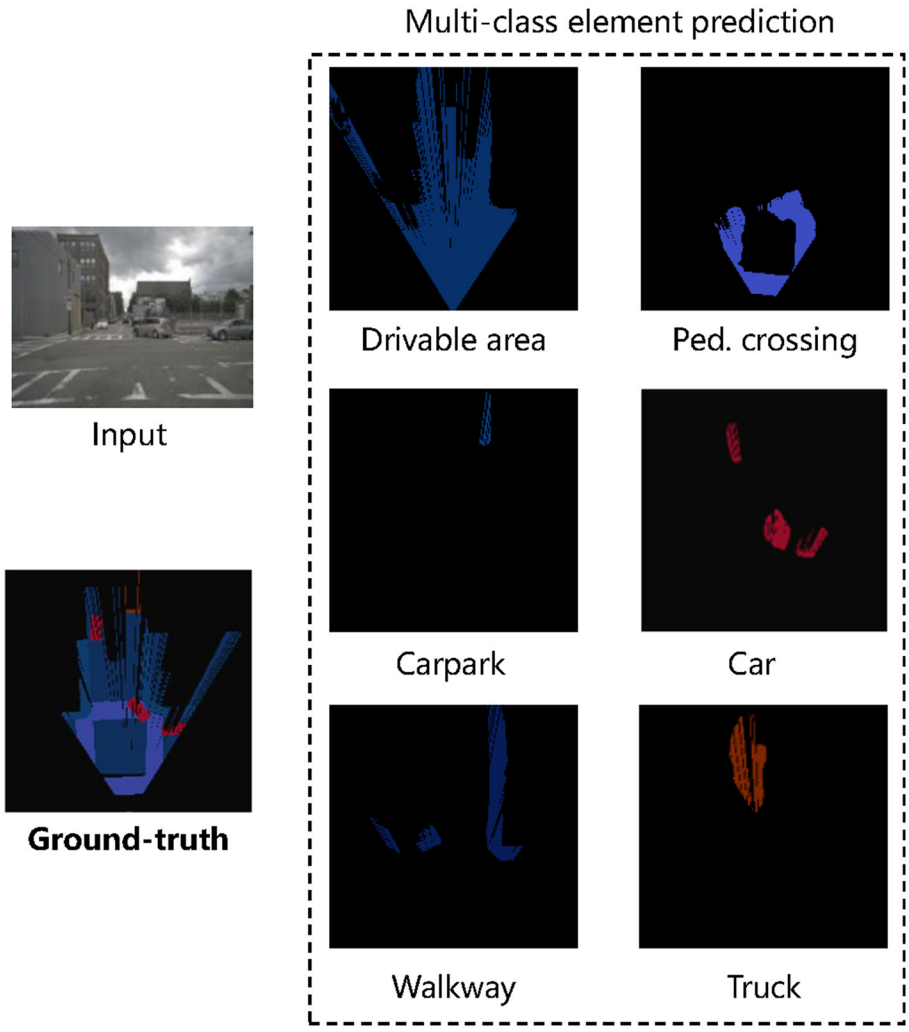


FIGURE 7
Visualization of the multi-class element prediction results on the NuScenes. An autonomous vehicle predicts a BEV representation with six elements in a cloudy scene. The BEV representation of the six elements is present separately.

and (3) build a Transformer-based temporal framework. We argue that the boost in performance of BEV representation prediction depends on spatiotemporal relationship mining, and balancing between data-driven approaches and performance-boosting techniques is key for deep learning.

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

JY: conceptualization, methodology, software, and writing—original draft. WZ: software, validation, formal analysis, and writing—original draft. YC: conceptualization, supervision, writing—review and editing, and funding acquisition. YZ: data curation and validation. RH: project administration. All authors contributed to the article and approved the submitted version.

Funding

The authors would like to acknowledge the support from the Shenzhen Science and Technology Program (JCYJ20210324115604012, JSGG20220606140201003, JCYJ20220818103006012, and ZDSYS20220606100601002), the Guangdong Basic and Applied Basic Research Foundation (2021B1515120008 and 2023A1515011347), and the Institute of Artificial Intelligence and Robotics for Society.

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

Ba, J. L., Kiros, J. R., and Hinton, G. E. (2016). Layer normalization. *arXiv*. [preprint]. doi: 10.48550/arXiv.1607.06450

Caesar, H., Bankiti, V., Lang, A. H., Vora, S., Liong, V. E., Xu, Q., et al. (2020). "NuScenes: a multimodal dataset for autonomous driving," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (Seattle, WA: IEEE), 11621–11631. doi: 10.1109/CVPR42600.2020.01164

Cao, H., Chen, G., Li, Z., Feng, Q., Lin, J., Knoll, A., et al. (2022a). "Efficient grasp detection network with gaussian-based grasp representation for robotic manipulation," in *IEEE/ASME Transactions on Mechatronics*, *Vol. 28* (IEEE), 1384–1394. doi: 10.1109/TMECH.2022.3224314

Cao, H., Chen, G., Li, Z., Hu, Y., and Knoll, A. (2022b). Neurograsp: multimodal neural network with euler region regression for neuromorphic vision-based grasp pose estimation. *IEEE Trans. Instrum. Meas.* 71, 1–11. doi: 10.1109/TIM.2022.3179469

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S., et al. (2020). "End-to-end object detection with transformers," in *European Conference on Computer Vision* (Cham: Springer), 213–229. doi: 10.1007/978-3-030-58452-8_13

Chang, M.-F., Lambert, J., Sangkloy, P., Singh, J., Bak, S., Hartnett, A., et al. (2019). "Argoverse: 3d tracking and forecasting with rich maps," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (Long Beach, CA: IEEE), 8748–8757. doi: 10.1109/CVPR.2019.00895

Chen, C.-F. R., Fan, Q., and Panda, R. (2021a). "Crossvit: cross-attention multi-scale vision transformer for image classification," in *Proceedings of the IEEE/CVF International Conference on Computer Vision* (Montreal, QC: IEEE), 357–366. doi: 10.1109/ICCV48922.2021.00041

Chen, G., Cao, H., Conradt, J., Tang, H., Rohrbein, F., Knoll, A., et al. (2020). Event-based neuromorphic vision for autonomous driving: a paradigm shift for bio-inspired visual sensing and perception. *IEEE Signal Process. Mag.* 37, 34–49. doi: 10.1109/MSP.2020.2985815

Chen, X., Xie, S., and He, K. (2021b). "An empirical study of training self-supervised vision transformers," in *Proceedings of the IEEE/CVF International Conference on Computer Vision* (Montreal, QC: IEEE), 9640–9649. doi: 10.1109/ICCV48922.2021.00950

Deng, L., Yang, M., Li, H., Li, T., Hu, B., Wang, C., et al. (2019). Restricted deformable convolution-based road scene semantic segmentation using surround view cameras. *IEEE Trans. Intell. Transp. Syst.* 21, 4350–4362. doi: 10.1109/TITS.2019.2939832

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]. doi: 10.48550/arXiv.2010.11929

Han, K., Wang, Y., Chen, H., Chen, X., Guo, J., Liu, Z., et al. (2022). A survey on vision transformer. *IEEE Trans. Pattern Anal. Mach. Intell.* 45, 87–110. doi: 10.1109/TPAMI.2022.3152247

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. doi: 10.1109/CVPR.2016.90

Hendy, N., Sloan, C., Tian, F., Duan, P., Charchut, N., Xie, Y., et al. (2020). Fishing net: future inference of semantic heatmaps in grids. *arXiv*. [preprint]. doi: 10.48550/arXiv.2006.09917

Hu, Y., Chen, G., Li, Z., and Knoll, A. (2022). Robot policy improvement with natural evolution strategies for stable nonlinear dynamical system. *IEEE Trans. Cybern.* 53, 4002–4014. doi: 10.1109/TCYB.2022.3192049

Hu, Y., Li, Z., and Yen, G. G. (2023). A knee-guided evolutionary computation design for motor performance limitations of a class of robot with strong nonlinear dynamic coupling. *IEEE Trans. Syst. Man Cybern. Syst.* 53, 4429–4441. doi: 10.1109/TSMC.2023.3249123

Jiang, Y., Chang, S., and Wang, Z. (2021). Transgan: two pure transformers can make one strong gan, and that can scale up. *Adv. Neural Inf. Process. Syst.* 34, 14745–14758. doi: 10.48550/arXiv.2102.07074

Kim, H., Papamakarios, G., and Mnih, A. (2021). "The lipschitz constant of self-attention," *in International Conference on Machine Learning* (PMLR), 5562–5571. Available online at: https://proceedings.mlr.press/v139/kim21i.html

Lee, K., Chang, H., Jiang, L., Zhang, H., Tu, Z., Liu, C., et al. (2021). Vitgan: training gans with vision transformers. *arXiv*. [preprint]. doi: 10.48550/arXiv.2107.04589

Li, X., Hou, Y., Wang, P., Gao, Z., Xu, M., Li, W., et al. (2022). Trear: transformer-based RGB-D egocentric action recognition. *IEEE Trans. Cogn. Dev. Syst.* 14, 246–252. doi: 10.1109/TCDS.2020.3048883

Lin, T.-Y. Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S. (2017). "Feature pyramid networks for object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Honolulu, HI: IEEE), 2117–2125. doi: 10.1109/CVPR.2017.106

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., et al. (2021). "Swin transformer: hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF International Conference on Computer Vision* (Montreal, QC: IEEE), 10012–10022. doi: 10.1109/ICCV48922.2021.00986

Lu, Y., Chen, Y., Zhao, D., Liu, B., Lai, Z., Chen, J., et al. (2021). CNN-G: convolutional neural network combined with graph for image segmentation with theoretical analysis. *IEEE Trans. Cogn. Dev. Syst.* 13, 631–644. doi: 10.1109/TCDS.2020.2998497

Mani, K., Daga, S., Garg, S., Narasimhan, S. S., Krishna, M., Jatavallabhula, K. M., et al. (2020). "Monolayout: amodal scene layout from a single image," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision* (Snowmass, CO: IEEE), 1689–1697. doi: 10.1109/WACV45572.2020.9093519

Mazzia, V., Angarano, S., Salvetti, F., Angelini, F., and Chiaberge, M. (2022). Action transformer: a self-attention model for short-time pose-based human action recognition. *Pattern Recognit.* 124, 108487. doi: 10.1016/j.patcog.2021.108487

Misra, I., Girdhar, R., and Joulin, A. (2021). "An end-to-end transformer model for 3d object detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision* (Montreal, QC: IEEE), 2906–2917. doi: 10.1109/ICCV48922.2021.00290

Ohn-Bar, E., Prakash, A., Behl, A., Chitta, K., and Geiger, A. (2020). "Learning situational driving," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (Seattle, WA: IEEE), 11296–11305. doi: 10.1109/CVPR42600.2020.01131

Pan, B., Sun, J., Leung, H. Y. T., Andonian, A., and Zhou, B. (2020). Cross-view semantic segmentation for sensing surroundings. *IEEE Robot. Autom. Lett.* 5, 4867–4873. doi: 10.1109/LRA.2020.3004325

Philion, J., and Fidler, S. (2020). "Lift, splat, shoot: encoding images from arbitrary camera rigs by implicitly unprojecting to 3D," in *European Conference on Computer Vision* (Cham: Springer), 194–210. doi: 10.1007/978-3-030-58568-6_12

Roddick, T., and Cipolla, R. (2020). "Predicting semantic map representations from images using pyramid occupancy networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (Seattle, WA: IEEE), 11138–11147. doi: 10.1109/CVPR42600.2020.01115

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., et al. (2017). Attention is all you need. *Adv. Neural Inf. Process. Syst.* 30, 1–11.

Wang, D., Devin, C., Cai, Q.-Z. Krähenbühl, P., and Darrell, T. (2019). "Monocular plan view networks for autonomous driving," *in 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Macau: IEEE), 2876–2883. doi: 10.1109/IROS40897.2019.8967897

Wang, K., Ma, S., Chen, J., Ren, F., and Lu, J. (2022). Approaches, challenges, and applications for deep visual odometry: toward complicated and emerging areas. *IEEE Trans. Cogn. Dev. Syst.* 14, 35–49. doi: 10.1109/TCDS.2020.3038898

Wu, W., Sun, W., Wu, Q. M. J., Zhang, C., Yang, Y., Yu, H., et al. (2021). Faster single model vigilance detection based on deep learning. *IEEE Trans. Cogn. Dev. Syst.* 13, 621–630. doi: 10.1109/TCDS.2019.2963073

Yang, M., Yu, K., Zhang, C., Li, Z., and Yang, K. (2018). "Denseaspp for semantic segmentation in street scenes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Salt Lake City, UT: IEEE), 3684–3692. doi: 10.1109/CVPR.2018.00388

Yao, Y., Atkins, E., Johnson-Roberson, M., Vasudevan, R., and Du, X. (2021). Bitrap: bi-directional pedestrian trajectory prediction with multi-modal goal estimation. *IEEE Robot. Autom. Lett.* 6, 1463–1470. doi: 10.1109/LRA.2021.3056339

Yi, D., Fang, H., Hua, Y., Su, J., Quddus, M., Han, J., et al. (2021). Improving synthetic to realistic semantic segmentation with parallel generative ensembles

for autonomous urban driving. *IEEE Trans. Cogn. Dev. Syst.* 14, 1496–1506. doi: 10.1109/TCDS.2021.3117925

Yu, J., Gao, H., Chen, Y., Zhou, D., Liu, J., Ju, Z., et al. (2022). Deep object detector with attentional spatiotemporal lstm for space human?robot interaction. *IEEE Trans. Hum. Mach. Syst.* 52, 784–793. doi: 10.1109/THMS.2022.3144951

Yu, J., Gao, H., Sun, J., Zhou, D., and Ju, Z. (2021). Spatial cognition-driven deep learning for car detection in unmanned aerial vehicle imagery. *IEEE Trans. Cogn. Dev. Syst.* 14, 1574–1583. doi: 10.1109/TCDS.2021.3124764

Zheng, S., Lu, J., Zhao, H., Zhu, X., Luo, Z., Wang, Y., et al. (2021). "Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (Nashville, TN: IEEE), 6881–6890. doi: 10.1109/CVPR46437.2021.00681

Zhu, X., Yin, Z., Shi, J., Li, H., and Lin, D. (2018). "Generative adversarial frontal view to bird view synthesis," in *2018 International conference on 3D Vision (3DV)* (Verona: IEEE), 454–463. doi: 10.1109/3DV.2018.00059

frontiers | Frontiers in Neuroscience

TYPE Original Research
PUBLISHED 05 July 2023
DOI 10.3389/fnins.2023.1190515

Check for updates

OPEN ACCESS

EDITED BY
Simon D. Levy,
Washington and Lee University, United States

REVIEWED BY
Adam Safron,
Johns Hopkins University, United States
Xiangyu Deng,
Dalian University of Technology, China

*CORRESPONDENCE
Nicole Sandra-Yaffa Dumont
✉ ns2dumont@uwaterloo.ca

RECEIVED 21 March 2023
ACCEPTED 16 June 2023
PUBLISHED 05 July 2023

CITATION
Dumont NS-Y, Furlong PM, Orchard J and Eliasmith C (2023) Exploiting semantic information in a spiking neural SLAM system. *Front. Neurosci.* 17:1190515. doi: 10.3389/fnins.2023.1190515

COPYRIGHT
© 2023 Dumont, Furlong, Orchard and Eliasmith. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

Exploiting semantic information in a spiking neural SLAM system

Nicole Sandra-Yaffa Dumont*, P. Michael Furlong, Jeff Orchard and Chris Eliasmith

Centre for Theoretical Neuroscience, University of Waterloo, Waterloo, ON, Canada

To navigate in new environments, an animal must be able to keep track of its position while simultaneously creating and updating an internal map of features in the environment, a problem formulated as simultaneous localization and mapping (SLAM) in the field of robotics. This requires integrating information from different domains, including self-motion cues, sensory, and semantic information. Several specialized neuron classes have been identified in the mammalian brain as being involved in solving SLAM. While biology has inspired a whole class of SLAM algorithms, the use of semantic information has not been explored in such work. We present a novel, biologically plausible SLAM model called SSP-SLAM—a spiking neural network designed using tools for large scale cognitive modeling. Our model uses a vector representation of continuous spatial maps, which can be encoded via spiking neural activity and bound with other features (continuous and discrete) to create compressed structures containing semantic information from multiple domains (e.g., spatial, temporal, visual, conceptual). We demonstrate that the dynamics of these representations can be implemented with a hybrid oscillatory-interference and continuous attractor network of head direction cells. The estimated self-position from this network is used to learn an associative memory between semantically encoded landmarks and their positions, i.e., an environment map, which is used for loop closure. Our experiments demonstrate that environment maps can be learned accurately and their use greatly improves self-position estimation. Furthermore, grid cells, place cells, and object vector cells are observed by this model. We also run our path integrator network on the NengoLoihi neuromorphic emulator to demonstrate feasibility for a full neuromorphic implementation for energy efficient SLAM.

KEYWORDS

simultaneous localization and mapping, semantic SLAM, path integration, spiking neural networks, neuromorphic, hyperdimensional computing, neural engineering framework, semantic mapping

1. Introduction

Simultaneous localization and mapping (SLAM) is the computational process of keeping track of one's location while navigating an unknown environment (i.e., *localization*) and, simultaneously, creating a map of the environment (i.e., *mapping*). Accurate localization is required for building metric map from egocentric observations, but errors in localization accumulate when relying solely on internally generated signals or self-motion (i.e., path integration or dead reckoning). An allocentric environment map can be used to correct these errors, making localization and mapping interdependent processes. SLAM is a core problem in mobile robotics, particularly in applications where high-precision GPS data is not available, such as in autonomous underwater vehicles or planetary exploration (Kim and Eustice, 2013; Palomeras et al., 2019; Geromichalos et al., 2020).

Biological systems have evolved to solve these problems. Animals are capable of navigating and creating maps of novel environments, deducing their current location, and retracing their steps. Considerable research has been conducted to investigate the neural mechanisms underlying spatial cognition in animals. It is known that many animals—including rodents (Mittelstaedt and Mittelstaedt, 1982; Etienne, 1987; Benhamou, 1997), bats (Aharon et al., 2017), and humans (Mittelstaedt and Mittelstaedt, 2001) – are capable of path integration. Tolman (1948) proposed that animals construct "cognitive maps": internal mental constructs used to retain and retrieve information about the relative locations and features of an environment. Such maps are widely believed to be used to discover novel shortcuts and provide corrections to path integration, much like SLAM systems in robots. Indeed, animals have access to a plethora of external sensory information, such as visual landmarks and odor trails, which can be used to correct the errors that would accumulate when using path integration alone. The hippocampal formation is believed to be crucial for such computations, with place cells, head direction cells, and grid cells thought to play significant roles. In fact, Safron et al. (2022) have characterized the hippocampal-entorhinal system as "the most sophisticated of all biological SLAM mechanisms".

While SLAM is a well-studied problem, and modern mobile robots are capable of performing SLAM, animal navigational abilities are still superior; they are more robust, efficient and adaptive, making them more useful in challenging real-world environments. Animals can use information from multiple sensory modalities (e.g., visual, olfactory, auditory, magnetoreception, and idiothetic cues) for navigation. Additionally, animals are able to navigate and map their environment in real-time using power-constrained computational resources, which is something that robots are still not able to achieve—brains are far more energy efficient than the GPUs or CPUs used to execute typical SLAM algorithms. The brain consumes around 20 Watts of energy while a single modern graphics card requires around 350 Watts. By taking inspiration from biology, researchers are trying to develop SLAM algorithms that are optimized for online processing, and that can run on resource-constrained platforms. For instance, neuromorphic hardware—designed to mimic the functionality of biological neural networks—is particularly well-suited for resource-constrained computing because it is designed to be energy-efficient and can perform brain-like computations using minimal resources (Bersuker et al., 2018; Thakur et al., 2018; Rathi et al., 2021).

Biology has influenced the development of a new category of SLAM models that includes RatSLAM (Milford et al., 2004), DolphinSLAM (Silveira et al., 2015), and NeuroSLAM (Yu et al., 2019), among others. Remarkably, some of these models have demonstrated performance comparable to contemporary state-of-the-art approaches. However, this is still an active area of inquiry, with questions remaining regarding scalability and biological plausibility of these approaches, as well as their deployability on neuromorphic hardware. While these types of SLAM algorithms have made notable progress, they have yet to fully explore the wealth of knowledge available from neuroscience and cognitive science. Animals extract and make use of higher-order semantic information about their environment and landmarks from raw sensory inputs while navigating. Recent advancements in robotics have successfully incorporated semantic information into SLAM models (Bowman et al., 2017; Zhang et al., 2018; Chen et al., 2019; Fan et al., 2022). Semantic SLAM models use deep neural networks to extract semantic information to build environment maps. By utilizing higher-level conceptualization of states grounded in cognitive meaning, these models can augment and improve upon purely metric SLAM. Consequently, the construction of maps containing semantic representations empowers such SLAM systems to interact with environments in sophisticated and intelligent ways.

In the same way that biology can aid in the development of AI and robotics, computational modeling can also provide valuable insights into biological research questions. By creating computational models of SLAM that are constrained to be biologically plausible, we can gain a deeper understanding of the neural algorithms that may underlie spatial cognition in animals. For example, we can investigate hypotheses on how exactly cognitive maps may be learned, stored, and used to assist in navigation. Or how multi-modal sensory information is integrated during the construction of cognitive maps. Or how such maps may be accessed and queried to reason about space.

In this work, we unite biologically inspired and semantic SLAM in our model SSP-SLAM, and consider how our computational model can explain neuroscientific observations. Specifically, we present a novel spiking neural network SLAM system, called SSP-SLAM. This model is built using the Neural Engineering Framework (NEF) (Eliasmith and Anderson, 2003) and the Semantic Pointer Architecture (SPA) (Eliasmith, 2013). The NEF provides a systematic method for embedding a state space model into a spiking neural network that can run on neuromorphic hardware. The SPA, which includes Spatial Semantic Pointers (SSPs), provides an approach for representing and processing symbol-like information in connectionist systems. The SPA provides an architecture and "semantic pointer" representations, for characterizing neural processing, including that of symbols, as manipulation of high-dimensional vectors. This enables the development of systems that can learn and reason about symbolic information in a scalable, differentiable, and compositional manner. These methods are used in SSP-SLAM to build environment maps. These maps are core to the functioning of SSP-SLAM, as they integrate semantic information, while being combined with an SSP-based path integrator. The resulting model provides the following contributions:

- We propose and implement a novel spiking neural network SLAM model.
- We constrain our model to only use quantities that are known to be represented in hippocampus, like spatial representations of head direction cells, object vector cells, place cells, and grid cells. Furthermore, biologically plausible, Hebbian-like rules are used to learn an environment map in the form of an associative memory.
- We explore compositional semantic map representations using the SPA and the principles of vector symbolic architectures more broadly. We demonstrate how such a map can be queried post-training to recall what landmarks were in particular areas, recall where landmarks of certain types or

colors were located, and compute (online) the vector between self-position and landmarks in memory.
- We illustrate first steps toward a neuromorphic implementation of our model, showing that the path integration component of SSP-SLAM can be run on an emulator of Intel's Loihi neuromorphic chip.

2. Materials and methods

2.1. The semantic pointer architecture

Our computational SLAM model is built using the tools and principles of the semantic pointer architecture (SPA). This framework has been used to model various cognitive processes, such as action selection (Stewart et al., 2012b), planning (Blouw et al., 2016), memory and free recall (Gosmann and Eliasmith, 2021), and reinforcement learning (Rasmussen and Eliasmith, 2014; Duggins et al., 2022). Furthermore, it has been used to construct a large-scale functional brain model, Spaun (Eliasmith et al., 2012; Choo, 2018), with over 6 million neurons and 20 billion connections. The SPA proposes that *semantic pointers* are the fundamental representations of biological cognition (Eliasmith, 2013). Semantic pointers are spiking neural implementations of high-dimensional vectors that are defined by their compression relations to other neural representations. In the case of cognitive semantic pointers, they can be used to represent concepts, objects, or states, and can be combined in a distributed and compositional manner to represent more complex meanings or structures. By means of the neural engineering framework (NEF), semantic pointers in the SPA are generated by the activities of a collection of spiking neurons. Operations on the underlying vectors are then performed through setting the connections within the spiking neural network. As such, the SPA provides a means to translate symbolic cognitive models into biologically plausible spiking neural networks, and is an approach to neurosymbolic AI. As we will demonstrate, it can be used to build a spiking neural network SLAM model that is deployable on neuromorphic hardware.

2.1.1. Algebra of cognition

The cognitive representations in the SPA are based on *hyperdimensional computing*, also known as *vector symbolic architectures* (VSAs), which bridge symbolic and connectionist approaches to AI. A VSA is any computing framework in which symbols and structured compositions of symbols are represented as high-dimensional vectors. The VSA includes a set of algebraic operations, defined over the vector space, that correspond to operations on the underlying symbols, effectively creating an algebraic language for cognition. The key operations that define this algebra are a similarity measure, a hiding operation, a bundling operation, a binding operation, and an inverse operation. The specification of these operations differentiates particular VSAs. In this work, we implement the SPA using Holographic Reduced Representations (HHRs; Plate, 1995), realized in spiking neural networks.

The *similarity measure* between two semantic pointers indicates the semantic similarity of the symbols they represent. This is given by the cosine similarity (or dot product), which is also the measure for semantic similarity used in many vector encodings in machine learning (Mikolov et al., 2013). The *bundling* operation is addition, and is used to group semantic pointers in a set. The *binding* and *hiding* operations are used to combine symbols together (e.g., combining a slot and filler, to have a single slot-filler representation). In HRRs, binding and hiding are done by one operation, circular convolution,

$$A \circledast B = \mathcal{F}^{-1}\{\mathcal{F}\{A\} \odot \mathcal{F}\{B\}\}, \tag{1}$$

where $\mathcal{F}$ is the Discrete Fourier Transform (DFT), and $\odot$ is the Hadamard product. The *inverse* operation takes a single input vector and produces a single output vector that reverses the effect of binding with the input vector, $(A \circledast B) \circledast B^{-1} = A$. In HRRs, an easy-to-compute and numerically stable approximate inverse (involution) is frequently used. It is defined as $B^{-1} = [B_1, B_d, B_{d-1}, \ldots, B_2]$.

To understand how these operations are used to compose and reason about structured representations, consider a concrete example. Let X denote the semantic pointer representation of the concept X. The sentence, "a brown cow jumped over the moon", can thus be represented via binding and bundling operations as follows:

$$\begin{aligned}\texttt{SUBJECT} \circledast (\texttt{COLOR} \circledast \texttt{BROWN} + \texttt{ANIMAL} \\ \circledast \texttt{COW}) + \texttt{VERB} \circledast \texttt{JUMP} + \texttt{OBJECT} \circledast \texttt{MOON}\end{aligned} \tag{2}$$

The semantic pointer representations of various slots (e.g., subject, color, verb) are bound with the semantic pointers representations of various fillers (e.g., cow, jump), all of which are summed together to represent their collection in a single sentence. The final vector can be queried via the inverse operation to retrieve information. For example, by binding the final vector with $\texttt{VERB}^{-1}$ we can approximately obtain the semantic pointer JUMP.

Typically, VSAs have been used to represent discrete symbols with a one-to-one mapping used to translate between symbols and vectors. Random high-dimensional vectors are often used. Certain models have employed machine learning techniques to obtain vector embeddings with desired similarity characteristics (Mitrokhin et al., 2020). In recent years, VSAs have been extended to represent continuous features using a mapping conceived as a fractional version of the binding operator.

2.1.2. Spatial semantic pointers

Spatial Semantic Pointers (SSPs) extend VSAs to support representation of continuous features (Komer et al., 2019). Here, the mapping from input features to an output vector, $\mathbb{R}^m \rightarrow \mathbb{R}^d$, is explicitly defined. A d-dimensional SSP representing an m-dimensional variable $\mathbf{x}$ is given by

$$\phi(\mathrm{x}) = \mathcal{F}^{-1}\left\{e^{iA\mathrm{x}}\right\} \tag{3}$$

where $A \in \mathbb{R}^{d \times m}$ is the encoding matrix of the representation, $A\mathrm{x}$ is a d-vector, and $e^{iA\mathrm{x}}$ is a vector of d complex numbers. The dot products of x with a fixed set of d vectors—the rows of the encoding matrix—are cast as the phases of complex exponentials to obtain the high dimensional SSP useful for hyperdimensional

computing. There is freedom in the selection of this matrix. However, to ensure the SSP is real-valued, the encoding matrix must be chosen so that e^{iAx} is conjugate symmetric. Though originally SSPs were developed as a fractional extension to the binding operator of HRRs, the resulting mapping is similar to the encoding used in Random Fourier Features (RFF), a popular method for approximating kernels in machine learning (Rahimi and Recht, 2007; Furlong and Eliasmith, 2022).

A useful property of SSPs is that binding in the SSP space is equivalent to addition in the variable space,

$$\phi(\mathrm{x}) \circledast \phi(\mathrm{x}') = \mathcal{F}^{-1}\left\{e^{iA\mathrm{x}} \odot e^{iA\mathrm{x}'}\right\} = \phi(\mathrm{x}+\mathrm{x}'). \tag{4}$$

Thus, it is easy to "update" SSP representations without any decoding. For instance, it is easy to "move" an object located somewhere with one or more binding operations.

Generally, the SSP representation of a number is similar to nearby numbers (in terms of Euclidean distance), and dissimilar to distant ones—with some rippling effects. As a result, similarity between SSPs provides a method for visualizing these high-dimensional vectors (see Figure 1). The similarities between a particular SSP and a set of SSPs that represent points gridded over m-dimensional space can be computed and plotted. We refer to such plots as similarity maps. For example, a similarity map of an SSP, ϕ', representing a 1−D variable is a plot of x vs. $\phi' \cdot \phi(x)$, which has been shown to be a sinc function in the limit $d \to \infty$ (Voelker, 2020). For SSPs representing 2D variables, a similarity map can be depicted as a surface plot or a heat map as in Figure 1.

A primary advantage of the SSP representation is that it can be used in combination with semantic pointers encoding discrete symbols. Figure 1 provides a concrete demonstration of such a representation. Consider a simple 2D environment consisting of different objects and landmarks: a robot, two boxes, and a wall (see Figure 1A). The position of the robot, $(x_1,\ y_1)$, can be encoded as a SSP, $\phi(x_1,\ y_1)$. This, in turn, can be bound (i.e., circularly convolved) with the semantic pointer representing the *concept* of a robot, ROBOT, to obtain $\mathsf{ROBOT} \circledast \phi(x_1,\ y_1)$—this represents a robot at a particular location. Likewise, the semantic pointer for a tool box can be bound with SSPs encoding their locations, $(x_2,\ y_2)$ and $(x_3,\ y_3)$, to obtain $\mathsf{BOX} \circledast \big(\phi(x_2,\ y_2) + \phi(x_3,\ y_3)\big)$; in this case, the sum of SSPs is used to represent a set of locations. The wall in the environment covers an area D, which can be represented by integrating the SSP encoding over that area, $\iint_D \phi(x,\ y)dxdy$. All together, the complete environment can be represented by adding all of these object-location vector encodings:

$$M = \mathsf{ROBOT} \circledast \phi(x_1,\ y_1) + \mathsf{BOX} \circledast \big(\phi(x_2,\ y_2) + \phi(x_3,\ y_3)\big) + \mathsf{WALL} \circledast \iint_D \phi(x,\ y)dxdy \tag{5}$$

This vector was constructed and "queried" for different locations with approximate unbinding. The results of this unbinding are shown in Figures 1B–D. The high-dimensional SSPs are visualized in this figure via their similarity to neighboring points.

2.1.3. Probability representations

Recent work has demonstrated that the algebra of cognition defined by VSAs and SSPs has a probabilistic interpretation (Furlong and Eliasmith, 2022). Using the tools provided by the NEF, it is possible to construct spiking neural networks that embody probability distributions and perform computations related to probability, such as determining entropy and mutual information (Furlong et al., 2021; Furlong and Eliasmith, 2023).

In particular, SSPs can be used for kernel density estimation (KDE), a non-parametric method used for estimating a probability density function of a random variable X. To approximate a PDF f given a set of samples, $\{\mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_n\}$, drawn from an unknown distribution, one can average kernel functions around each data point, $k(\mathrm{x} - \mathrm{x}_i)$, to obtain a smooth estimate, $\hat{f}$, of the underlying PDF:

$$\hat{f}(\mathrm{x}) = \frac{1}{n}\sum_{i=1}^{n} k(\mathrm{x} - \mathrm{x}_i). \tag{6}$$

KDE has the advantage of being non-parametric and flexible, allowing the estimation of complex and multi-modal distributions. However, the choice of the kernel function is crucial for the accuracy and smoothness of the estimate. Common kernel functions used in KDE include the Gaussian, Epanechnikov, and triangle kernels.

The similarity, or dot product, between SSPs approximates a sinc kernel function. Consequently, we can define $k(\mathrm{x} - \mathrm{x}_i) = \phi(\mathrm{x}) \cdot \phi(\mathrm{x}_i)$. Our KDE is given by $\frac{1}{n}\sum_{i=1}^{n} \phi(\mathrm{x}) \cdot \phi(\mathrm{x}_i) = \phi(\mathrm{x}) \cdot M_n$, where $M_n = \frac{1}{n}\sum_{i=1}^{n} \phi(\mathrm{x}_i)$ is the average over datapoint SSP representations. Unlike the kernels listed above, the normalized sinc can take on negative values, but it can be used to obtain probability densities with a simple correction,

$$\hat{f}(\mathrm{x}) \approx (\phi(\mathrm{x}) \cdot M_n - \xi)^+ \tag{7}$$

where ξ is a fixed scalar chosen so that $\int_{-\infty}^{\infty}(\phi(\mathrm{x}) \cdot M_n - \xi)^+ d\mathrm{x} = 1$ (Glad et al., 2003, 2007). Note that this is simply a ReLU neuron with bias ξ, and either weights M_n and input $\phi(\mathrm{x})$, or vice versa—weights $\phi(\mathrm{x})$ and input M_n. In the former case, a population of many such neurons (with varying incoming synaptic weights M_n) can be interpreted as estimating the probability of a query $\phi(\mathrm{x})$ under different distributions. In the later case, the activities of a population of neurons would represent the probabilities of different sample points x under a given input distribution represented by SSPs, M_n. Notably, the sinc estimate is often more accurate than the "optimal" Epanechinikov estimate (Section 1.3, Tsybakov, 2009).

Using SSPs for neural probability computations in this way results in different interpretations of the standard VSA operations, which are useful in the context of SLAM. Under this interpretation, bundling is used to add new datapoints to a running mean M_n, and is a kind of belief update, binding can be used for multivariate KDE, and the inverse operation that performs unbinding is analogous to conditioning.

2.1.4. The neural engineering framework

The SPA is not just a VSA, but rather a full architecture that includes a variety of functional components, as well as the neural instantiation of a VSA. To create spiking neural networks that implement algorithms involving VSAs and SSPs, we require methods to embed vector representations into the

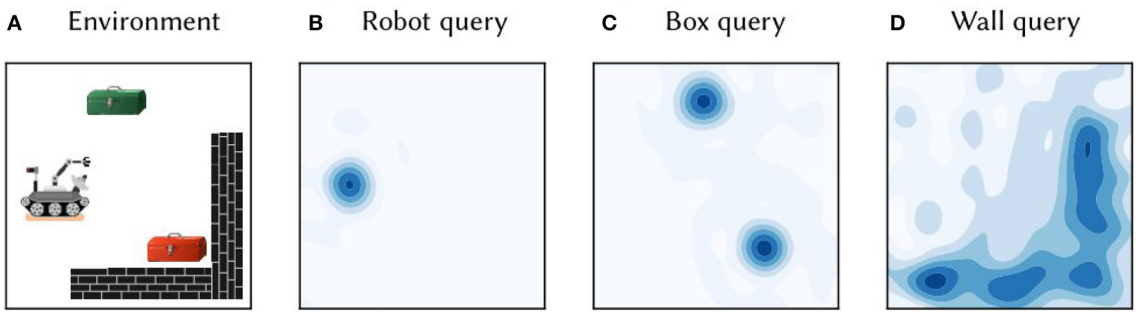


FIGURE 1
(A) A toy 2D environment consisting of a robot, boxes, and walls. Information about the objects and their locations was encoded in single vector M, as per Equation (5). **(B)** The vector M was queried for the location of the robot by approximate unbinding: $M \circledast \text{ROBOT}^{-1} \approx \phi(x_1, y_1)$. The heat map shows the cosine similarity between the query output and SSP representations of points gridded over 2D space. **(C)** The similarity map obtained from querying the map M for the location of boxes. **(D)** The similarity map obtained from querying for the wall area.

activity of spiking neurons, and to be able to perform computations on these vectors via projections between neural populations. The NEF provides such methods, which are described by three primary principles.

The first principle of *representation* specifies how the collective neural activity of a population encodes a vector and vectors can be decoded out of spike trains. The activity of neuron i in a population encoding a vector, $\phi \in \mathbb{R}^d$, is given by,

$$a_i(t) = \mathcal{G}_i \left[\alpha_i \boldsymbol{e}_i \cdot \phi + \beta_i\right], \tag{8}$$

where $\alpha_i > 0$ is the neuron's gain, β_i is its bias, $\boldsymbol{e}_i$ is its encoder, and $\mathcal{G}_i$ is a non-linear function—in this work, the leaky-integrate-and-fire (LIF) function. The gain and bias parameters vary amongst neurons to create a heterogeneous population. Encoders determine the type of input a specific neuron is responsive to, thus capturing the neuron's "receptive field". In the case of a neural population representing SSPs, it is reasonable to set encoders as SSPs that represent random points in space. This produces a population of neurons that are sensitive to specific spatial locations—i.e., place cells. Other types of spatial sensitive neurons can be constructed using different neural encoders and SSP encoding matrices. In Dumont and Eliasmith (2020), grid cells were obtained this way.

A vector represented by the activity of a population of N neurons can be decoded from a linear combination of the spiking neural activity after post-synaptic filtering:

$$\hat{\phi} = \sum_{i=1}^{N} a_i(t) * h(t)\boldsymbol{d}_i, \tag{9}$$

where $*$ is convolution and $\boldsymbol{d}_i \in \mathbb{R}^d$ are called the decoders of the population. Least-squares optimization is typically used to solve for the decoders. The function $h(t)$ is a post-synaptic filter and is parameterized by τ_{syn}, the post-synaptic time constant:

$$h(t) = \begin{cases} e^{-t/\tau_{syn}} & \text{if } t > 0 \\ 0 & \text{otherwise.} \end{cases} \tag{10}$$

The second principle of the NEF, *transformation*, provides the method for setting weights between two neural populations to compute a desired function. Assume a population of N neurons representing a vector, ϕ, is fully connected to a different population of N' neurons. Suppose we would like second population to represent some function of the vector, $f(\phi)$. This function can be decoded out of the first population's activity,

$$\hat{f}(\phi) = \sum_{i=1}^{n} a_i(t) * h(t)\boldsymbol{d}_i^{(f)}. \tag{11}$$

These function-specified decoders, $\boldsymbol{d}_i^{(f)}$, can be solved for using samples of the desired function output or, if sample outputs are not available, decoders can be learned online in response to error signals (see Section 2.1.5). Decoding the output of the first population and encoding it into the activity of the second population is equivalent to multiplying the filtered activities of the first population with a weight matrix and feeding that current into the second population, which will have activities given by

$$b_j(t) = \mathcal{G}_i\left[\sum_{i=1}^{N} w_{ij} a_i(t) + \beta_j\right], \quad w_{ij} = \alpha_j \boldsymbol{e}_j \times \boldsymbol{d}_i^{(f)} \tag{12}$$

where $\times$ is an outer product.

The result is a standard neural network, with populations connected via weighted synapses. The NEF provides a method to generate the weight matrices that are the outer product between the decoders of the first population (which are optimized) and the encoders of the second (which are pre-set, usually to match biological tuning curves).

The last principle of the NEF is *dynamics*. Dynamical systems can be embedded into in a recurrently connected population of spiking neurons using this principle. The NEF proposes that to implement a non-linear dynamical system $\dot{\phi} = f(\phi) + g(u)$ (where u is some input signal), the incoming connection from the population representing the input u must compute the transform $\tau g(u)$ (where τ is the post-synaptic time constant), and the recurrent connection from the population representing S to itself must compute the transform $\tau f(\phi) + \phi$. This is due to the use of post-synaptic filters. This principle allows us to embed a wide variety of non-linear dynamical systems into spiking neural networks, which we exploit in Section 2.2.1.

2.1.5. Learning rules

Biologically plausible learning rules that only use local information can be used in the NEF for modifying synaptic weights online. The Prescribed Error Sensitivity (PES) (MacNeil and Eliasmith, 2011) is an error-driven learning rule in which, to learn a connection between a pre- and post-population of neurons, the pre-population's decoders are modified in response to an error signal:

$$\Delta \boldsymbol{d}_i = \kappa \boldsymbol{E} a_i, \tag{13}$$

which is equivalent to modifying weights by

$$\Delta w_{ij} = -\kappa \alpha_j \boldsymbol{e}_j \cdot \boldsymbol{E} a_i \tag{14}$$

where κ is a learning rate, a_i are pre-population neural activities (filtered spikes), α_j are post-population gains, $\boldsymbol{e}_j$ are the post-population encoders, and $\boldsymbol{E}$ is an error signal we seek to minimize. This signal may be computed by other neural populations in a model. Biologically, we can think of those populations as dopaminic neurons that can modify weights in this way via dopamine levels. Real data of spike timing dependent plasticity is matched by PES when used in combination with the unsupervised Bienenstock, Cooper, Munro (BCM) learning rule, which sparsifies weights (Bekolay et al., 2013).

Another, unsupervised, learning rule is the Oja learning rule (Oja, 1982), which modifies the Hebbian learning rule in order to improve stability. The vector version of this rule, the "Voja" learning rule, shifts encoders so that neurons fire selectively at particular inputs and activity is sparsified:

$$\Delta \boldsymbol{e}_i = \kappa a_i (\mathrm{x} - \boldsymbol{e}_i). \tag{15}$$

This rule has been used for training heteroassociative memory networks (Voelker et al., 2014), and is used in SSP-SLAM, along with the PES rule, to train an associative memory.

2.2. The SSP-SLAM model

In this paper, we develop a spiking neural network SLAM model using semantic pointers, SSPs and the NEF. The model, SSP-SLAM, consists of six main neural populations, grouped into four modules, that provide all the necessary functionality.

- **Localization module**
  - *Path integrator:* A network maintaining an allocentric self-position estimate, represented as a SSP $\hat{\phi}(\mathrm{x}(t))$, that is dynamically updated using a velocity signal. Specifically, this is a recurrent neural network, consisting of many sub-populations representing controlled oscillators that contain heading direction cells.
  - *Grid cell (GC) population:* A population representing a "cleaned-up" version of the SSP self-position estimate, $\phi(\hat{\mathrm{x}}(t))$.
- **Landmark perception module**
  - *Object vector cell (OVC) population:* A population that encodes the SSP representation of distances and directions to landmarks and environmental features in view—i.e., an egocentric representation of feature locations.
  - *Object location (OL) population:* A population that performs circular convolution to obtain an allocentric SSP representation of feature locations.
- **Environment map module**
  - *Associative memory (AN) network:* A network that learns a mapping between landmarks and locations using the biologically plausible PES and Voja learning rules.
- **Loop closure module**
  - *Map estimate (ME) population:* A population that performs circular convolution to obtain an alternative estimation of self-position using the environment map. This provides corrections to the path integrator.

Each element of the SSP-SLAM model is described in more detail below and a high-level overview of the model is given in Figure 2.

2.2.1. Localization module

In prior work, we have used SSPs to maintain a neural estimate of an agent's self-position while navigating an environment (Voelker et al., 2021; Dumont et al., 2022). To build a network that maintains an encoding of position, consider how $\phi(\mathrm{x})$ changes if x is a function of time. We can relate the rate of change of ϕ to the velocity $\dot{\mathrm{x}}(t)$:

$$\dot{\phi}(\mathrm{x}(t)) = \mathcal{F}^{-1}\{e^{iA\mathrm{x}(t)} \odot iA\dot{\mathrm{x}}(t)\}, \tag{16}$$

where $\odot$ is element-wise multiplication. Now consider the dynamics of an SSP in the Fourier domain. Taking the Fourier transform of Equation (16), we get,

$$\mathcal{F}\{\dot{\phi}(\mathrm{x}(t))\} = \left(iA\dot{\boldsymbol{x}}\right) \odot \mathcal{F}\{\phi(\mathrm{x}(t))\}. \tag{17}$$

Note that the dynamics of the Fourier components of an SSP are independent of one another. The dynamics of the j^{th} Fourier coefficient of the SSP can be written as

$$\frac{d}{dt}\begin{bmatrix} \mathrm{Re}\mathcal{F}\{\phi(\mathrm{x})\}_j \\ \mathrm{Im}\mathcal{F}\{\phi(\mathrm{x})\}_j \end{bmatrix} = \begin{bmatrix} 0 & -\omega_j \\ \omega_j & 0 \end{bmatrix} \begin{bmatrix} \mathrm{Re}\mathcal{F}\{\phi(\mathrm{x})\}_j \\ \mathrm{Im}\mathcal{F}\{\phi(\mathrm{x})\}_j \end{bmatrix}, \quad \text{where } \omega_j \equiv A_{j,:} \cdot \dot{\mathrm{x}}(t) \tag{18}$$

Each Fourier coefficient of the SSP is thus a simple harmonic oscillator. The real and imaginary components of the Fourier coefficients of the SSP oscillate about the unit circle with time-varying frequency $\omega_j = A_{j,:} \cdot \dot{\mathrm{x}}(t)$. The oscillator frequencies are modulated by the velocity; in other words, they are velocity controlled oscillators (VCOs). In our model, we modify the

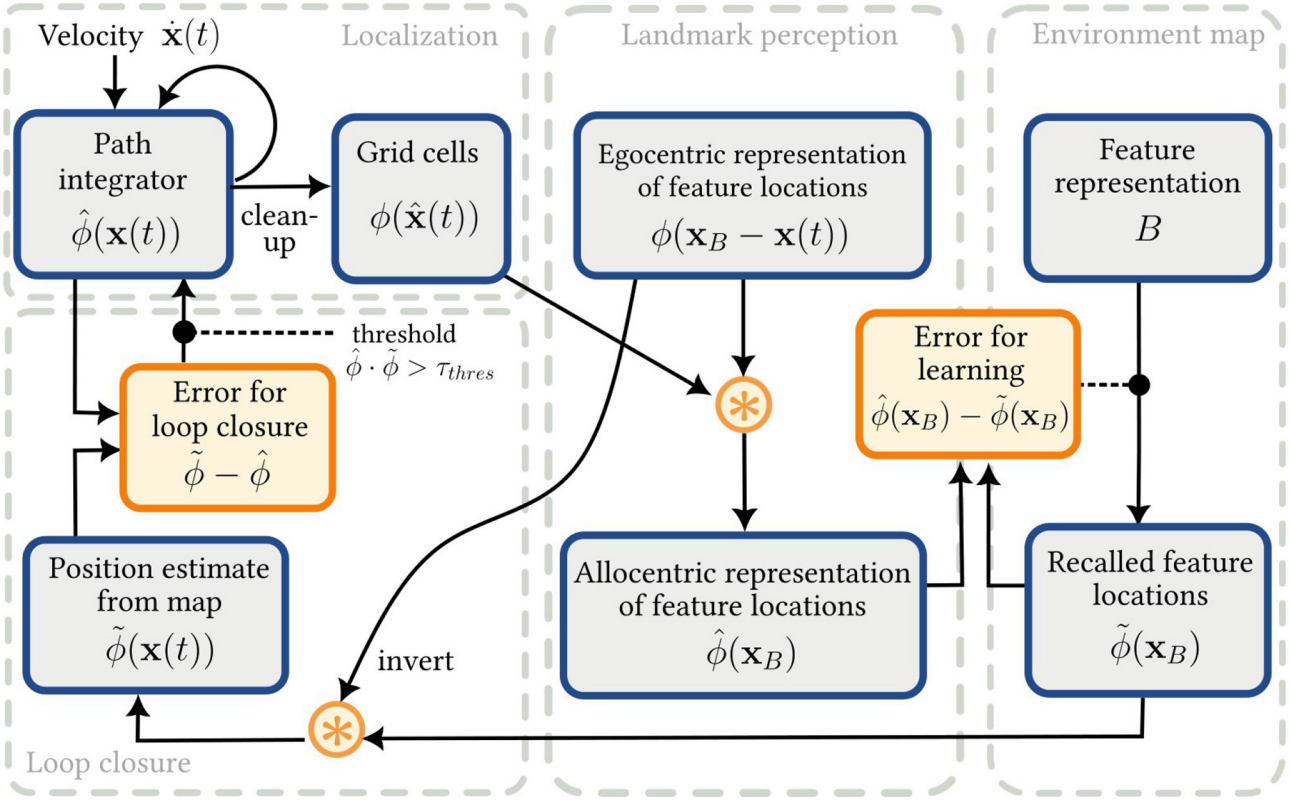


FIGURE 2
The SSP-SLAM model. Output of the localization module is used (along with the egocentric feature locations encoded by the OVC population) to train an associative memory network, which can be thought of as an environment map. The output of this map is, in turn, used for error correction of the PI model.

dynamics of Equation (18) so that the unit circle is an attractor and the oscillators self-stabilize:

$$\frac{d}{dt}\begin{bmatrix}\text{Re}\mathcal{F}\{\phi(\text{x})\}_j \\ \text{Im}\mathcal{F}\{\phi(\text{x})\}_j\end{bmatrix} = \begin{bmatrix}-\omega_j \text{Im}\mathcal{F}\{\phi(\text{x}(t))\}_j + \frac{1-r_j^2}{r_j}\text{Re}\mathcal{F}\{\phi(\text{x})\}_j \\ \omega_j \text{Re}\mathcal{F}\{\phi(\text{x}(t))\}_j + \frac{1-r_j^2}{r_j}\text{Im}\mathcal{F}\{\phi(\text{x})\}_j\end{bmatrix}, \tag{19}$$

where $r_j \equiv |\mathcal{F}\{\phi\}_j|$.

This reduces drift and ensures the entire SSP vector remains unit length. Thus, our path integrator is a hybrid between continuous attractor and oscillatory inference models of path integration.

To realize this representation and dynamics in a spiking neural network, we use the tools of the NEF as described above. The SSP estimate of self-position is encoded in $\lfloor \frac{d}{2} \rfloor$ recurrent populations of spiking neurons, each of which is a VCO. Only $\lfloor \frac{d}{2} \rfloor$ VCO populations are needed since the Fourier transform of the SSP has conjugate symmetry (half of its Fourier components can be computed from the other half).

To compute the non-linearities bet ween the frequency and SSP Fourier coefficients, we must represent both in a single population, as is standard in the NEF. The vector being represented by the collective activity of the j^{th} VCO population is,

$$\left[\omega_j \ \text{Re}\mathcal{F}\{\hat{\phi}(\text{x})\}_j \ \text{Im}\mathcal{F}\{\hat{\phi}(\text{x})\}_j\right]^T. \tag{20}$$

We write $\hat{\phi}(\text{x})$ here to emphasize that this is an *estimate* of $\phi(\text{x})$. Due to noise inherent in neural encoding and the dynamics being approximated by recurrent connections rather than being computed exactly, this estimate will drift from the SSP encoding of the ground truth position over time. Indeed, the vector encoded by the path integrator will even drift from the sub-space of SSP vectors in $\mathbb{R}^d$ without some form of correction.

A population of speed- and heading-direction cells that encode the agent's velocity projects onto the VCO populations. The connection weights compute the linear transform needed to obtain the input frequencies, $A\dot{\text{x}}(t) = \boldsymbol{\omega}$. Each VCO neural population is recurrently connected to itself with weights optimized by least squares to implement the dynamics of Equation (19).

An advantage of this model is its ability to perform localization in space of different dimensionality without major modification. Consider the SSP representation of a $\text{x}(t) \in \mathbb{R}^m$, given by $\phi(\text{x}(t)) \in \mathbb{R}^d$, compared to an SSP of the same dimension d, but encoding a higher dimension variable, $\phi(\boldsymbol{y}(t))$ where $\boldsymbol{y}(t) \in \mathbb{R}^p$ and $p \neq m$. In either case, the dynamics of the SSP are given by Equation (18). The same set of VCO populations can model the dynamics of $\phi(\text{x}(t))$ and $\phi(\boldsymbol{y}(t))$ – the only difference between their computations is the calculation of the frequencies, ω_j, used in the VCOs. These frequencies are input to the path integrator, with incoming synaptic weights performing the linear transformation from self-motion to frequencies, $A\dot{\text{x}}(t)$. The same path integrator network can receive input from multiple sources, with synaptic

gating used to switch between localization in different dimensional spaces and coordinate frames.

Note that the VCO populations consist of spatially sensitive neurons, but these neurons will not resemble place or grid cells. Each oscillator is a population representing a frequency (derived from velocity) and a single Fourier coefficient of the SSP. This results in neurons with conjunctive sensitivity to heading direction, speed, and spatial position (in a periodic fashion, resembling a plane wave). Their firing patterns are velocity dependent bands or stripes. Banded cells have been predicted by other VCO models (Burgess, 2008) and have been a point of contention since reports of band cells in the hippocampal formation are limited, and their existence is controversial (Krupic et al., 2012; Navratilova et al., 2016). Additionally, grid cells do not intrinsically emerge from PI in the model presented in this section. Nevertheless, SSPs naturally represent grid cells, and we use such a population to represent the collective output of all VCOs after a clean-up operation and provide a better basis for the downstream construction of place cells and spatial maps (Orchard et al., 2013; Dumont and Eliasmith, 2020). This is not unwarranted, given the observations from the MEC. The deeper layers of the MEC receive hippocampal output [along with input from many other cortical areas (Czajkowski et al., 2013)], and is where head-direction cells, speed cells, and conjunctive grid cells are primarily located (Witter and Moser, 2006). The superficial layers of the MEC, specifically layer II, mainly provide input to the hippocampus and consist mostly of "pure" grid cells (Sargolini et al., 2006). This suggests that the deeper layers and head direction cells may play a crucial role in integrating external input, much like the path integrator network in SSP-SLAM. The output of this integration is then processed into more stable, purely spatial representations in the superficial layers, like the grid cell population in SSP-SLAM, which are used for downstream tasks. However, this narrative is subject to debate, and not universally accepted.

As described in Section 2.1.3, SSPs can be used to construct probability distributions. When performing path integration, we are interested in obtaining an estimate of the agent's position at a given point in time. Let $\hat{\phi}(\mathrm{x}(t))$ be the vector represented by the path integrator network at time t. The network is initialized to encode the SSP $\phi(\mathrm{x}(0))$, from which a prior probability distribution can be computed. At every simulation time step this belief state is updated according to the dynamics given in Equation (19). Then, the probability density of the agent being at a location $\hat{\mathrm{x}}$ is $\hat{f}(\hat{\mathrm{x}}) \approx (\phi(\hat{\mathrm{x}}) \cdot \hat{\phi}(\mathrm{x}(t)) - \xi)^{+}$. The position estimate of the path integration model is taken to be the $\hat{\mathrm{x}}$ that maximizes this posterior distribution, i.e., the maximum a posteriori probability (MAP) estimate. A simple example path decoded in this manner is shown in Figure 3. The SSP representation of the MAP estimate, $\phi(\hat{\mathrm{x}})$, is computed as a part of the "clean-up" process applied to the output of the VCOs to obtain the input to the grid cell population.

2.2.2. Landmark perception module

In the SSP-SLAM model, the agent not only receives a self-velocity signal as input, but additionally receives observations of its local environment. As an animal moves through space, sensory systems and other brain regions provide information about its surroundings. The inferotemporal cortex, for example, plays a vital role in object recognition (Rajalingham and DiCarlo, 2019), and populations in the medial entorhinal cortex (MEC) appear to encode vectors to nearby objects (Høydal et al., 2019). It is possible to create a spiking neural network that uses raw sensory data to recognize objects and estimate their displacement from the observer, though it remains an active area of research. For example, Osswald et al. (2017) presented a spiking neural network model and neuromorphic demonstration of stereo-correspondence in 3D space. Spiking neural algorithms for object detection (Kim et al., 2020b) and place recognition (Hussaini et al., 2022) have also been developed. Moreover, deep learning has proven to be highly successful in computer vision tasks such as semantic segmentation (Lateef and Ruichek, 2019), and these pre-trained artificial neural networks can be converted to spiking neural networks (Cao et al., 2015). However, in this work, visual processing of raw sensory data is out of scope. Instead, we assume that information regarding distance to landmarks and landmark identity is provided directly as input to SSP-SLAM.

Specifically, we let $\{B_1, B_2, \dots\}$ be a set of semantic pointers representing features or landmarks in an environment, at locations $\{\mathrm{x}_1, \mathrm{x}_2, \dots\}$. The input to SSP-SLAM uses these representations to determine the SSP representation of the vector from the agent to each landmark within the agent's field of view, $\phi(\mathrm{x}_i - \mathrm{x}(t))$. In short, the input is represented in a population that encodes an egocentric representation of landmark locations that will change over time as the agent passes by landmarks. The neurons in this population have activity patterns like those of object vector cells (OVCs) in the MEC, so we call the population the OVC population. The output of the path integrator and OVC populations are bound together to compute allocentric features locations, $\hat{\phi}(\mathrm{x}(t)) \circledast \phi(\mathrm{x}_i - \mathrm{x}(t)) = \hat{\phi}(\mathrm{x}_i) \approx \phi(\mathrm{x}_i)$. This is stored in the object location (OL) population.

As with path integration positions, the allocentric SSP estimate of an landmark location, $\hat{\phi}(\mathrm{x}_i)$, can be converted into probabilities. The probability density of landmark B_i being at a location x is $(\phi(\mathrm{x}) \cdot \hat{\phi}(\mathrm{x}_i) - \xi)^{+}$ (see Figure 4 for examples).

2.2.3. Environment map module

In SSP-SLAM, an environment map is stored in the weights of a heteroassociative memory network. This memory network architecture was first presented in Voelker et al. (2014). It is a neural population that maps input to some desired association. The PES learning rule, given in Equation (13), is used to train the decoders (i.e., the outgoing synaptic weights) of the population. Concurrently, the Voja learning rule, given in Equation (15), is used to modify the population's encoders. This shifts neurons' encoders to be more similar to input they receive. It results in sparser representations in the population, which helps prevent catastrophic forgetting or interference.

Networks that map between landmarks and locations can be thought of as encoding a cognitive map. In SSP-SLAM, several landmark-location mappings are of interest. The associative memory network just described maps features in the agent's field of view (e.g., objects, landmarks, barriers, colors, etc.) to the current estimate of those feature's locations as SSPs, $\hat{\phi}(\mathrm{x}_i)$. Notably, these environmental features can be structured representations. For example, vector representations

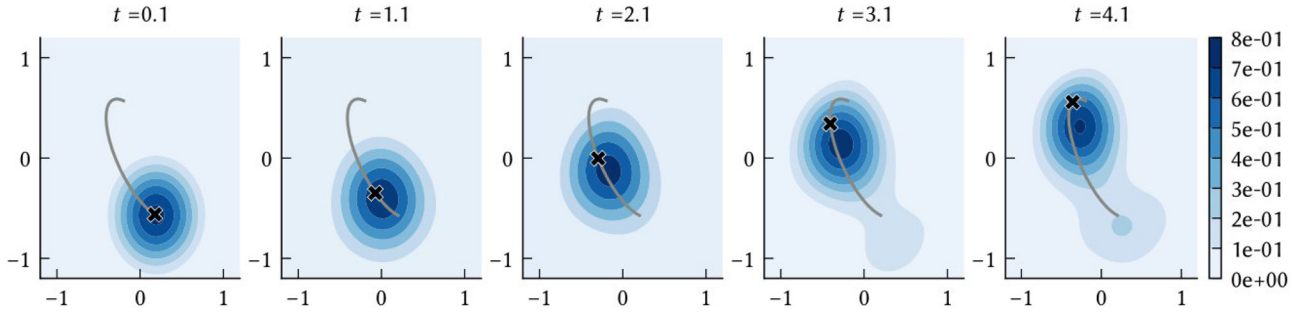


FIGURE 3
The time evolution of the posterior distribution $f(\mathrm{x}) = (\phi(\mathrm{x}) \cdot \phi(t) - \xi)^{+}$, where $\phi(t)$ is the output of the vector encoded by the collective activities of neurons in the path integrator network. The black "x" shows the ground truth x(t) at the sampled times, and the grey line shows the complete ground truth path.

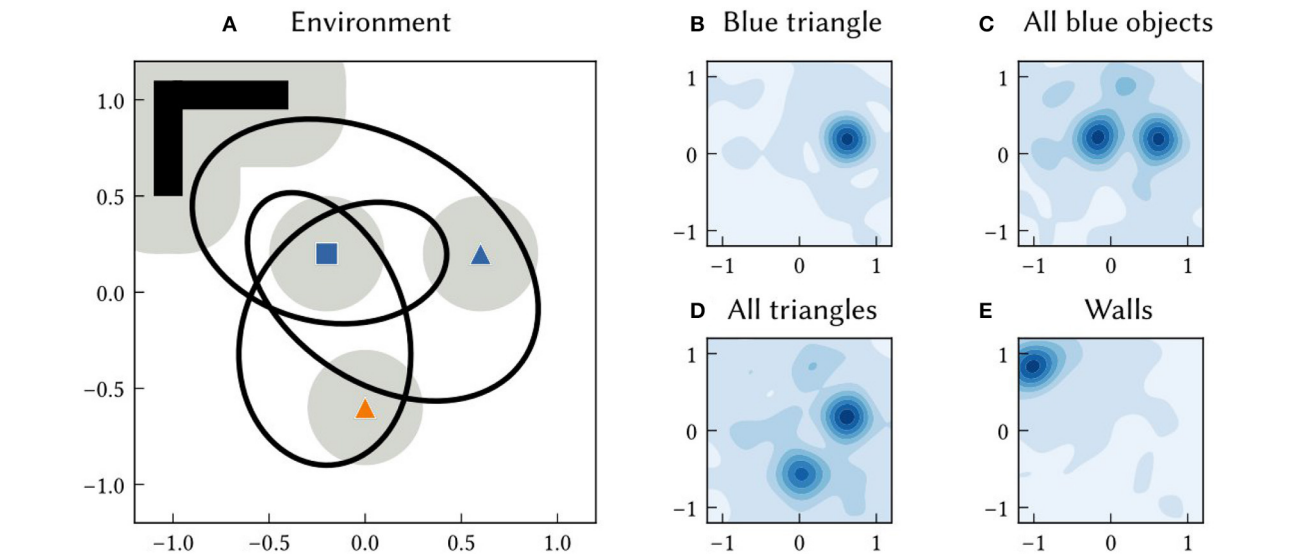


FIGURE 4
(A) An example 2D environment consisting of three point landmarks (a blue square, blue triangle, and orange triangle) and a wall region. An agent transverses a 60 s long trajectory through the environment (the black line). The agent has a limited field of view. The grey circles around landmarks indicates the distance at which the landmark is visible to the agent. SSP-SLAM is used to estimate the agent's location and learn a map of this environment. **(B–E)** The environment map network, trained to map features to locations, is probed at the end of the simulation. In **(B)**, the location of the blue triangle is recalled by passing in $B \circledast T$ to the environment map network. The heat map shows the similarity of the output of the network to SSPs representing points over the 2D. This represents the probability over locations. **(C)** The recall of the location of all blue landmarks, plotted as a similarity map. **(D)** The recall of the location of all triangle landmarks, plotted as a similarity map. **(E)** The recall of the wall area.

of a color, smell, and shape can be bound or bundled together to create a multi-sensory landmark. Using such representations, complex semantic environment maps can be learned.

Other mappings can be used as well. For example, a network can be trained to map feature locations $\hat{\phi}(\mathrm{x}_i)$, to feature symbols. Or, alternatively, a mapping from feature locations to feature symbols bound with their location, $\hat{\phi}(\mathrm{x}_i) \circledast B$, can be learned. Given an SSP input that represents the whole area of an environment, the network will approximately recall $\sum_i \phi(\mathrm{x}_i) \circledast B_i$, and so a single vector representation of a complete map can be recovered. We demonstrate a variety of these mappings in the Section 3.

2.2.4. Loop closure module

The combination of the PI model (presented in Section 2.2.1) and the associative memory network (for environment mapping) provides the core components of a SLAM model. As landmarks are discovered, their perception drives the training of a memory network, which learns a mapping from a symbol-like representation of features, B_j, to their locations, $\hat{\phi}(\mathrm{x}_i)$. When landmarks are re-encountered, the past estimate of their location is recalled by the memory network. This might be different than the current estimate of their locations computed in the OL population, due to errors accumulating in the PI computation. The difference in estimations is used to correct the PI model. This full loop is shown in Figure 2.

TABLE 1 The hyperparameters used for experiments with SSP-SLAM, exceptions are noted in the text.

Parameter	Default value
Number of neurons	
PI	45,000
GC	1,000
OVC	1,000
OL	27,000
AM	1,000
ME	27,000
Dim of SSPs, d	181
View radius of agent	0.3 × env. radius
Post-synaptic time constant, τ_{syn}	0.05
Max firing rate of LIF neurons	200–400 Hz
Proportion of active neurons	0.1
Voja learning rate	5×10^{-3}
PES learning rate	1×10^{-2}

3. Results

3.1. Mapping in 2D environments

In this section, we focus on a single example environment to demonstrate map querying and accuracy in SSP-SLAM. As shown in Figure 4A, we use a simple 2D environment that contains three point landmarks (a blue square, blue triangle, and orange triangle) as well as a wall region. To provide a path, we generate a random, frequency-limited trajectory through the environment and use finite differences to obtain velocities along the path (see Figure 4). The velocity input signal is represented by a spiking neural population, introducing noise to the signal. Model parameters used in this and subsequent experiments (unless stated otherwise) are given Table 1.

The environment map network is trained to map semantic pointers representing environment features to the feature locations as SSPs. Given the map in Figure 4, it ideally learns the following associations:

$$\texttt{BLUE} \circledast \texttt{SQUARE} \rightarrow \phi([0.6, 0.2]) \tag{21}$$

$$\texttt{BLUE} \circledast \texttt{TRIANGLE} \rightarrow \phi([0.0, -0.6]) \tag{22}$$

$$\texttt{ORANGE} \circledast \texttt{TRIANGLE} \rightarrow \phi([-0.2, 0.2]) \tag{23}$$

$$\texttt{WALL} \rightarrow \int_{0.5}^{1.1}\int_{-1.1}^{-0.95} \phi(x, y)dxdy + \int_{0.95}^{1.1}\int_{-1}^{-0.4} \phi(x, y)dxdy \tag{24}$$

where BLUE is a semantic pointer representing the color "blue", SQUARE is the semantic pointer representing the shape "square", etc.

At the end of the simulation, the actual mapping learned by the environment map network is probed. The locations of particular point landmarks is recalled by feeding in semantic pointer input, e.g., BLUE ⊛ TRIANGLE as shown in Figure 4B. Additionally, the map was queried for locations of all landmarks sharing certain characteristics. For example, the locations of all blue landmarks was queried by giving the network input BLUE ⊛ (SQUARE + TRIANGLE) (see Figure 4C).

In Figure 5A, the MAP estimates of point landmark locations at the end of the simulation are shown. Also plotted is the output of a biologically plausible computation of the vector from the model's self-position estimate to all recalled landmark locations. The output from querying the environment map network for each landmark's SSP location, $\hat{\phi}(\mathrm{x}_i)$, is combined with the output of the localization module to compute these vectors over the simulation run time. This is done by taking the inverse of the SSP output of the localization module, $\phi(\hat{\mathrm{x}}(t))^{-1}$, and binding it with recalled locations from the associative memory, $\phi(\hat{\mathrm{x}}(t))^{-1} \circledast \hat{\phi}(\mathrm{x}_i) = \hat{\phi}(\mathrm{x}_i - \mathrm{x}(t)) \approx \phi(\mathrm{x}_i - \mathrm{x}(t))$. This produces an estimate of the vector distance between the agent and landmark i – a useful quantity for navigation. The error in this computation is plotted in Figures 5B, C. At the beginning of the simulation, environment map has not yet been learned and so the output $\hat{\phi}(\mathrm{x}_i - \mathrm{x}(t))$ is inaccurate. After an item has been encountered, the error drops.

An associative memory that maps landmark location SSPs to landmark semantic pointers is also trained in this experiment. After learning, SSPs are passed into this network to recall the semantic pointers of landmarks or features at particular locations or over particular areas. An example of querying an area is shown in Figure 6.

3.2. Maintaining neural activity patterns

The activity patterns of spiking neurons in various components of the SSP-SLAM are presented and discussed here. SSP-SLAM is run on a 150 s path, recorded from a rat by Sargolini et al. (2006), with ten landmarks at random locations added to the environment for our experiment. Spike trains are recorded from neurons in the path integrator network, GC population, OVC population, and the associative memory network during the simulation. Activity patterns from certain example neurons are shown in Figure 7.

In Figure 7A, we see that a neuron in the GC population indeed has hexagonally patterned activity, as expected. However, this pattern deteriorates when using the path integrator alone. The corrections computed using the trained environment map module ensure the pattern's stability. This environment map is learned by modifying the outgoing connection weights in the associative memory population using the PES rule, while the Voja learning rule is used to modify the encoders of the associative memory population. This results in neurons developing selective sensitivity to particular encountered landmarks, similar to hippocampal place cells (Geiller et al., 2017; Kim et al., 2020a). This is apparent in Figure 7C. In Figure 7B, the activity of a neuron from the OVC population is shown and, as expected, its activity is like that of the object-vector cells of the MEC.

Activity from an example neuron from a VCO population in the path integrator is shown in Figure 7D. Here the spatial sensitivity of the neuron is not discernible. There are no obvious stripe or band patterns, due to the neuron's conjunctive sensitivity to velocity. In a non-random path with correlation between path velocity and position, a stripe pattern would be more apparent (for example,

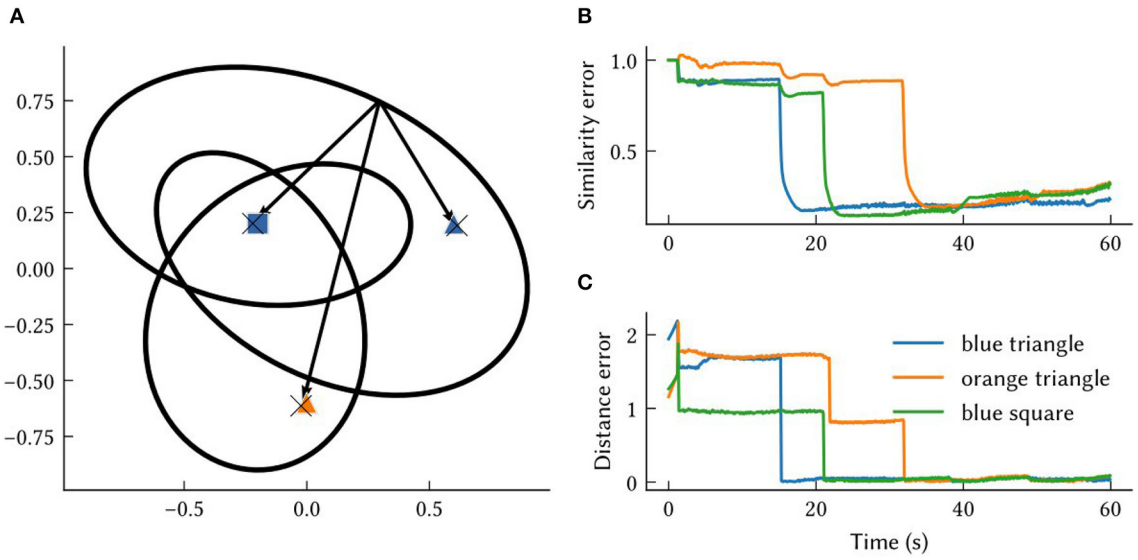


FIGURE 5
The results from querying vectors to landmarks in the same environment from Figures 4, 6. **(A)** Each "X" marks the model's MAP estimate of a point landmark's location at the end of the simulation. The arrows are estimates of the vectors between self-position and recalled landmarks at the end of the simulation. These approximate vectors are estimated from $\hat{\phi}(x_i - x(t))$, obtained by binding the model's other SSP estimates, $\phi(\hat{x}(t))^{-1} \circledast \hat{\phi}(x_i) \approx \phi(x_i - x(t))$. **(B)** The similarity error, $1 - \phi(x_i - x(t)) \cdot \hat{\phi}(x_i - x(t))$, over the simulation time t. **(C)** The distance between the MAP estimate obtained from $\hat{\phi}(x_i - x(t))$ and the ground truth vector between self-position at time t and landmark locations, $x_i - x(t)$.

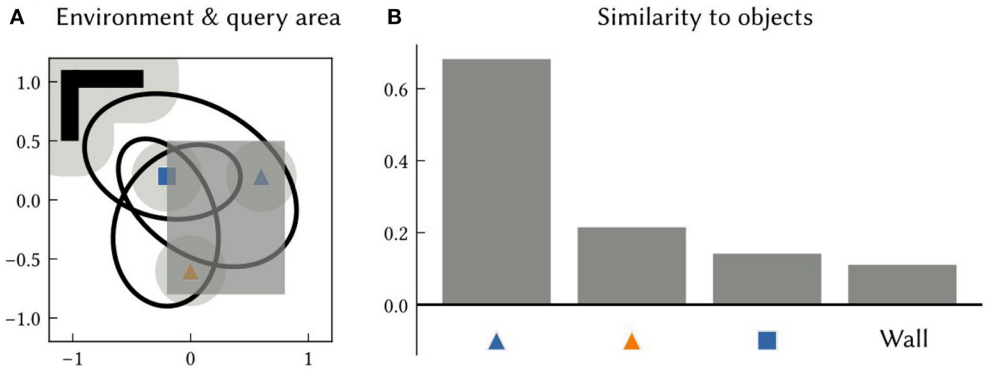


FIGURE 6
(A) An example 2D environment and a query area (the dark grey shaded region). The SSP representing the query area is given as input to an associative memory network that learned to map object location SSPs to object features, using the output of SSP-SLAM's path integrator and OB network components. **(B)** The similarity of the output of the associative memory network to all object semantic pointers in the environment. The results indicate that the orange triangle and blue square are within the queried area.

the spiral path example used in Dumont et al., 2022). However, the histogram in Figure 7D showing the distribution of spike counts by heading direction shows that the neuron has selective sensitivity to heading directions between 337.5° and 360° from north. Thus, this neuron is not unlike the head direction cells with conjunctive sensitivity to velocity and position found in the MEC in Sargolini et al. (2006).

3.3. Localization in 2D environments

In this experiment, the accuracy of localization in SSP-SLAM is explored. SSP-SLAM is tested on ten different environments. In each environment, ten random locations were chosen for point landmarks, and a two minute-long path generated. The paths are randomly generated from band-limited white noise signals. The model is initialized with the SSP representation of the starting point of the path, and receives the velocity along the path (computed using finite differences) as input over the simulation run time.

To determine the accuracy of the model, the raw spiking data is interpreted as a position estimate as follows (see Figure 8). The vector represented by the path integrator network, $\hat{\phi}(x(t))$, is decoded from neural activities. Then the $\hat{x}$ that maximizes $(\phi(\hat{x}) \cdot \hat{\phi}(x(t)) - \xi)^+$ is computed. This is the MAP estimate of self-position.

The average accuracy of SSP-SLAM localization output is shown in Figure 8. Plotted are similarity and distance errors.

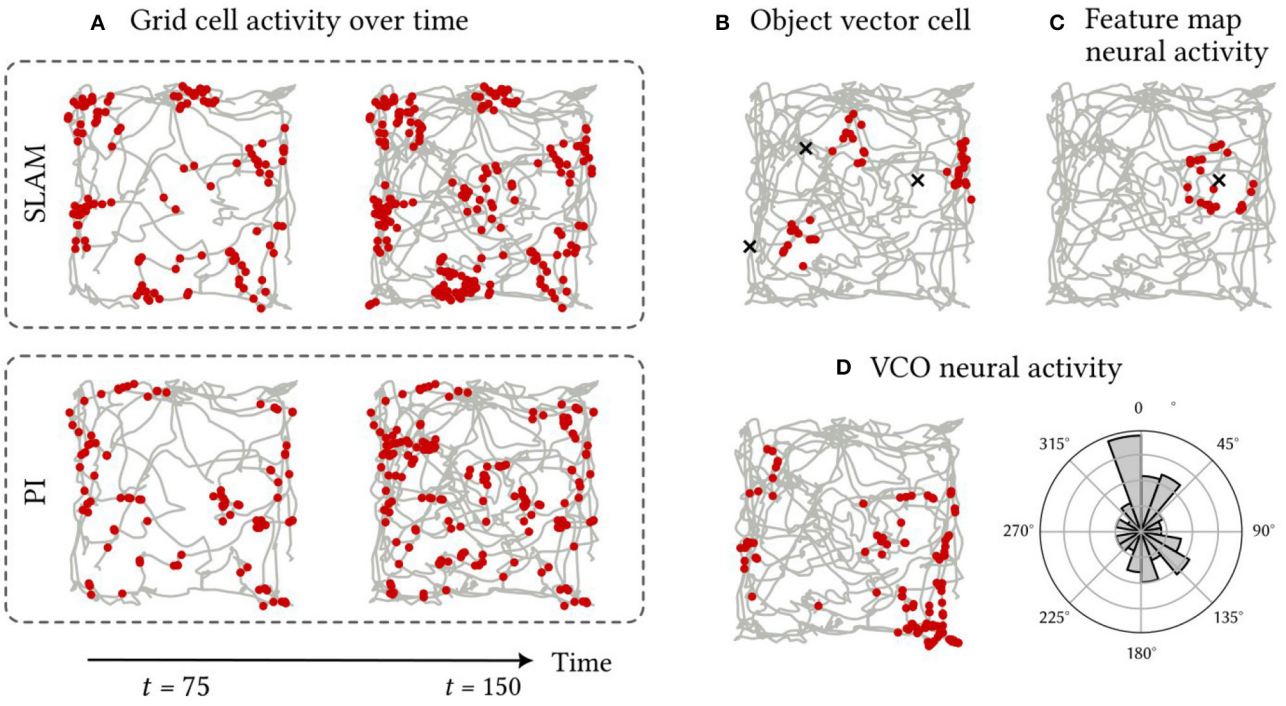


FIGURE 7
Firing patterns of neurons in SSP-SLAM. The path used for the simulation is shown in grey [obtained from Sargolini et al. (2006)]. Red dots indicate the positions at which a neuron fired. **(A)** A neuron from the GC population encoding $\phi(\hat{x}(t))$. Spikes are recorded with the normal functioning of SSP-SLAM in the top row. For the second row, error correction from the learned environment map is turned off (i.e., the GC population was representing the output of path integration alone). The first column of both rows presents result half way through the simulation, while the second column displays the complete recording. The corrections from the environment map help maintain the grid-cell-like activity pattern over time. **(B)** A neuron from the OVC population encoding the SSP representation of the vector between $x(t)$ and any landmarks in view. Object locations are marked with an "x". For this simulation, only three landmarks are included for clarity of the visualization. This neuron fires when an landmark is east of the agent. **(C)** A neuron from the associative memory population. The Voja learning rule shifts the neuron's encoder toward its input, resulting in the neuron firing when a particular landmark (marked with an "x") is in view. **(D)** A neuron from a VCO population in the path integrator. Neurons in this population have conjunctive sensitivity to velocity and position. On the right panel, spike counts are binned by the heading direction of the path, demonstrating the neuron's preference for a particular head direction.

The increasing similarity error for SSP-SLAM shows that it is not perfectly representing the SSP encoding of the ground truth. However, the low distance error indicates that an accurate position estimate can be decoded from the output of SSP-SLAM. The absolute trajectory error (the average deviation from ground truth trajectory per time-step) for SSP-SLAM is 0.0529 ± 0.0315 in these experiments. For the PI model alone, this error is 0.7876 ± 0.2958 Integrating the RMSE between SSP-SLAM's MAP estimate and the ground truth over the entire simulation time yields 5.758 ± 3.704 for SSP-SLAM and 73.728 ± 33.69 for PI. The error corrections provided by the environment map in SSP-SLAM result in a more than ten-fold improvement in localization error.

Figure 9 shows examples of the path estimate of SSP-SLAM compared to the exact path and the path integrator network alone (i.e., dead reckoning); the full SSP-SLAM model accurately follows the true path for the entire trajectory. In contrast, the results from the path integrator alone are very poor in these experiments due to the length of the paths and the number of neurons used. Early on in the simulation, the vector represented by the path integrator leaves the manifold in $\mathbb{R}^d$ of the SSPs. Since it is no longer representing a valid SSP, an accurate position cannot be decoded and so the position estimate jumps wildly in the space. In contrast, the corrections computed using the environment map in SSP-SLAM keep the path integrator output near the ideal result.

3.4. Localization in 3D environments

While we have focused on 2D environments in this work, the model and all representations naturally generalize to any number of dimensions. In Figure 10, we show how the same model structure using 3D SSPs can be used to accurately perform 3D localization. There are no differences between this and the 2D models, other than using SSP vectors, $\phi(x)$, encoding $x \in \mathbb{R}^3$.

3.5. Neuromorphic simulation of dead reckoning

To investigate the feasibility of deploying SSP-SLAM on neuromorphic hardware, we simulated the path integrator network on the NengoLoihi emulator. This Python package allows spiking neural network models built in Nengo to be run on Intel's Loihi

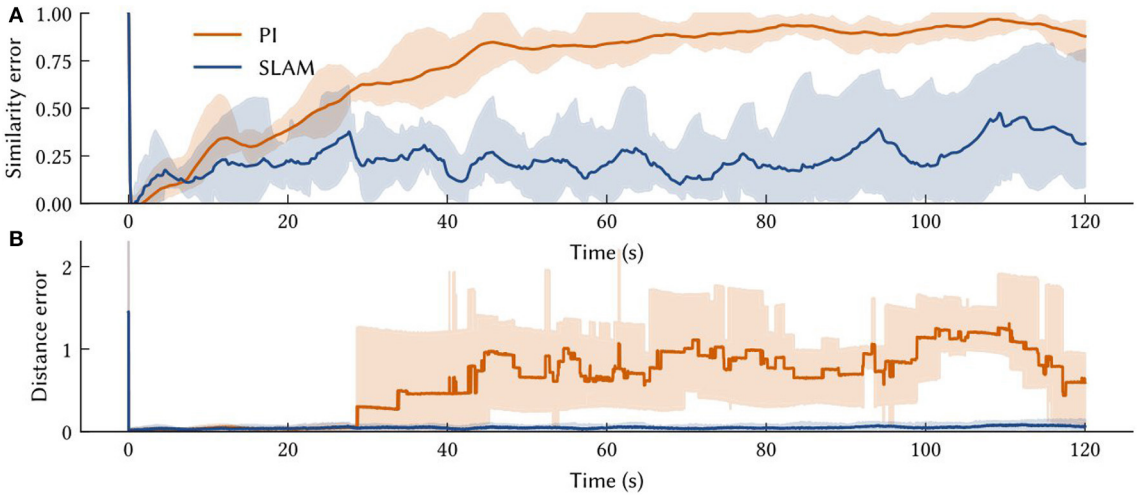


FIGURE 8
The solid line is the performance measure averaged over ten trials of different paths. Also shown are shaded error bars. **(A)** The similarity error, $1 - \phi(\mathbf{x}(t)) \cdot \hat{\phi}(\mathbf{x}(t))$, over the simulation time t – i.e., how far the off the vector output of the path integrator is from the SSP encoding of the ground truth. **(B)** The distance between the model's MAP estimate of self-position and the ground truth over the simulation time.

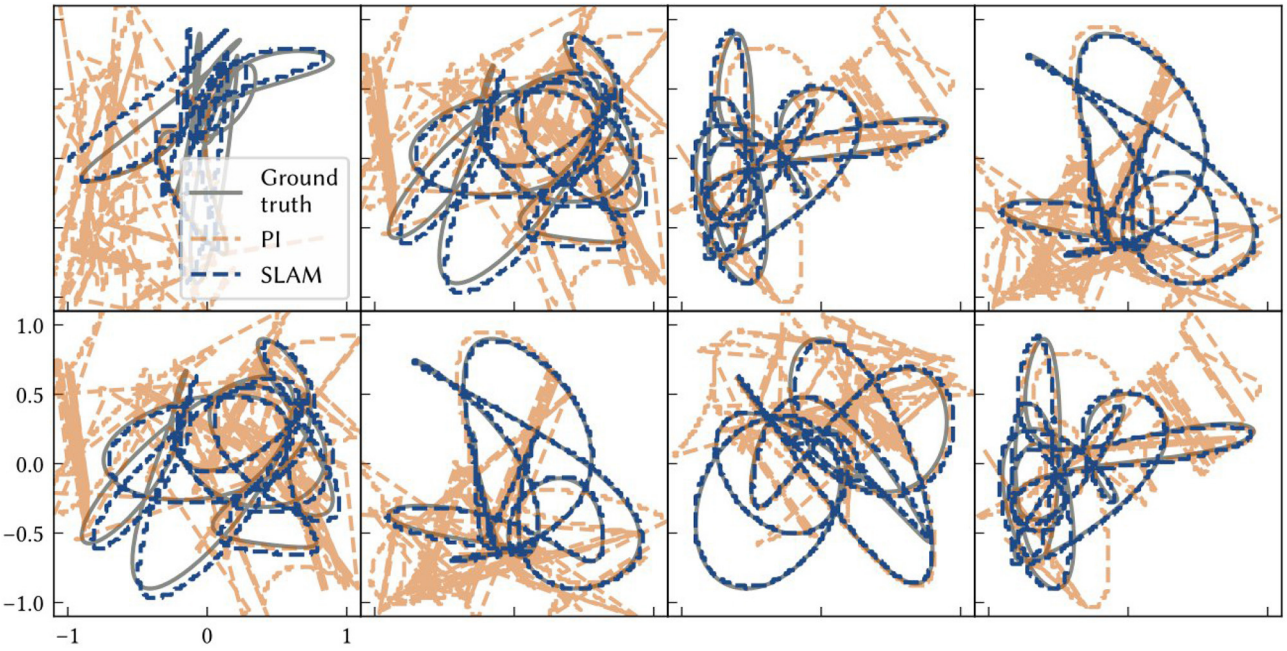


FIGURE 9
Each panel shows model results for a different environment/ trial. The ground truth paths are plotted as grey solid lines. The dashed blue line is the location estimate from SSP-SLAM. The dashed orange line is the estimate from the path integration network without any corrections from the environment map network (i.e., dead reckoning).

architecture. It includes both support for running models on the Loihi hardware and a Loihi emulator, which we used for these experiments. In this experiment, we run the model on paths derived from the KITTI odometry benchmark (Geiger et al., 2012). However, we do not use raw visual input from the KITTI datasets, as our model does not support visual SLAM. Rather, we use velocity signals computed via finite differences on the ground truth paths and represented by a neural population. To compensate for the absence of the landmark perception, environment map, and loop closure modules, the total number of neurons in the path integrator was increased to 90,000 to reduce drift. Results are shown in Figure 11.

Notably, the NengoLoihi emulator implements the same limited precision mathematics as the actual hardware, using 8-bit weights and a quantized neuron response function. Figure 11 shows that the path integrator network is robust to these additional

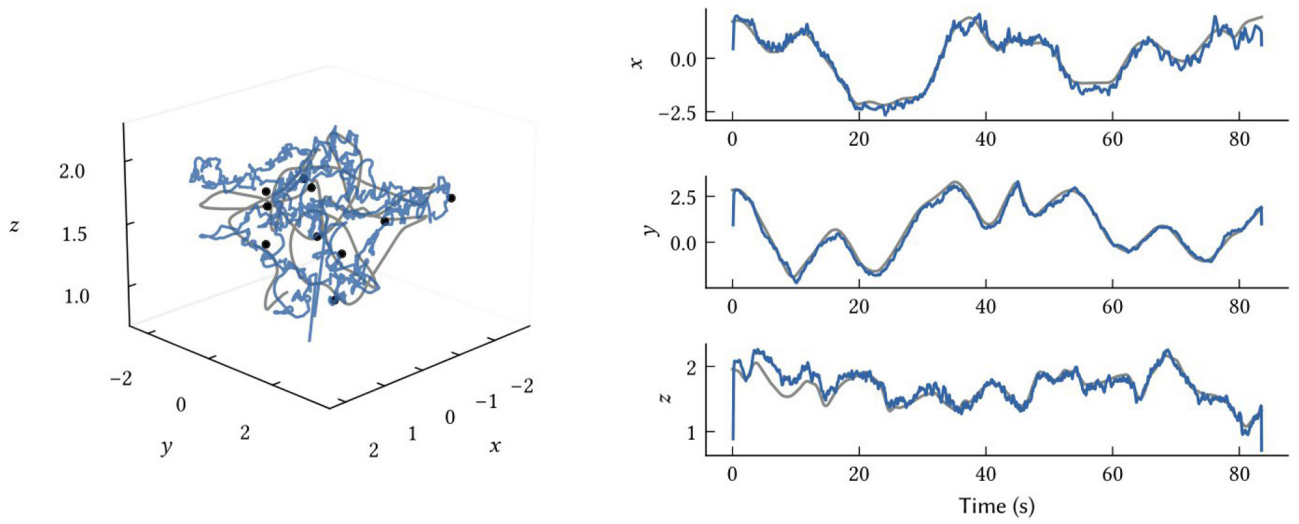


FIGURE 10
Results from 3D SLAM. The ground truth path is shown in grey, the SSP-SLAM MAP estimate is in blue, and the locations of ten point landmarks are given by the black dots.

constraints, and continues to perform largely as expected, although with more error compared to the typical performance of the full SSP-SLAM model (Section 3.3).

4. Discussion

4.1. Prior research

The development and implementation of SLAM algorithms for mobile robots has garnered significant attention in academic and engineering communities. Approaches generally involve recursive Bayesian estimation—via various kinds of Kalman Filters (Smith et al., 1990; Brossard et al., 2018), Particle Filters (Montemerlo et al., 2002; Sim et al., 2005), or occupancy grid methods (Stachniss et al., 2004)—or graph optimization (Thrun and Montemerlo, 2006; Sünderhauf and Protzel, 2012). In recent years, researchers have focused on incorporating semantic information into SLAM systems, using deep artificial neural networks, particularly convolutional or recurrent neural networks for object detection and semantic segmentation. The use of semantic information in SLAM has been found to improve performance and robustness of robot localization (Frost et al., 2016; Stenborg et al., 2018; Bowman, 2022). Furthermore, robots equipped with semantic SLAM hold the promise of performing higher-level tasks, such as planning paths based on human instructions that reference objects in the environment. Concurrently, an alternative approach to SLAM, drawing inspiration from the brain, has continued to develop novel algorithms with the goal of improving efficiency and robustness (Milford et al., 2004, 2016; Silveira et al., 2015; Yu et al., 2019). In this line of research, models of neural path integration inspired by hippocampal cells are used for localization. Coupling such neural algorithms with recent developments in neuromorphic hardware, as we have done here, aims to both improve our understanding of how the brain accomplishes SLAM and to improve the power efficiency of engineered solutions.

Neural localization models used in this alternative approach can generally be divided into two categories: Continuous Attractor Network (CAN) models (Samsonovich and McNaughton, 1997; Tsodyks, 1999; Conklin and Eliasmith, 2005) and Oscillator-Interference (OI) models (O'Keefe and Burgess, 2005; Burgess et al., 2007; Hasselmo et al., 2007; Welday et al., 2011). In CAN models, path integration is performed by a recurrently connected neural sheet, whose dynamics sustain a single Gaussian-like activity bump that represents the self-position estimate of an agent. In contrast, in OI models, the self-position estimate is encoded by the phase differences between Velocity-Controlled Oscillators (VCOs)—oscillators whose frequency is modulated by a velocity signal.

The seminal application of neural-inspired methods to SLAM is RatSLAM, in which visual odometry is used to drive a CAN (Milford et al., 2004, 2016). The CAN consists of "pose cells" (similar to the place and head direction cells found in the hippocampal formation) and maintains an estimate of self-position and orientation. Sensor data is processed outside the neural network to create a template array (for example, raw visual input is converted to an intensity profile vector). When a novel template is observed, a new "local view cell" (similar to the spatial view cells in the hippocampus) is added to the network. The population of these cells is sparsely connected to the CAN, with associations learned via Hebbian learning. Additionally, a graph is constructed and updated with a graph relaxation algorithm online to create a topological environment map. Its nodes store experiences in the form of activity of pose cells and local view cells along with robot pose estimates.

In contrast, a hybrid OI-CAN model is used for path integration in SSP-SLAM and a graphical environment map is not learned—instead, the outgoing connection weights from the

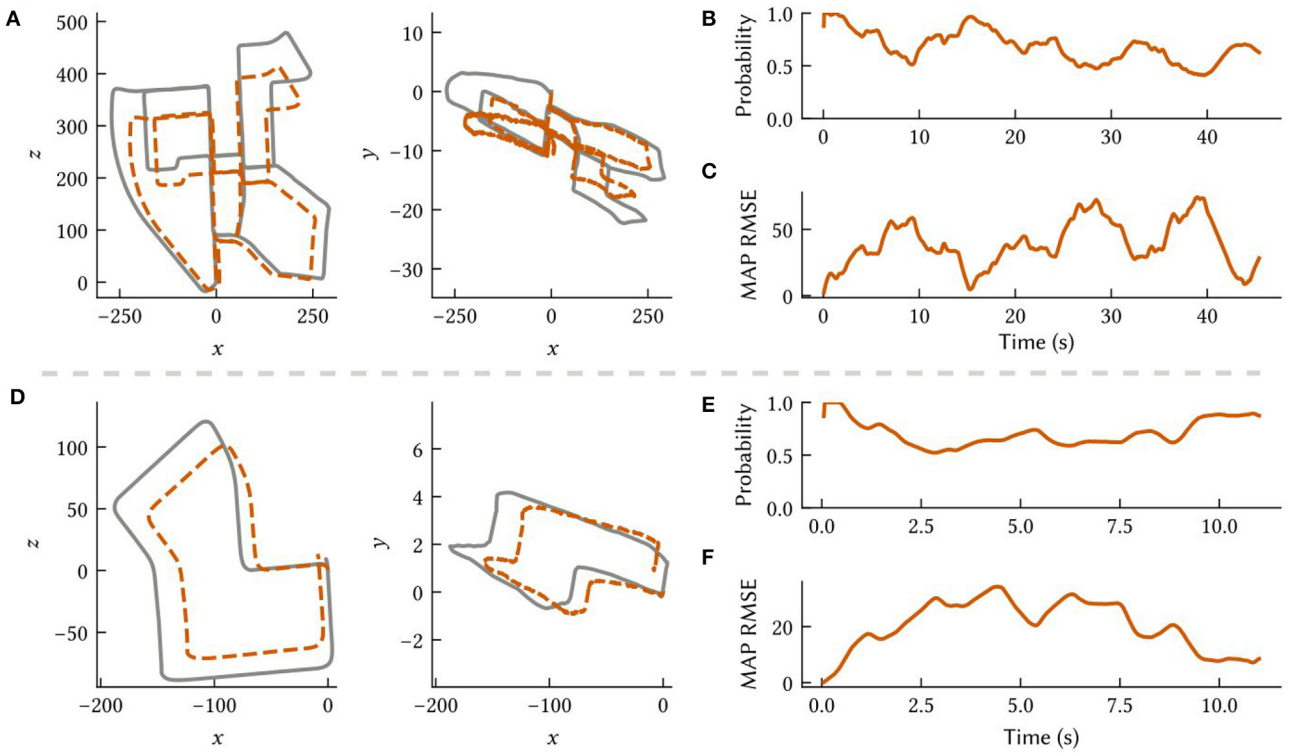


FIGURE 11
Results from the path integrator network simulated on the NengoLoihi emulator using two different 3D paths constructed from the KITTI odometry dataset (Geiger et al., 2012). **(A, D)** Comparison of the ground truth paths (in grey) to our model's MAP estimate (in orange) along different dimensions. **(B, E)** The probability the model assigns to the ground truth over time. **(C, F)** The RMSE between the MAP estimate and ground truth.

memory network implicitly store a map which can be retrieved by querying the network. The Voja rule, which is used to shift the associative memory population encoders toward observed input in SSP-SLAM, plays a similar role to the template novelty detection and addition of local view cells that occurs in RatSLAM. Furthermore, we have not implemented an external module for pre-processing of sensory data, and we use landmark semantic pointers and displacement SSPs in lieu of templates. Object detection and depth estimation algorithms would be required to obtain this input from visual data.

Many models have since extended the original RatSLAM. CAN SLAM models with place cell-like activity were also used by BatSLAM (Steckel and Peremans, 2013), an extension to RatSLAM for handling environment information from sonar sensors, and DolphinSLAM (Silveira et al., 2015), developed for 3D SLAM in underwater environments. A CAN consisting of conjunctive grid cells was used in the SLAM model presented in Zeng and Si (2017). Three-dimensional SLAM in realistic environments with grid cells was also explored in NeuroSLAM (Yu et al., 2019). Unlike our work, none of these models use spiking neural networks.

More recent research has focused on developing spiking networks for SLAM and testing them on neuromorphic hardware. Spiking 2D SLAM models were presented in Tang and Michmizos (2018), Tang et al. (2019), and Kreiser et al. (2020a,b). In Kreiser et al. (2020a), a SLAM system on the Loihi chip was used to estimate the head position of an iCub robot as it visually explored a wall with a dot pattern acting as the environment. Tang et al. (2019) made use of a depth camera and Bayesian updates on a posterior distribution represented by neural population. They found that their SLAM system, when run on Loihi, was more energy efficient by two orders of magnitude compared to a baseline method on a CPU. While the models discussed here use raw sensory input, it should be noted that non-spiking visual modules are used to process this information and obtain input for SLAM. For instance, intensity profile vectors or feature colors and distances from the observer are used. In contrast to SSP-SLAM, none of the models mentioned incorporate any elements of OI to perform path integration, or perform 3D SLAM. Furthermore, some of these models employ "localist"/discrete representations, such as using one neuron to represent each integer value for heading direction or discretized distance to features. This approach does not support generalization and does not scale well to higher dimensional representations, unlike SSPs.

Taken together, and summarized in Table 2, past work provides examples of spiking and non-spiking networks, using CANs for path integration. However, unlike SSP-SLAM, none of these approaches provides a methodology for incorporating semantic information or for online learning of semantic environmental maps. In addition, none of these employ SSPs, or the same combination of a OI-CAN network in a fully spiking model capable of functioning equally well in both 2D and 3D spatial environments, as demonstrated above.

TABLE 2 Comparison of bio-inspired SLAM models.

Model	Sensors	Input representation	Dim.	Localization	Env. map	Cells	Experiment scale	Spiking	Neuromorhpic hardware
SSP-SLAM	None	Displacement to features as an SSP and feature identities as SPs	Any, tested on 2D & 3D	OI-CAN hybrid	Weights between landmark population to landmark locations	HDC, GC, landmark cells, OVC	Small	Yes	Partially
RatSLAM (Milford et al., 2004, 2016)	Monocular camera	Greyscale image intensity profile	2D	CAN	Topological map associating local views with position stored as a graph	Pose cells, local view cells	Large	No	No
BatSLAM (Steckel and Peremans, 2013)	Biomimetic sonar	Intensity difference between left and right Echolocation Related Transfer Functions	2D	CAN	Topological map local views with position stored as a graph	Pose cells, local view cells	Small	No	No
DolphinSLAM (Silveira et al., 2015)	Sonar & visual	One-hot representation obtained from FabMAP algorithm on top of a Bag of Words model	3D	CAN	Graph with nodes storing local view, place cell and position while edges store displacements	3D PC, local view cells	Small	No	No
NeuroSLAM (Yu et al., 2019)	Panoramic camera	Greyscale image intensity profile	3D	CAN	Topological map storing activities of local view cells, GCs, HDCs, and estimated pose	3D PC, conjuctive 3D GC and HDC, local view cells	Large	No	No
Kreiser et al. (2020a)	Event-based camera	Detection of blinking LEDs at different frequencies	2D	CAN	Weights from landmark population to a HDC population	HDC, landmark cells	Small	Yes	Fully
Tang et al. (2019)	RGB-Depth camera	Discretized distances to landmarks	2D	CAN	Weights from PC to a displacement-from-border population	2D PC, HDC, border cells, Bayesian cells	Small	Yes	Fully

10.3389/fnins.2023.1190515

4.2. Performance

We have presented the results of several experiments on SSP-SLAM to assess its performance and utility. The model demonstrates accurate localization capabilities on different paths, both two-dimensional and three-dimensional. To achieve this, a hybrid OI-CAN model is employed for path integration. Notably, this is the only SLAM model (to our knowledge) that uses OI techniques for localization. This approach has the advantage of easy generalization to higher dimensional spaces. Typically, CAN models describe a neural population as a 2D sheet or 3D array (often with periodic boundary conditions), where the geometry specifies the recurrent connectivity pattern required for localization. However, this only supports unimodal position estimates, and the connectivity pattern must be modified and made more complicated to move to higher dimensional path integration.[1] In contrast, in our approach the recurrent connectivity of the path integrator network remains the same regardless of spatial dimensionality. This allows the same model to switch seamlessly between SLAM in different spaces and domains.

Furthermore, SSP-SLAM encodes environment maps in the outgoing connections of an associative memory network, which are learned online using biologically plausible learning rules. The map generated is a semi-metric, semantic map that uses symbol-like vector representations that have been leveraged in a variety of large-scale cognitive models (Eliasmith, 2013; Arora et al., 2018; Kajić et al., 2019; Kelly et al., 2020; Gosmann and Eliasmith, 2021). By working in the SSP and VSA paradigm, we are able to formulate the problem in such a way that unites metric and semantic SLAMs. This approach unites analytical models of vehicle motion and map construction with neural networks, resulting in a formulation that is compatible with modern ML approaches to robotics, while still maintaining the explainability of the system. These feature distinguishes SSP-SLAM from other bio-inspired SLAM models and makes it the first spiking semantic SLAM model to our knowledge.

This inclusion of semantic information helps SSP-SLAM be more accurate. Specifically, SSP-SLAM performs loop closure via corrections to the PI network provided by the environment map, which leads to significant improvements in localization accuracy. After training, the map can be queried to obtain object locations given their symbol-like representation as a semantic pointer. Alternatively, item representations can be obtained by querying specific areas, or vectors between the agent and landmarks can be computed. These kinds of direct queries of semantic map knowledge cannot be easily made with past spiking network map representations.

Finally, a key element of the model, the path integrator, was tested on a neuromorphic emulator. The results indicate that the model can maintain expected accuracy (given the absence of error correction mechanisms) on neuromorphic hardware. Notably, all additional operations used in the model have been implemented on neuromorphic hardware in other work (Knight et al., 2016; Mundy, 2017), so we believe this demonstration strongly suggests that a full neuromorphic implementation is achievable. Overall, this study presents a novel and promising approach to SLAM based on a fully spiking neural network.

4.3. Limitations

This study presents a novel model that employs biologically-inspired mechanisms to solve SLAM. However, SSP-SLAM has several limitations. First, the full SSP-SLAM model has not been tested on a neuromorphic chip emulator nor has the model been deployed on an actual neuromorphic hardware platform. Second, the model was tested on a small scale and artificial environments, which restricts what conclusions we can draw as to its generalizability to more complex, real-world environments.

To improve the model's utility, it is essential to test it on real-world input and integrate it with a network that can process raw sensory data. Such an approach would enhance the model's ability to handle more complex and diverse environmental conditions. Moreover, the current model's accuracy is inferior to that of non-biologically inspired SLAM methods, which limits its usefulness to mobile robotics. This accuracy drop and the use of small scale test environments is true of current spiking SLAM models more generally. Despite this, the use of neuromorphic computing and hardware has the potential to improve energy efficiency of SLAM systems, which is particularly useful in mobile robotics applications. This encourages further research into spiking SLAM systems. Reduced power demands permits the deployment of SLAMs in progressively more power-constrained environments, such as edge computing or operations in GPS-denied settings, like space or sub-sea exploration. An increasing number of algorithms have harnessed the advantages of spike-based computing to make gains in efficiency and speed (Yakopcic et al., 2020; Davies et al., 2021; Yan et al., 2021).

Therefore, while the current model shows promise in enabling biologically-inspired SLAM, its limitations in terms of testing and accuracy should be addressed before considering its wider application in real-world scenarios. Further research could focus on testing the model on larger networks and more complex environments, as well as investigating ways to improve its accuracy.

4.4. Future work

One clear direction for future work is ameliorating the limitations discussed in the previous section. Beyond this, there are several other directions that warrant further exploration—for example, explicit modeling of sensor uncertainties using SSPs, introducing coupling dynamics to increase localization accuracy, higher-dimensional SLAM, and integration with other cognitive models.

Accurate of localization is vital and phase drift is one of the main factors contributing to SSP inaccuracy. As path integration

1 Recent research has explored variants to traditional CANs that overcome these limitations. A multimodal CAN model was presented in Wang and Kang (2022) and research exploring CANs with arbitrary dimensional attractor manifolds and more biologically realistic asymmetries in synaptic connectivity was presented in Darshan and Rivkind (2022). Such CAN variants have not been used in SLAM systems.

progresses, errors can accumulate in the phases of the velocity-controlled oscillators (VCOs), resulting in inconsistencies that degrade the spatial information (e.g., see Figure 8). The loop-closure error corrections (Figure 2) can shift the phases toward the true values, but the phase inconsistencies would still be present. However, one could take advantage of the redundancy in the SSP representation by adding coupling between the VCOs that enforce their proper phase relationships (Orchard et al., 2013).

Additionally, higher-dimensional SLAM could be a promising area of investigation. The proposed model can be extended to localization and mapping in any dimension of space by modifying the input without changing the model or hyperparameters. Although SLAM is mainly applied to navigation and mapping in physical spaces, operating in dimensions equal to or less than three, it is possible that the same neural mechanisms underlying spatial navigation and mapping could be applicable to non-spatial domains, such as mapping in high-dimensional conceptual space. The idea that similar computations to those behind SLAM may be understood as core cognitive processes has been proposed in Safron et al. (2022).

The application of SSP-SLAM to localization and mapping in various spaces (including non-spatial ones) via interactions with other cognitive systems is promising area for future research. By employing control mechanisms to manipulate the input to SSP-SLAM, it may be possible to model different cognitive functions. For instance, one could switch between motion input from sensory systems to perform localization and input from memory and cognitive maps to simulate path replay or planning. This could be realized by integrating SSP-SLAM with more complex memory, action selection, and reasoning systems. Since the proposed model was developed using the SPA, it fits naturally within the context of NEF and other SPA models, including Spaun (Stewart et al., 2012a; Choo, 2018). Integration of the proposed SLAM model with other models constructed with these tools could be used to develop systems equipped with more sophisticated cognitive capabilities and able to tackle multiple tasks. Exploiting memory and reasoning capabilities in large spatial environments remains a challenge for models of biological cognition.

4.5. Summary

In conclusion, we have proposed a novel spiking semantic SLAM model, SSP-SLAM, which is inspired by the hippocampal formation in the mammalian brain. The model is unique in its integration of a hybrid OI-CAN path integrator, online biologically-plausible learning of an environment map, and use of symbol-like object representations in a spiking network. This combination enables the model to perform SLAM accurately in small scale environments and learn representations that can be queried in powerful ways. For example, it can provide information about what is located in a particular area of the map, report vectors between landmarks, and identify the location of objects based on their properties, such as their color. Furthermore, these techniques advance the sophistication of biologically plausible SLAM networks, showing a wide variety of previously identified cell types while demonstrating functionality in 2D and 3D environments.

Finally, we have tested a core component of the network on a neuromorphic hardware emulator, which represents an important step toward achieving a full system running on neuromorphic hardware. While significant work remains to achieve this goal, we believe that the methods and components employed in this study provide a foundation for future research in this area. With continued progress, this spiking semantic SLAM model could have important applications in a wide range of fields, including robotics, artificial intelligence, and neuroscience.

Data availability statement

The raw data supporting the conclusions of this article can be found here: https://github.com/nsdumont/Semantic-Spiking-Neural-SLAM-2023.

Author contributions

ND, JO, and CE contributed to the theoretical development and conception of this work. ND wrote the code used in experiments, generated results and figures, and wrote the first draft of the manuscript. PF developed the probabilistic interpretation of SSPs used in this manuscript. All authors contributed to manuscript revision, read, and approved the submitted version.

Funding

This work was supported by CFI (52479-10006) and OIT (35768) infrastructure funding as well as the Canada Research Chairs program, NSERC Discovery grant 261453, AFOSR grant FA9550-17-1-0026, NRC grants AI4L-116 and AI4L-117, the NRC-Waterloo Collaboration Centre, an Intel Neuromorphic Research Community grant, and the Barbara Hayes-Roth Award for Women in Math and Computer Science. The funders had no role in the direction of this research, in the analyses or interpretation of data, in the writing of the manuscript, or in the decision to publish the results.

Conflict of interest

CE has a financial interest in Applied Brain Research, Incorporated, holder of the patents related to the material in this paper patent 63/110,231. The company or this cooperation did not affect the authenticity and objectivity of the experimental results of this work.

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

Aharon, G., Sadot, M., and Yovel, Y. (2017). Bats use path integration rather than acoustic flow to assess flight distance along flyways. *Curr. Biol.* 27, 3650–3657. doi: 10.1016/j.cub.2017.10.012

Arora, N., West, R., Brook, A., and Kelly, M. (2018). Why the common model of the mind needs holographic a-priori categories. *Proc. Comput. Sci.* 145, 680–690. doi: 10.1016/j.procs.2018.11.060

Bekolay, T., Kolbeck, C., and Eliasmith, C. (2013). "Simultaneous unsupervised and supervised learning of cognitive functions in biologically plausible spiking neural networks," in *Proceedings of the Annual Meeting of the Cognitive Science Society* (Berlin), Vol. 35.

Benhamou, S. (1997). Path integration by swimming rats. *Anim. Behav.* 54, 321–327. doi: 10.1006/anbe.1996.0464

Bersuker, G., Mason, M., and Jones, K. L. (2018). *Neuromorphic Computing: The Potential for hIgh-Performance Processing in Space*. Game Changer. Arlington, VA: Aerospace Center for Space Policy and Strategy 1–12.

Blouw, P., Eliasmith, C., and Tripp, B. (2016). "A scaleable spiking neural model of action planning," in *Proceedings of the 38th Annual Conference of the Cognitive Science Society*, eds D. Grodner, A. P. Dan Mirman, and J. Trueswell (Philadelphia, PA: Cognitive Science Society), 1583–1588.

Bowman, S. L. (2022). *Semantic Simultaneous Localization and Mapping* (PhD thesis). Philadelphia, PA: University of Pennsylvania.

Bowman, S. L., Atanasov, N., Daniilidis, K., and Pappas, G. J. (2017). "Probabilistic data association for semantic slam," in *2017 IEEE International Conference on Robotics and Automation (ICRA)* (Singapore: IEEE), 1722–1729.

Brossard, M., Bonnabel, S., and Barrau, A. (2018). "Invariant kalman filtering for visual inertial slam," in *2018 21st International Conference on Information Fusion (FUSION)* (Cambridge, UK: IEEE), 2021–2028.

Burgess, N. (2008). Grid cells and theta as oscillatory interference: theory and predictions. *Hippocampus* 18, 1157–1174. doi: 10.1002/hipo.20518

Burgess, N., Barry, C., and O'Keefe, J. (2007). An oscillatory interference model of grid cell firing. *Hippocampus* 17, 35–53. doi: 10.1002/hipo.20327

Cao, Y., Chen, Y., and Khosla, D. (2015). Spiking deep convolutional neural networks for energy-efficient object recognition. *Int. J. Comput. Vis.* 113, 54–66. doi: 10.1007/s11263-014-0788-3

Chen, X., Milioto, A., Palazzolo, E., Giguere, P., Behley, J., and Stachniss, C. (2019). "Suma++: Efficient lidar-based semantic slam," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Macau: IEEE), 4530–4537.

Choo, F.-X. (2018). *Spaun 2.0: Extending the World's Largest Functional Brain Model*. Waterloo: University of Waterloo.

Conklin, J., and Eliasmith, C. (2005). A controlled attractor network model of path integration in the rat. *J. Comput. Neurosci.* 18, 183–203. doi: 10.1007/s10827-005-6558-z

Czajkowski, R., Sugar, J., Zhang, S.-J., Couey, J. J., Ye, J., and Witter, M. P. (2013). Superficially projecting principal neurons in layer v of medial entorhinal cortex in the rat receive excitatory retrosplenial input. *J. Neurosci.* 33, 15779–15792. doi: 10.1523/JNEUROSCI.2646-13.2013

Darshan, R., and Rivkind, A. (2022). Learning to represent continuous variables in heterogeneous neural networks. *Cell Rep.* 39, 110612. doi: 10.1016/j.celrep.2022.110612

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

Duggins, P., Stewart, T. C., and Eliasmith, C. (2022). "Reinforcement learning, social value orientation, and decision making: computational models and empirical validation," in *Proceedings of the Annual Meeting of the Cognitive Science Society* (Toronto, CA), Vol. 44.

Dumont, N. S.-Y., and Eliasmith, C. (2020). "Accurate representation for spatial cognition using grid cells," in *42nd Annual Meeting of the Cognitive Science Society* (Toronto, ON: Cognitive Science Society), 2367–2373.

Dumont, N. S.-Y., Orchard, J., and Eliasmith, C. (2022). "A model of path integration that connects neural and symbolic representation," in *Proceedings of the Annual Meeting of the Cognitive Science Society, Vol. 44* (Toronto, ON: Cognitive Science Society).

Eliasmith, C. (2013). *How to Build a Brain: A Neural Architecture for Biological Cognition*. New York, NY: Oxford University Press.

Eliasmith, C., and Anderson, C. H. (2003). *Neural Engineering*. Cambridge, MA: The MIT Press.

Eliasmith, C., Stewart, T. C., Choo, X., Bekolay, T., DeWolf, T., Tang, Y., et al. (2012). A large-scale model of the functioning brain. *Science* 338, 1202–1205. doi: 10.1126/science.1225266

Etienne, A. (1987). "The control of short-distance homing in the golden hamster," in *Cognitive Processes and Spatial Orientation in Animal and Man* (Dordrecht: Springer), 36, 233–251. doi: 10.1007/978-94-009-3531-0_19

Fan, Y., Zhang, Q., Tang, Y., Liu, S., and Han, H. (2022). Blitz-slam: a semantic slam in dynamic environments. *Pattern Recognit.* 121, 108225. doi: 10.1016/j.patcog.2021.108225

Frost, D. P., Kähler, O., and Murray, D. W. (2016). "Object-aware bundle adjustment for correcting monocular scale drift," in *2016 IEEE International Conference on Robotics and Automation (ICRA)* (Stockholm: IEEE), 4770–4776.

Furlong, P. M., and Eliasmith, C. (2022). "Fractional binding in vector symbolic architectures as quasi-probability statements," in *44th Annual Meeting of the Cognitive Science Society* (Toronto, CA: Cognitive Science Society).

Furlong, P. M., and Eliasmith, C. (2023). *Modelling Neural Probabilistic Computation Using Vector Symbolic Architectures*. London: Centre for Theoretical Neuroscience, University of Waterloo.

Furlong, P. M., Stewart, T. C., and Eliasmith, C. (2021). "Fractional binding in vector symbolic representations for efficient mutual information exploration," in *ICRA Workshop: Towards Curious Robots: Modern Approaches for Intrinsically-Motivated Intelligent Behavior* (Virtual).

Geiger, A., Lenz, P., and Urtasun, R. (2012). "Are we ready for autonomous driving? the kitti vision benchmark suite," in *Conference on Computer Vision and Pattern Recognition (CVPR)* (Providence).

Geiller, T., Fattahi, M., Choi, J.-S., and Royer, S. (2017). Place cells are more strongly tied to landmarks in deep than in superficial ca1. *Nat. Commun.* 8, 14531. doi: 10.1038/ncomms14531

Geromichalos, D., Azkarate, M., Tsardoulias, E., Gerdes, L., Petrou, L., and Perez Del Pulgar, C. (2020). Slam for autonomous planetary rovers with global localization. *J. Field Robot.* 37, 830–847. doi: 10.1002/rob.21943

Glad, I. K., Hjort, N. L., and Ushakov, N. (2007). *Density Estimation Using the Sinc Kernel*. Trondheim: Norwegian University of Science and Technology.

Glad, I. K., Hjort, N. L., and Ushakov, N. G. (2003). Correction of density estimators that are not densities. *Scand. J. Stat.* 30, 415–427. doi: 10.1111/1467-9469.00339

Gosmann, J., and Eliasmith, C. (2021). CUE: A unified spiking neuron model of short-term and long-term memory. *Psychol. Rev.* 128, 104–124. doi: 10.1037/rev0000250

Hasselmo, M. E., Giocomo, L. M., and Zilli, E. A. (2007). Grid cell firing may arise from interference of theta frequency membrane potential oscillations in single neurons. *Hippocampus* 17, 1252–1271. doi: 10.1002/hipo.20374

Høydal, Ø. A., Skytøen, E. R., Andersson, S. O., Moser, M.-B., and Moser, E. I. (2019). Object-vector coding in the medial entorhinal cortex. *Nature* 568, 400–404. doi: 10.1038/s41586-019-1077-7

Hussaini, S., Milford, M., and Fischer, T. (2022). Spiking neural networks for visual place recognition via weighted neuronal assignments. *IEEE Robot. Automat. Lett.* 7, 4094–4101. doi: 10.1109/LRA.2022.3149030

Kajić, I., Schröder, T., Stewart, T. C., and Thagard, P. (2019). The semantic pointer theory of emotion: Integrating physiology, appraisal, and construction. *Cogn. Syst. Res.* 35–53. doi: 10.1016/j.cogsys.2019.04.007

Kelly, M. A., Arora, N., West, R. L., and Reitter, D. (2020). Holographic declarative memory: distributional semantics as the architecture of memory. *Cogn. Sci.* 44, e12904. doi: 10.1111/cogs.12904

Kim, A., and Eustice, R. M. (2013). Real-time visual slam for autonomous underwater hull inspection using visual saliency. *IEEE Transact. Robot.* 29, 719–733. doi: 10.1109/TRO.2012.2235699

Kim, S., Jung, D., and Royer, S. (2020a). Place cell maps slowly develop via competitive learning and conjunctive coding in the dentate gyrus. *Nat. Commun.* 11, 4550. doi: 10.1038/s41467-020-18351-6

Kim, S., Park, S., Na, B., and Yoon, S. (2020b). "Spiking-yolo: spiking neural network for energy-efficient object detection," in *Proceedings of the AAAI Conference on Artificial Intelligence, Vol.* 34 (New York, NY), 11270–11277.

Knight, J., Voelker, A. R., Mundy, A., Eliasmith, C., and Furber, S. (2016). "Efficient spinnaker simulation of a heteroassociative memory using the neural engineering framework," in *2016 International Joint Conference on Neural Networks (IJCNN)* (Vancouver, CA: IEEE), 5210–5217.

Komer, B., Stewart, T. C., Voelker, A. R., and Eliasmith, C. (2019). "A neural representation of continuous space using fractional binding," in *41st Annual Meeting of the Cognitive Science Society* (Montreal, QC: Cognitive Science Society).

Kreiser, R., Renner, A., Leite, V. R., Serhan, B., Bartolozzi, C., Glover, A., et al. (2020a). An on-chip spiking neural network for estimation of the head pose of the icub robot. *Front. Neurosci.* 14, 551. doi: 10.3389/fnins.2020.00551

Kreiser, R., Waibel, G., Armengol, N., Renner, A., and Sandamirskaya, Y. (2020b). "Error estimation and correction in a spiking neural network for map formation in neuromorphic hardware," in *2020 IEEE International Conference on Robotics and Automation (ICRA)* (Virtual: IEEE), 6134–6140.

Krupic, J., Burgess, N., and O'Keefe, J. (2012). Neural representations of location composed of spatially periodic bands. *Science* 337, 853–857. doi: 10.1126/science.1222403

Lateef, F., and Ruichek, Y. (2019). Survey on semantic segmentation using deep learning techniques. *Neurocomputing* 338, 321–348. doi: 10.1016/j.neucom.2019.02.003

MacNeil, D., and Eliasmith, C. (2011). Fine-tuning and the stability of recurrent neural networks. *PLoS ONE* 6, e22885. doi: 10.1371/journal.pone.0022885

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv:1301.3781*.

Milford, M., Jacobson, A., Chen, Z., and Wyeth, G. (2016). "Ratslam: using models of rodent hippocampus for robot navigation and beyond," in *16th International Symposium of Robotics Research, ISRR '13* (Springer), 467–485.

Milford, M. J., Wyeth, G. F., and Prasser, D. (2004). "Ratslam: a hippocampal model for simultaneous localization and mapping," in *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004, Vol. 1* (New Orleans, LA: IEEE), 403–408.

Mitrokhin, A., Sutor, P., Summers-Stay, D., Fermüller, C., and Aloimonos, Y. (2020). Symbolic representation and learning with hyperdimensional computing. *Front. Robot.* AI, 7. doi: 10.3389/frobt.2020.00063

Mittelstaedt, H., and Mittelstaedt, M.-L. (1982). "Homing by path integration," in *International Symposium on Avian Navigation (ISAN)*, eds F. Papi and H. G. Wallraff (Pisa: Springer), 290–297.

Mittelstaedt, M.-L., and Mittelstaedt, H. (2001). Idiothetic navigation in humans: estimation of path length. *Exp. Brain Res.* 139, 318–332. doi: 10.1007/s002210100735

Montemerlo, M., Thrun, S., Koller, D., and Wegbreit, B. (2002). "Fastslam: A Factored Solution to the Simultaneous Localization and Mapping Problem," in *Proceedings of the AAAI Conference on Artificial Intelligence* (Palo Alto, CA: AAAI Press), 18, 593.

Mundy, A. (2017). *Real Time Spaun on Spinnaker Functional Brain Simulation on a Massively-Parallel Computer Architecture*. Manchester: The University of Manchester.

Navratilova, Z., Godfrey, K. B., and McNaughton, B. L. (2016). Grids from bands, or bands from grids? an examination of the effects of single unit contamination on grid cell firing fields. *J. Neurophysiol.* 115, 992–1002. doi: 10.1152/jn.00699.2015

Oja, E. (1982). Simplified neuron model as a principal component analyzer. *J. Math. Biol.* 15, 267–273. doi: 10.1007/BF00275687

O'Keefe, J., and Burgess, N. (2005). Dual phase and rate coding in hippocampal place cells: theoretical significance and relationship to entorhinal grid cells. *Hippocampus* 15, 853–866. doi: 10.1002/hipo.20115

Orchard, J., Yang, H., and Ji, X. (2013). Does the entorhinal cortex use the Fourier transform? *Front. Comput. Neurosci.* 7, 179. doi: 10.3389/fncom.2013.00179

Osswald, M., Ieng, S.-H., Benosman, R., and Indiveri, G. (2017). A spiking neural network model of 3d perception for event-based neuromorphic stereo vision systems. *Sci. Rep.* 7, 40703. doi: 10.1038/srep40703

Palomeras, N., Carreras, M., and Andrade-Cetto, J. (2019). Active slam for autonomous underwater exploration. *Remote Sens.* 11, 2827. doi: 10.3390/rs11232827

Plate, T. A. (1995). Holographic reduced representations. *IEEE Transact. Neural Netw.* 6, 623–641. doi: 10.1109/72.377968

Rahimi, A., and Recht, B. (2007). Random features for large-scale kernel machines. *Adv. Neural Inf. Process. Syst.* 20, 1177–1184.

Rajalingham, R., and DiCarlo, J. J. (2019). Reversible inactivation of different millimeter-scale regions of primate it results in different patterns of core object recognition deficits. *Neuron* 102, 493–505. doi: 10.1016/j.neuron.2019.02.001

Rasmussen, D., and Eliasmith, C. (2014). "A neural model of hierarchical reinforcement learning," in *Proceedings of the 36th Annual Conference of the Cognitive Science Society*, eds P. Bello, M. Guarini, M. McShane, and B. Scassellati (Austin: Cognitive Science Society), 1252–1257.

Rathi, N., Agrawal, A., Lee, C., Kosta, A. K., and Roy, K. (2021). "Exploring spike-based learning for neuromorphic computing: prospects and perspectives," in *2021 Design, Automation & Test in Europe Conference & Exhibition (DATE)* (Grenoble: IEEE), 902–907.

Safron, A., Catal, O., and Verbelen, T. (2022). Generalized simultaneous localization and mapping (g-slam) as unification framework for natural and artificial intelligences: towards reverse engineering the hippocampal/entorhinal system and principles of high-level cognition. *Front. Syst. Neurosci.* 16, 787659. doi: 10.3389/fnsys.2022.787659

Samsonovich, A., and McNaughton, B. L. (1997). Path integration and cognitive mapping in a continuous attractor neural network model. *J. Neurosci.* 17, 5900–5920. doi: 10.1523/JNEUROSCI.17-15-05900.1997

Sargolini, F., Fyhn, M., Hafting, T., McNaughton, B. L., Witter, M. P., Moser, M.-B., et al. (2006). Conjunctive representation of position, direction, and velocity in entorhinal cortex. *Science* 312, 758–762. doi: 10.1126/science.1125572

Silveira, L., Guth, F., Drews-Jr, P., Ballester, P., Machado, M., Codevilla, F., et al. (2015). An open-source bio-inspired solution to underwater slam. *IFAC PapersOnLine* 48, 212–217. doi: 10.1016/j.ifacol.2015.06.035

Sim, R., Elinas, P., Griffin, M., Shyr, A., and Little, J. J. (2005). "Vision-based slam using the rao-blackwellised particle filter," in *IJCAI Workshop on Reasoning with Uncertainty in Robotics, Vol. 14* (Edinburgh: Citeseer), 9–16.

Smith, R., Self, M., and Cheeseman, P. (1990). Estimating uncertain spatial relationships in robotics. *Autonom. Robot Vehicles* 167–193. doi: 10.1007/978-1-4613-8997-2_14

Stachniss, C., Hahnel, D., and Burgard, W. (2004). "Exploration with active loop-closing for fastslam," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566), Vol. 2* (Sendai: IEEE), 1505–1510.

Steckel, J., and Peremans, H. (2013). Batslam: Simultaneous localization and mapping using biomimetic sonar. *PLoS ONE* 8, e54076. doi: 10.1371/journal.pone.0054076

Stenborg, E., Toft, C., and Hammarstrand, L. (2018). "Long-term visual localization using semantically segmented images," in *2018 IEEE International Conference on Robotics and automation (ICRA)* (Brisbane: IEEE), 6484–6490.

Stewart, T., Choo, F.-X., and Eliasmith, C. (2012a). "Spaun: a perception-cognition-action model using spiking neurons," in *Proceedings of the Annual Meeting of the Cognitive Science Society* (Sapporo), Vol. 34.

Stewart, T. C., Bekolay, T., and Eliasmith, C. (2012b). Learning to select actions with spiking neurons in the basal ganglia. *Front. Neurosci.* 6, 2. doi: 10.3389/fnins.2012.00002

Sünderhauf, N., and Protzel, P. (2012). "Switchable constraints for robust pose graph slam," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems* (Vilamoura-Algarve: IEEE), 1879–1884.

Tang, G., and Michmizos, K. P. (2018). "Gridbot: An autonomous robot controlled by a spiking neural network mimicking the brain's navigational system," in *Proceedings of the International Conference on Neuromorphic Systems* (Knoxville), 1–8.

Tang, G., Shah, A., and Michmizos, K. P. (2019). "Spiking neural network on neuromorphic hardware for energy-efficient unidimensional slam," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Macau: IEEE), 4176–4181.

Thakur, C. S., Molin, J. L., Cauwenberghs, G., Indiveri, G., Kumar, K., Qiao, N., et al. (2018). Large-scale neuromorphic spiking array processors: a quest to mimic the brain. *Front. Neurosci.* 12, 891. doi: 10.3389/fnins.2018.00891

Thrun, S., and Montemerlo, M. (2006). The graph slam algorithm with applications to large-scale mapping of urban structures. *Int. J. Rob. Res.* 25, 403–429. doi: 10.1177/0278364906065387

Tolman, E. C. (1948). Cognitive maps in rats and men. *Psychol. Rev.* 55, 189. doi: 10.1037/h0061626

Tsodyks, M. (1999). Attractor neural network models of spatial maps in hippocampus. *Hippocampus* 9, 481–489. doi: 10.1002/(SICI)1098-1063(1999)9:4<481::AID-HIPO14>3.0.CO;2-S

Tsybakov, A. B. (2009). *Introduction to Nonparametric Estimation*. Berlin: Springer.

Voelker, A. R. (2020). A short letter on the dot product between rotated fourier transforms. *arXiv:2007.13462*.

Voelker, A. R., Blouw, P., Choo, X., Dumont, N. S.-Y., Stewart, T. C., and Eliasmith, C. (2021). Simulating and predicting dynamical systems with spatial semantic pointers. *Neural Comput.* 33, 2033–2067. doi: 10.1162/neco_a_01410

Voelker, A. R., Crawford, E., and Eliasmith, C. (2014). "Learning large-scale heteroassociative memories in spiking neurons," in *Unconventional Computation and Natural Computation, 13th International Conference, UCNC 2014*, London, ON, Canada. doi: 10.1007/978-3-319-08123-6

Wang, R., and Kang, L. (2022). Multiple bumps can enhance robustness to noise in continuous attractor networks. *PLoS Comput. Biol.* 18, e1010547. doi: 10.1371/journal.pcbi.1010547

Welday, A. C., Shlifer, I. G., Bloom, M. L., Zhang, K., and Blair, H. T. (2011). Cosine directional tuning of theta cell burst frequencies: evidence

for spatial coding by oscillatory interference. *J. Neurosci.* 31, 16157–16176. doi: 10.1523/JNEUROSCI.0712-11.2011

Witter, M. P., and Moser, E. I. (2006). Spatial representation and the architecture of the entorhinal cortex. *Trends Neurosci.* 29, 671–678. doi: 10.1016/j.tins.2006.10.003

Yakopcic, C., Rahman, N., Atahary, T., Taha, T. M., and Douglass, S. (2020). "Solving constraint satisfaction problems using the loihi spiking neuromorphic processor," in *2020 Design, Automation & Test in Europe Conference & Exhibition (DATE)* (Grenoble: IEEE), 1079–1084.

Yan, Y., Stewart, T. C., Choo, X., Vogginger, B., Partzsch, J., Höppner, S., et al. (2021). Comparing loihi with a spinnaker 2 prototype on low-latency keyword spotting and adaptive robotic control. *Neuromor. Comp. Eng.* 1, 014002. doi: 10.1088/2634-4386/abf150

Yu, F., Shang, J., Hu, Y., and Milford, M. (2019). Neuroslam: a brain-inspired slam system for 3d environments. *Biol. Cybern.* 113, 515–545. doi: 10.1007/s00422-019-00806-9

Zeng, T., and Si, B. (2017). Cognitive mapping based on conjunctive representations of space and movement. *Front. Neurorobot.* 11, 61. doi: 10.3389/fnbot.2017.00061

Zhang, L., Wei, L., Shen, P., Wei, W., Zhu, G., and Song, J. (2018). Semantic slam based on object detection and improved octomap. *IEEE Access* 6, 75545–75559. doi: 10.1109/ACCESS.2018.2873617

frontiers | Frontiers in Neuroscience

TYPE Original Research
PUBLISHED 14 July 2023
DOI 10.3389/fnins.2023.1221740

Check for updates

OPEN ACCESS

EDITED BY
Alois C. Knoll,
Technical University of Munich, Germany

REVIEWED BY
Yingbai Hu,
Technical University of Munich, Germany
Wang Juan,
South China University of Technology, China

*CORRESPONDENCE
Dan Huang
✉ dan78huang@163.com

RECEIVED 12 May 2023
ACCEPTED 20 June 2023
PUBLISHED 14 July 2023

CITATION
Qiu H, Huang D, Zhang B and Wang M (2023) A novel multidimensional uncalibration method applied to six-axis manipulators.
Front. Neurosci. 17:1221740.
doi: 10.3389/fnins.2023.1221740

COPYRIGHT
© 2023 Qiu, Huang, Zhang and Wang. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

A novel multidimensional uncalibration method applied to six-axis manipulators

Haitao Qiu[1], Dan Huang[2]*, Bo Zhang[3] and Ming Wang[3]

[1]School of Electric Power Engineering, South China University of Technology, Guangzhou, China, [2]School of Mechanical and Automotive Engineering, South China University of Technology, Guangzhou, China, [3]School of Mechanical Engineering, Ningxia University, Yinchuan, China

This study proposes a multidimensional uncalibrated technique for tracking and grasping dynamic targets by a robotic arm in the eye-in-hand mode. This method avoids complex and cumbersome calibration processes, enabling machine vision tasks to be adaptively applied in a variety of complex environments, which solved the problem of traditional calibration methods being unstable in complex environments. The specific method used in this study is first, in the eye-in-hand mode, the robotic arm moves along the *x*, *y*, and *z* axes in sequence, and images are taken before and after each movement. Thereafter, the image Jacobian matrix is calculated from the three (or more) sets of images collected. Finally, the robotic arm converts the target coordinates in the real-time captured images by the camera into coordinates in the robotic arm coordinate system through the image Jacobian matrix and performs real-time tracking. This study tests the dynamic quasi-Newton method for estimating the Jacobian matrix and optimizes the initialization coupling problem using the orthogonal moving method. This optimization scheme significantly shortens the iteration process, making the uncalibrated technology more fully applied in the field of dynamic object tracking. In addition, this study proposes a servo control algorithm with predictive compensation to mitigate or even eliminate the systematic error caused by time delay in dynamic target tracking in robot visual servo systems.

KEYWORDS

image Jacobian matrix, machine vision, uncalibrated visual servo, dynamic quasi-Newton algorithm, robot

1. Introduction

In the 1960s, due to the development of robotics and computer technology, people began to study robots with visual functions, and in the 1980s, the concept of robot visual servo was proposed. In the following decades, robot visual servoing underwent rapid development. Visual servo control mainly inputs visual information provided by visual sensors into the control system, enabling the control system to process external information. Traditional robot visual servo systems are mostly implemented based on system model calibration technology (Gans, 2003; Huang et al., 2022), which mainly involves models such as camera models, robot models, and target object models. The camera model refers to the internal and external parameters of the camera; the robot model generally refers to the robot kinematics model; the target model mainly refers to the depth information from the target to the end of the robotic arm, as well as the pose and motion parameters of the target in a fixed coordinate system. In the traditional robot visual servo system, the first step is to complete the calibration of the camera and the calibration between the camera and the robot (Hutchinson et al., 1996) to obtain an accurate conversion matrix between the

image coordinate system and the robot coordinate system. Then, based on the calibrated transformation matrix, the coordinates of the target object in the image captured by the visual system are converted to obtain the pose of the robot in the coordinate system. Finally, the robot tracks, locates, and grasps the target object in the camera's field of view based on the converted coordinate information (Kang et al., 2020). Throughout the entire work process, the accuracy of the transformation matrix between the image coordinate system and the robot coordinate system is heavily dependent (Malis, 2004). The calibration work between the camera and the robot is extremely cumbersome, requiring data such as the internal and external parameters of the camera, the motion model of the robot model, and the position relationship between the camera and the fixed position of the robot. However, in practical applications, replacing the camera or camera lens, or loosening the installation position between the camera and the robot can cause deviation in the calibration results, requiring complex calibration work to be carried out again. The traditional calibration methods for robot visual servo systems make it difficult for them to operate in complex working environments, which is currently a bottleneck limiting the development of robot visual servo systems.

To break the bottleneck, researchers have begun to focus on studying the "eye-in-hand" structure visual servo control method for calculating the image Jacobian matrix without knowing system parameters. The robot visual servo system still needs to overcome many technical difficulties to be put into normal use in various complex production environments.

The development of uncalibrated technology between cameras and robots without knowing system parameters can be divided into multiple stages: 1. The robot visual servo system achieves precise positioning and grasping of static targets through uncalibrated technology; 2. the robot visual servo system achieves tracking and positioning of dynamic targets through uncalibrated technology; and 3. the robot visual servo system achieves practical production applications with low latency and high accuracy in complex environments.

The fundamental goal of implementing a robot visual servo system is to achieve precise positioning and grasping of static targets. Hosoda and Asada first proposed the exponential weighted recursive least squares method to obtain the Jacobian matrix. This method achieves servo tracking and positioning of stationary targets in an uncalibrated state, but there are still shortcomings in terms of system stability and accuracy of image feature extraction (Hosoda and Asada, 1994; Cao et al., 2022a,b). Yoshimi and Allen introduced an additional robotic arm to explore motion and observed corresponding changes in image features during each calculation cycle. Then, they combined the least square method to calculate the Jacobian matrix of the current image, achieving more accurate two-dimensional target tracking. However, this method is too cumbersome and lacks real-time performance, making it difficult to apply in practical work (Yoshimi and Allen, 1995). In addition, many researchers have obtained the image Jacobian matrix by converting the online estimation of the Jacobian matrix into system state observation (Jianbo, 2004) or recursive formula calculation (Longjiang et al., 2003) and tested the algorithm from four aspects: initial value, operating range, stability, and robustness. Simulation experiments have been conducted to verify the reliability of the algorithm (Hao and Sun, 2007). At this stage, it is possible to use robot visual servo systems for positioning and grasping static targets in industrial production applications that meet various requirements (Singh et al., 1998). Compared to traditional calibration methods (Jingmei et al., 2014), it avoids the tedious process of repeated calibration.

With the development of production technology, the function of only achieving precise positioning and grasping static targets no longer meets the production needs of enterprises. Therefore, Piepmeier proposed the Broyden method to estimate the image Jacobian matrix, thereby achieving tracking and positioning of moving targets. However, when the deviation of image features is large, the performance of the control system will decrease, even leading to control failure (Piepmeier and Lipkin, 2003). When the robot visual servo system tracks irregularly moving targets (Haifeng et al., 2010), it is necessary to improve the real-time performance of the system (Zaien et al., 2014) and the convergence speed of the image Jacobian matrix (Chang et al., 2020). However, while ensuring the real-time performance of the system, it can also lead to problems such as slow recognition speed and low accuracy of the visual system during high-speed movement. Many researchers have combined BP neural networks and genetic algorithms (Samad and Haq, 2016; Chen et al., 2020; Yuhan et al., 2021; Wu et al., 2022) and applied them to real-time image processing in the visual system, improving the processing speed of the visual system, improving the processing speed of the visual system. In addition, it is necessary to improve the robustness of the robot's visual servo system (Li et al., 2009; Hao et al., 2020) to adapt to stable operation in various complex environments. For example, in the field of medical equipment, the robot servo system needs to operate absolutely accurately and stably (Piepmeier, 2003; Gu et al., 2018; Zhang et al., 2020), thus improving the robustness and anti-interference of the system is very important (Cao et al., 2021; Gao and Xiao, 2021).

This study researches the application background of tracking and trajectory coverage of irregular dynamic targets. First, an online estimation test of the dynamic quasi-Newtonian Jacobian matrix was conducted in the simulation system. After analyzing the simulation test results, the system initialization process was targeted and optimized, significantly improving the convergence speed of Jacobian matrix iteration. In addition, this study also proposes a predictive compensation Jacobian matrix PI control algorithm to solve the lag problem of the visual system in the dynamic tracking process, effectively improving the accuracy of the robot servo system in the dynamic tracking process.

The remainder of this article is structured as follows. In Section 2, a detailed introduction is given to the control system. This includes the hardware composition of the control system, theoretical deduction of uncalibrated technology, and an introduction to servo control algorithms. In Section 3, we present the experimental results and discuss them. These results include the iterative process for the proposed uncalibrated visual servo system and the optimized iterative process. In addition, a comparative analysis of the research and experimental data conducted in this study is also presented in Section 4.

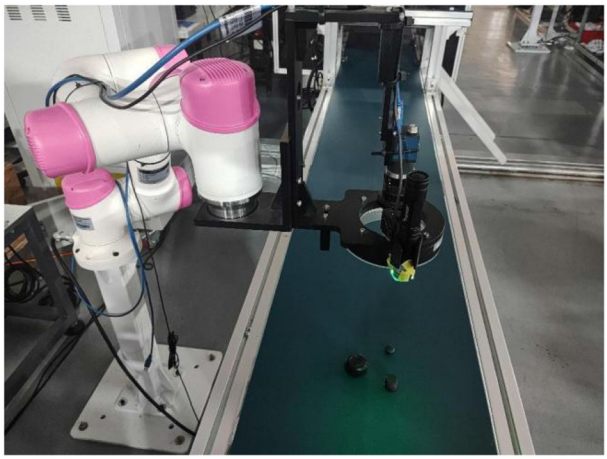


FIGURE 1
Bozhilin 6-axis robotic arm platform.

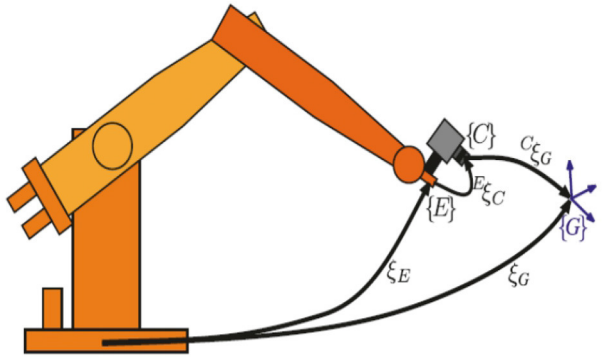


FIGURE 2
Schematic diagram of the eye-in-hand model.

2. Control system

2.1. Operating platform

The robot uncalibrated servo technology reviewed in this study is based on the application of tracking and coating trajectories to moving targets. The technology analyzed in this study can be applied to different fields such as the application of mobile robots to building cracks and robot welding. The robot platform used in this study is a six-axis industrial robot independently developed by Bozhilin, as shown in Figure 1. A Daheng high-speed industrial camera is installed at the end of the robotic arm to collect image information within the working range of the robotic arm. The camera used needs to have a large field of view, as the target object cannot leave the camera's field of view during uncalibrated initialization; otherwise, it will cause the Jacobian matrix error to increase. The camera and robot are installed in the eye-in-hand mode, and the model diagram is shown in Figure 2.

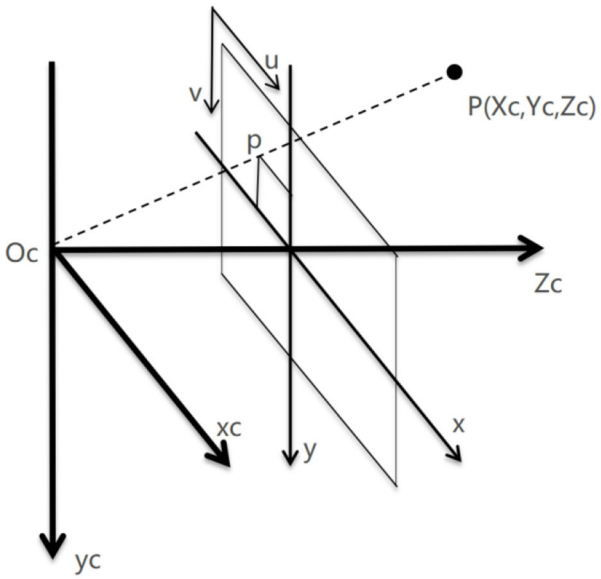


FIGURE 3
Camera pinhole imaging model.

2.2. Process of uncalibration

Uncalibration technology, such as traditional calibration techniques, is used to describe the relationship between the speed of robot end effectors and the rate of feature change in the image. Assuming a point P in three-dimensional space, based on the traditional camera pinhole imaging model as shown in Figure 3, it can be concluded that

$$\begin{cases} x_i = \frac{f}{z_c} x_c \\ y_i = \frac{f}{z_c} y_c \end{cases} \tag{1}$$

$P_c(x_c, y_c, z_c)$ is the coordinate of point P in the camera coordinate system, $P_w(x_w, y_w, z_w)$ is the Cartesian coordinate of point P in the world coordinate system (robotic arm base coordinate system), $P_I(x_i, y_i)$ is the projection coordinate of point P in the camera plane coordinate system, and (u_i, v_i) is the pixel coordinate in the pixel plane coordinate system.

The relationship between the camera imaging plane coordinate $P_I(x_i, y_i)$ and the pixel plane coordinate (u_i, v_i) is

$$\begin{cases} u_i = \frac{x_i}{dx} + u_0 \\ v_i = \frac{y_i}{dy} + v_0 \end{cases} \tag{2}$$

In the above equation, u_0 and v_0 are the pixel coordinates of the penetration point of the camera's optical axis in the pixel plane, while dx and dy represent the spatial distance represented by a single pixel in the X and Y directions in the pixel plane, respectively.

Convert the above equation into a matrix equation as follows:

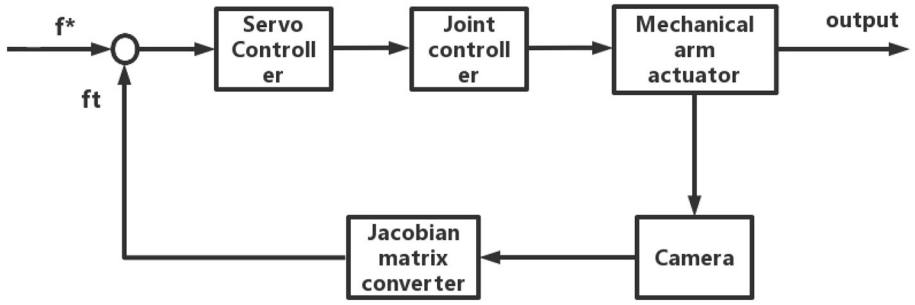


FIGURE 4
Framework diagram of the robot servo system.

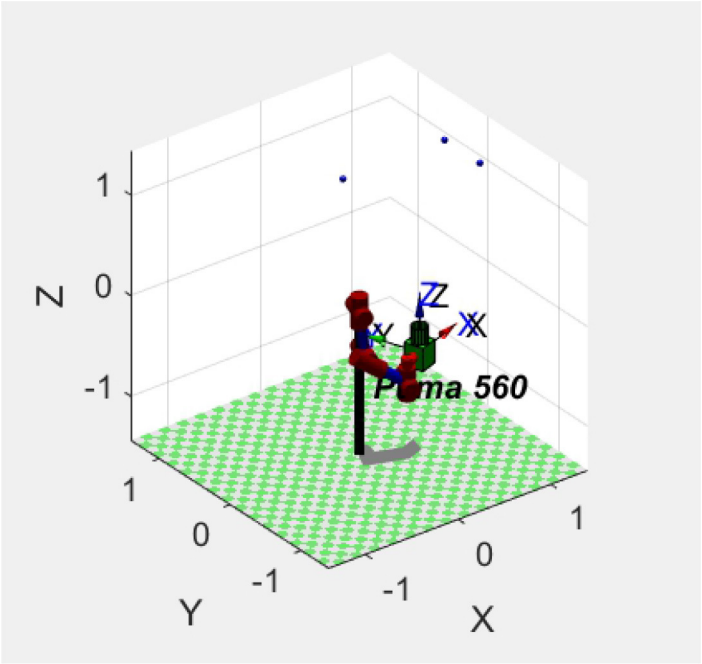


FIGURE 5
Simulation model of Puma560 Robot Arm Servo System.

$$\begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{dx} & 0 & u_0 \\ 0 & \frac{1}{dy} & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} \tag{3}$$

Assuming that the focal length of the camera is f, under the ideal pinhole model of the eye-in-hand system, the conversion relationship between the camera coordinate system and the pixel coordinate system is

$$\begin{bmatrix} u_i \\ v_i \end{bmatrix} = \frac{f}{z_c} \begin{bmatrix} x_c \\ y_c \end{bmatrix} \tag{4}$$

According to the motion equation of the robot's end effector, we have

$$P^c = \Omega^c * P^c + T^c \tag{5}$$

$$\begin{cases} x_c = z_c w_y + T_x - \frac{v_i z_c}{f} w_z \\ y_c = \frac{u_i z_c}{f} w_z - z_c w_x + T_y \\ z_c = \frac{v_i z c}{f} w_x - \frac{u_i z_c}{f} w_y + T_z \end{cases} \tag{6}$$

Converting the above equation into a matrix equation, we obtain as follows:

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \frac{f}{z_c} & 0 & -\frac{u_i}{z_c} & -\frac{u_i v_i}{f} & \frac{f^2+u_i^2}{f} & -v_i \\ 0 & -\frac{f}{z_c} & -\frac{v_i}{z_c} & -\frac{f^2+V_i^2}{f} & \frac{u_i v_i}{f} & u_i \end{bmatrix} \cdot \begin{bmatrix} T^c \\ \Omega^c \end{bmatrix} \tag{7}$$

In practical applications, it is impossible to obtain the transformation matrix between (u_i, v_i) and $[T^c, \Omega^c]^T$ by measuring

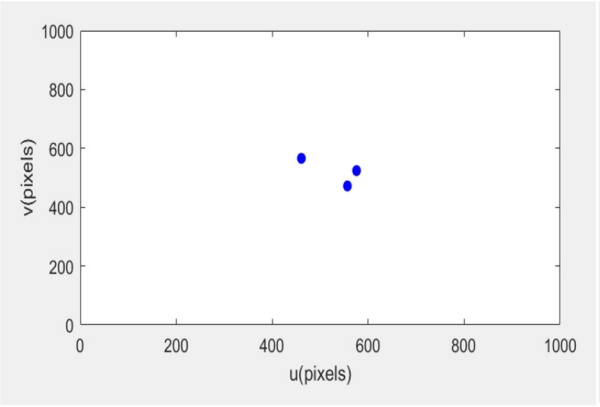


FIGURE 6
Initial position of the servo system.

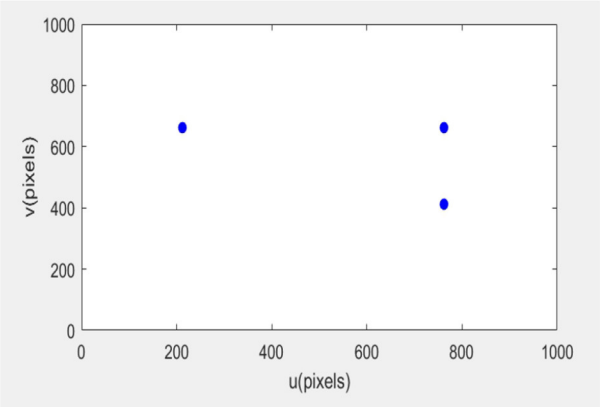


FIGURE 7
Position of the servo system in the end.

each variable in the above equation. Therefore, the variables in the matrix are considered unknown:

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} & a_{16} \\ b_{11} & b_{12} & b_{13} & b_{14} & b_{15} & b_{16} \end{bmatrix} \cdot \begin{bmatrix} T_x \\ T_y \\ T_z \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \tag{8}$$

On the six-axis robotic arm platform, a single feature pixel does not meet the dimensional requirements, so three feature points are taken and stacked up and down:

$$\begin{bmatrix} \dot{u_1} \\ \dot{v_1} \\ \dot{u_2} \\ \dot{v_2} \\ \dot{u_3} \\ \dot{v_3} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} & a_{16} \\ b_{11} & b_{12} & b_{13} & b_{14} & b_{15} & b_{16} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} & a_{26} \\ b_{21} & b_{22} & b_{23} & b_{24} & b_{25} & b_{26} \\ a_{31} & a_{32} & a_{33} & a_{34} & a_{35} & a_{36} \\ b_{31} & b_{32} & b_{33} & b_{34} & b_{35} & b_{36} \end{bmatrix} \cdot \begin{bmatrix} T_x \\ T_y \\ T_z \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \tag{9}$$

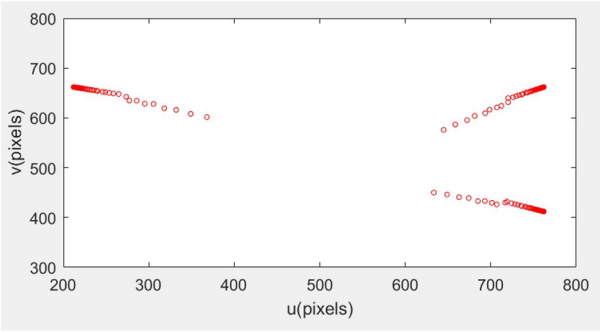


FIGURE 8
Motion trajectory of feature points in the image plane.

$\dot{F}$ represents the rate of change of image features, J_0 represents the Jacobian transformation matrix, and $\dot{P}$ represents the motion vector of the robotic arm end effector. The above equation can be expressed as follows:

$$\dot{F} = J_0 \dot{P} \tag{10}$$

In practical applications, we need to convert the two change rates $\dot{F}$ of image features to obtain the motion vector $\dot{P}$ of the robotic arm end effector, so we need to inverse the Jacobian matrix $J = {J_0}^{-1}$.

$$\dot{P} = J_0 \dot{F} \tag{11}$$

In application, two change rates of image features are obtained from two adjacent images, so discretization of equations is also required. In the process of high-frequency camera image retrieval, we assume that the Jacobian matrix of adjacent two frames of images remains approximately unchanged. The discrete equation can be obtained as follows:

$$F_{(n+1)} \approx F_{(n)} + J_{(n)} \cdot \Delta P_{(n)} \tag{12}$$

$$P_{(n+1)} \approx P_{(n)} + {J_{(n)}}^{-1} \cdot \Delta F_{(n)} \tag{13}$$

$$J = \Delta F \cdot \Delta P^{-1} \tag{14}$$

During the initialization process of the robot visual servo system, there is a coupling relationship between multiple movements of the robot, which can lead to the irreversibility and solvability of the Jacobian matrix. In order to obtain a more accurate Jacobian matrix, this article optimized the initialization process of the robot visual servo system. Therefore, by standardizing the movement direction of the robotic arm during the initialization process, the obtained feature point set is naturally linearly uncorrelated by decomposing the movement of the robotic arm into independent movements of each degree of freedom $\begin{bmatrix} T_x & T_y & T_z & \omega_x & \omega_y & \omega_z \end{bmatrix}$ during the initialization process. When moving in the independent T_x direction, we get

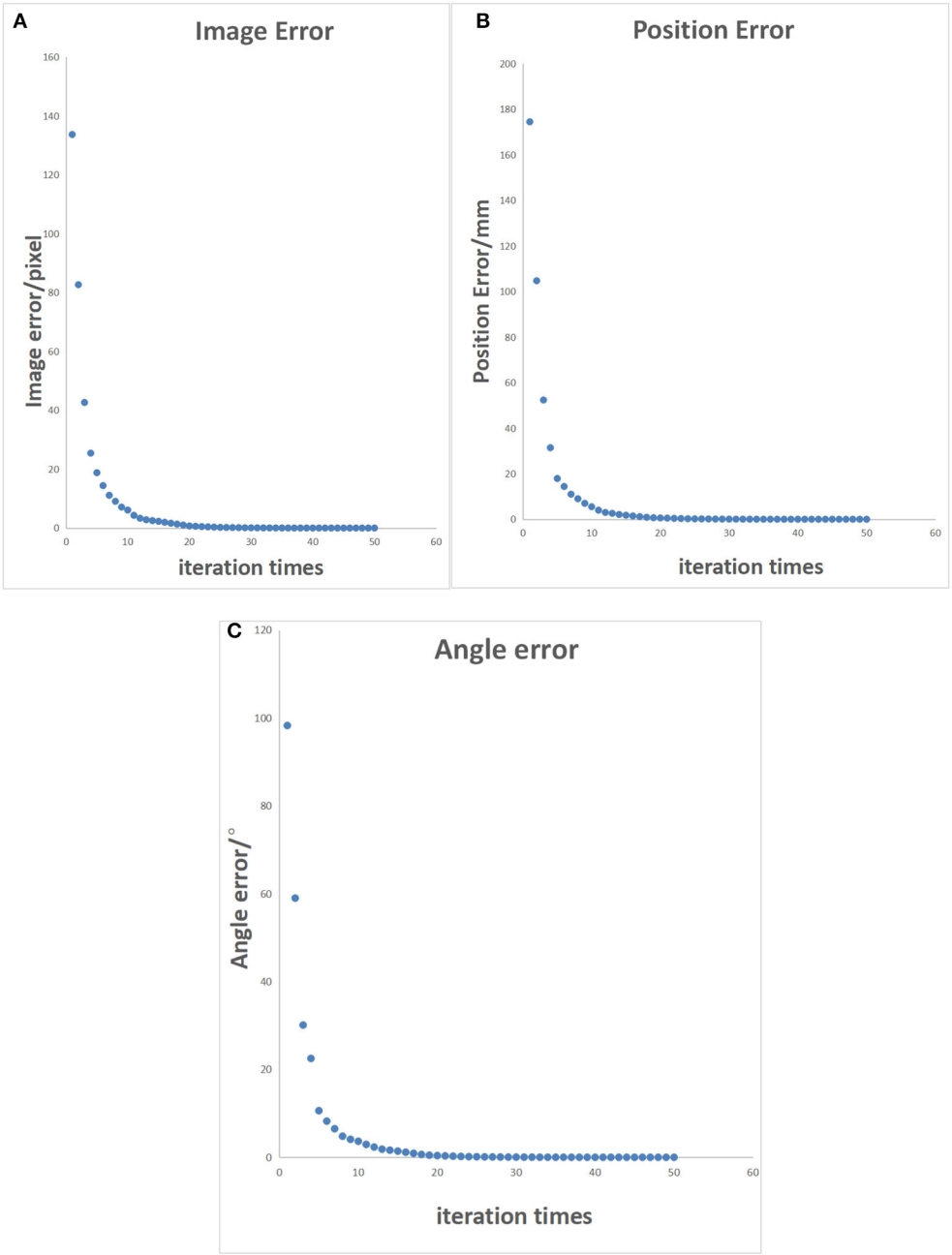


FIGURE 9
Characteristic points error control chart. **(A)** Pixel error of characteristic points on the image. **(B)** Position error of characteristic points in reality. **(C)** Angle error of the robotic arm.

$$\begin{bmatrix} \dot{u_1} \\ \dot{v_1} \\ \dot{u_2} \\ \dot{v_2} \\ \dot{u_3} \\ \dot{v_3} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} & a_{16} \\ b_{11} & b_{12} & b_{13} & b_{14} & b_{15} & b_{16} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} & a_{26} \\ b_{21} & b_{22} & b_{23} & b_{24} & b_{25} & b_{26} \\ a_{31} & a_{32} & a_{33} & a_{34} & a_{35} & a_{36} \\ b_{31} & b_{32} & b_{33} & b_{34} & b_{35} & b_{36} \end{bmatrix} \cdot \begin{bmatrix} T_x \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (15) \qquad \dot{F} = T_x \begin{bmatrix} a_{11} \\ b_{11} \\ a_{21} \\ b_{21} \\ a_{31} \\ b_{31} \end{bmatrix} + 0 * \begin{bmatrix} a_{12} \\ b_{12} \\ a_{22} \\ b_{22} \\ a_{32} \\ b_{32} \end{bmatrix} + 0 * \begin{bmatrix} a_{13} \\ b_{13} \\ a_{23} \\ b_{23} \\ a_{33} \\ b_{33} \end{bmatrix}$$

$$+0*\begin{bmatrix} a_{14} \\ b_{14} \\ a_{24} \\ b_{24} \\ a_{34} \\ b_{34} \end{bmatrix} + 0*\begin{bmatrix} a_{15} \\ b_{15} \\ a_{25} \\ b_{25} \\ a_{35} \\ b_{35} \end{bmatrix} + 0*\begin{bmatrix} a_{16} \\ b_{16} \\ a_{26} \\ b_{26} \\ a_{36} \\ b_{36} \end{bmatrix} \tag{16}$$

After completing the initialization of the image Jacobian matrix, it is necessary to update and iterate the matrix in real time to ensure accuracy during the robot operation process. In the image plane, the difference between the actual feature and the expected feature is $f(\theta, t) = y(\theta, t) - y^*$, where θ is the joint angle and t is time. Taylor expansion is performed on the deviation function $f(\theta, t)$ and the radiation model is defined as $m(\theta, t)$.

$$m(\theta, t) = f(\theta_k, t_k) + J(\theta - \theta_k) + \frac{\partial f_k}{\partial t}(t - t_k) \tag{17}$$

At moment *k-1*, we get

$$\begin{aligned} &f(\theta_{k-1}, t_{k-1}) = m(\theta_{k-1}, t_{k-1}) \\ &= f(\theta_k, t_k) + J_k(\theta_{k-1} - \theta_k) + \frac{\partial f_k}{\partial t}(t_{k-1} - t_k) \end{aligned} \tag{18}$$

The iterative equation can be obtained as follows:

$$J_k = J_{k-1} + \frac{(\Delta f_k - J_{k-1}\Delta\theta - \frac{\partial f_k}{\partial t}\Delta t)\Delta\theta^t}{\Delta\theta^t \Delta\theta} \tag{19}$$

2.3. Servo control algorithm

The process of running a robot visual servo control system is as follows: first, the visual system captures images and processes them, and then the processed image information inputs into the robot controller to start the robot moving. There is a time delay between the visual system capturing images and the robot starting to move, which can cause systematic errors in the robot's tracking of dynamic targets. Therefore, in the process of robot motion control, this study designs a Jacobian matrix PI control algorithm with predictive compensation to reduce systematic errors caused by the time lag.

Assuming that the expected image feature of the moving target is $f^*(u^*, v^*)$ and the actual feature of the robot pose after the image Jacobian matrix transformation is $f_t(u_t, v_t)$ the actual pose and expected pose feature error of the system are as follows:

$$e(t) = f^* - f_t \tag{20}$$

In order to improve the real-time performance of the system and ensure that the target motion speed is fast and can complete effective tracking tasks, a predictive compensation method is introduced into the Jacobian matrix control algorithm on the inverse Jacobian matrix visual servo control algorithm, and a Jacobian matrix PI control algorithm with predictive compensation is designed. We define the system image feature error as follows:

$$e_h(t) = f^d - f^h{}_t \tag{21}$$

TABLE 1 Characteristic point iteration error data table.

Iterations	Image error/pixel	Position error/mm	Angle error/°
1	133.68860	174.48133	98.2602
5	18.79910	17.89122	10.6121
10	6.10400	5.50026	3.6511
15	2.29155	1.76988	1.4074
20	0.70234	0.57714	0.3932
25	0.23001	0.18867	0.1286
30	0.07536	0.06163	0.0420
35	0.02469	0.02002	0.0137
40	0.00809	0.00639	0.0044
45	0.00265	0.00193	0.0014
50	0.00087	0.00047	0.0004

TABLE 2 Optimized feature point iteration error data table.

Iterations	Image error/pixel	Position error/mm	Angle error/°
1	133.68860	174.48133	98.2602
2	7.1235	7	4.1
3	2.52	2.1	1.62
4	0.70234	0.57714	0.3932
5	0.4	0.32	0.2
25	0.0043	0.0037	0.0018
50	0.00087	0.00047	0.0004

In the above equation, $f^h{}_t$ is the current image feature, and f^d is the expected image feature. The predicted compensation amount ξ is defined as follows:

$$\xi = kV_{image} \tag{22}$$

V_{image} is the rate of change in image features, and k is the compensation coefficient.

In the process of dynamic target tracking, in order to reduce system tracking error, the PI control algorithm is introduced into the inverse Jacobian matrix control algorithm, with a control amount of

$$u_h(n) = \Delta f^h(n+1) = f^h(n+1) - f^h(n) \tag{23}$$

In order to reduce the impact of system image processing time delay on the system, the compensation amount will be predicted ξ bringing it into the control algorithm to obtain the final visual servo control algorithm:

$$u_h(n+1) = J(K_P e_h(n) + K_I \sum_{i=0}^{n} e_h(n)) + kV_{image} \tag{24}$$

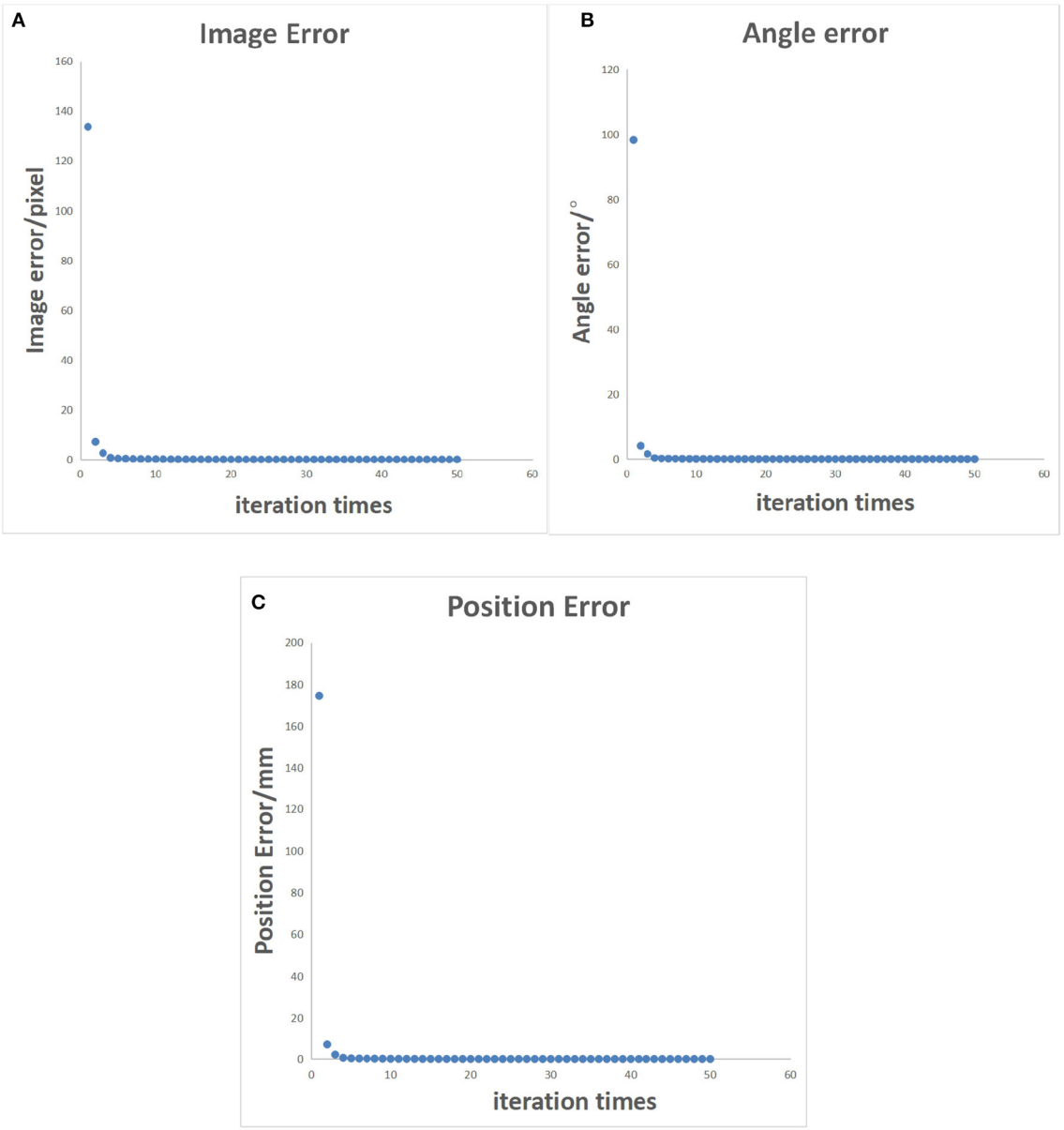


FIGURE 10
Error control chart after decoupled optimization. **(A)** Pixel error of characteristic points on the image. **(B)** Position error of characteristic points in reality. **(C)** Angle error of the robotic arm.

K_P and K_I represent the proportional and differential coefficients, while k represents the predictive compensation coefficient of the system, which is related to the rate of change of image features. As shown in Figure 4, the robot control system is combined with the visual system to form a closed-loop robot visual servo system.

3. Experimental results

3.1. Simulation test

To verify the correctness of the uncalibrated visual servo algorithm, a robotic arm model, a monocular camera model, and a target object model were established in the simulation platform MATLAB by simulating real robotic arm servo experiments. A camera robotic arm model with "eyes in hand" was adopted, and the Jacobian online estimation algorithm using the dynamic quasi-Newton method was used for visual feedback. By using a visual controller, the control amount is calculated using image feature deviation to drive the end of the robotic arm to move toward the target. Finally, the effectiveness of the uncalibrated visual servo algorithm was verified through simulation experiments, providing a theoretical basis for practical development work.

We established a robotic arm model, monocular camera model, and target object model in the simulation platform MATLAB. The robotic arm is a six-axis Puma560 robotic arm. The camera has a

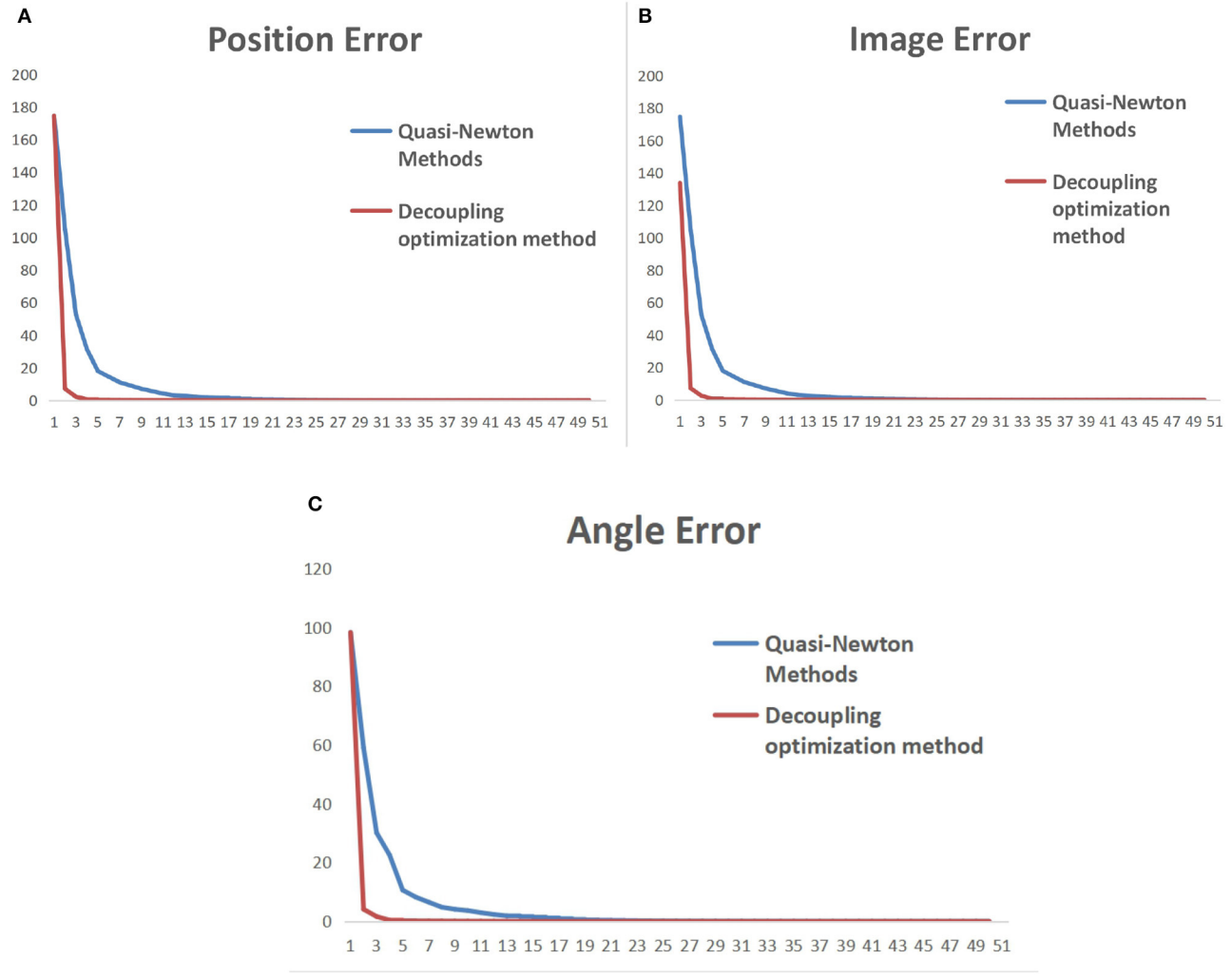


FIGURE 11
Comparison of iterative convergence before and after initialization optimization. **(A)** Pixel error of characteristic points on the image. **(B)** Position error of characteristic points in reality. **(C)** Angle error of the robotic arm.

resolution of 1,024 * 1,024, a focal length of 8 mm, and is installed at the end of the robotic arm (eye in hand). The target object is three small balls located above the robotic arm.

At the initial moment, the end of the robotic arm undergoes six exploratory movements. As shown in Figure 5, it is a simulation model of the servo system. The robotic arm is Puma560, and the camera is installed at the end of the robotic arm in green. The three blue balls in the picture are the target objects. Robot movement generates displacement ΔP_0 at the end of the robotic arm and the displacement of feature points ΔF_0 within the image plane. The initial value of the Jacobian matrix is

$$J_0 = \Delta F_0 \cdot \Delta P_0^{-1} \tag{25}$$

Using the dynamic quasi-Newton method to update the Jacobian matrix, the update frequency of the robotic arm is set to 0.1–0.2 mm per movement until the pixel error of the image reaches the range.

3.2. Experimental results of the dynamic quasi-Newton method

The error of the robotic arm in this experiment after 20 iterations is 0.31. After 35 iterations, the error is 0.011. After 56 iterations, the error was 0.0001, and the final image coordinates of the small ball were 761.999661.999, 761.999412.0, and 212.0661.999, respectively. The initial expected pixel coordinates were 762662, 762412, and 212662.

The initial posture of the robotic arm servo system and the pixel coordinates of three small balls are shown in Figure 6. The posture and ball pixel coordinates at the end of the servo are shown in Figure 7. The motion trajectories of the feature points of three small balls in the image plane are shown in Figure 8. The error of the entire process (image error and robotic arm end pose error) varies with the number of cycles, as shown in Figure 9.

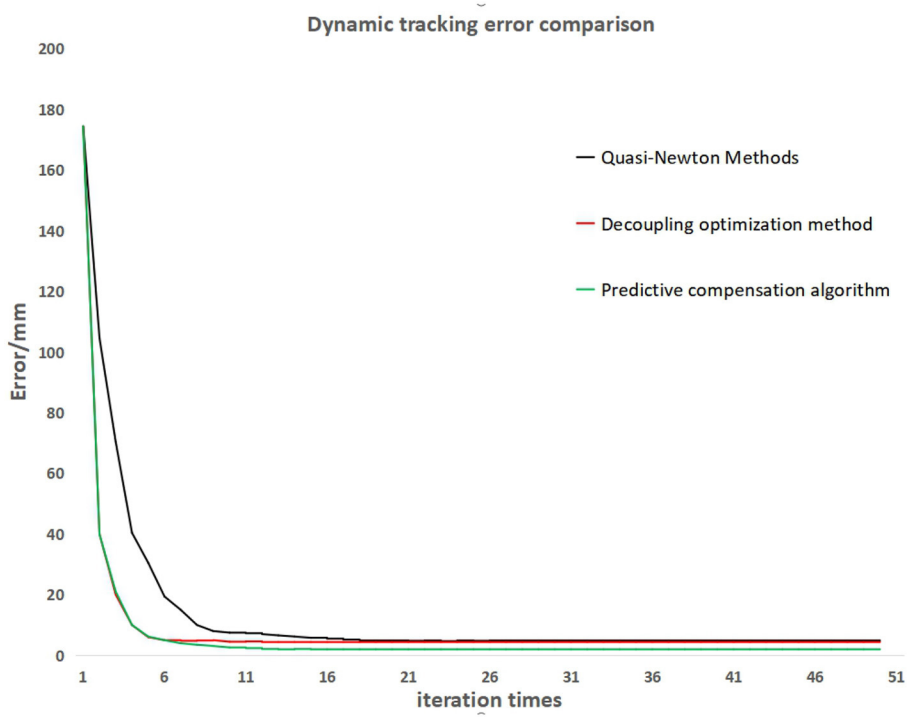


FIGURE 12
Dynamic tracking error comparison.

3.3. Orthogonal initialization method test results

According to the simulation experiment results shown in Table 1, it can be seen that the uncalibrated system requires multiple iterations to achieve the specified accuracy. However, in actual production environments, there is no enough time for iterative optimization. Looking at the simulation results data, it was found that the Jacobian matrix obtained from the uncalibrated initialization of the original scheme had a significant error in conversion. Through analysis, it was found that during the initialization process, images before and after movement were obtained by moving the robotic arm. In this process, there is coupling in the movement of the manipulator, which will lead to an irreversible and unsolvable Jacobian matrix.

To make the multiple sets of image feature points obtained after the robotic arm moves linearly uncorrelated, it is necessary to decouple the collected feature point set. This will be an incredibly complex and cumbersome task. Therefore, by standardizing the movement direction of the robotic arm during the initialization process, the obtained feature point set is naturally linearly uncorrelated. The iterative process error data of the uncalibrated system after the decoupling optimization initialization process is shown in Table 2, and the error of the entire process varies with the number of cycles, as shown in Figure 10. In Figure 11, it can be seen that the iterative speed of the Jacobian matrix after decoupling optimization has been significantly improved. Faster iterative convergence speed can effectively improve the real-time performance of robot visual servo systems during dynamic tracking.

In Figure 11, the vertical axis represents the error during the robot iteration process, and the horizontal axis represents the number of iterations. The update cycle for each iteration of the robot is not fixed. The iterative process includes camera shooting, image processing, and robot motion. Due to the different amount of information in each cycle, the iteration period will fluctuate between 20 and 30 ms.

3.4. Comparison of experimental results

The above simulation tests have verified the reliability of the dynamic quasi-Newton method and the iterative algorithm after decoupling optimization. Next, the two algorithms mentioned above and the servo control algorithm with predictive compensation will be tested on the robotic arm. During the testing process, the robot dynamically tracks the target ball moving on the conveyor belt. The tracking process error data is recorded by identifying the distance between the centroid position of the target ball in the photos captured by the camera during the tracking process and the laser point position vertically shot by the robot arm. The tracking error curves of the three algorithms are shown in Figure 12.

From the tracking error curve in Figure 12, it can be observed that the iterative algorithm after decoupling optimization and the servo control algorithm with predictive compensation have a faster

convergence speed than the dynamic equal Newton method. The servo control algorithm with predictive compensation can further reduce the tracking error in the convergence state.

4. Conclusion

This study investigates the application of the dynamic equal Newton method, the iterative algorithm after decoupling optimization, and the servo control algorithm with predictive compensation in robot uncalibrated visual servo systems. However, due to the dynamic equal Newton method requiring multiple iterations to obtain an accurate Jacobian matrix, a decoupling optimization method for the initialization process was proposed by analyzing the entire process of the uncalibrated robot visual servo system. The iterative algorithm after decoupling optimization can effectively reduce the number of iterations and improve the convergence speed of the Jacobian matrix through simulation testing. Therefore, this algorithm has a high practical value in production applications.

Due to the time lag that cannot be completely eliminated when moving from the visual system to the robot's active position information in the eye-in-hand mode, this study proposes a method called the servo control algorithm with predictive compensation to weaken or even eliminate the tracking error caused by the time lag. It showed a very significant effect on the experimental test results.

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

HQ conducted theoretical research, algorithm design, and paper writing for the article. DH conducted simulation testing and built the framework of the robot visual servo system. BZ completed the collection and analysis of experimental data. MW has completed the optimization and revision of the paper content. All authors contributed to the article and approved the submitted version.

Funding

The study was supported by the Science and Technology Planning Project of Guangzhou City (Grant No. 2023A04J1691).

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

Cao, H., Chen, G., Li, Z., Feng, Q., Lin, J., and Knoll, A. (2022a). Efficient grasp detection network with gaussian-based grasp representation for robotic manipulation. *IEEE ASME Trans. Mech*. doi: 10.1109/TMECH.2022.3224314

Cao, H., Chen, G., Li, Z., Feng, Q., Lin, J., and Knoll, A. (2022b). NeuroGrasp: multimodal neural network with euler region regression for neuromorphic vision-based grasp pose estimation. *IEEE Trans. Instrument. Measure.* 71, 1–11. doi: 10.1109/TIM.2022.3179469

Cao, H., Chen, G., Xia, J., Zhuang, G., and Knoll, A. (2021). Fusion-based feature attention gate component for vehicle detection based on event camera. *IEEE Sensors J.* 21, 24540–24548. doi: 10.1109/JSEN.2021.3115016

Chang, Y., Li, L., Wang, Y., and You, K. (2020). Toward fast convergence and calibration-free visual servoing control: a new image based uncalibrated finite time control scheme. *IEEE Access* 8, 88333–88347. doi: 10.1109/ACCESS.2020.2993280

Chen, G., Cao, H., Conradt, J., Tang, H., Rohrbein, F., and Knoll, A. (2020). Event-based neuromorphic vision for autonomous driving: a paradigm shift for bio-inspired visual sensing and perception. *IEEE Signal Process. Magazine* 37, 34–49. doi: 10.1109/MSP.2020.2985815

Gans, N. R. (2003). Performance tests for visual servo control systems, with application to partitioned approaches to visual servo control. *Int. J. Robot. Res.* 22, 955–981. doi: 10.1177/027836490302210011

Gao, Q., and Xiao, W. (2021). *Research on the Robot Uncalibrated Visual Servo Method Based on the Kalman Filter With Optimized Parameters*. Singapore: Springer.

Gu, J., Wang, W., Zhu, M., Lv, Y., Huo, Q., and Xu, Z. (2018). "Research on a technology of automatic assembly based on uncalibrated visual servo system," in *2018 IEEE International Conference on Mechatronics and Automation (ICMA). IEEE.*

Haifeng, L., Jingtai, L., Yan, L., Xiang L, Lei S. (2010). *Visual Servoing With an Uncalibrated Eye-in-Hand Camera*. Technical Committee on Control Theory, Chinese Association of Automation (Beihang University Press), 3741–3747.

Hao, M., and Sun, Z. (2007). "Uncalibrated eye-in-hand visual servoing using recursive least squares," in *IEEE International Conference on Systems, Man and Cybernetics, IEEE* (2007).

Hao, T., Wang, H., Xu, F., Wang, J., and Miao, Y. (2020). "Uncalibrated visual servoing for a planar two link rigid-flexible manipulator without joint-space-velocity measurement," in *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 1–13.

Hosoda, K., and Asada, M. (1994). "Versatile visual servoing without knowledge of true Jacobian. Intelligent robots and system' 94'. 'Advanced robotic systems and the real world',IROS' 94," in *Proceedings of the IEEE /RSJ/GI International Conference on IEEE*, Vol. 1, 486–193

Huang, H., Bian, X., Cai, F., Li, J., Jiang, T., Zhang, Z., et al. (2022). A review on visual servoing for underwater vehicle manipulation systems automatic control and case study. *Ocean Eng.* 260, 112065. doi: 10.1016/j.oceaneng.2022.112065

Hutchinson, S., Hager, G. D., and Corke, P. I. (1996). A tutorial on visual servo control. *IEEE Trans. Robot Automat.* 12, 651–670. doi: 10.1109/70.538972

Jianbo, S. (2004). *Uncalibrated Robotic Hand-Eye Coordination of Full Degree-of-Freedoms Based on Fuzzy Neural NetWork* (苏 剑 波), 42–44. doi: 10.13245/j.hust.2004.s1.012

Jingmei, Z., Pengfei, D., and Tie, Z. (2014). Positioning and grasping system design of industrial robot based on visual guidance. *Machine Design Res.* 30, 45–49. doi: 10.13952/j.cnki.jofmdr.2014.0135

Kang, M., Chen, H., and Dong, J. (2020). Adaptive visual servoing with an uncalibrated camera using extreme learning machine and Q-leaning. *Neurocomputing* 402, 384–394. doi: 10.1016/j.neucom.2020.03.049

Li, Y. X., Mao, Z. Y., and Tian, L. F. (2009). Visual servoing of 4DOF using image moments and neural network. *Control Theory Appl. 26*, 1162–1166.

Longjiang, X., Bingyu, S., Dingyu, X., and Xinhe, X. (2003). Model independent uncallbration visual servo control. *Robot* 25, 424–427. doi: 10.13973/j.cnki.robot.2003.05.009

Malis, E. (2004). Visual servoing invarant to changes in camera-intrinsic parameters. *IEEE Trans. Robot. Automat.* 20, 72–81. doi: 10.1109/TRA.2003.820847

Piepmeier, J. A. (2003). "Experimental results for uncalibrated eye-in-hand visual servoing," in *IEEE*, 335–339.

Piepmeier, J. A., and Lipkin, H. (2003). Uncalibrated eye-in-hand visual servoing. *Int. J. Robot. Res.* 22, 805–819. doi: 10.1177/027836490302210002

Samad, A. A. I., and Haq, M. Z. (2016). *Uncalibrated Visual Servoing Using Modular MRAC Architecture*. doi: 10.13140/RG.2.2.25994.34244

Singh, R., Voyles, R. M., Littau, D., and Papanikolopoulos, N. P. (1998). "Grasping real objects using virtual images," in *IEEE Conference on Decision and Control (Cat. No.98CH36171), Tampa, FL, USA*, Vol. 3, 3269–3274.

Wu, W., Su, H., and Gou, Z. (2022). "Research on precision motion control of micro-motion platform based on uncalibrated visual servo," in *2022 4th International Conference on Control and Robotics (ICCR), Guangzhou, China*, 77–81.

Yoshimi, B. H., and Allen, P.K. (1995). Alignment using an uncalibrated camera system. *IEEE Trans. Robot. Automat.* 11, 516–521. doi: 10.1109/70.406936

Yuhan, D., Lisha, H., and Shunlei, L. (2021). Research on computer vision enhancement in intelligent robot based on machine learning and deep learning. *Neural Comput. Appl.* doi: 10.1007/S00521-021-05898-8

Zaien, Y., Xueliang, P., Zhengyang, L., Yi, J., and Shenglong, C. (2014). The simulation and reconstruction of the complex robot trajectories based on visual tracking. *Machine Design Res.* 30, 39–46. doi: 10.13952/j.cnki.jofmdr.2014.01.038

Zhang, K., Yang, X., Wang, J., Song, S., and Meng, M. Q. H. (2020). "Eye-in-hand uncalibrated visual servoing of concentric tube robot," in *2020 IEEE International Conference on Real-time Computing and Robotics (RCAR). IEEE.*

frontiers | Frontiers in Neuroscience

TYPE Original Research
PUBLISHED 19 July 2023
DOI 10.3389/fnins.2023.1201088

Check for updates

OPEN ACCESS

EDITED BY
Manning Wang,
Fudan University, China

REVIEWED BY
Xiang Xie,
Karlsruhe Institute of Technology (KIT), Germany
Haohao Hu,
Karlsruhe Institute of Technology (KIT), Germany

*CORRESPONDENCE
Wei Tian
✉ tian_wei@tongji.edu.cn

RECEIVED 06 April 2023
ACCEPTED 18 May 2023
PUBLISHED 19 July 2023

CITATION
Tian W, Gao Z and Tan D (2023) Single-view multi-human pose estimation by attentive cross-dimension matching.
Front. Neurosci. 17:1201088.
doi: 10.3389/fnins.2023.1201088

COPYRIGHT
© 2023 Tian, Gao and Tan. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

Single-view multi-human pose estimation by attentive cross-dimension matching

Wei Tian*, Zhong Gao and Dayi Tan

Institute of Intelligent Vehicles, School of Automotive Studies, Tongji University, Shanghai, China

Vision-based human pose estimation has been widely applied in tasks such as augmented reality, action recognition and human-machine interaction. Current approaches favor the keypoint detection-based paradigm, as it eases the learning by circumventing the highly non-linear problem of direct regressing keypoint coordinates. However, in such a paradigm, each keypoint is predicted based on its small surrounding region in a Gaussian-like heatmap, resulting in a huge waste of information from the rest regions and even limiting the model optimization. In this paper, we design a new k-block multi-person pose estimation architecture with a voting mechanism on the entire heatmap to simultaneously infer the key points and their uncertainties. To further improve the keypoint estimation, this architecture leverages the SMPL 3D human body model, and iteratively mines the information of human body structure to correct the pose estimation from a single image. By experiments on the 3DPW dataset, it improves the state-of-the-art performance by about 8 mm on MPJPE metric and 5 mm on PA-MPJPE metric. Furthermore, its capability to be employed in real-time provides potential applications for multi-person pose estimation to be conducted in complex scenarios.

KEYWORDS

attentive learning, multi-person pose estimation, single-image pose estimation, keypoint prediction, cross-dimension matching

1. Introduction

Vision-based human pose estimation has been favored in tasks of augmented reality, action recognition, human-machine interaction, etc. However, estimating human poses from a single image is a persistent challenge for the research community. In traditional algorithms, manually designed human body models are adopted to obtain local representations and global pose structures. However, the complexity of the human pose is far beyond the representation ability of hand-crafted features. In recent years, various human pose estimation technologies have been progressed driven by deep learning algorithms and large datasets.

The current mainstream 2D Human Pose Estimation (HPE) models can be divided into two categories: regression-based method and detection-based method. The former attempts to learn the direct mapping from an image to human keypoint (e.g, joint) coordinates (Toshev and Szegedy, 2014), which is yet a highly nonlinear problem and difficult to learn. The latter has dominated HPE for years due to high performance and intends to predict location heatmaps of parts or key points (Newell et al., 2016; Chu et al., 2017). However, the heatmaps are typically with low feature resolution and each keypoint only focuses on a small local region, resulting in a large waste of propagated gradients from the rest regions during model optimization.

Considering that current methods do not make full use of the information of human body structure, we propose a new k-block human pose estimation approach. Given a forecasted heatmap, this approach employs a voting mechanism over the entire heatmap to calculate keypoint coordinates and their corresponding uncertainties. Thus, compared to the traditional form, more feature information can be utilized through the increased number

of back-propagated gradients, and non-informative key points (e.g., by occlusion) will be given less attention during learning.

Due to the lack of depth information, the traditional 2D pose estimation often yields keypoint ambiguity. However, the human body structure based on 3D coordinates can better alleviate this problem. Leveraging the Skinned Multi-Person Linear (SMPL) 3D structure model of human body (Loper et al., 2015), we design an iterative coordinate matching strategy between 2D and 3D key points. The matching is optimized by using the Singular Value Decomposition (SVD) algorithm. The 2D coordinates can be corrected based on the predicted 3D key points and the optimized corresponding Euclidean transforms.

Compared with other human pose estimation schemes, we focus on mining the prior structure information of the human body itself and use the information of key points to reconstruct the pose model. With the new designed k-block module and corresponding optimization algorithm, the human body pose information can be iteratively corrected and the final output is based on the combination of the predicted human 2D/3D pose estimation.

2. Related works

2.1. 2D human pose estimation

As aforementioned, the direct regression learning of keypoint coordinates is difficult because it is a highly nonlinear problem, which lacks learning robustness. In comparison, the heatmap learning has a dense pixel information supervision, but the resolution of the heatmap is usually low due to downsampling operations such as pooling and strided convolution in the model, which limits the accuracy of the final estimated coordinates. A typical effort to this problem is the design of Hourglass module (Newell et al., 2016). It uses an hourglass-shaped model to gradually restore the features compressed in high-dimensional space to the original scale. Detail information such as faces and hands are captured by local features, which are restored and fused in the corresponding heatmaps with the same dimensions of features. Further efforts such as data stream adjustment (Bulat et al., 2020) and high-resolution (Sun et al., 2019) are also proposed to improve the network efficiency.

In addition to the keypoint detection, another problem that should be faced in the multi-person pose estimation is how to divide a large number of recognized pose key points into corresponding human bodies. The existing solutions are mainly divided into the top-down and the bottom-up paradigms. The former is achieved with a two-stage pipeline, which firstly employs off-the-shelf detectors on the input image to locate region of interests (RoI, denoted by bounding boxes) of human bodies, which are then individually processed by single-person pose estimators. But such approaches may be suboptimal since the pose estimation results are significantly affected by the detection accuracy, the focus of these methods is on the exploration of more efficient detectors (He et al., 2017; Ren et al., 2017). In contrast, the bottom-up methods firstly predict the key points of all persons in the image and then group them into different human bodies. The difficulty lies in how to correctly assemble the joint points. A typical approach is the OpenPose (Cao et al., 2017). It uses the Part Affinity Fields (PAF) module to predict the Part Confidence Maps and Part Affinity Fields on the entire image, which are further matched based on the learned local association fields. In other approaches, Newell et al. performed simultaneous detection and grouping with the Associative Embedding (Newell et al., 2017). They designed a new deep network structure to generate location heatmaps and associative embedding tags for each joint, distinguishing between different human bodies by tags. Although the processing speed of bottom-up methods is relatively fast and even real-time applicable (Cao et al., 2017; Nie et al., 2018), their performance is greatly affected by the complex backgrounds or occlusions. Therefore, motion information has been considered in recent works (Ohashi et al., 2020; Wang et al., 2020), which yet require video frames instead of a single image as inputs.

2.2. 3D human pose estimation

In mainstream models, the 3D human pose estimation is defined as the estimation of 3D human joint points. Related methods are mainly divided into two strategies: one-stage estimation and two-stage estimation. The one-stage methods directly estimate 3D poses from the input image in the presentations such as 3D heatmaps (Pavlakos et al., 2017), position maps (Sun Y. et al., 2021), and depth information (Liu et al., 2021). In contrast, the two-stage methods firstly estimate 2D human poses and then uplift them to the 3D space via pre-learned structural information (Zhou et al., 2016, 2017) or regression models (Martinez et al., 2017; Sun et al., 2017). Since two-stage methods are highly dependent on accurate 2D pose estimators, the combination of powerful backbone networks (Simonyan and Zisserman, 2015; Sun S. et al., 2021) became a trend in achieving impressive performance. However, as the human body structure information is implicitly modeled by neural networks, there is no guarantee that the output 3D skeleton in these methods is consistent with the real ones.

Aside from the 3D skeletons, the prior statistics about human body structure have also drawn increased research attention. A representative is the SMPL human body model (Loper et al., 2015), which is utilized to parameterize the output targets in model-based 3D pose estimation methods. Compared with model-free methods, these approaches directly predict controllable parameters, which facilitates an end-to-end 3D pose estimation without secondary adjustment, such as the SMPLify model proposed by Bogo et al. (2016). Since the mapping from an image to the shape space and the relative rotation of body parts is hard to learn, forms of intermediate representations and supervision are chosen to alleviate this problem, such as contours, semantic part segmentation, and 2D heatmaps. For example, Kanazawa et al. (2018) designed the adversarial priors and iterative error feedback (IEF) loops to reduce the difficulty of regression. Arnab et al. (2019) exploited temporal context information. Guler and Kokkinos (2019) used partial voting expressions and post-processing to improve regression networks. Kolotouros et al. (2019) leveraged an optimization paradigm to provide additional 3D supervision from unlabeled images. The hybrid inverse kinematics solution (HybrIK) (Li et al., 2021) leveraged the twist-and-swing decomposition to transform the 3D joints to shape estimation via both Kinematics and inverse Kinematics modeling

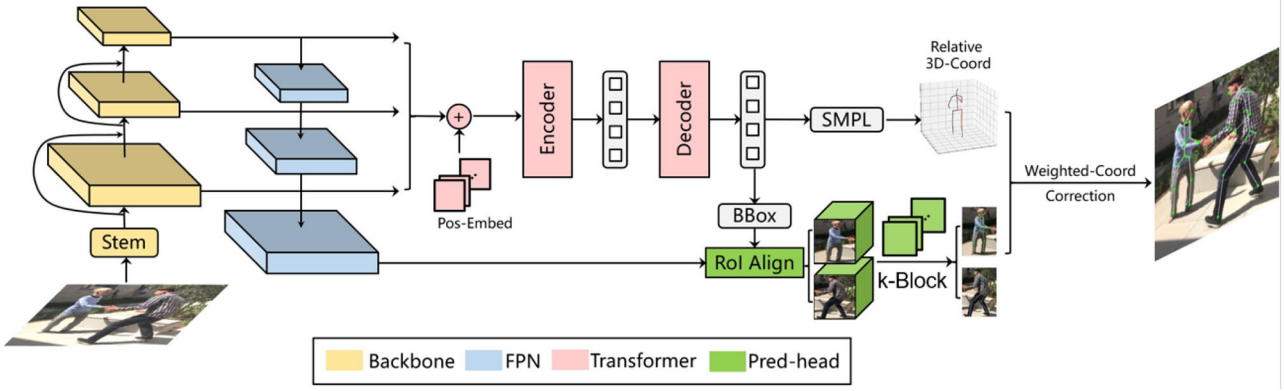


FIGURE 1
An overview of proposed multi-person pose estimation framework. Reproduced with permission from the official 3DPW benchmark, available at https://virtualhumans.mpi-inf.mpg.de/3DPW/.

and circumvented direct learning the abstract parameters of the general human body models.

In this paper, we propose a novel monocular multi-person pose estimation framework by exploiting the advantages of both 2D and 3D strategies. For backbone, this framework employs the Deformable DETR model (Zhu et al., 2021) (left part of Figure 1). It serves as a multi-person detector as well as a provider of reference regions and image features for the k-block module, which covers the entire heatmap information by a voting mechanism. Additionally, the k-block introduces uncertainties to 2D keypoint estimation, so that occluded joint points are given lower weights in the learning process, as they are less informative and can be inaccurately estimated, resulting in higher uncertainties. We also leverage an SMPL-based parametric model with a 2D–3D iterative optimization process. The core of our optimization algorithm is to estimate the optimal transform matrix and depths through iterative fitting between 2D and 3D relative coordinates. In this way, an accurate pose estimation can be obtained step by step without requring depth information.

3. Proposed method

3.1. 2D human characterization based on k-block structure

As previously introduced, the existing detection-based 2D pose estimation paradigm is designed to predict the location heatmap of key points, but is limited by the insufficient computational resolution. Moreover, most values on the heatmap are set to zero except for small local region surrounding the joints (Figure 4B), thus having no effect on the estimation of joint point coordinates. This fact forces a lot of back-propagated gradients to suppress predictions at non-joint positions, not only leading to a less efficient overall learning, but also making the model preferentially predict zero values.

To address these problems, we propose the k-block-based single-person pose estimation module, as illustrated in Figure 2. The input image is firstly processed by the backbone network to extract a feature tensor with a size of $w \times h$ pixels and l channels. With a further convolution in the channel dimension, a new tensor is predicted with k channels, which is equal to the number of to be predicted joint points. The tensor is further fed into the k-block module to generate the voting matrices. The joint points of the human body are finally predicted according to the corresponding voting results. The detailed calculation process is shown in Figure 3.

Here, we denote the i-th channel of input tensor as a heatmap (with a size of $w \times h$). The k-block module firstly accumulates heatmap values in both u- and v-directions. The obtained vectors $\boldsymbol{z}_u$ and $\boldsymbol{z}_v$ are then considered as the coordinate voting weights of the corresponding joint point in the u- and v-direction. By applying the *Softmax* operation on both weighting vectors, the normalized weight distributions $\boldsymbol{z}'_u$ and $\boldsymbol{z}'_v$ are obtained. Given a vector with a length of n, it generates an enumeration vector $\boldsymbol{e} = [1, 2, ...n]$, which corresponds to the sequence of row or column IDs. The element-wise product of the normalized weight distribution $\boldsymbol{z}^*$ and the enumeration sequence $\boldsymbol{e}$ is thus the distribution of corresponding voting values. The predicted joint coordinates can be calculated by summing up of the voting values. Additionally, we denote the joint coordinate uncertainty $c_{i,u/v}$ as the standard deviation of the voting values, i.e., the more concentrated the vote distribution is, the lower the uncertainty will be.

A comparison of Gaussian heatmap used in traditional methods and the k-block weights predicted in our approach is illustrated in Figure 4. In order to achieve a sufficient accuracy for the joint location, Gaussian heatmaps often require a larger resolution (e.g., 128 × 128 pixels). The non-joint areas are indicated in black in Figure 4B, in which a large number of gradients are used to suppress non-zero predictions. This part of the gradients has little effect on the prediction of joint points, resulting in a slow convergence of the model. Moreover, it still consumes a lot of computation in these areas in the forward inference stage, although their predictions are not considered. However, for heatmaps with larger Gaussian kernels, although more pixels are involved in the joint point estimation, the location accuracy can be reduced due to the reduction of the gap between predicted values.

In this paper, a new k-block structure is designed and the coordinate values of human joints are calculated from all heatmap

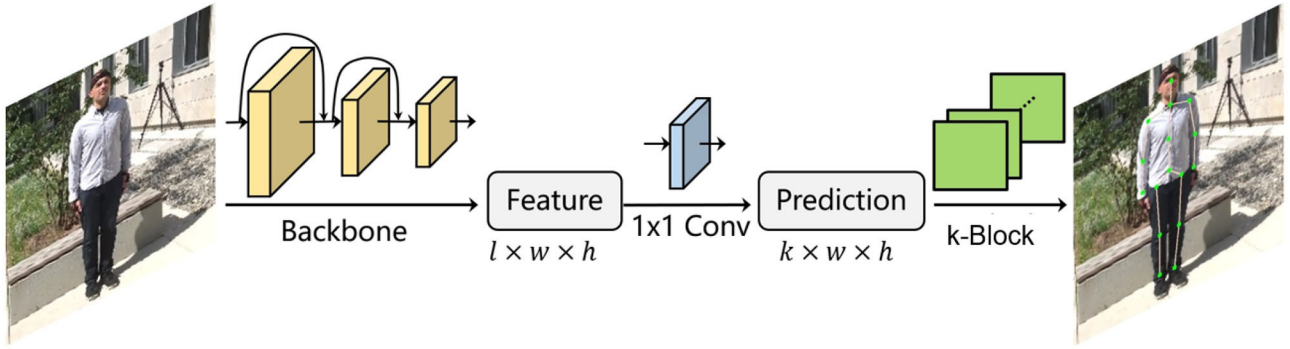


FIGURE 2
k-block-based 2D single-person pose estimation.

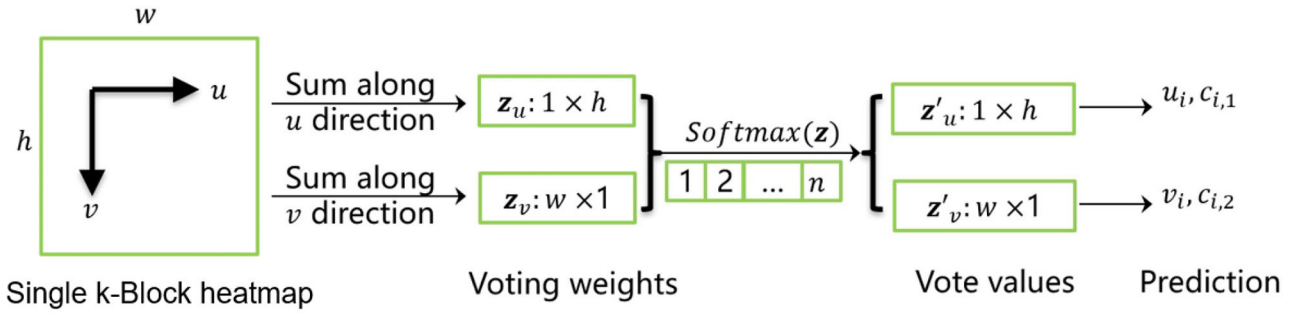


FIGURE 3
k-block prediction process.

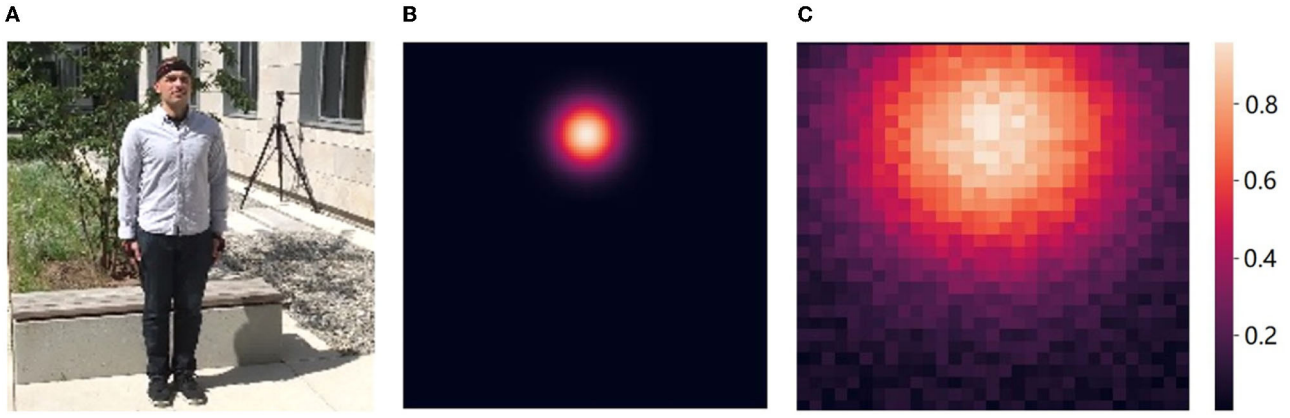


FIGURE 4
Comparison of Gaussian heatmap and k-block weights. **(A)** Input. **(B)** Gaussian (128 × 128). **(C)** k-Block (32 × 32).

elements at the same time, which greatly reduces the waste of gradients based on Gaussian heatmap prediction, so that it can use less computation (e.g., with a resolution of 32 × 32 pixels, which is yet still larger than the small local joint region in Gaussian heatmap) to obtain more gradient propagation to achieve similar accuracy.

In our proposed approach, each joint point estimation is regarded as a Gaussian distribution. Given an estimated coordinate x_i (i.e., u_i or v_i) and its ground truth $\hat{x}_i$, the estimation error $f_e(x_i)$ follows the Gaussian distribution, interpreted as

$$f_e(x_i) = \frac{1}{\sqrt{2\pi}\, c_i} \exp\left(-\frac{(x_i - \hat{x}_i)^2}{2c_i^2}\right) \quad (1)$$

with the standard deviation c_i. By applying the logarithm form of (1) and considering all joint points, the loss for k-block module is

expressed as

$$L_{KB} = \sum_i (\log(\sqrt{2\pi}c_i) + \frac{(x_i - \hat{x})^2}{2c_i^2}) + \omega_c \sum_i \frac{1}{2}c_i^2, \quad (2)$$

where ω_c represents the weight of the additional regularization term and is empirically set to 0.2. The set of inferred 2D joint points are denoted as $\mathcal{P}_{2D} = \{\boldsymbol{p}_{2D,1}, ..., \boldsymbol{p}_{2D,k}\}$.

3.2. 3D human characterization based on SMPL parameters

The SMPL (Loper et al., 2015) is a vertex-based three-dimensional model containing a fixed set of parameterized expressions based on the statistics of a large amount of real human body data. In this paper, the SMPL model is selected as the prior structure of the rigid human body, since it can accurately express different postures and movements. It should be noted that the original SMPL model also needs a set of root coordinates to further determine the 3D coordinates of the joint point. In this paper, we focus on the spatial relation between the 3D coordinates (e.g., relative to the body center), thus it requires no additional corresponding root points. Here, we implement an additional output head after the decoder of Derformable DETR (Zhu et al., 2021) to infer both the human body shape parameter $\boldsymbol{\beta}$ and the pose parameter $\boldsymbol{\theta}$ from an input image, as illustrated in the middle part of Figure 1.

The complete shape parameters consist of a total of 50 items with only the first 10 open-sourced. Statistics show that most of the parameter values are in the range from −1.5 to +1.5. This paper chooses the Smooth-L1 loss as the shape loss function and adjusts its second-order loss range to (−1.5, 1.5), interpreted as

$$L_{shape} = \sum_i \begin{cases} \frac{2}{9}(\beta_i - \hat{\beta}_i)^2, & |\beta_i - \hat{\beta}_i| \leq 1.5 \\ |\frac{2}{3}(\beta_i - \hat{\beta}_i)| - 0.5, & |\beta_i - \hat{\beta}_i| > 1.5 \end{cases}, \quad (3)$$

where β_i is the predicted i-th element of shape parameter $\boldsymbol{\beta}$ in the SMPL model and the symbol ˆ indicates the ground truth.

Additionally, we introduce the Quaternion notation to avoid the ambiguity problem induced by Euler angles used in the original SMPL. Let the normalized vector of the rotation axis be (x', y', z') and the rotation angle be $\alpha \in (-\pi, \pi]$. The pose parameter of SMPL can be expressed as

$$\boldsymbol{\theta} = (x' \sin\frac{\alpha}{2}, y' \sin\frac{\alpha}{2}, z' \sin\frac{\alpha}{2}, \cos\frac{\alpha}{2}). \quad (4)$$

Considering that the Quaternion representation is a normalized vector and its element value is in the range of $(-1, 1)$, the loss function of the pose parameter is selected as an L1 loss with an additive regularization term:

$$L_{pose} = \|\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}\|_1 + \omega_p \left|1 - \|\boldsymbol{\theta}\|_2^2\right|, \quad (5)$$

where $\boldsymbol{\theta}_i$ represents the i-th element of $\boldsymbol{\theta}$ and ω_p denotes the weight of the regularization term and empirically set to 1.

Based on the inferred shape parameter $\boldsymbol{\beta}$ and pose parameter $\boldsymbol{\theta}$, we can estimate the 3D joint point coordinates according to the SMPL model. The computation details can be referred to work (Loper et al., 2015). The point set is coordinate-normalized (by removing the mean and rescaling with the reciprocal of standard deviation) and denoted as $\mathcal{Q}_{3D} = \{\boldsymbol{q}_{3D,1}, ..., \boldsymbol{q}_{3D,k}\}$.

3.3. 2D-3D keypoint optimization

To correct the prediction results, especially for 2D joint points, we resort to the idea of 3D point matching. Generally, given two sets of matched 3D points $\mathcal{P} = \{\boldsymbol{p}_1, \boldsymbol{p}_2, ..., \boldsymbol{p}_k\}$ and $\mathcal{Q} = \{\boldsymbol{q}_1, \boldsymbol{q}_2, ..., \boldsymbol{q}_k\}$, the aim is to find a set of Euclidean transforms $\{\boldsymbol{R}, \boldsymbol{t}\}$ to minimize their alignment errors. The optimal transform $\{\boldsymbol{R}^*, \boldsymbol{t}^*\}$ can be obtained by solving the Least Squares problem as

$$(\boldsymbol{R}^*, \boldsymbol{t}^*) = \arg\min \sum_i^k \frac{1}{2}\|\boldsymbol{R}\boldsymbol{p}_i + \boldsymbol{t} - \boldsymbol{q}_i\|_2^2. \quad (6)$$

If the mean values of both sets $\mathcal{P}$ and $\mathcal{Q}$ are removed, which means their center are aligned at the origin, we obtain

$$\boldsymbol{t}^* = \boldsymbol{t} = 0. \quad (7)$$

Thus, Eq. (6) can be reformulated as

$$\boldsymbol{R}^* = \arg\min \sum_i^k \frac{1}{2}\|\boldsymbol{R}\boldsymbol{p}_i - \boldsymbol{q}_i\|_2^2. \quad (8)$$

The square term of above equation can be calculated as

$$\|\boldsymbol{R}\boldsymbol{p}_i - \boldsymbol{q}_i\|_2^2 = \boldsymbol{p}_i^\top \boldsymbol{p}_i - \boldsymbol{p}_i^\top \boldsymbol{R}^\top \boldsymbol{q}_i - \boldsymbol{q}_i^\top \boldsymbol{R}\boldsymbol{p}_i + \boldsymbol{q}_i^\top \boldsymbol{q}_i. \quad (9)$$

Noting that $(\boldsymbol{q}_i^\top \boldsymbol{R}\boldsymbol{p}_i)^\top = \boldsymbol{p}_i^\top \boldsymbol{R}^\top \boldsymbol{q}_i$, by discarding constant terms, Eq. (8) can be further simplified as

$$\begin{aligned} \boldsymbol{R}^* &= \arg\max \sum_i^k \boldsymbol{q}_i^\top \boldsymbol{R}\boldsymbol{p}_i = \arg\max \mathrm{tr}(\boldsymbol{Q}^\top \boldsymbol{R}\boldsymbol{P}) \\ &= \arg\max \mathrm{tr}(\boldsymbol{R}\boldsymbol{P}\boldsymbol{Q}^\top), \end{aligned} \quad (10)$$

where $\boldsymbol{P}$ and $\boldsymbol{Q}$ denote the matrix forms of point sets. Leveraging the SVD decomposition, it obtains $\boldsymbol{P}\boldsymbol{Q}^\top = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top$. Equation (10) can then be reformed as

$$\boldsymbol{R}^* = \arg\max \mathrm{tr}(\boldsymbol{R}\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top) = \arg\max \mathrm{tr}(\boldsymbol{\Sigma}\boldsymbol{V}^\top \boldsymbol{R}\boldsymbol{U}). \quad (11)$$

Since $\boldsymbol{R}$, $\boldsymbol{U}$, and $\boldsymbol{V}$ are all orthogonal matrices, the matrix $\boldsymbol{M} = \boldsymbol{V}^\top \boldsymbol{R}\boldsymbol{U}$ is also orthogonal. Thus, we obtain

$$1 = \boldsymbol{m}_i^\top \boldsymbol{m}_i = \sum_j \boldsymbol{m}_{i,j}^2 \rightarrow \boldsymbol{m}_{i,j}^2 \leq 1 \rightarrow |\boldsymbol{m}_{i,j}| \leq 1, \quad (12)$$

where $\boldsymbol{m}_i$ is the i-th row of $\boldsymbol{M}$ and $\boldsymbol{m}_{i,j}$ is the j-th element of $\boldsymbol{m}_i$. As $\boldsymbol{\Sigma} = \mathrm{diag}[\sigma_1, ..., \sigma_k]$ is a diagonal matrix, there is

$$\mathrm{tr}(\boldsymbol{\Sigma}\boldsymbol{M}) = \sum_i \sigma_i \boldsymbol{m}_{i,i} \leq \sum_i \sigma_i. \quad (13)$$

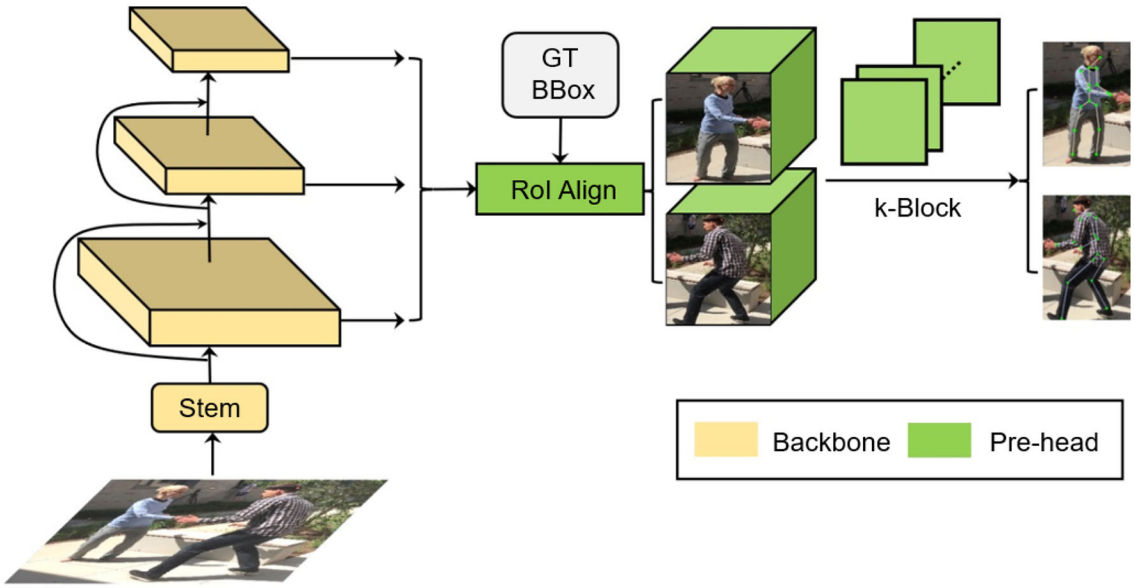


FIGURE 5
Multi-scale information for multi-person pose estimation.

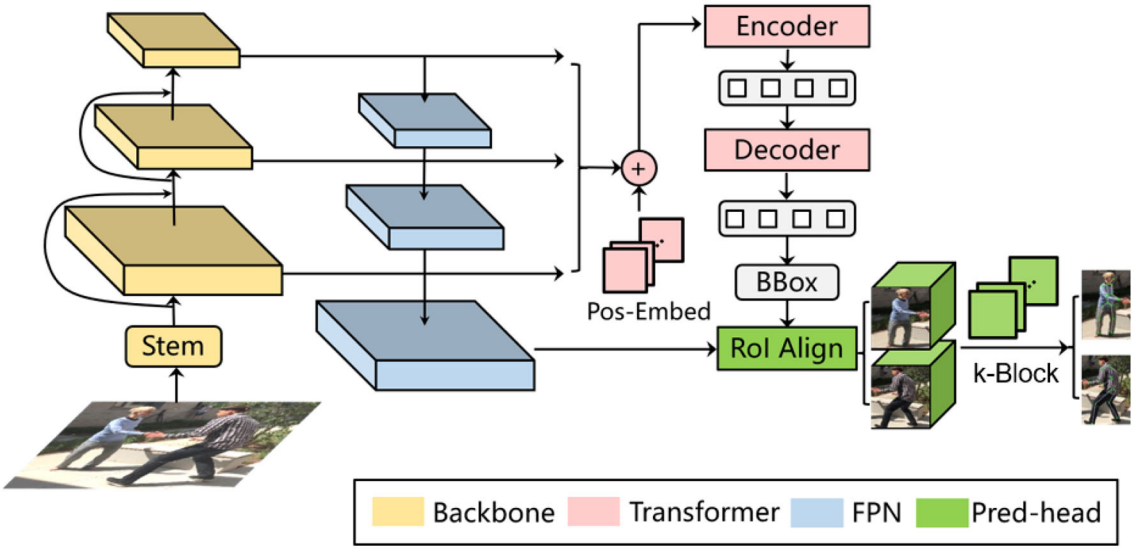


FIGURE 6
Deformable DETR-based multi-person pose estimation.

Obviously, only with $\boldsymbol{m}_{i,i} = 1$ can $\mathrm{tr}(\boldsymbol{\Sigma M})$ be maximized. Then, $\boldsymbol{M}$ becomes a unit matrix, which is

$$\boldsymbol{I} = \boldsymbol{M} = \boldsymbol{V}^{\top}\boldsymbol{R}^{*}\boldsymbol{U}. \tag{14}$$

By solving the above equation, we obtain the optimal rotation matrix $\boldsymbol{R}^{*} = \boldsymbol{V}\boldsymbol{U}^{\top}$.

If the depths of 2D joint points are known, with the above solution, we can correct the 2D joint points with their corresponding 3D coordinates estimated by the SMPL model, as illustrated in the right part of Figure 1. This is based on the fact that the SMPL is built on the statistics of a large set of real human bodies. Thus, its representation about the spatial relation between joint points should be more consistent with the real ones compared to the k-block-based estimation. Since the predicted 2D joint points are depthless, we consider their depths as additional to be optimized parameters in the entire optimization process. The main idea is to firstly lift the 2D key points into 3D space by assigning them with initial depth values, which are then gradually optimized by the 3D matching according to the solved rotation matrix. With iterations in this process, the accuracy of the estimated depth, the solved rotation matrix and the corresponding 2D coordinates of 3D

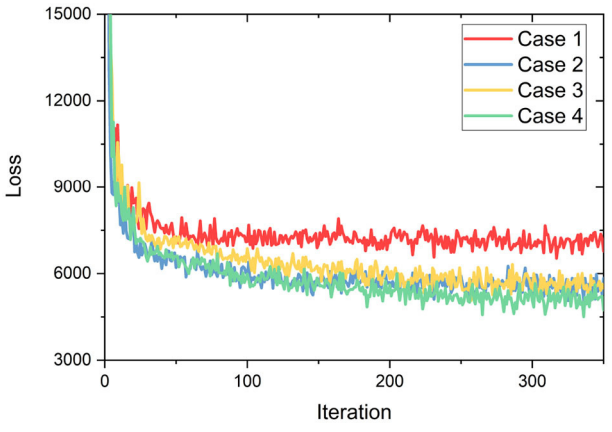


FIGURE 7
Trend of loss with different hyperparameters set (Case1: $\lambda_{shape} = 0.4$, $\lambda_{pose} = 0.4$, $\lambda_{SMPL} = 0.4$, $\lambda_{DET} = 0.25$, $\lambda_{KB} = 0.25$; Case2: $\lambda_{shape} = 0.25$, $\lambda_{pose} = 0.25$, $\lambda_{SMPL} = 0.25$, $\lambda_{DET} = 0.25$, $\lambda_{KB} = 0.25$; Case3: $\lambda_{shape} = 0.25$, $\lambda_{pose} = 0.25$, $\lambda_{SMPL} = 0.25$, $\lambda_{DET} = 0.25$, $\lambda_{KB} = 0.4$; Case4: $\lambda_{shape} = 0.2$, $\lambda_{pose} = 0.25$, $\lambda_{SMPL} = 0.15$, $\lambda_{DET} = 0.4$, $\lambda_{KB} = 0.3$).

TABLE 1 Exploration on performance of different multi-person pose estimation strategies with ↓ indicating that lower values are better.

Multi-scale	FPN	Correction	MPJPE (mm)↓
✓			58.5
	✓		57.9
	✓	✓	57.2

key points are progressively improved. Here the z-axis is defined as aligned with the depth direction, which is perpendicular to the image plane.

During the optimization, we also introduce the uncertainties of estimated 2D keypoint locations by the k-block module. Since the joint points in occluded or low-light areas are often estimated more inaccurately due to less information, their uncertainties will be high and their matching errors should be less weighted. Thus, Eq. (8) can rewritten as

$$\boldsymbol{R}^* = \arg\min \sum_i \frac{1}{2} w_i \|\boldsymbol{R}\boldsymbol{p}_i + \boldsymbol{q}_i\|^2. \tag{15}$$

The weight w_i is set to $1/c_i$, which is the reciprocal of the uncertainty c_i. We further define a diagonal weight matrix $\boldsymbol{W} = \mathrm{diag}[w_1, ..., w_k]$. Leveraging Eq. (10), the above equation can be reformed as

$$\boldsymbol{R}^* = \arg\max \mathrm{tr}(\boldsymbol{R}\boldsymbol{P}^\top \boldsymbol{W}\boldsymbol{Q}) = \arg\max \mathrm{tr}(\boldsymbol{\Sigma}\boldsymbol{V}^\top \boldsymbol{R}\boldsymbol{U}) \tag{16}$$

with the new SVD decomposition $\boldsymbol{P}^\top \boldsymbol{W}\boldsymbol{Q} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top$. This can be considered as a weighted 2D coordinate correction process based on SMPL parameters. Detailed steps of this process are listed in Algorithm 1, where the iteration number is empirically set to 3.

Input:
2D keypoint set $\mathcal{P}_{2D}$ with coordinate matrix $\boldsymbol{P}_{2D}$, diagonal uncertainty matrix $\boldsymbol{C}_{2D}$; Normalized 3D keypoint set $\mathcal{Q}_{3D}$ with coordinate matrix $\boldsymbol{Q}_{3D}$.
Output:
Corrected 2D keypoint coordinates $\boldsymbol{P}_{2D}$.
Calculate mean vector $\bar{\boldsymbol{p}}_{2D}$ and standard deviation vector $\boldsymbol{\sigma}_{2D}$ of $\boldsymbol{P}_{2D}$.
Lift $\boldsymbol{P}_{2D}$ to 3D space by assigning initial depth $z = 0$;
Normalize $\boldsymbol{P}_{2D}$ to the same center and scale of $\boldsymbol{Q}_{3D}$;
Calculate product: $\boldsymbol{S} = \boldsymbol{P}_{2D}^\top \boldsymbol{C}_{2D}^{-1} \boldsymbol{Q}_{3D}$;
SVD decomposition: $\boldsymbol{S} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top$, and obtain rotation matrix $\boldsymbol{R}^* = \boldsymbol{V}\boldsymbol{U}^\top$;
Ensure $\det(\boldsymbol{R}^*) = 1$, so that $\boldsymbol{R}^*$ is a rotation matrix;
Correct z-coordinates of $\boldsymbol{P}_{2D}$ based on $\boldsymbol{Q}_{3D}$ and $\boldsymbol{R}^*$;
Repeat step 4 to 7 to optimize $\boldsymbol{R}^*$ and z-coordinates of $\boldsymbol{P}_{2D}$;
Restore non-normalized $\boldsymbol{P}_{2D}$ based on $\boldsymbol{Q}_{3D}$, $\boldsymbol{R}^*$, $\bar{\boldsymbol{p}}_{2D}$ and $\boldsymbol{\sigma}_{2D}$;
return Corrected $\boldsymbol{P}_{2D}$.

Algorithm 1. Weighted 2D coordinate correction based on SMPL parameters.

3.4. Multi-person detection and pose estimation

Since pedestrians can appear in the image with different scales due to their sizes or distances in the 3D world space, the representation ability of features only extracted from a single layer of neural network becomes insufficient. Hence, the multi-person pose estimation scheme should be adapted to multi-scale image information. Considering the multi-layer convolution characteristics of the deep neural network itself, the deeper the layer is, the greater information amount a single neuron will capture, i.e., a deeper layer corresponds to a greater receptive field. Therefore, we can extract features from different layers of the backbone network to obtain the multi-scale information, as shown in Figure 5. Although such a multi-scale feature manipulation yields mere computational overhead, it has shortcomings like that the features from shallow layers are with relative low semantic information, limiting the prediction performance, while the deep layers are with relatively low resolution, leading to insufficient information amount within an RoI.

Referring to the Feature Pyramid Network model (FPN), we add an additional information recovery branch to the backbone (i.e., the ResNet). As shown on the left side of Figure 6, the bottom-up process indicates the feedforward feature calculation in the original model. As the layer deepens, the corresponding feature map gradually becomes downsampled. The top-down process is the gradual feature restoration toward the original image size. By fusing the information from different levels, the shallower layer obtains both higher resolution and richer semantic features. For inference, according to bounding box sizes, feature maps from the corresponding FPN layer are selected to be cropped and sent to

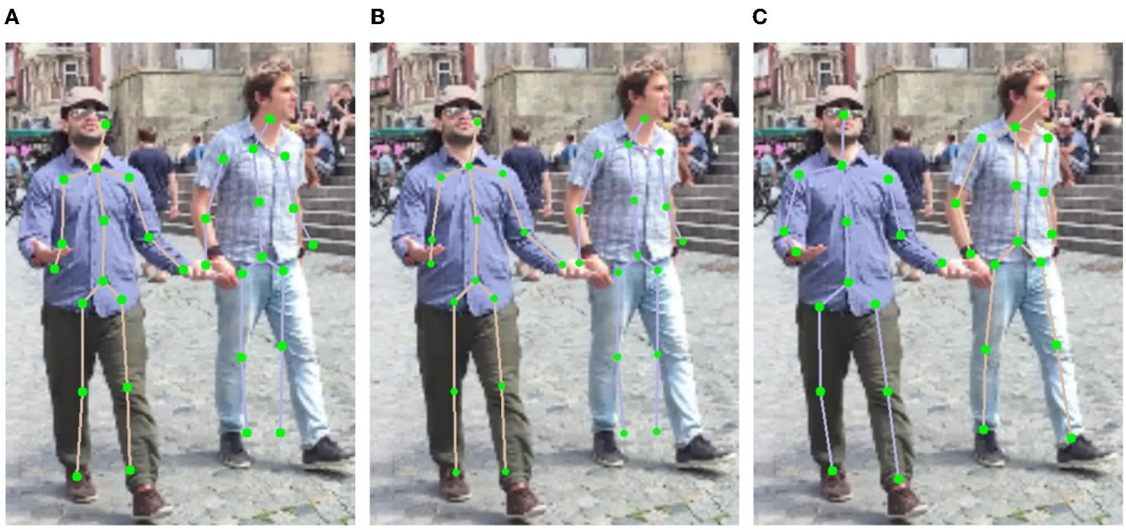


FIGURE 8
Examples of results under different multi-person pose estimation strategies. **(A)** Multi-scale. **(B)** FPN. **(C)** FPN and correction.

the k-block module to estimate the pose of each individual person. Additionally, we adopt the RoI Align (He et al., 2017) to avoid the dislocation of feature tensors caused by quantization operations.

To further improve the pedestrian detector performance, we employ the Deformable DETR framework (Zhu et al., 2021), as illustrated in Figure 6. In terms of single-frame pose estimation, the Deformable DETR model provides the candidate regions of detected persons and their corresponding image features for the k-block module. Thus, a simultaneous multi-person detection and pose estimation can be achieved. In addition to the detection bounding boxes, we also introduce another output head to the original Deformable DETR to regress the shape and pose parameters $\boldsymbol{\beta}$ and $\boldsymbol{\theta}$ of the SMPL model. The SMPL model is further applied in the iterative optimization process introduced in Section 3.3 to correct the predicted 2D key points, resulting in the final architecture proposed in this paper as shown in Figure 1. The total loss function for training the entire architecture is interpreted as

$$L_{total} = \lambda_{shape}L_{shape} + \lambda_{pose}L_{pose} + \lambda_{SMPL}L_{SMPL} + \lambda_{KB}L_{KB} + \lambda_{DET}L_{DET}, \quad (17)$$

where L_{DET} denotes the object detection loss defined in the Deformable DETR (Zhu et al., 2021), L_{SMPL} represents the squared errors of keypoint coordinates predicted by the SMPL, and the subscripted term λ indicates the corresponding weight of each loss.

4. Experiments and evaluations

4.1. Experimental setups

Here we choose two mainstream datasets, i.e., 3DPW (Von Marcard et al., 2018) and Human3.6M (Ionescu et al., 2011, 2014), for experiments. The 3DPW is a single-view multi-person 3D pose dataset containing 60 video sequences (24 for training, 24 for test, and 12 for validation) shot in outdoor environments such as forests, streets, playgrounds, etc. This dataset also includes a large number of 2D/3D pose annotations, 3D body scans, and SMPL parameters. The Human 3.6M is a multi-view single-person 3D pose dataset captured in an indoor space. It contains 3.6 million 3D human poses and corresponding videos (50 FPS) from 15 scenes, with keypoint annotations of both 2D/3D positions and angles. For evaluation, the video is downsampled at a ratio of 5/64 to eliminate redundancy.

TABLE 2 Comparison with state-of-the-art multi-person pose estimators.

Model	MPJPE (mm)↓	PA-MPJPE (mm)↓
HMR (Kanazawa et al., 2018)	130.0	81.3
SPIN (Kolotouros et al., 2019)	96.9	59.2
ROMP (Sun Y. et al., 2021)	76.7	47.3
HybrIK (Li et al., 2021)	74.1	45.0
DynBOA (Huang et al., 2020)	65.5	40.4
Ours	57.2	35.5

Since our proposed method adopts predicted 3D key points to assist the correction of predicted 2D keypoint coordinates, 3D annotations are employed in supervising the module for 3D keypoint prediction learning, which is also one of the main reasons in choosing above datasets for evaluation. In experiments, the proposed architecture is implemented by the PyTorch on a computer platform with a CPU of Intel i9@3.50 GHz, a GPU of NVIDIA RTX 3090 and a memory of 32 GB. During training, we adopt the Adam optimizer with a learning rate of 1e-3. The manual selection of hyperparameters, based on experience, has a substantial effect on the outcome of training. Consequently, various hyperparameters were designed and promptly evaluated with a

FIGURE 9
Example of prediction results by different multi-person pose estimators. **(A)** HMR. **(B)** SPIN. **(C)** HybrIK. **(D)** Ours.

TABLE 3 Runtime comparisons with different estimators.

Method	FPS	Backbone	Device
RepNet (Wandt and Rosenhahn, 2019)	10	Stacked hourglass network	NVIDIA TITAN X
VIBE (Kocabas et al., 2020)	10.9	ResNet-50	1070Ti GPU
ROMP (Sun Y. et al., 2021)	20.8	HRNet-32	1070Ti GPU
ROMP (Sun Y. et al., 2021)	30.9	ResNet-50	1070Ti GPU
Ours	9	DETR	NVIDIA RTX 3090

consistent number of iterations in order to choose the suitable configuration. It can be seen in Figure 7 that when the weights λ_{shape}, λ_{pose}, and λ_{SMPL} of 3D pose estimation are relatively small and the weight λ_{DET} of the human detection box is relatively large, there is a minimum loss trend (case 4). This may be due to the fact that the human detection box is the foundation of the top-down approach and its accuracy will directly influence the subsequent 2D/3D pose estimation. To this end, the weights for loss terms are empirically set as: $\lambda_{shape} = 0.2$, $\lambda_{pose} = 0.25$, $\lambda_{SMPL} = 0.15$, $\lambda_{DET} = 0.4$ and $\lambda_{KB} = 0.3$.

TABLE 4 Comparison with state-of-the-art single-person pose estimators.

	Model	MPJPE (mm)↓	Input frames	Training ratio
Video	VIBE (Kocabas et al., 2020)	65.6	16	50%
	Bundle (Arnab et al., 2019)	63.3	190	100%
	Att3DPose (Liu et al., 2020)	45.1	243	100%
Single Img.	RepNet (Wandt and Rosenhahn, 2019)	89.9	1	100%
	SMPLify (Bogo et al., 2016)	80.7	1	50%
	HMR (Kanazawa et al., 2018)	56.8	1	50%
	Ours	65.8	1	10%

For evaluation, we choose metrics of Mean Per Joint Position Error (MPJPE) and the Procrustes Alignment Mean Per Joint Position Error (PA-MPJPE), calculated as follows:

$$MPJPE = \frac{1}{k}\sum_{i}^{k} \|\boldsymbol{p}_i - \bar{\boldsymbol{p}}_i\|_2, \tag{18}$$

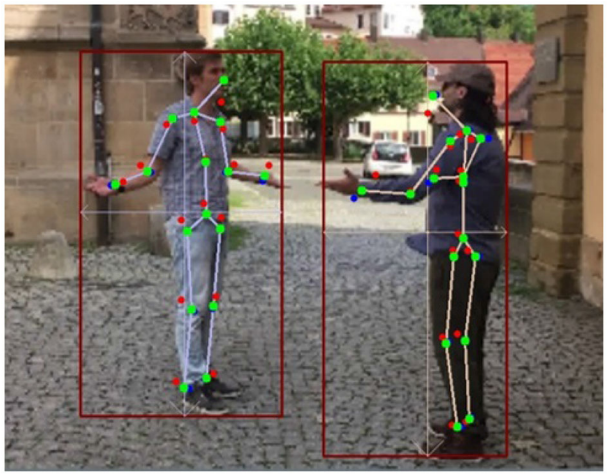


FIGURE 10
Example of multi-person pose estimation results (Red points: original output coordinates of the k-block module; Blue points: results by unweighted 3D correction algorithm; Green points: results by uncertainty-based weighted correction algorithm).

$$PA - MPJPE = \frac{1}{k}\sum_{i}^{k} \|\boldsymbol{p}_i' - \overline{\boldsymbol{p}}_i'\|_2, \tag{19}$$

where $\boldsymbol{p}_i$ refers to the predicted position of the i-th joint point while $\overline{\boldsymbol{p}}_i$ indicates the corresponding ground truth. The $\boldsymbol{p}_i'$ also denotes the position of the i-th joint point, yet with the predicted skeleton firstly aligned to its ground truth by rotation, translation and scaling. To facilitate a fair comparison with other mainstream pose estimators on above benchmarks, we calculate the corresponding 3D coordinates of predicted 2D key points by using the optimized depths and the given camera parameters. Thus, position errors can be measured in the 3D space.

4.2. Evaluation on multi-person pose estimation

In the first experiment, we explore the performance of different strategies for multi-person pose estimation introduced in Section 3.4, i.e., the direct multi-scale information fusing scheme, the FPN-based scheme and the SMPL correction-based scheme. For a fair comparison, all schemes adopt the Deformable DETR as base-detector and are evaluated on the 3DPW dataset. The results are reported w.r.t. the MPJPE metric in Table 1.

Obviously, introducing FPN module improves the mean joint position error by 0.6 mm according to the MPJPE metric, which proves that the top-down feature restoration process in the FPN is more efficient than the direct feature combination of different scales. By integrating the SMPL correction algorithm, the MPJPE is further reduced by 0.7 mm, demonstrating the benefit of 3D human body structure prior in the 2D keypoint prediction task. The processing speed of our entire architecture is about eight–nine FPS, which can be applied in real-time use cases. A qualitative comparison is also shown in Figure 8. As depicted, the direct multi-scale information fusion yields relative large estimation errors (Figure 8A). By only introducing the FPN module, the improvement is limited (Figure 8B). By further deploying the SMPL correction algorithm, the estimation errors at the end of the torso, on the arms and on the legs are compensated (Figure 8C).

We also compare the pose estimation results of our proposed architecture with those by other mainstream multi-person pose estimators including HMR (Kanazawa et al., 2018), SPIN (Kolotouros et al., 2019), ROMP (Sun Y. et al., 2021), HybrIK (Li et al., 2021) and DynBOA (Huang et al., 2020). Results of compared methods are listed in Table 2. It can be seen that the model based on k-block and SMPL parameter estimation proposed in this paper has reached a new level of state-of-the-art performance on the 3DPW dataset. It outperforms other approaches by an error reduction of about 5 mm w.r.t. the PA-MPJPE metric. In terms of the MPJPE metric, a larger accuracy gain is obtained, which is 8.3 mm. Examples of pose prediction results are shown in Figure 9. To be noted, since some of compared methods are not open-sourced, we only illustrate the prediction results of methods whose codes are available. As can be seen, in complex activities such as couple dancing, the key points at the end of body parts (e.g., arms and legs) can be easily misdetected in mainstream pose estimators while our method can still accurately locate these key points, proving its strong scene adaptability. Furthermore, we compare the inference time of the proposed method to the published results of other approaches, whose specific results are presented in Table 3. The use of DETR, with its large number of network parameters, inevitably sacrifices inference speed in order to achieve good results.

4.3. Evaluation on single-person pose estimation

Although our proposed architecture is designed aiming at the multi-person pose estimation task, it can still be applied for single-person pose estimation. Here, we evaluate our architecture on the Human3.6M dataset. As this dataset consists of millions of images and our computation resources are limited, we train our approach only on 10% of the training set. The evaluation results are reported in Table 4. As can be seen, the video-based pose estimators generally outperform single-view-based approaches. This can be attributed to additional motion information extracted from consecutive frames. However, the increased accuracy comes at the cost of processing a large number of frames, such as the top-ranked method Att3DPose, which requires 243 input frames. As to our method, its performance is comparable to the video-based VIBE (Kocabas et al., 2020) and Bundle (Arnab et al., 2019), and surpasses the singe-view-based RepNet (Wandt and Rosenhahn, 2019) and SMPLify (Bogo et al., 2016). Although the SMPLify is also an SMPL-based model, we achieve a position error reduction of about 15 mm by adopting the iterative optimization of 2D–3D key points, further demonstrating its advantages. However, our method is still with an error gap of 9 mm to the method HMR (Kanazawa et al., 2018), which is learned on half of the training data. As our model is only learned on 10% of the training data, there is still potential to improve its performance.

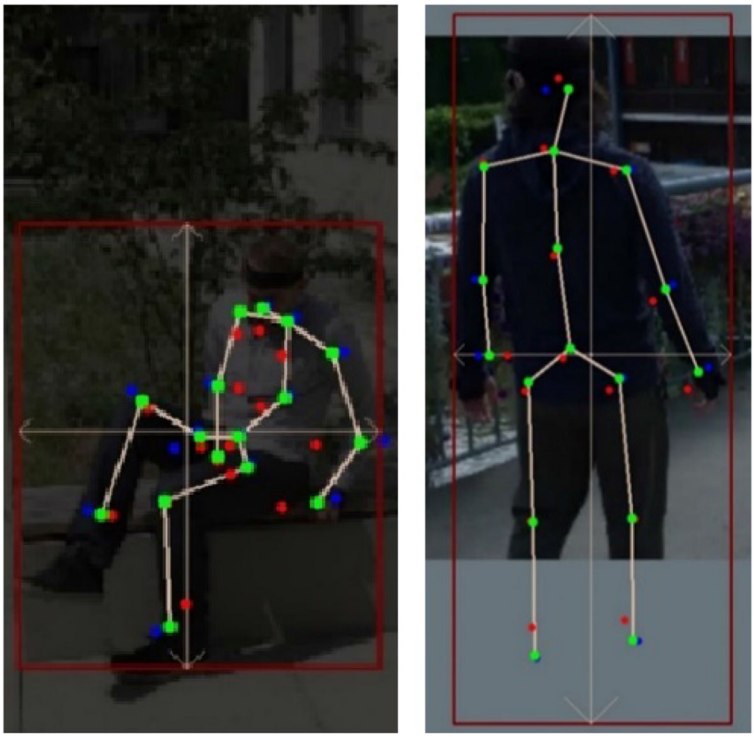


FIGURE 11
Prediction results under low-illumination **(left)** and with occlusion **(right)**.

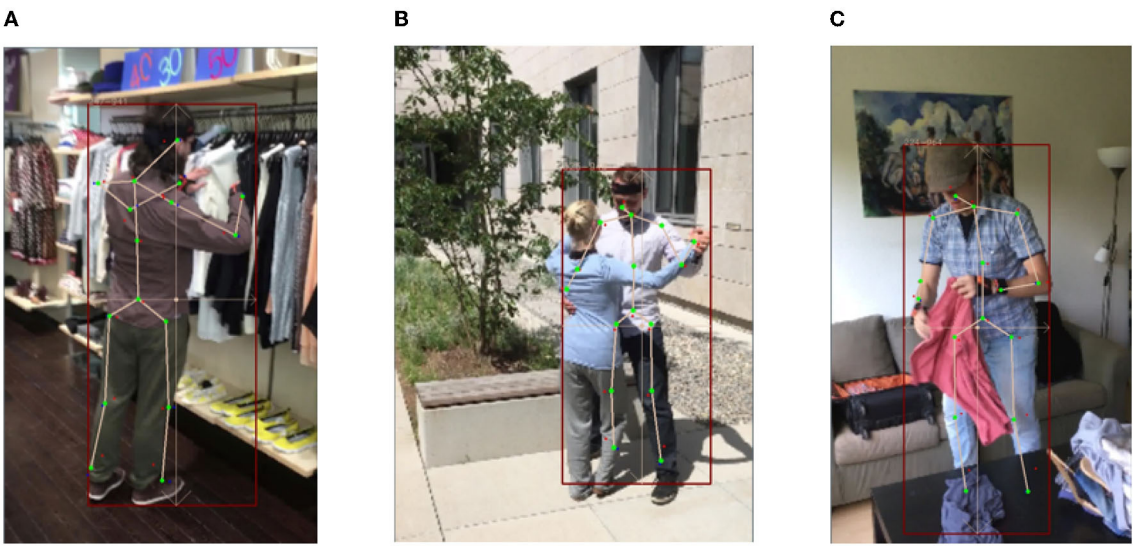


FIGURE 12
Example of negative results caused by occlusion with significant errors on legs. **(A)** Self-occlusion. **(B)** Occlusion by other people. **(C)** Occlusion by object.

4.4. Exploration on uncertainty weighting

The essence of k-block module is not only to predict the 2D key points but also to estimate their uncertainties based on the large heatmap information. In this experiment, we qualitatively explore its influence on the keypoint weighting in the optimization process. As illustrated in Figure 10, we depict the key points directly predicted by the k-block module in red, the ones corrected by SMPL yet without considering uncertainties in blue, and those corrected by the uncertainty-based weighted optimization in green. As can be seen, key points directly predicted by the k-block module are with obvious errors such at the head,

elbows, writs, and ankles. By applying the correction algorithm with the 3D SMPL model, the keypoint errors at the end of body parts are only reduced to some extent (e.g., the hand of the right person in Figure 10). By introducing uncertainty-based weighting in the optimization process, the keypoint errors are further reduced and the estimated skeleton looks more realistic. The uncertainty-based weighting is also beneficial to use cases under low-illumination or with occlusion, where individual key points become difficult to predict due to deteriorated image information. However, by considering uncertainties in the optimization, we can still obtain relative accurate keypoint prediction by fitting the informative body parts with the 3D shape and pose estimated by the SMPL model (Figure 11), validating the proposed approach.

5. Conclusion and discussion

In this paper, we present a new single-view multi-person pose estimation approach. It manifests improvements over existing approaches in two main aspects: Firstly, it proposes a k-block module to simultaneously calculate the 2D key point coordinates and their uncertainties, which improves the extraction of heatmap features and facilitates the attentive learning of more informative key points. Secondly, it employs a 3D shape and pose estimation based on the SMPL model and further proposes an uncertainty-weighted correction algorithm to iteratively align the estimated 3D coordinates with the predicted 2D key points. By experiments on the 3DPW benchmark, it surpassing state-of-the-art approaches by a gain of about 8 mm on MPJPE metric and 5 mm on PA-MPJPE metric. Additionally, it is real-time applicable and preforms robust against complex scenarios. Nonetheless, when the human body is subjected to self-occlusion or occlusion (see Figure 12), there is an ambiguity in depth estimation, which has a consequential impact on 3D pose estimation. Therefore, it is worth noting several important considerations for the future work: (1) incorporating an angle-axis representation or a regularization term to represent rotation; (2) improving the model accuracy for node coordinates through the utilization of multi-perspective images and designing a lighter, more compact model through network coding schemes.

Data availability statement

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

Author contributions

WT contributed to the conceptualization, methodology, supervision, writing, and review of the paper. ZG contributed to the methodology, experiments, and writing of the paper. DT contributed to the methodology, writing, and review of the paper. All authors contributed to the article and approved the submitted version.

Funding

The project was supported by the National Natural Science Foundation of China (No. 52002285), the Shanghai Science and Technology Commission (No. 21ZR1467400), the original research project of Tongji University (No. 22120220593), and the National Key R&D Program of China (No. 2021YFB2501104).

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

Arnab, A., Doersch, C., and Zisserman, A. (2019). "Exploiting temporal context for 3D human pose estimation in the wild," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (Long Beach, CA: IEEE), 3390–3399. doi: 10.1109/CVPR.2019.00351

Bogo, F., Kanazawa, A., Lassner, C., Gehler, P., Romero, J., Black, M. J., et al. (2016). "Keep it SMPL: automatic estimation of 3D human pose and shape from a single image," in *European Conference on Computer Vision (ECCV)*, Vol. 9909 (Cham: Springer), 561–578. doi: 10.1007/978-3-319-46454-1_34

Bulat, A., Kossaifi, J., Tzimiropoulos, G., and Pantic, M. (2020). "Toward fast and accurate human pose estimation via soft-gated skip connections," in *IEEE International Conference on Automatic Face and Gesture Recognition (FG)* (Buenos Aires: IEEE), 8–15. doi: 10.1109/FG47880.2020.00014

Cao, Z., Simon, T., Wei, S-. E., and Sheikh, Y. (2017). "Realtime multi-person 2D pose estimation using part affinity fields," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (Honolulu, HI: IEEE), 1302–1310. doi: 10.1109/CVPR.2017.143

Chu, X., Yang, W., Ouyang, W., Ma, C., Yuille, A. L., Wang, X., et al. (2017). "Multi-context attention for human pose estimation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (Honolulu, HI: IEEE), 5669–5678. doi: 10.1109/CVPR.2017.601

Guler, R. A., and Kokkinos, I. (2019). "HoloPose: holistic 3D human reconstruction in-the-wild," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (Long Beach, CA: IEEE), 10876–10886. doi: 10.1109/CVPR.2019.01114

He, K., Gkioxari, G., Dollar, P., and Girshick, R. (2017). "Mask R-CNN," in *IEEE International Conference on Computer Vision (ICCV)* (Venice: IEEE), 2980–2988. doi: 10.1109/ICCV.2017.322

Huang, C., Jiang, S., Li, Y., Zhang, Z., Traish, J., Deng, C., et al. (2020). "End-to-end dynamic matching network for multi-view multi-person 3D pose estimation,"

in *European Conference on Computer Vision (ECCV)*, Vol. 12373 (Cham: Springer), 477–493. doi: 10.1007/978-3-030-58604-1_29

Ionescu, C., Li, F., and Sminchisescu, C. (2011). "Latent structured models for human pose estimation," in *International Conference on Computer Vision* (Barcelona: IEEE). doi: 10.1109/ICCV.2011.6126500

Ionescu, C., Papava, D., Olaru, V., and Sminchisescu, C. (2014). 'Human3.6m: large scale datasets and predictive methods for 3D human sensing in natural environments. *IEEE Trans. Pattern Anal. Mach. Intell.* 36, 1325–1339. doi: 10.1109/TPAMI.2013.248

Kanazawa, A., Black, M. J., Jacobs, D. W., and Malik, J. (2018). "End-to-end recovery of human shape and pose," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (Salt Lake City, UT: IEEE), 7122–7131. doi: 10.1109/CVPR.2018.00744

Kocabas, M., Athanasiou, N., and Black, M. J. (2020). "VIBE: video inference for human body pose and shape estimation," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (Seattle, WA: IEEE), 5252–5262. doi: 10.1109/CVPR42600.2020.00530

Kolotouros, N., Pavlakos, G., Black, M., and Daniilidis, K. (2019). "Learning to reconstruct 3D human pose and shape via model-fitting in the loop," in *IEEE/CVF International Conference on Computer Vision (ICCV)* (Seoul: IEEE), 2252–2261. doi: 10.1109/ICCV.2019.00234

Li, J., Xu, C., Chen, Z., Bian, S., Yang, L., Lu, C., et al. (2021). "HybrIK: a hybrid analytical-neural inverse kinematics solution for 3D human pose and shape estimation," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (Nashville, TN: IEEE), 3383–3393. doi: 10.1109/CVPR46437.2021.00339

Liu, J., Tsujinaga, S., Chai, S., Sun, H., Tateyama, T., Iwamoto, Y., et al. (2021). Single image depth map estimation for improving posture recognition. *IEEE Sens. J.* 21, 26997–27004. doi: 10.1109/JSEN.2021.3122128

Liu, R., Shen, J., Wang, H., Chen, C. Cheung, S.-c., and Asari, V. (2020). "Attention mechanism exploits temporal contexts: real-time 3D human pose reconstruction," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (Seattle, WA: IEEE), 5063–5072. doi: 10.1109/CVPR42600.2020.00511

Loper, M., Mahmood, N., Romero, J., Pons-Moll, G., and Black, M. J. (2015). SMPL: a skinned multi-person linear model. *ACM Trans. Graph.* 34, 1–16. doi: 10.1145/2816795.2818013

Martinez, J., Hossain, R., Romero, J., and Little, J. J. (2017). "A simple yet effective baseline for 3D human pose estimation," in *IEEE International Conference on Computer Vision (ICCV)* (Venice: IEEE), 2659–2668. doi: 10.1109/ICCV.2017.288

Newell, A., Huang, Z., and Deng, J. (2017). "Associative embedding: end-to-end learning for joint detection and grouping," in *Advances in Neural Information Processing Systems*, Vol. 30, eds I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan (Long Beach, CA), 2277–2287.

Newell, A., Yang, K., and Deng, J. (2016). "Stacked hourglass networks for human pose estimation," in *European Conference on Computer Vision (ECCV)*, Vol. 9912 (Cham: Springer), 483–499. doi: 10.1007/978-3-319-46484-8_29

Nie, X., Feng, J., Xing, J., and Yan, S. (2018). "Pose partition networks for multi-person pose estimation," in *European Conference on Computer Vision (ECCV)*, Vol. 11209(Cham: Springer), 705–720. doi: 10.1007/978-3-030-01228-1_42

Ohashi, T., Ikegami, Y., and Nakamura, Y. (2020). Synergetic reconstruction from 2d pose and 3D motion for wide-space multi-person video motion capture in the wild. *Image Vis. Comput.* 104, 104028. doi: 10.1016/j.imavis.2020.104028

Pavlakos, G., Zhou, X., Derpanis, K. G., and Daniilidis, K. (2017). "Coarse-to-fine volumetric prediction for single-image 3D human pose," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (Honolulu, HI: IEEE), 1263–1272. doi: 10.1109/CVPR.2017.139

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

Simonyan, K., and Zisserman, A. (2015). "Very deep convolutional networks for large-scale image recognition," in *International Conference on Learning Representations (ICLR)* (San Diego, CA: ACM).

Sun, K., Xiao, B., Liu, D., and Wang, J. (2019). "Deep high-resolution representation learning for human pose estimation," in *IEEE International Conference on Computer Vision and Pattern Recognition CVPR* (Long Beach, CA: IEEE), 3349–3364. doi: 10.1109/CVPR.2019.00584

Sun, S., Akhtar, N., Song, H., Mian, A., and Shah, M. (2021). Deep affinity network for multiple object tracking. *IEEE Trans. Pattern Anal. Mach. Intell.* 43, 104–119. doi: 10.1109/TPAMI.2019.2929520

Sun, X., Shang, J., Liang, S., and Wei, Y. (2017). "Compositional human pose regression," in *IEEE International Conference on Computer Vision (ICCV)* (Venice: IEEE), 2621–2630. doi: 10.1109/ICCV.2017.284

Sun, Y., Bao, Q., Liu, W., Fu, Y., Black, M. J., Mei, T., et al. (2021). "Monocular, one-stage, regression of multiple 3D people," *in Proc. International Conference on Computer Vision (ICCV)* (Montreal, QC: IEEE), 11179–11188. doi: 10.1109/ICCV48922.2021.01099

Toshev, A., and Szegedy, C. (2014). "DeepPose: human pose estimation via deep neural networks," in *IEEE Conference on Computer Vision and Pattern Recognition* (Columbus, OH: IEEE), 1653–1660. doi: 10.1109/CVPR.2014.214

Von Marcard, T., Henschel, R., Black, M., Rosenhahn, B., and Pons-Moll, G. (2018). "Recovering accurate 3D human pose in the wild using IMUs and a moving camera," in *European Conference on Computer Vision (ECCV)* (Cham: Springer), 601–617. doi: 10.1007/978-3-030-01249-6_37

Wandt, B., and Rosenhahn, B. (2019). "RepNet: weakly supervised training of an adversarial reprojection network for 3D human pose estimation," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (Long Beach, CA: IEEE), 7774–7783. doi: 10.1109/CVPR.2019.00797

Wang, M., Tighe, J., and Modolo, D. (2020). "Combining detection and tracking for human pose estimation in videos," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (Seattle, WA: IEEE), 11085–11093. doi: 10.1109/CVPR42600.2020.01110

Zhou, X., Zhu, M., Leonardos, S., and Daniilidis, K. (2017). Sparse representation for 3D shape estimation: a convex relaxation approach. *IEEE Trans. Pattern Anal. Mach. Intell.* 39, 1648–1661. doi: 10.1109/TPAMI.2016.2605097

Zhou, X., Zhu, M., Leonardos, S., Derpanis, K. G., and Daniilidis, K. (2016). "Sparseness meets deepness: 3d human pose estimation from monocular video," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (Las Vegas, NV: IEEE), 4966–4975. doi: 10.1109/CVPR.2016.537

Zhu, X., Su, W., Lu, L., Li, B., Wang, X., Dai, J., et al. (2021). Deformable DETR: deformable transformers for end-to-end object detection. *arXiv* 2010.04159. doi: 10.48550/arXiv.2010.04159

frontiers | Frontiers in Neuroscience

TYPE Original Research
PUBLISHED 30 August 2023
DOI 10.3389/fnins.2023.1254088

Check for updates

OPEN ACCESS

EDITED BY
Alois C. Knoll,
Technical University of Munich, Germany

REVIEWED BY
Yisen Huang,
The Chinese University of Hong Kong, China
Keli Shen,
National Institute of Advanced Industrial Science and Technology (AIST), Japan

*CORRESPONDENCE
Elena M. Gutierrez-Farewik
✉ lanie@kth.se

RECEIVED 06 July 2023
ACCEPTED 11 August 2023
PUBLISHED 30 August 2023

CITATION
Zhang L, Zhang X, Zhu X, Wang R and Gutierrez-Farewik EM (2023) Neuromusculoskeletal model-informed machine learning-based control of a knee exoskeleton with uncertainties quantification. *Front. Neurosci.* 17:1254088. doi: 10.3389/fnins.2023.1254088

COPYRIGHT
© 2023 Zhang, Zhang, Zhu, Wang and Gutierrez-Farewik. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

Neuromusculoskeletal model-informed machine learning-based control of a knee exoskeleton with uncertainties quantification

Longbin Zhang[1], Xiaochen Zhang[1], Xueyu Zhu[2], Ruoli Wang[1] and Elena M. Gutierrez-Farewik[1]*

[1]KTH MoveAbility Lab, Department of Engineering Mechanics, KTH Royal Institute of Technology, Stockholm, Sweden, [2]Department of Mathematics, University of Iowa, Iowa City, IA, United States

Introduction: Research interest in exoskeleton assistance strategies that incorporate the user's torque capacity is growing rapidly. However, the predicted torque capacity from users often includes uncertainty from various sources, which can have a significant impact on the safety of the exoskeleton-user interface.

Methods: To address this challenge, this paper proposes an adaptive control framework for a knee exoskeleton that uses muscle electromyography (EMG) signals and joint kinematics. The framework predicted the user's knee flexion/extension torque with confidence bounds to quantify the uncertainty based on a neuromusculoskeletal (NMS) solver-informed Bayesian Neural Network (NMS-BNN). The predicted torque, with a specified confidence level, controlled the assistive torque provided by the exoskeleton through a TCP/IP stream. The performance of the NMS-BNN model was also compared to that of the Gaussian process (NMS-GP) model.

Results: Our findings showed that both the NMS-BNN and NMS-GP models accurately predicted knee joint torque with low error, surpassing traditional NMS models. High uncertainties were observed at the beginning of each movement, and at terminal stance and terminal swing in self-selected speed walking in both NMS-BNN and NMS-GP models. The knee exoskeleton provided the desired assistive torque with a low error, although lower torque was observed during terminal stance of fast walking compared to self-selected walking speed.

Discussion: The framework developed in this study was able to predict knee flexion/extension torque with quantifiable uncertainty and to provide adaptive assistive torque to the user. This holds significant potential for the development of exoskeletons that provide assistance as needed, with a focus on the safety of the exoskeleton-user interface.

KEYWORDS

machine learning, data-driven biomechanical models, inverse dynamics, neuromusculoskeletal modeling, uncertainty quantification

1. Introduction

Exoskeletons have enormous potential to enhance movement and to contribute to neuromuscular rehabilitation in persons with motor disorders such as stroke, cerebral palsy, and spinal cord injury (Sartori et al., 2016; Li et al., 2019, 2021; Liu et al., 2019; Zhang et al., 2019). Exoskeleton-assisted rehabilitation training involves the use of control algorithms aimed at improving muscle strength, neuroplasticity, and movement enhancement in

users (Fujii et al., 2017). These control strategies can be classified into three types: passive control, triggered passive control, and assist-as-needed control (Marchal-Crespo and Reinkensmeyer, 2009; Meng et al., 2015; Proietti et al., 2016). Passive control refers to a technique in which the exoskeleton is in charge and guides the user to follow predefined trajectories or assistive forces/torques that have been extracted from healthy populations. The user is passive in the movement and does not actively control the exoskeleton. This type of control is often used in the initial stages of therapy to re-acquaint a limb to movement. Triggered passive control is a variant of passive control, where the user initiates the exoskeleton's assistance. Once activated, the user is again passive in the movement as the exoskeleton moves along pre-determined trajectories. This technique is often used to incorporate the brain-machine interface into the control process, providing assistance to individuals with irreversible impairments, such as tetraplegia (Proietti et al., 2016). Assist-as-needed control, also known as "user-in-charge" or "active control," empowers the user to perform daily tasks with the aid of an exoskeleton. The exoskeleton provides assistance based on the user's ability and intention to generate torque, with the aim of promoting neuroplasticity and user autonomy. This active control technique is typically applied in persons with residual motor function (Chen et al., 2016; Durandau et al., 2017; Li et al., 2018; Yao et al., 2018). The primary focus of this paper is on active control techniques that seek to supplement the user's insufficient muscle contributions with assistance from an exoskeleton. Providing torque assistance based on the user's movement intention requires precise and robust decoding of motor function, which can be achieved through recording of underlying neuromuscular activities, such as brain and nerve signals and muscle electromyography (EMG) signals. EMG signals, which capture the electrical excitation of muscles, are a commonly used method for predicting joint torques, as they are easy to obtain and offer crucial insights into human motion (Sartori et al., 2018; Huang et al., 2019; Mounis et al., 2019).

Joint torque prediction is crucial in the control of exoskeleton-assisted rehabilitation and has frequently been achieved through two methods: physics-based neuromusculoskeletal (NMS) modeling and artificial neural networks (ANNs) (Pizzolato et al., 2015, 2019; Zhang et al., 2020, 2021). To improve prediction accuracy, ANNs have been integrated into NMS models in recent research. In our recent study (Zhang et al., 2022), an NMS solver-informed ANN model was developed to estimate ankle joint torque by combining features from an NMS model with a standard ANN, based on measured joint angles and muscle EMG signals during gait and isokinetic motions. This hybrid model was overall more accurate than the NMS or standard ANN models alone, but still showed poor prediction performance in one subject during gait, possibly due to incorporating less informative or misleading input features from the NMS model. This highlights the necessity of quantifying the uncertainty of joint torque predictions for safe and efficient human-exoskeleton interaction; accurate estimation of joint torque is crucial for determining the appropriate level of assistance from an exoskeleton.

A Bayesian Neural Network (BNN) is a well-established type of ANN for making predictions with uncertainties and has great potential in safe and efficient exoskeleton control (Cursi et al., 2021; Wei et al., 2021; Zhong et al., 2021). Unlike conventional ANNs (Cao et al., 2022b; Hu et al., 2022), BNNs incorporate probability distributions to represent prediction uncertainty and provide a probability distribution indicating the likelihood of different outcomes (Cursi et al., 2021). This characteristic makes BNNs useful for decision-making in various fields, including biomechanics, meteorology, and robotics. For instance, Zhong et al. (2021) developed a BNN-based framework for predicting the environmental context of lower limb prostheses. The quantified prediction uncertainty could lead to context recognition strategies that enhance reliable decision-making, efficient sensor fusion, and improved design of intelligent systems for various applications. Another popular technique for making predictions with uncertainties is the Gaussian Process (GP) (Chen et al., 2013; Yun et al., 2014; Maritz et al., 2018; Guo et al., 2019; Cao et al., 2022a). GP models the output as a Gaussian distribution with mean and covariance parameters, wherein the uncertainty is expressed by the covariance. Liang et al. (2021), for example, developed a GP model to estimate knee joint angles and uncertainties from EMG signals during walking and running movements. As both BNNs and GPs can estimate prediction uncertainty, it is of interest to compare the two methods in the context of safe and efficient human-exoskeleton interaction.

The objective of this study was thus to develop an NMS uncertainty-informed adaptive control framework for a knee exoskeleton. The framework aimed to provide accurate predictions of the user's knee flexion/extension (F/E) physiological torque, while also quantifying the level of estimation uncertainty. To achieve this, an NMS solver was employed to inform the machine learning models, which would subsequently adjust the assistance level based on the level of uncertainty. Another aim was to compare the predictions with uncertainties from the NMS solver-informed BNN (NMS-BNN) model with those from the NMS solver-informed GP (NMS-GP) model.

2. Methods

We developed an adaptive control framework for a knee exoskeleton based on an NMS-BNN model (Figure 1). The NMS-BNN takes two types of inputs: (1) experimental measurements—muscle signals and joint angles, and (2) informative physical features extracted from the underlying NMS solver, such as individual muscle force and joint torque. The NMS-BNN outputs knee joint torque with uncertainty quantification in the form of confidence bounds. The predicted torque with a specified confidence level is then used to control the assistive torque provided by the knee exoskeleton through a TCP/IP data stream. The study results consist of two key components: (1) an analysis of the NMS-BNN model's prediction accuracy and uncertainty, compared to the traditional NMS model and to the NMS-GP model; (2) an evaluation of the tracking performance of the assistive torque provided by the knee exoskeleton.

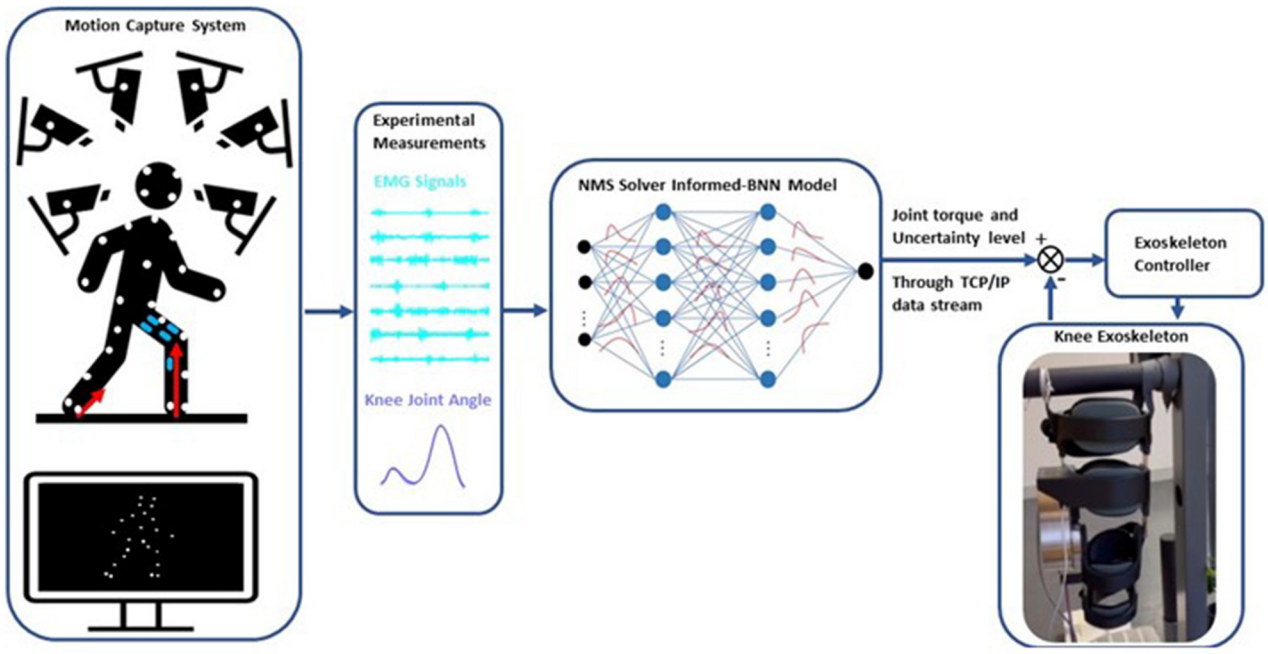


FIGURE 1
Schematic of the adaptive control framework for a knee exoskeleton based on an NMS solver-informed BNN (NMS-BNN) model. The inputs to the NMS-BNN include observed muscle signals and joint angles, as well as physical features derived from the NMS solver such as individual muscle force and joint torque. The NMS-BNN outputs knee joint torque with uncertainty quantification in the form of confidence bounds. The predicted torque with a specified confidence level is then used to control the assistive torque provided by the knee exoskeleton through a TCP/IP data stream.

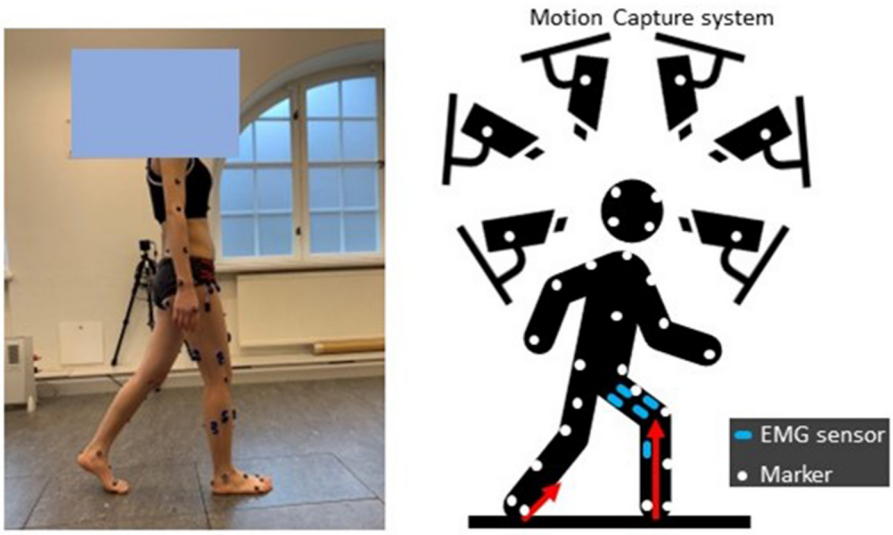


FIGURE 2
Experimental setup: subjects equipped with EMG sensors and markers, performed movements in an instrumented motion lab.

2.1. Data collection and processing

Eight able-bodied subjects (sex: 4F/4M; height: 168.1 ± 9.4 cm; weight: 65.2 ± 17.8 kg; age: 29 ± 4 years) were recruited. The Swedish Ethical Review Authority (Dnr. 2020-02311) approved this study, and all subjects provided informed written consent documents. All participants were asked to do five movement types (Figure 2), specifically slow walking, normal walking, fast walking, sit-to-stand, and stand-to-sit. During the experiments, each movement was repeated at least ten times. The sequence of movements was randomized.

Surface EMG signals (aktos nano, myon, Schwarzenberg, Switzerland) from vastus medialis (VM), vastus lateralis (VL), rectus femoris (RF), semitendinosus (ST), biceps femoris (BF), gastrocnemius medialis (GM), and gastrocnemius lateralis (GL) of each participant's one randomly-selected leg were measured at 1,000 Hz. EMGs were post-processed by bandpass filtering (30–300 Hz), rectifying, low pass filtering (6 Hz), and normalizing to the maximum EMG value among all movement trials (Sartori et al., 2016; Pizzolato et al., 2017; Hoang et al., 2018).

Marker trajectories were recorded at 100 Hz using a 3D motion capture system (V16, Vicon, Oxford, UK), with marker placement based on the CGM2.3 model (Leboeuf et al., 2019). Ground reaction forces (GRFs) were measured at 100 Hz with three force plates (AMTI, MA, USA). Kinematics were calculated via inverse kinematics by solving a weighted least square optimization problem to minimize the discrepancy between virtual x_i and measured x_i^{exp} marker trajectories (Lu and O'connor, 1999), Equation (1).

$$\min_{q}\left(\sum_{i}^{N}\theta_i\|x_i^{exp}-x_i\|^2\right), \tag{1}$$

where q represents the generalized coordinates of the model and θ_i is the weight of ith marker. Kinetics were computed via inverse dynamics by solving for joint torques in the dynamic equations of motion (Pandy, 2001) (Equation 2),

$$M(\mathbf{q})\ddot{q}+C(q,\dot{q})+\mathbf{G}(\mathbf{q})+R(\mathbf{q})\mathbf{F^{mt}}+\mathbf{F_e}=0 \tag{2}$$

where $\mathbf{q},\dot{q},\ddot{q}$ are the vector of generalized position, velocity and acceleration, respectively; $M(\mathbf{q})$ is mass matrix and $M(\mathbf{q})\ddot{q}$ is a vector of inertial forces and torques; $C(q,\dot{q})$ is the vector of centripetal and Coriolis forces and torques; $\mathbf{G}(\mathbf{q})$ is the vector of gravitational forces and torques; $R(\mathbf{q})$ is the matrix of muscle moment arms; $\mathbf{F^{mt}}$ is a vector of musculotendon forces and $R(\mathbf{q})\mathbf{F^{mt}}$ is the vector of musculotendon torques; $\mathbf{F_e}$ is the vector of external force and torques (i.e., GRFs in this paper). A low-pass fourth-order zero-lag Butterworth filter (6 Hz) was used to filter joint kinematics and kinetics (Winter et al., 1974; Mantoan et al., 2015; Derrick et al., 2020).

2.2. EMG-driven neuromusculoskeletal model

The EMG-driven NMS model used in this study was the open-source CEINMS model (Pizzolato et al., 2015) (Figure 3). This model includes four components: musculotendon kinematics, muscle activation dynamics, muscle contraction dynamics, and joint dynamics relationships (Sartori et al., 2011). The musculotendon kinematics component calculates moment arms and musculotendon lengths, while the muscle activation dynamics component computes muscle activation based on the available EMG information. The relationship between EMG excitation, $e(t)$, and neural activation, $u(t)$, is expressed in Equation (3) (Lloyd and Besier, 2003):

$$u(t)=\alpha\cdot e(t-\tau)-\beta_1\cdot u(t-1)-\beta_2\cdot u(t-2) \tag{3}$$

where α is the muscle gain parameter, β_1 and β_2 are the recursive parameters [$\beta_1=C_1+C_2$, $\beta_2=C_1\cdot C_2$, with $|C_1|<1$, $|C_2|<1$, and $\alpha-\beta_1-\beta_2=1$ for a stable solution (Lloyd and Besier, 2003; Buchanan et al., 2004; Pizzolato et al., 2015)], and τ is the electromechanical delay. Muscle activation, $a(t)$, is described by Equation (4):

$$a(t)=\frac{e^{Au(t)}-1}{e^{A}-1} \tag{4}$$

where A is the shape factor (Buchanan et al., 2004; Hoang et al., 2018).

The muscle contraction dynamics component calculates the muscle force with a Hill-type muscle model, represented by Equation (5):

$$F=F_0^m[F_{al}(l)\cdot F_v(v)\cdot a+F_{pl}(l)+d_p\cdot v]cos(\theta) \tag{5}$$

where F_0^m is the muscle's maximum isometric force, $F_{al}(l)$ describes the relationship between active muscle force and fiber length l, $F_{pl}(l)$ describes the relationship between passive muscle force and fiber length, $F_v(v)$ describes the relationship between muscle force and fiber contraction velocity v, θ is the fiber pennation angle, and d_p is the muscle damping parameter.

Finally, the joint dynamics component computes joint torque by multiplying muscle forces and moment arms.

The parameters were calibrated as outlined by Pizzolato et al. (2015), with optimal fiber length and tendon slack length adjusted within ±15% of initial values, coefficients C_1 and C_2 limited to values between −1 and 1, and parameter A bounded between $(-3,0)$. The maximum isometric force was determined using a strength coefficient with a range of 0.5 to 2.5. The optimization process focused on minimizing the error between predicted and actual joint torques (computed via inverse dynamics) during the calibration procedure. This optimization task was achieved by employing a simulated annealing algorithm, which iteratives to refine the parameter values. The algorithm was executed until the average change in the objective function's value reached a tolerance level of 10^{-5}.

2.3. NMS-BNN model

The NMS-BNN models consist of an input neural layer, 3 hidden layers, and an output neural layer. The inputs, $\mathbf{x}=[x_1,x_2,\ldots,x_m]^T$ where $m=21$, were augmented with two types of features (Figure 4): (1) Muscle EMG signals and joint kinematics (knee F/E angle) from a 3D motion capture system, and (2) physical features such as muscle forces and NMS torque from an underlying NMS solver, to increase the model's accuracy by providing more information about the system being modeled. Each hidden layer has 40 neurons. The estimated knee torque with uncertainties bound was determined in the output layer.

In BNNs, weights are treated as probability distributions rather than as single point estimates, as in standard neural networks (Figure 4). These distributions are used to reflect the uncertainty

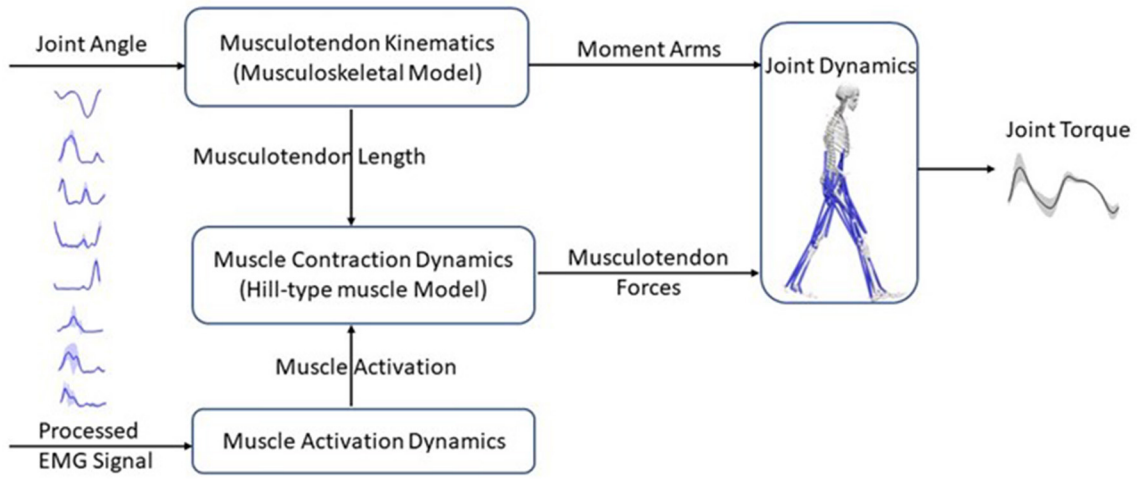


FIGURE 3
Schematic structure of an EMG-driven neuromusculoskeletal model with four components: the musculotendon kinematics component calculates musculotendon lengths and moment arms; the muscle activation dynamics component uses the EMG information to compute muscle activation; the muscle contraction dynamics component, predicts musculotendon force using musculotendon length and muscle activation based on a Hill-type muscle model; and finally, the joint dynamics component computes joint torques with musculotendon forces and moment arms as inputs.

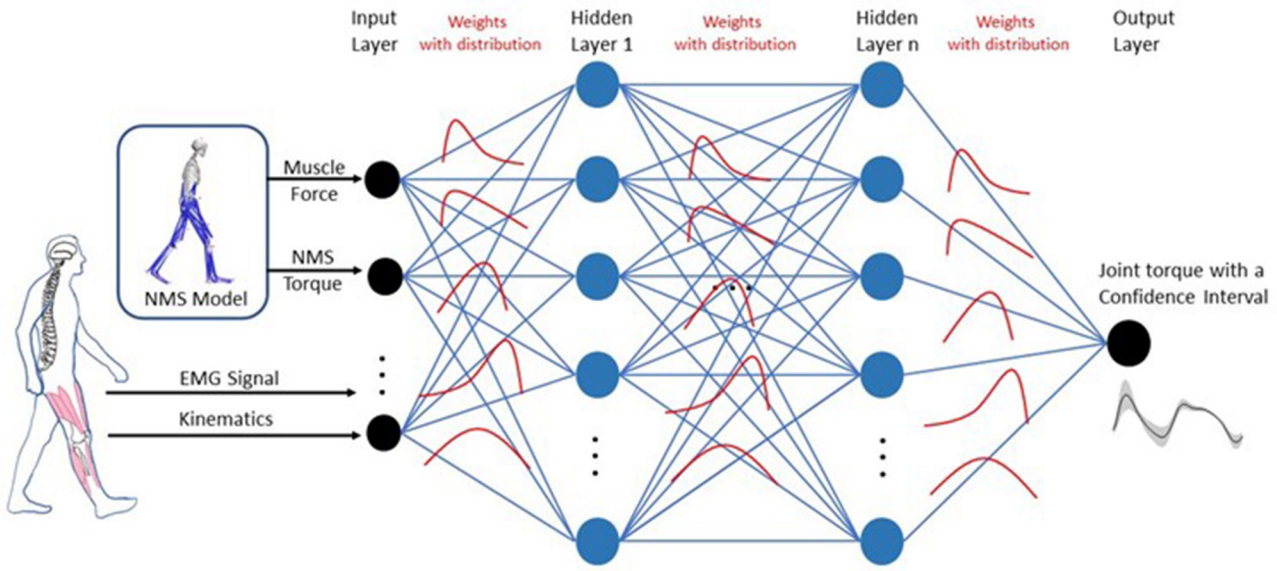


FIGURE 4
Architecture for the NMS-BNN model. The NMS-BNN models consist of an input neural layer, 3 hidden layers, and an output neural layer. The inputs, $\mathbf{x} = [x_1, x_2, \ldots, x_m]^T$ where $m = 21$, were augmented with two types of features: (1) Muscle EMG signals and joint kinematics from a 3D motion capture system, and (2) Physical features such as muscle forces and NMS torque from an underlying NMS solver, to increase the model's accuracy by providing more information about the system being modeled. Each hidden layer has 40 neurons. The estimated knee torque with uncertainties bound was determined in the output layer. Weights are treated as probability distributions rather than as single-point estimates as in standard neural networks. These distributions are used to reflect the uncertainty in weights and predictions.

in weights and predictions. The posterior probability of weights, $P(W|X)$, is computed using Bayes theorem as follows (Equation 6):

$$P(W|X) = \frac{P(X|W)P(W)}{P(X)} \tag{6}$$

where X is the data, $P(X|W)$ represents the likelihood of the data given weights W, and $P(W)$ is the prior probability of the weights. The denominator, $P(X)$, represents the probability of the data, which is obtained by integrating the likelihood of the data given weights and the prior probability of weights over all possible values of weights, represented by Equation (7):

$$P(X) = \int P(X|W)P(W)dW \tag{7}$$

The BNN package TensorBNN, developed by Kronheim et al. (2022), was used in this study. The hyper-parameters of the BNN models were determined using a "coarse-to-fine" random search method (Bergstra and Bengio, 2012). During training, the mean square error was used as the loss function and a batch size of 32 was applied. Three hidden layers were included. Each hidden layer has 40 neurons with a tanh activation function. A Gaussian likelihood with a standard deviation of 0.1 was employed. Prior to sampling, the model was pre-trained using the AMSGrad optimizer with learning rates of 0.01, 0.001, and 0.0001, with a patience of 10. To obtain a point cloud of the posterior density of neural network parameters, Hamiltonian Monte Carlo (HMC) sampling was used to compute the likelihood function. HMC is a Markov chain Monte Carlo method that leverages a fictitious potential energy function derived from the posterior density of the neural network parameters. Numerical approximation was conducted using the leapfrog method, with the number of leapfrog steps and step size determining the distance traveled to the next proposed point. The number of steps for the HMC hyper-parameter sampler remained constant, while the step size was adapted using the Dual-Averaging algorithm based on the acceptance rate of the sample during 80% of the burn-in period. It is worth noting that selecting suitable values for the number of steps and step size can be challenging, and TensorBNN incorporates the parameter adapter algorithm to automatically optimize these parameters (Wang et al., 2013; Kronheim et al., 2022).

2.4. NMS-GP model

The NMS-GP model was developed using the same input data as the NMS-BNN model, which comprised experimentally obtained muscle signals and joint kinematics, as well as physical features such as muscle forces and NMS torque extracted from the underlying NMS solver. The NMS-GP model $f(\mathbf{x})$ was specified by a mean function $\mu(\mathbf{x})$ and a covariance function $k(\mathbf{x}, \mathbf{x}^{'})$, as expressed in equation (8) (Rasmussen, 2004).

$$f(\mathbf{x}) \sim GP(\mu(\mathbf{x}), k(\mathbf{x}, \mathbf{x}^{'})) \tag{8}$$

where the mean function $\mu(\mathbf{x})$ provides an estimate of the expected value of the model at a given input, while the covariance function, also referred to as the kernel function, quantifies the similarity between two inputs. The Gaussian process model offers various kernel functions to capture the underlying structure of data. Among them, the radial basis function kernel (RBF) is widely used due to its smoothness and infinite differentiability, as shown in Equation (9),

$$k(\mathbf{x}, \mathbf{x}') = \exp(-\frac{|\mathbf{x} - \mathbf{x}'|^2}{2l^2}) \tag{9}$$

where l controls the length-scale of the kernel, and $|\mathbf{x} - \mathbf{x}'|$ is the Euclidean distance between inputs $\mathbf{x}$ and $\mathbf{x}'$.

Another popular kernel function is the Matern kernel, which is a generalization of the RBF kernel and is defined as Equation (10),

$$k(\mathbf{x}, \mathbf{x}') = \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu}}{l} |\mathbf{x} - \mathbf{x}'| \right)^{\nu} K_{\nu} \left(\frac{\sqrt{2\nu}}{l} |\mathbf{x} - \mathbf{x}'| \right) \tag{10}$$

where ν determines the smoothness of the kernel, K_{ν} is the modified Bessel function, $\Gamma(\nu)$ is the Gamma function, and l controls the scale of the kernel.

For modeling noise in data, the White Noise Kernel, as shown in Equation (11), is commonly used,

$$k(\mathbf{x}, \mathbf{x}') = \sigma^2 \delta(\mathbf{x} - \mathbf{x}') \tag{11}$$

where σ^2 is the noise variance parameter that determines the amplitude of the noise, and $\delta(\mathbf{x} - \mathbf{x}')$ is the Dirac delta function. This function equals one when $\mathbf{x} = \mathbf{x}'$ and zero otherwise, ensuring that the kernel function is non-zero only at the diagonal of the input space.

The Linear kernel is another widely used kernel that models a linear relationship between the input and output variables. Specifically, it can be formulated as depicted in Equation (12):

$$k(\mathbf{x}, \mathbf{x}') = \sigma^2 \mathbf{x}^T \mathbf{x}' \tag{12}$$

where σ^2 represents the variance parameter.

In Gaussian process modeling, the combination of different kernel functions can improve the performance of the model. In this regard, we selected a combination of Matern, White Noise, and Linear kernels after conducting extensive testing to obtain the most accurate predictions. The Matern kernel was used to capture non-linear patterns, while the White Noise kernel accounted for measurement errors and uncertainty, and the Linear kernel modeled linear relationships between the input and output variables. Hyperparameters such as length scale, signal variance, noise variance, and others were optimized during the training process to enhance the model's performance. In the Matern kernel, we set ν to 3/2, and the length scale was bounded between the range of (0.01, 200), with variance confined within the range of $(10^{-3}, 10^5)$. Similarly, the White Noise kernel had a noise variance bounded between (0.03, 100), while the Linear kernel had a variance range of $(10^{-3}, 10^5)$.

2.5. Knee exoskeleton

The knee exoskeleton hardware consists of a drive unit (Gen.1, Maxon, Switzerland), a 3D-printed thigh-shank frame, and thigh and shank straps (Orliman 94260, Spain). The drive unit features a brushless DC motor (EC90 flat), a MILE encoder with 4,096 counts per turn, a three-stage planetary gearbox with an 18-bit SSI absolute encoder, and an EPOS4 position controller. The drive unit is capable of providing a continuous torque of 54 Nm and a maximum torque of 120 Nm on a 20% duty cycle. The system can operate on a DC power supply ranging from 10 to 50vV and its actuation speed can reach up to 22 rpm.

2.6. Evaluation protocol

2.6.1. Joint torque prediction

The prediction accuracy of the knee joint torque for NMS, NMS-GP, and NMS-BNN models was investigated. The uncertainty of the predicted torque by NMS-GP and NMS-BNN models

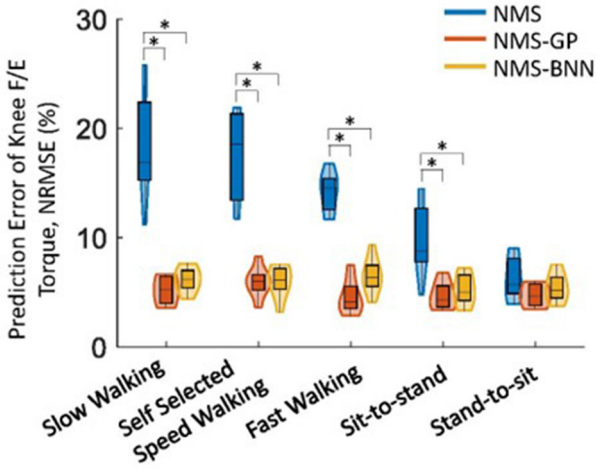


FIGURE 5
The distributions of NRMSE between estimated and measured/actual knee joint torques across subjects in NMS, NMS-GP, and NMS-BNN models during five daily activities. The violin plots depict the probability distributions of NRMSE using kernel density plots, and the box plots represent the minimum, lower quartile, median, upper quartile, and maximum values of NRMSE. A significant difference between any two models is indicated by an asterisk (∗), based on paired *t*-test (for normally distributed data) or Wilcoxon signed-rank test (for non-normally distributed data) with Bonferroni correction.

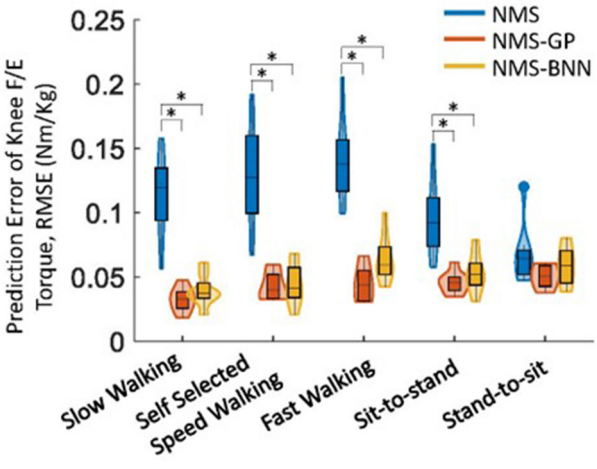


FIGURE 6
The distributions of RMSE between estimated and measured/actual knee joint torques across subjects in NMS, NMS-GP, and NMS-BNN models during five daily activities. The violin plots depict the probability distributions of NRMSE using kernel density plots, and the box plots represent the minimum, lower quartile, median, upper quartile, and maximum values of NRMSE. A significant difference between any two models is indicated by an asterisk (∗), based on paired *t*-test (for normally distributed data) or Wilcoxon signed-rank test (for non-normally distributed data) with Bonferroni correction.

was also analyzed. The prediction accuracy and uncertainty quantification was compared in five cases: $Gait_{slow}$, $Gait_{self}$, $Gait_{fast}$, *SitToStand*, and *StandToSit*, which were trained using data from each movement separately. NMS-GP and NMS-BNN models were trained using 80% of the data and evaluated on the remaining data, while NMS models were calibrated using three trials of each movement and tested on the same data as NMS-GP and NMS-BNN models. The input data from each trial consisted of approximately 100 time-series data points and 21 dimensions.

Two prediction error metrics were evaluated: the Normalized Root Mean Square Error (NRMSE, E_{NRMS}) and the Root Mean Square Error (RMSE, E_{RMS}). A low prediction error indicated a high prediction accuracy. NRMSE was calculated by dividing the RMSE (between the predicted and actual torque) by the range of joint torque observed during the corresponding motion:

$$E_{RMS} = \sqrt{\frac{1}{N}\sum_{n=1}^{N}(y_{p,n} - y_n)^2} \tag{13}$$

$$E_{NRMS} = \frac{E_{RMS}}{(y_{max} - y_{min})} \times 100\% \tag{14}$$

where y_n and $y_{p,n}$ are the measured/actual and predicted torque respectively; and y_{min} and y_{max} are the minimum and maximum measured torque in corresponding movements. The RMSE and NRMSE were calculated for each subject and the average was obtained across eight subjects. The results section presents the average values of RMSE and NRMSE.

The normality of the data distribution was evaluated using Shapiro-Wilk tests ($p < 0.05$ significance level). To determine the differences among the NRMSEs and RMSEs estimated by the three approaches, pairwise comparisons were performed using either paired *t*-tests for normally distributed data or Wilcoxon signed-rank tests for non-normally distributed data, both with Bonferroni correction applied and significance level of $p < 0.05$.

The uncertainty of the predicted torque by NMS-GP and NMS-BNN models was quantified by using a 95% confidence level (CL), which means that there is a 95% probability that the true value of the function being modeled falls within the predicted interval. A high uncertainty value indicates low confidence in the predicted value.

2.6.2. Exoskeleton assistive torque tracking performance

We also evaluate the tracking performance of the knee exoskeleton's assistive torque provided by the adaptive control framework during five daily activities by using the two metrics: NRMSE and RMSE (between desired and actual torque provided by the knee exoskeleton). The assistance level A_L of the knee torque provided by the adaptive control framework is adapted/determined by the uncertainties U quantified by the NMS-BNN model, as described by the following equation:

$$A_L = \begin{cases} 0.8 & \text{if } U < 0.05 \\ 0.6 & \text{if } 0.05 \le U < 0.1 \\ 0.4 & \text{if } 0.1 \le U < 0.15 \\ 0.3 & \text{if } 0.15 \le U < 0.2 \\ 0.1 & \text{if } U \ge 0.2 \end{cases} \tag{15}$$

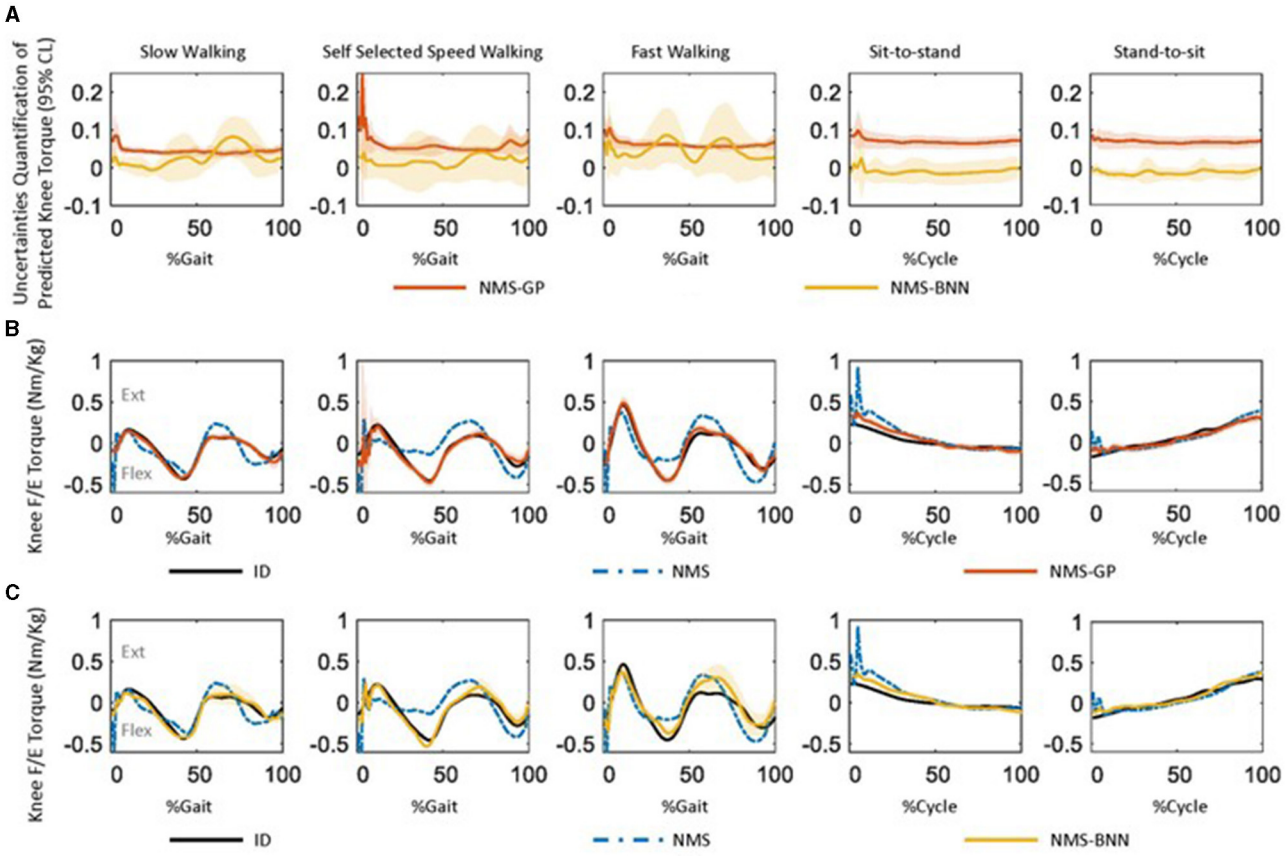


FIGURE 7
(A) The uncertainty quantification of predicted knee joint torque by the NMS-GP and NMS-BNN models across subjects during five daily activities, as the mean ± 1 standard deviation of all subjects. The uncertainty was quantified using a 95% confidence level, meaning that there is a 95% probability that the true value falls within the predicted interval. A high uncertainty value indicates low confidence in the prediction. **(B)** One example of measured knee flex/extension torques (Nm/kg) by inverse dynamics (ID) and predicted values by both NMS and NMS-GP models during five daily activities. The standard deviation in the NMS-GP models highlights the uncertainties from the expected mean value. **(C)** One example of predicted knee flex/extension torques by NMS-BNN models was presented and compared with the same example from ID and NMS models in **(B)**. The standard deviation in the NMS-BNN models also highlights the uncertainties from the expected mean value.

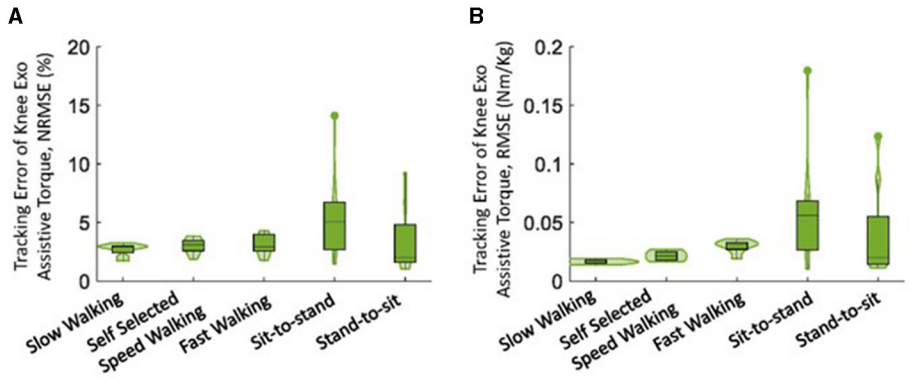


FIGURE 8
The tracking error of the knee exoskeleton's assistive torque provided by the adaptive control framework during five daily activities, presented as **(A)** NRMSE and **(B)** RMSE. The violin plots illustrate the probability distributions of prediction error through kernel density plots, and the box plots depict the minimum, first quartile, median, third quartile, and maximum values of the prediction error.

3. Results

3.1. Joint torque prediction

Overall, both NMS-BNN and NMS-GP models accurately predicted knee joint torque with relatively low error (RMSE: NMS-GP $\leq$ 0.05 Nm/kg, NMS-BNN $\leq$ 0.07 Nm/kg; NRMSE: NMS-GP $\leq$ 5.9%, NMS-BNN $\leq$ 6.8%). The errors were considerably lower than those of NMS models (RMSE:$\leq$ 0.14 Nm/kg, NRMSE: $\leq$ 18.3%, Figures 5, 6).

The NRMSE prediction error for the NMS-GP and NMS-BNN models was significantly lower than that of the NMS models in all cases, except the *StandToSit* case ($Gait_{slow}$: $p < 0.01$ and $p < 0.01$; $Gait_{self}$: $p < 0.01$ and $p < 0.01$; $Gait_{fast}$: $p < 0.01$ and $p < 0.01$, *SitToStand*: $p < 0.01$ and $p < 0.01$, *StandToSit*: $p = 0.08$ and $p = 1.45$; Figure 5). Similar findings were also observed in the RMSE.

Among the NRMSE predicted by NMS models in five cases, the NRMSE in the *StandToSit* case was the lowest ($\leq$ 7.2%). No significant differences were observed in the *StandToSit* case among NMS, NMS-GP, and NMS-BNN models (NMS:$\leq$ 7.2%, NMS-GP: $\leq$ 5.5%; NMS-BNN: $\leq$ 6.8%).

Both the NMS-GP and NMS-BNN models had relatively high uncertainties in the predicted knee torque at the beginning of each movement, particularly in the $Gait_{self}$ case (Figure 7A). In the NMS-GP model, high uncertainties were observed during terminal stance and terminal swing in the $Gait_{self}$ case. On the other hand, the NMS-BNN model had high uncertainties during terminal stance, initial swing, and terminal swing in all gait speeds.

The predicted torque by the NMS models had a poorer agreement with the measured torque compared to the NMS-GP and NMS-BNN models (Figures 7B, C). Relatively high offsets were observed at the beginning of each movement in NMS models.

3.2. Exoskeleton assistive torque tracking performance

Overall, the knee exoskeleton accurately provided the required assistive torque with relatively low error (RMSE:$\leq$ 0.06 Nm/kg, NRMSE: $\leq$ 5.6%, Figure 8). Among the five movements, the NRMSE was evenly distributed among all subjects for walking movements, while one outlier was noted in both sit-to-stand and stand-to-sit movements. The sit-to-stand movement had the highest tracking error among the five movements.

Generally, the actual assistive torque provided by the knee exoskeleton matched the desired torque well (Figure 9). However, it is important to note that limited torque was provided at the start of the sit-to-stand movement. Additionally, relatively low assistive torque was observed during the terminal stance of fast walking compared to self-selected speed walking.

4. Discussion

We developed an NMS-BNN-based adaptive control framework for a knee exoskeleton using muscle EMG signals and joint kinematics. We also compared the predictions with uncertainties from the NMS-BNN model with those from the NMS-GP model. We observed both NMS-BNN and NMS-GP models showed accurate predictions of knee joint torque with low error, outperforming traditional NMS models, indicating the benefits of incorporating NMS features into machine learning models. High uncertainties, however, were observed at the beginning of each movement and at terminal stance and terminal swing in the self-selected speed walking in both NMS-BNN and NMS-GP models. The knee exoskeleton provided the desired assistive torque accurately, with a relatively low error. Lower levels of torque were observed during terminal stance in fast walking compared to self-selected walking speed. The level of assistive torque was determined and adjusted based on the uncertainty in the NMS-BNN predictions, promoting the safety of the exoskeleton-user interface.

Incorporating the user's physiological joint torque into exoskeleton control strategies has in recent years become feasible, and has vast potential to improve task performance and rehabilitation outcomes. Among common techniques for predicting joint torque, EMG-driven NMS models require expertise and complex calibration, whereas machine learning models are more accessible but considered as black boxes (Hoang et al., 2018; Ezati et al., 2019; Soleimani and Nazerfard, 2021). To improve prediction accuracy, ANNs have been integrated into NMS models, allowing the advantages of both approaches to be leveraged. However, ensuring the safety and efficiency of exoskeleton control is also crucial, particularly when using predicted torque as inputs for the exoskeleton. To address this, this study integrated NMS with machine learning models with uncertainty quantification for joint torque prediction. As mentioned earlier, Both BNNs and GP models can provide predictions with associated uncertainties. While BNN models incorporate Bayesian methods to quantify the uncertainty in predictions, GP models are based on Gaussian processes and provide a probabilistic model for predictions with uncertainties. In the current study, we also compared the predictions with uncertainties between NMS-BNN and NMS-GP models. We found both NMS-BNN and NMS-GP models accurately predicted knee joint torque with relatively low error (RMSE: NMS-GP $\leq$ 0.05 Nm/kg, NMS-BNN $\leq$ 0.07 Nm/kg; NRMSE: NMS-GP $\leq$ 5.9%, NMS-BNN $\leq$ 6.8%), and were found to be superior to traditional NMS models (RMSE $\leq$ 0.14 Nm/kg, NRMSE $\leq$ 18.3%). The results are attributed to the addition of machine learning layers, which further train the model by minimizing the error between measured and predicted joint torque.

The quantification of uncertainty by either the NMS-BNN or NMS-GP models can supply the exoskeleton controller with valuable data for decision-making, which could enhance safety in the exoskeleton-user interaction. For instance, we observed high uncertainties at the beginning of each movement in both NMS-BNN and NMS-GP models (Figure 7). This is likely due to the physical characteristics adopted from NMS models, which show a noticeable offset at the start of each movement. In NMS models, two prior time steps of neural activation from each MTU are required to calculate muscle neural activation (Zhang et al., 2022). At the beginning of a cycle, these past two neural activation values are not yet obtainable and are approximated using EMG signals from

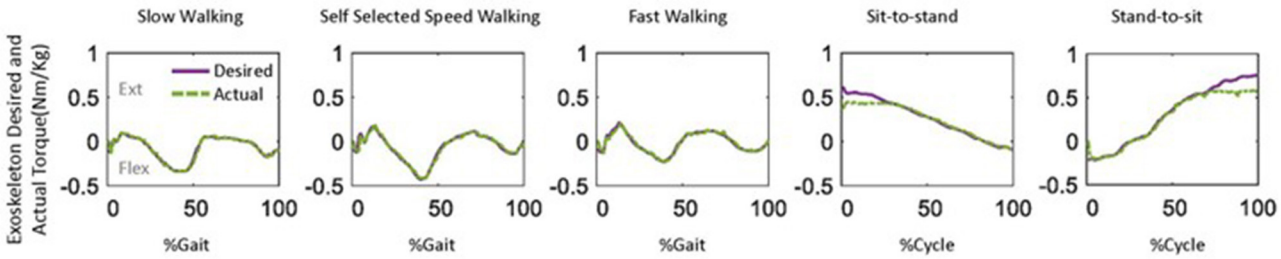


FIGURE 9
One example of the desired and actual assistive torque provided by the knee exoskeleton during five daily activities.

two previous time steps, potentially leading to initial inaccuracies in predicted torque. Furthermore, high uncertainties were observed during the terminal stance and terminal swing in self-selected speed walking in both NMS-BNN and NMS-GP models. This may be attributed to the of transitions between the stance and swing phases of gait.

The knee exoskeleton provided the desired assistive torque accurately, with a relatively low error (RMSE:$\leq$ 0.06 Nm/kg, NRMSE: $\leq$ 5.6%, Figure 8). The assistive torque was achieved through current control in the motor, a widely used closed-loop control technique (Zhang et al., 2018; Azocar et al., 2020; Nuckols et al., 2021). The current control system aims to maintain a consistent current in the motor, even as its speed and load conditions vary. Precise control over the motor's torque production can be achieved through current control, though accuracy may be influenced by factors such as the quality of current sensing and the speed of the control loop's response. To estimate output torque, the control system uses the measured current as feedback, as the current is proportional to the torque produced by the motor (Azocar et al., 2020). This allows the control system to determine the amount of torque produced and adjust the exoskeleton accordingly. We observed lower levels of torque during the terminal stance of fast walking compared to self-selected walking speed (Figure 9). This discrepancy may be due to the increased uncertainties present during fast walking, which in turn led to a lower level of assistance torque being assigned according to our control strategy (Equation 15). It is worth noting that an outlier was observed in the sit-to-stand and stand-to-sit movements (Figure 8). This deviation may be attributed to the limited torque capacity of the exoskeleton at the beginning of the sit-to-stand movement and at the end of the stand-to-sit movement (Figure 9).

This study focused on evaluating the feasibility of the NMS-BNN framework by implementing a basic current control strategy. The objective was to assess the overall viability of the framework. However, future studies are necessary to investigate more advanced control techniques, such as impedance control. The current control strategy may result in less smooth assistive torque. Therefore, in future studies, we recommend incorporating an improved control strategy that takes into account both uncertainties and the closest points of predicted torque to enhance the smoothness and improve the user-exoskeleton interface. Furthermore, while our current study centers on the knee joint, it is important to note that the approach can be adapted and extended to other joints as well. Additionally, it is worth mentioning that the maximum torque that can be generated by the system is 54 Nm, which may also impact the smoothness of the assistive torque. Thus, this should be considered in future control strategies. It should be noted that this study did not involve testing the performance of the NMS-BNN-based adaptive framework on real users for practical applications. Further research is essential to address this issue and ascertain the practicality of the framework.

5. Conclusion

In this study, we proposed an NMS-BNN-based adaptive control framework for a knee exoskeleton that uses muscle EMG signals and joint kinematics. The NMS-BNN model combines a traditional NMS model with modern machine learning techniques and includes uncertainty quantification. The proposed framework also measures uncertainty in predictions and incorporates it into the control design to ensure safety of the exoskeleton-user interface. We also compared the performance of the NMS-BNN model to an NMS-GP model, which also predicts uncertainties. Detailed information relating to how to combine traditional models with machine learning models with uncertainties can provide useful guidance for designing exoskeleton control strategies.

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 humans were approved by Swedish Ethical Review Authority. The studies were conducted in accordance with the local legislation and institutional requirements. The participants provided their written informed consent to participate in this study. Written informed consent was obtained from the individual(s) for the publication of any potentially identifiable images or data included in this article.

Author contributions

LZ: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Software, Validation, Visualization, Writing—original draft, Writing—review and editing. XZha: Writing—review and editing, Formal analysis, Methodology, Investigation. XZhu: Methodology, Supervision, Writing—review and editing. RW: Methodology, Supervision, Writing—review and editing. EG-F: Conceptualization, Data curation, Funding acquisition, Methodology, Project administration, Resources, Supervision, Writing—review and editing.

Funding

We would like to acknowledge financial support from Promobilia Foundation (A22078, 18014, 21302, and 18200), Swedish Research Council (2018-04902 and 2018-00750), and Simons Foundation (504054).

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

Azocar, A. F., Mooney, L. M., Duval, J. F., Simon, A. M., Hargrove, L. J., and Rouse, E. J. (2020). Design and clinical implementation of an open-source bionic leg. *Nat. Biomed. Eng.* 4, 941–953. doi: 10.1038/s41551-020-00619-3

Bergstra, J., and Bengio, Y. (2012). Random search for hyper-parameter optimization. *J. Mach. Learn. Res.* 13, 1–25.

Buchanan, T. S., Lloyd, D. G., Manal, K., and Besier, T. F. (2004). Neuromusculoskeletal modeling: estimation of muscle forces and joint moments and movements from measurements of neural command. *J. Appl. Biomech.* 20, 367–395. doi: 10.1123/jab.20.4.367

Cao, H., Chen, G., Li, Z., Feng, Q., Lin, J., and Knoll, A. (2022a). Efficient grasp detection network with Gaussian-based grasp representation for robotic manipulation. *IEEE/ASME Trans. Mechatron.* 28, 1384–1394. doi: 10.1109/TMECH.2022.3224314

Cao, H., Chen, G., Li, Z., Hu, Y., and Knoll, A. (2022b). Neurograsp: multimodal neural network with Euler region regression for neuromorphic vision-based grasp pose estimation. *IEEE Trans. Instrument. Meas.* 71, 1–11. doi: 10.1109/TIM.2022.3179469

Chen, B., Ma, H., Qin, L.-Y., Gao, F., Chan, K.-M., Law, S.-W., et al. (2016). Recent developments and challenges of lower extremity exoskeletons. *J. Orthopaed. Transl.* 5, 26–37. doi: 10.1016/j.jot.2015.09.007

Chen, N., Qian, Z., Nabney, I. T., and Meng, X. (2013). Wind power forecasts using Gaussian processes and numerical weather prediction. *IEEE Trans. Power Syst.* 29, 656–665. doi: 10.1109/TPWRS.2013.2282366

Cursi, F., Modugno, V., Lanari, L., Oriolo, G., and Kormushev, P. (2021). Bayesian neural network modeling and hierarchical MPC for a tendon-driven surgical robot with uncertainty minimization. *IEEE Robot. Automat. Lett.* 6, 2642–2649. doi: 10.1109/LRA.2021.3062339

Derrick, T. R., van den Bogert, A. J., Cereatti, A., Dumas, R., Fantozzi, S., and Leardini, A. (2020). ISB recommendations on the reporting of intersegmental forces and moments during human motion analysis. *J. Biomech.* 99, 1–10. doi: 10.1016/j.jbiomech.2019.109533

Durandau, G., Farina, D., and Sartori, M. (2017). Robust real-time musculoskeletal modeling driven by electromyograms. *IEEE Trans. Biomed. Eng.* 65, 556–564. doi: 10.1109/TBME.2017.2704085

Ezati, M., Ghannadi, B., and McPhee, J. (2019). A review of simulation methods for human movement dynamics with emphasis on gait. *Multibody Syst. Dyn.* 47, 265–292. doi: 10.1007/s11044-019-09685-1

Fujii, K., Abe, T., Kubota, S., Marushima, A., Kawamoto, H., Ueno, T., et al. (2017). The voluntary driven exoskeleton hybrid assistive limb (HAL) for postoperative training of thoracic ossification of the posterior longitudinal ligament: a case report. *J. Spinal Cord Med.* 40, 361–367. doi: 10.1080/10790268.2016.1142056

Guo, H., Meng, Z., Huang, Z., Kang, L. W., Chen, Z., Meghjani, M., et al. (2019). "Safe path planning with gaussian process regulated risk map," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Macau: IEEE), 2044–2051.

Hoang, H. X., Pizzolato, C., Diamond, L. E., and Lloyd, D. G. (2018). Subject-specific calibration of neuromuscular parameters enables neuromusculoskeletal models to estimate physiologically plausible hip joint contact forces in healthy adults. *J. Biomech.* 80, 111–120. doi: 10.1016/j.jbiomech.2018.08.023

Hu, Y., Chen, G., Li, Z., and Knoll, A. (2022). Robot policy improvement with natural evolution strategies for stable nonlinear dynamical system. *IEEE Trans. Cybernet.* 53, 4002–4014. doi: 10.1109/TCYB.2022.3192049

Huang, Y., Song, R., Argha, A., Savkin, A. V., Celler, B. G., and Su, S. W. (2019). Continuous description of human 3d motion intent through switching mechanism. *IEEE Trans. Neural Syst. Rehabil. Eng.* 28, 277–286. doi: 10.1109/TNSRE.2019.2949203

Kronheim, B., Kuchera, M. P., and Prosper, H. B. (2022). Tensorbnn: Bayesian inference for neural networks using tensorflow. *Comput. Phys. Commun.* 270, 1–8. doi: 10.1016/j.cpc.2021.108168

Leboeuf, F., Baker, R., Barré, A., Reay, J., Jones, R., and Sangeux, M. (2019). The conventional gait model, an open-source implementation that reproduces the past but prepares for the future. *Gait Post.* 69, 235–241. doi: 10.1016/j.gaitpost.2019.04.015

Li, Z., Huang, B., Ye, Z., Deng, M., and Yang, C. (2018). Physical human–robot interaction of a robotic exoskeleton by admittance control. *IEEE Trans. Indus. Electron.* 65, 9614–9624. doi: 10.1109/TIE.2018.2821649

Li, Z., Li, G., Wu, X., Kan, Z., Su, H., and Liu, Y. (2021). Asymmetric cooperation control of dual-arm exoskeletons using human collaborative manipulation models. *IEEE Trans. Cybernet.* 52, 12126–12139. doi: 10.1109/TCYB.2021.3113709

Li, Z., Yuan, Y., Luo, L., Su, W., Zhao, K., Xu, C., et al. (2019). Hybrid brain/muscle signals powered wearable walking exoskeleton enhancing motor ability in climbing stairs activity. *IEEE Trans. Med. Robot. Bionics* 1, 218–227. doi: 10.1109/TMRB.2019.2949865

Liang, J., Shi, Z., Zhu, F., Chen, W., Chen, X., and Li, Y. (2021). Gaussian process autoregression for joint angle prediction based on sEMG signals. *Front. Public Health* 9, 685596. doi: 10.3389/fpubh.2021.685596

Liu, Z., Zhong, B., Zhong, W., Guo, K., and Zhang, M. (2019). "A new trajectory determination method for robot-assisted ankle ligament rehabilitation," in *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society* (Berlin: IEEE), 5390–5393.

Lloyd, D. G., and Besier, T. F. (2003). An EMG-driven musculoskeletal model to estimate muscle forces and knee joint moments *in vivo*. *J. Biomech.* 36, 765–776. doi: 10.1016/S0021-9290(03)00010-1

Lu, T.-W., and O'connor, J. (1999). Bone position estimation from skin marker co-ordinates using global optimisation with joint constraints. *J. Biomech.* 32, 129–134.

Mantoan, A., Pizzolato, C., Sartori, M., Sawacha, Z., Cobelli, C., and Reggiani, M. (2015). MOtoNMS: a MATLAB toolbox to process motion data for neuromusculoskeletal modeling and simulation. *Source Code Biol. Med.* 10, 1–14. doi: 10.1186/s13029-015-0044-4

Marchal-Crespo, L., and Reinkensmeyer, D. J. (2009). Review of control strategies for robotic movement training after neurologic injury. *J. Neuroeng. Rehabil.* 6, 1–15. doi: 10.1186/1743-0003-6-20

Maritz, J., Lubbe, F., and Lagrange, L. (2018). A practical guide to Gaussian process regression for energy measurement and verification within the Bayesian framework. *Energies* 11, 1–12. doi: 10.3390/en11040935

Meng, W., Liu, Q., Zhou, Z., Ai, Q., Sheng, B., and Xie, S. S. (2015). Recent development of mechanisms and control strategies for robot-assisted lower limb rehabilitation. *Mechatronics* 31, 132–145. doi: 10.1016/j.mechatronics.2015.04.005

Mounis, S. Y. A., Azlan, N. Z., and Sado, F. (2019). Assist-as-needed control strategy for upper-limb rehabilitation based on subject's functional ability. *Meas. Control* 52, 1354–1361. doi: 10.1177/0020294019866844

Nuckols, R. W., Lee, S., Swaminathan, K., Orzel, D., Howe, R. D., and Walsh, C. J. (2021). Individualization of exosuit assistance based on measured muscle dynamics during versatile walking. *Sci. Robot.* 6, 1–11. doi: 10.1126/scirobotics.abj1362

Pandy, M. G. (2001). Computer modeling and simulation of human movement. *Annu. Rev. Biomed. Eng.* 3, 245–273. doi: 10.1146/annurev.bioeng.3.1.245

Pizzolato, C., Lloyd, D. G., Sartori, M., Ceseracciu, E., Besier, T. F., Fregly, B. J., et al. (2015). CEINMS: a toolbox to investigate the influence of different neural control solutions on the prediction of muscle excitation and joint moments during dynamic motor tasks. *J. Biomech.* 48, 3929–3936. doi: 10.1016/j.jbiomech.2015.09.021

Pizzolato, C., Reggiani, M., Modenese, L., and Lloyd, D. (2017). Real-time inverse kinematics and inverse dynamics for lower limb applications using opensim. *Comput. Methods Biomech. Biomed. Eng.* 20, 436–445. doi: 10.1080/10255842.2016.1240789

Pizzolato, C., Saxby, D. J., Palipana, D., Diamond, L. E., Barrett, R. S., Teng, Y. D., et al. (2019). Neuromusculoskeletal modeling-based prostheses for recovery after spinal cord injury. *Front. Neurorobot.* 13, 97. doi: 10.3389/fnbot.2019.00097

Proietti, T., Crocher, V., Roby-Brami, A., and Jarrasse, N. (2016). Upper-limb robotic exoskeletons for neurorehabilitation: a review on control strategies. *IEEE Rev. Biomed. Eng.* 9, 4–14. doi: 10.1109/RBME.2016.2552201

Rasmussen, C. E. (2004). Gaussian processes in machine learning. *Adv. Lectures Mach. Learn.* 3176, 63–71. doi: 10.1007/978-3-540-28650-9_4

Sartori, M., Durandau, G., Došen, S., and Farina, D. (2018). Robust simultaneous myoelectric control of multiple degrees of freedom in wrist-hand prostheses by real-time neuromusculoskeletal modeling. *J. Neural Eng.* 15, 1–15. doi: 10.1088/1741-2552/aae26b

Sartori, M., Lloyd, D. G., and Farina, D. (2016). Neural data-driven musculoskeletal modeling for personalized neurorehabilitation technologies. *IEEE Trans. Biomed. Eng.* 63, 879–893. doi: 10.1109/TBME.2016.2538296

Sartori, M., Reggiani, M., Lloyd, D. G., and Pagello, E. (2011). "A neuromusculoskeletal model of the human lower limb: towards EMG-driven actuation of multiple joints in powered orthoses," in *2011 IEEE International Conference on Rehabilitation Robotics* (Zürich: IEEE), 1–6.

Soleimani, E., and Nazerfard, E. (2021). Cross-subject transfer learning in human activity recognition systems using generative adversarial networks. *Neurocomputing* 426, 26–34. doi: 10.1016/j.neucom.2020.10.056

Wang, Z., Mohamed, S., and Freitas, N. (2013). "Adaptive Hamiltonian and Riemann manifold Monte Carlo," in *International Conference on Machine Learning* (Georgia: PMLR), 1462–1470.

Wei, W., Kaiming, Y., Yu, Z., Yuyang, Q., Chenhui, W., and Min, L. (2021). Walking speed estimation from a wearable insole pressure system embedded with an accelerometer using Bayesian neural network. *J. Eng. Sci. Med. Diagnost. Ther.* 4, 1–7. doi: 10.1115/1.4049964

Winter, D. A., Sidwall, H. G., and Hobson, D. A. (1974). Measurement and reduction of noise in kinematics of locomotion. *J. Biomech.* 7, 157–159.

Yao, S., Zhuang, Y., Li, Z., and Song, R. (2018). Adaptive admittance control for an ankle exoskeleton using an emg-driven musculoskeletal model. *Front. Neurorobot.* 12, 16. doi: 10.3389/fnbot.2018.00016

Yun, Y., Kim, H.-C., Shin, S. Y., Lee, J., Deshpande, A. D., and Kim, C. (2014). Statistical method for prediction of gait kinematics with Gaussian process regression. *J. Biomech.* 47, 186–192. doi: 10.1016/j.jbiomech.2013.09.032

Zhang, L., Li, Z., Hu, Y., Smith, C., Farewik, E. M. G., and Wang, R. (2020). Ankle joint torque estimation using an EMG-driven neuromusculoskeletal model and an artificial neural network model. *IEEE Trans. Automat. Sci. Eng.* 18, 564–573. doi: 10.1109/TASE.2020.3033664

Zhang, L., Zhu, X., Farewik, E. M. G., and Wang, R. (2021). "Estimation of ankle dynamic joint torque by a neuromusculoskeletal solver-informed NN model," in *2021 6th IEEE International Conference on Advanced Robotics and Mechatronics (ICARM)* (Chongqing: IEEE), 75–80.

Zhang, L., Zhu, X., Gutierrez-Farewik, E. M., and Wang, R. (2022). Ankle joint torque prediction using an NMS solver informed-ANN model and transfer learning. *IEEE J. Biomed. Health Inform.* 26, 5895–5906. doi: 10.1109/JBHI.2022.3207313

Zhang, M., McDaid, A., Veale, A. J., Peng, Y., and Xie, S. Q. (2019). Adaptive trajectory tracking control of a parallel ankle rehabilitation robot with joint-space force distribution. *IEEE Access* 7, 85812–85820. doi: 10.1109/ACCESS.2019.2925182

Zhang, T., Tran, M., and Huang, H. (2018). Design and experimental verification of hip exoskeleton with balance capacities for walking assistance. *IEEE/ASME Trans. Mechatron.* 23, 274–285. doi: 10.1109/TMECH.2018.2790358

Zhong, B., Silva, R. L., Li, M., Huang, H., and Lobaton, E. (2021). Environmental context prediction for lower limb prostheses with uncertainty quantification. *IEEE Trans. Automat. Sci. Eng.* 18, 458–470. doi: 10.1109/TASE.2020.2993399

frontiers | Frontiers in Neuroscience

TYPE Original Research
PUBLISHED 17 October 2023
DOI 10.3389/fnins.2023.1287053

Check for updates

OPEN ACCESS

EDITED BY
Alois C. Knoll,
Technical University of Munich, Germany

REVIEWED BY
Jiaqi Hu,
Rice University, United States
Baodan Bai,
Shanghai University of Medicine and Health Sciences, China

*CORRESPONDENCE
Shu Shen
✉ shens@njupt.edu.cn
Xinrong Chen
✉ chenxinrong@fudan.edu.cn

RECEIVED 01 September 2023
ACCEPTED 03 October 2023
PUBLISHED 17 October 2023

CITATION
Zhang X, Wang J, Dai X, Shen S and Chen X (2023) A non-contact interactive system for multimodal surgical robots based on LeapMotion and visual tags. *Front. Neurosci.* 17:1287053. doi: 10.3389/fnins.2023.1287053

COPYRIGHT
© 2023 Zhang, Wang, Dai, Shen and Chen. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

A non-contact interactive system for multimodal surgical robots based on LeapMotion and visual tags

Xinkang Zhang[1,2], Jie Wang[1,2], Xiaokun Dai[1,2], Shu Shen[3]* and Xinrong Chen[1,2]*

[1]Academy for Engineering and Technology, Fudan University, Shanghai, China, [2]Shanghai Key Laboratory of Medical Image Computing and Computer Assisted Intervention, Shanghai, China, [3]Jiangsu High Technology Research Key Laboratory for Wireless Sensor Networks, Nanjing University of Posts and Telecommunications, Nanjing, China

In recent years, the integration of robots in minimally invasive surgery has gained significant traction in clinical practice. However, conventional contact-based human-computer interaction poses the risk of bacterial infection, significantly limiting the role of robots in surgery. To address this limitation, we propose an innovative interaction method rooted in gestures and visual tags, allowing surgeons to control and fine-tune surgical robots without physical contact with the environment. By encoding the six gestures collected using LeapMotion, we can effectively control the surgical robot in a non-contact manner. Moreover, utilizing Aruco technology, we have accurately identified the 3D spatial position of the visual label, and developed 12 fine-tuning operations to refine surgical instruments. To evaluate the applicability of our proposed system in surgery, we designed a relevant experimental setup. In the experiment, we achieved enough precision. These results demonstrate that our system meets the clinical standard, providing doctors with a non-contact and flexible means of interacting with robots during surgery.

KEYWORDS

surgical robot, human-computer interaction, LeapMotion, Aruco, non-contact

1. Introduction

With the rapid advancement of sensor technology and computer technology in recent years, the capabilities of robots and manipulators have become more sophisticated and performed better. As a result, they have become increasingly prominent in various aspects of daily life and specific fields. This surge in demand for human-computer interaction and the development of human-computer interaction technology have led to a proliferation of mainstream methods, including joystick buttons, voice interaction, and gesture recognition, among others. In recent years, human-computer interaction has emerged as a popular research area in the fields of smart homes, customer service, remote control, and medicine.

In the medical field, human-computer interaction technology is widely utilized for rehabilitation purposes, assistance for those with disabilities, and surgical robots (Díaz et al., 2014; Ohmura et al., 2018; Nagyné Elek and Haidegger, 2019; Cao et al., 2022a). Chen C. et al. (2023) decompounds muscle EMG signals to quantify neural features and map them to

three-degree-of-freedom wrist movements through a multiple linear regression model. This method has shown great potential in the reconstruction process. Chen W. et al. (2023) employs myoelectric signals from the lower limbs to control exoskeletons. The quadriceps and hamstring muscles are selected to obtain gait information, and 16 dry electrodes measure electromyography signals transmitting information to a host computer via Bluetooth. The processed signals help users control their gait effectively. Ali et al. (2023) proposes a deep learning-based Thought-to-Text conversion for patients with neurodegenerative diseases like Alzheimer's disease type through EEG. Collected EEG signals are preprocessed with a band-pass filter and divided into five classifier tasks using XGBoost (Ogunleye and Wang, 2019) classifier. Finally, the CNN-LSTM deep neural network (Mutegeki and Han, 2020) learns advanced features from MI-EEG signals and translates them into corresponding alphabets. Coughlan et al. (2020) utilizes machine vision to provide voice prompts for visually impaired patients. Specifically, the Camera Input–Output (CamIO) augmented reality tool guides the patient in 3D space using a pen covered with 3D visual labels, leading the patient closer to the target.

In the realm of surgical robots, various information modalities will also be utilized to assist medical professionals (De Rossi et al., 2021; Long et al., 2021; Cao et al., 2022b). Surgical robots possess distinct advantages over humans, including unparalleled precision, exceptional stability, and rapid execution. As a result, they have become increasingly prominent in endoscopic surgery (Kang et al., 2009; Gifari et al., 2019) in recent years.

According to Jacob et al. (2011), a system is proposed to track the surgeon's hands, identify required surgical instruments, and have the robotic arm pass them to the doctor. However, this system is still slower than manual work, and nurses who pass instruments themselves do not need to leave the operating table, indicating a weak irreplaceability in surgery. Next, referring to Van Amsterdam et al. (2022), a video-based surgical assessment system is proposed, including automatic activity recognition, technical skill assessment, book assistance, and other functions. By integrating a multi-modal attention mechanism into a dual-stream temporal convolutional network, real-time dynamic weight kinematics and visual representation calculation improve fine-grained surgical data analysis accuracy. However, these methods have a common problem: complexity and single modality interaction. To address this, different interaction methods need to be introduced. Moving on, Cho et al. (2018) aims to establish a non-contact computer-assisted surgery system. LeapMotion's mature gesture recognition module is used to obtain hand gestures, and features are manually extracted before being aligned with support vector machine (SVM) classification (Chang and Lin, 2011). This creates a non-contact control interface with gesture recognition functionality. Similarly, Dwivedi et al. (2019) combines myoelectricity and visual tags to manipulate a robotic arm. Kam et al. (2018) provides three-dimensional attitude information to control the robotic arm's position, while processed myoelectric signals control the bionic hand at the arm's end for grasping actions. In our previous work (Wang et al., 2023), we explored gestural interactions with various countries to achieve seamless human-computer interaction and contactless operation. However, a common limitation is that users must maintain a predetermined posture to perform a single trigger on a specific action, which lacks the dynamic flexibility of more complex gestures.

Most of the existing systems rely on complex technology and have high equipment requirements. Therefore, finding a balance between simplicity and robustness is crucial. In this paper, we propose a straightforward method that integrates consumer-grade gesture recognition technology LeapMotion with accurate 3D space perception technology based on visual labels Aruco. This framework allows surgeons to interact with surgical robots without physical contact, leveraging the potential of surgical navigation systems. By doing so, we can maximize the capabilities of surgical robots in surgical procedures.

2. Materials and methods

2.1. System composition

The navigation operating system is primarily composed of a computer workstation, a gesture recognition module, an Aruco recognition module, a surgical navigator based on optical positioning, a surgical probe, a positioner, and a robot module comprising a seven-axis robotic arm and a control cabinet. All navigators and operating tables can be moved according to the position of the patient and doctor. One of the workstations is connected to the surgical navigation system and robotic system. The surgical navigation system tracks the precise position of surgical instruments and patients in real-time using the locator installed on the surgical instrument and the operating table, which consists of an array of reflective balls. Doctors can operate the collaborative surgical robot and its end-effectors with great flexibility through gesture recognition and tag-based recognition technologies. The entire surgical navigation system creates an enhanced surgical space, aligning the target area or target point in the preoperative medical image with the real patient's body to provide visual assistance for the doctor. In such cases, gesture and visual tag information can be used as a remote, aseptic method to adjust the position of surgical instruments (Figure 1).

2.2. System workflow

The proposed whole surgical system is depicted in Figure 2. The computer workstation acts as the central control unit to regulate the robot, generate enhanced surgical imaging data, and facilitate human-computer interaction. The hardware component of the collaborative interactive surgical robot system is represented by blue in the figure. In addition to the operator, surgical navigation interface, surgical robot movement/execution, patient, and contactless interaction system, the figure also depicts the data flow of hardware-participant interaction, indicated by black arrows, and the surgeon-centric flow of interaction, shown using red arrows. To achieve varying degrees of operational flexibility under non-contact conditions, a system employing non-contact human-computer interaction is proposed. The surgeon must adjust the position of the surgical instrument or operate the surgical instrument according to a predetermined trajectory during surgery. The surgeon can move and fine-tune the surgical robot through contactless gesture control and a special marker with visual markers. The article's core is to utilize two types of sensors and decoding technologies to enable seamless and flexible interaction between the surgeon and instrument during the procedure using non-contact means.

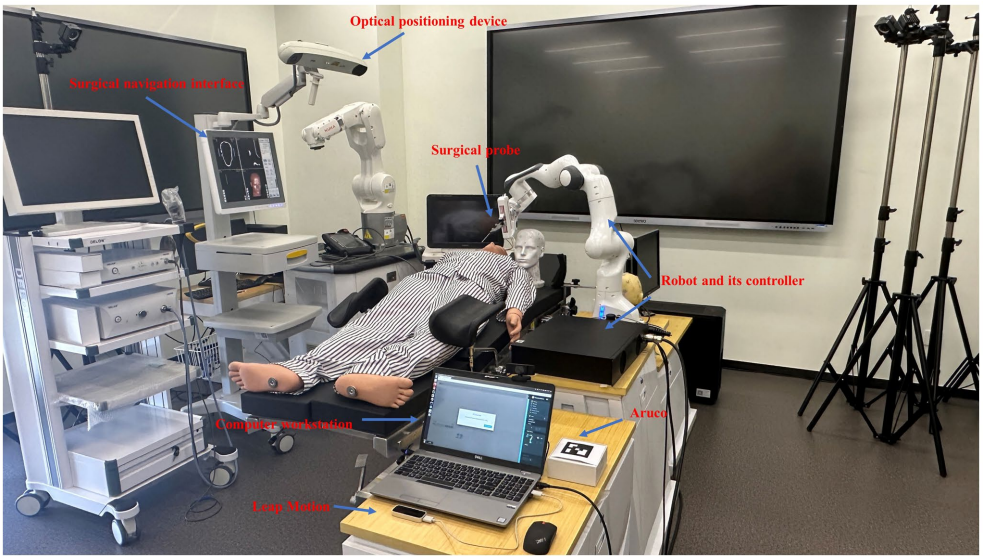


FIGURE 1
Overview of the surgical navigation robot system.

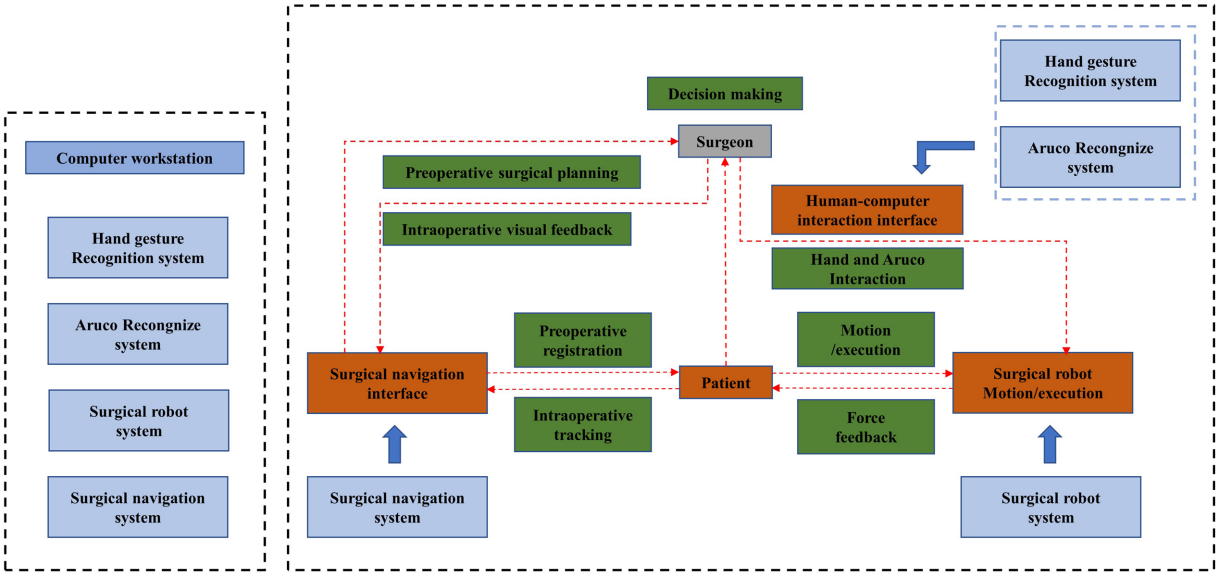


FIGURE 2
Workflow of the surgeon-robot system.

2.3. Gesture interaction based on LeapMotion

In this section, our attention is directed toward the gesture recognition module based on LeapMotion. The specific architecture of the model is illustrated in Figure 3.

Gesture information collected via LeapMotion is utilized to rapidly adjust the position of the surgical instrument held by the robotic arm. LeapMotion is a device used to collect gestures, and the collected information includes the overall position of the hand, the position of key points on the hand, the length and direction of the fingers, and other related data. In this work, LeapMotion will continuously obtain timing frames, and the pose data will be parsed and expressed in a form that is convenient for us. The posture data will then be decoded into instructions and transmitted to the surgical robot module to perform the corresponding operation. For the simplicity and robustness of the instructions, we have defined six instructions to control the robotic arm to move quickly and over a large range.

2.3.1. Introduction and principle of LeapMotion equipment

LeapMotion is a somatosensory device developed by LEAP, which specializes in recognizing the geometric information of the hand. The device consists of two cameras and three infrared LED lights, allowing it to obtain images from two angles and depth information using infrared light. The binocular camera's principle is based on the human

FIGURE 3
The LeapMotion used in this work.

TABLE 1 Accuracy of hand recognition with predefined categories.

Category	1	2	3	4	5	6
Accuracy	95%	97%	99%	96%	98%	98%

visual system, making LeapMotion more accurate and reliable in hand recognition. Its recognition accuracy reaches one hundredth of a millimeter, which is more accurate than Microsoft's Kinect and has more advantages in gesture interaction. Additionally, LeapMotion has a lower price compared to data gloves, making it a more cost-effective option in most scenarios, with sufficient precision (Table 1).

2.3.2. Reading and data processing of LeapMotion

LeapMotion's two cameras capture images from different angles and reconstruct the 3D information of the palm in space. The detection range is approximately 2.5–60 cm above the sensor. The coordinates of the entire space are as follows: the Cartesian coordinate system is centered on the sensor, and the *X*-axis of the coordinate's points to the right, the *Y*-axis points upward, and the *Z*-axis points away from the screen. The unit of the output distance value is millimeters. Each frame of information contains the position and orientation of the center of the palm, as well as the position and pose information of each key point on the hands and fingers. The approximate structure of the read structure is shown in Figure 3.

To represent manipulation commands, certain hand features must be extracted. In the interaction design of this article, specific keypoint distance and normal features are selected as features to reflect the uniqueness of different hand poses. In the interaction design, there are two overall postures of the hand: one where the palm is facing down, and the other where the palm is facing inward. The orientation can be determined by the normal vector of the hand. To determine the orientation, the method used is to calculate the cosine similarity between the vector and the standard coordinate axis vector. The similarity ranges from −1 to 1, and the greater the similarity, the closer the two vectors are aligned. From this, we can obtain the palm direction information.

To detect finger poses, distance features between key points are extracted, specifically, whether the fingers are bent or stretched based on the distance from the fingertip to the base of the palm. And determine the command according to the gesture of the finger.

The hand information in each frame can be extracted from the frames read out from the device, including the first 3D coordinate $x_{start-i} \in R^3$ and the tail 3D coordinate $x_{end-i} \in R^3$ of each finger. In addition, the position of the root $x_{root} \in R^3$ of the hand can also be obtained.

$$s_i = \begin{cases} \mathbf{1} & \text{if } \boldsymbol{d}\left(\boldsymbol{x_{start-i}}, \boldsymbol{x_{root-i}}\right) \geq \mathbf{threshold\_d} \\ \mathbf{0} & \text{if } \boldsymbol{d}\left(\boldsymbol{x_{start-i}}, \boldsymbol{x_{root-i}}\right) < \mathbf{threshold\_d} \end{cases} \quad (1)$$

where s_i indicates the state of the i-th finger, and **threshold_d** represents the threshold for judging the state of the finger, $\boldsymbol{d}$ is an operator used to calculate the Euclidean distance.

To take palm orientation into account, two vectors are used to represent.

$$d_L \in R^3 \quad (2)$$

$$d_R \in R^3 \quad (3)$$

where d_L indicates the palm facing direction of the left hand, while d_L indicates the palm facing direction of the right hand.

$$S_L = \left(s_0, s_1, s_2, s_3, d_L\right) \quad (4)$$

$$S_R = \left(s_0, s_1, s_2, s_3, d_R\right) \quad (5)$$

where S_L indicates the state of the left hand, while S_R indicates the state of the right hand.

The state of the two hands will be decoded into different operating instructions, corresponding to different actions.

2.4. Interaction based on Aruco

A second-stage fine-tuning of the manipulator pose and position is carried out using an Aruco-based pose estimation method. Aruco is a widely used label in computer vision localization tasks and augmented reality applications, first proposed in the paper (Garrido-Jurado et al., 2014) in 2014. By placing Aruco on the object to be estimated or tracked, attitude estimation and tracking of the object can be achieved through its posture. In this paper, Aruco is used as a means of manipulation, and the operation action is obtained by detecting its posture. Through rotation and displacement in different directions, a total of 12 operating instructions, corresponding to 12 distinct actions, are generated. These instructions are used for fine-tuning the robotic arm in a limited range.

2.4.1. Encoding

The full name of the Aruco code is the Augmented Reality University of Cordoba, which is visually represented as a square with a black background, and the grid pattern inside the square serves as a distinct identifier. The size of the detected square is a crucial reference information for estimating 3D pose from monocular RGB images. In image editing tasks, the position information provided by the Aruco tag enables accurate processing of the perspective relationship of the image. In scenarios where robots are required to be positioned, such as automated warehouses, the Aruco mark can serve as both a positioning mark for robots and an identification mark for designated areas.

2.4.2. Aruco generation

We use opencv to generate Aruco's markers. There are 25 predefined markup dictionaries. All Arucos in each dictionary contain the same number of blocks. According to different parameters, such as size, id, border width, etc., Aruco codes with predefined patterns and sizes are generated as needed for manipulation. As shown in Figure 4, in this work, we employ Aruco codes of size 10×10.

2.4.3. Pose estimation and parameter resolve

To detect the location and pose of tags, several steps are involved. First, the most prominent contour must be extracted using a local adaptive threshold method, which is highly robust to various lighting conditions. Next, contour extraction is performed, followed by a four-vertex polygon approximation. The resulting four-vertex polygon area is then passed through the homography matrix to eliminate perspective projection. Otsu's binarization method is used for thresholding. The interior of the polygon is then divided into a grid to assign 0 or 1 individually. Finally, the result is matched with the result in the dictionary to obtain the id of the tag. The pose relative to the camera is estimated by minimizing the reprojection error of the corners.

Here, we use python based programming language and opencv to obtain the position and pose of labels in detail. Opencv is a widely used open source library for computer vision and image processing. It contains a large number of image processing tools and can help developers create applications such as machine learning, target detection and tracking, and image processing. Opencv is heavily used in both academia and industry. In order to help users develop quickly and concisely, aruco's related interfaces have been integrated into the cv2.aruco library of opencv. We call several of the APIs to implement the label positioning function. Three APIs are mainly used: cv2.aruco.Dictionary_get() is used to obtain the dictionary of labels for corresponding label searches; cv2.aruco.detectMarkers() is used to detect labels from the image; and cv2.aruco.estimatePoseSingleMarkers() is used To estimate the pose of the label detected in the previous step and obtain the rotation and translation information.

We use these APIs to get the pose $\boldsymbol{R_{vec}} \in \mathbb{R}^3$ and position $\boldsymbol{T_{vec}} \in \mathbb{R}^3$ of Aruco. $\boldsymbol{R_{vec}}$ represents rotation in three directions, which are $\boldsymbol{Ang_{pitch}}$, $\boldsymbol{Ang_{roll}}$, and $\boldsymbol{Ang_{yaw}}$. $\boldsymbol{T_{vec}}$ represents displacement in three axes, which are $\boldsymbol{t_x}$, $\boldsymbol{t_y}$, and $\boldsymbol{t_y}$.

$$\boldsymbol{s_x} = \begin{cases} \mathbf{1} & \text{if } \boldsymbol{t_x} \geq \boldsymbol{t\_threshold\_pos} \\ \mathbf{0} & \text{otherwise} \\ \mathbf{-1} & \text{if } \boldsymbol{t_x} \leq \boldsymbol{t\_threshold\_neg} \end{cases} \tag{6}$$

$$\boldsymbol{s_{pitch}} = \begin{cases} \mathbf{1} & \text{if } \boldsymbol{Ang_{pitch}} \geq \boldsymbol{A\_threshold\_pos} \\ \mathbf{0} & \text{otherwise} \\ \mathbf{-1} & \text{if } \boldsymbol{Ang_{pitch}} \leq \boldsymbol{A\_threshold\_neg} \end{cases} \tag{7}$$

where $\boldsymbol{t\_threshold\_pos}$ indicates the positive threshold, and $\boldsymbol{t\_threshold\_neg}$ represents the negative threshold. $\boldsymbol{s_x}$ indicates whether there is a significant displacement on the x-axis. And $\boldsymbol{A\_threshold\_pos}$ indicates the positive threshold, and $\boldsymbol{A\_threshold\_neg}$ represents the negative threshold. $\boldsymbol{s_{pitch}}$ indicates whether there is a significant deflection in the pitch angle.

$$S = \left(s_x, s_y, s_z, s_{pitch}, s_{roll}, s_{yaw}\right) \tag{8}$$

where S indicates the status byte from Aruco. Among the six flag bits included in S, s_x is calculated as shown in the formula above. According to the translation of the tag in the x direction in the three-dimensional coordinate system, when the translation is greater than the positive direction threshold, this direction flag is set to 1. When the translation is less than the negative direction threshold, set the direction flag to -1. If neither of the above two conditions is met, set it to 0. In the same way, the calculation method of s_y is: compare the translation of the label in the y direction with the corresponding threshold, and obtain the flag position s_y in the y direction; the calculation method of s_z is: compare the displacement of the label in the z direction with the corresponding threshold. Compare and get the flag bit s_z in the z direction. Similarly, s_{pitch} is a flag bit obtained by comparing the rotation angle on the pitch axis with a preset threshold. Based on this, we can determine whether the label has a sufficient rotation angle in this direction. s_{roll} is a flag for rotation on the roll axis, indicating whether the label has sufficient rotation in this direction. s_{yaw} is a flag for rotation on the yaw axis, indicating whether the label has sufficient rotation in this direction.

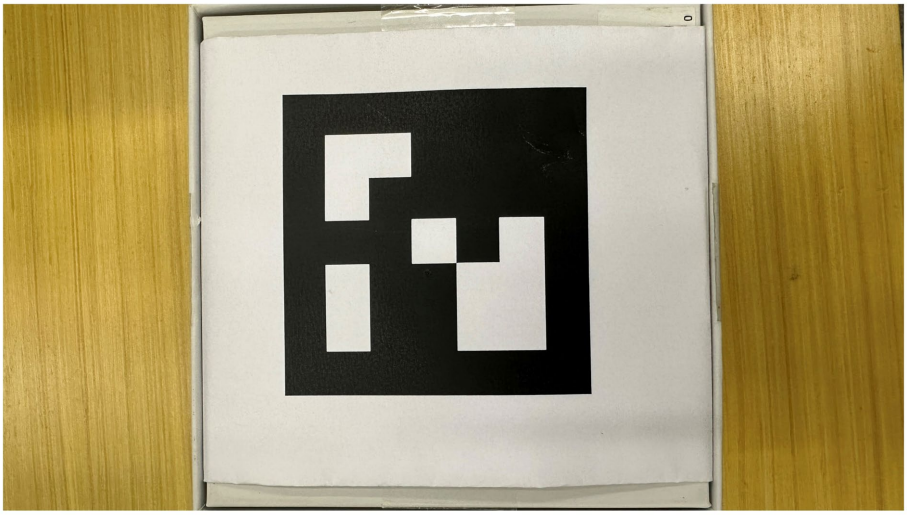


FIGURE 4
Aruco mark.

2.5. Interaction with surgical robot

In stage 1, when quickly adjusting the position, use gestures to adjust. For example, if you want to control the movement of the robotic arm, make a corresponding action above the Leap Motion. After detecting that the gesture and direction of the two hands meet the corresponding conditions, start counting. When the threshold is reached, set the preset action flag to True. Start to transmit signals to the surgical robot. When the robot reaches the designated position, change the gesture to clear the marked position.

2.5.1. Pose estimation and parameter resolve

In the second stage, the position of the Aruco tag is adjusted by translation, and its angle is adjusted by rotation, thereby triggering corresponding actions. Through the same mechanism, the instruction of the corresponding action is triggered, and the flag is set to true.

2.5.2. Aruco attitude command calculation

After obtaining the position and attitude of the tag, we need to generate the corresponding operation instructions. In this part, we use a state machine commonly used in control systems to control the triggering of operation instructions. As shown in the flow chart on the left side of Figure 5, we set a counter. When the status of the tag meets the trigger condition of a certain instruction, the corresponding counter value will increase. When the trigger condition is met for more than 30 consecutive frames, we will send the corresponding operation instructions. In order to prevent false triggering, our program stipulates that if a frame does not meet the conditions, the corresponding counter will be set to 0, thus ensuring the stability of the instruction triggering process.

2.5.3. Pose mapping to robot

After obtaining the operator's posture, it needs to be mapped to the robot end effector, and the corresponding inverse kinematics is calculated to obtain the joint angle of the target robot, and then the relevant operations are completed according to the operator's control intention. In the design of this paper, we divide it into two mapping relationships according to the speed of the operator's posture change. When the robot is far away from its operating object, the operator needs to control the robot to perform large movements, and the operator's posture changes relatively quickly. Accordingly, we designed the first mapping relationship of fast motion, as shown in Figure 6, including forward, backward, up, down, left, and right based on the end effector, and the speed is 0.5 m/s. This mapping relationship ensures the moving speed of the manipulator under absolute safety and can reach the vicinity of its operating object as soon as possible. When the robot reaches the vicinity of its operating object, it needs to adjust the posture of the end effector according to the operating task, and reach the target point at a relatively slow speed for related replacement. To this end, we designed a second fine-tuning mapping relationship as shown in Figure 7, including forward and backward, up, down, left, and right, and end-effector as a reference with a speed of 0.1 m/s. Fine-tuning of the direction of the actuator as a reference, including left pick, right pick, up pick, down pick, clockwise rotation, and counterclockwise rotation. The trajectory movement and attitude adjustment of the entire desired are based on the Franka open-source library and the ROS control package. In order to realize the robot's smooth response and real-time attitude tracking, a speed closed-loop controller suitable for the Franka robot is designed, and the control frequency is 1,000 Hz, which corresponds to the real-time communication frequency allowed by the Franka mechanical alarm system.

3. Results

The interaction model is implemented on the collaborative robotic arm, Franka Emikia, and its effectiveness is verified through experiments, following the process outlined in Figure 8.

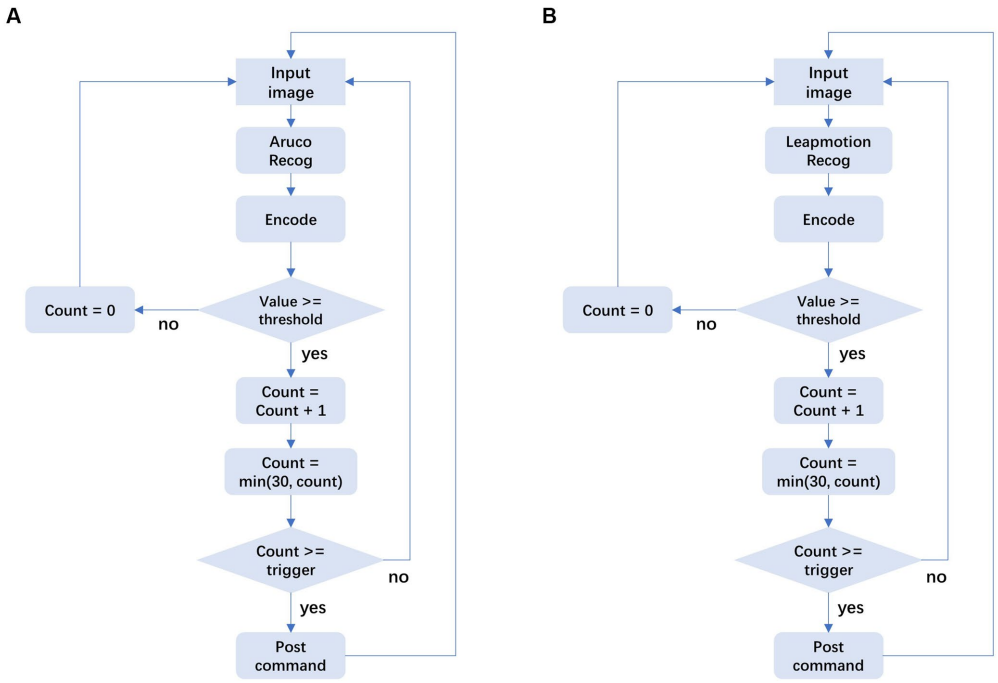


FIGURE 5
State machine design for HCI tasks.

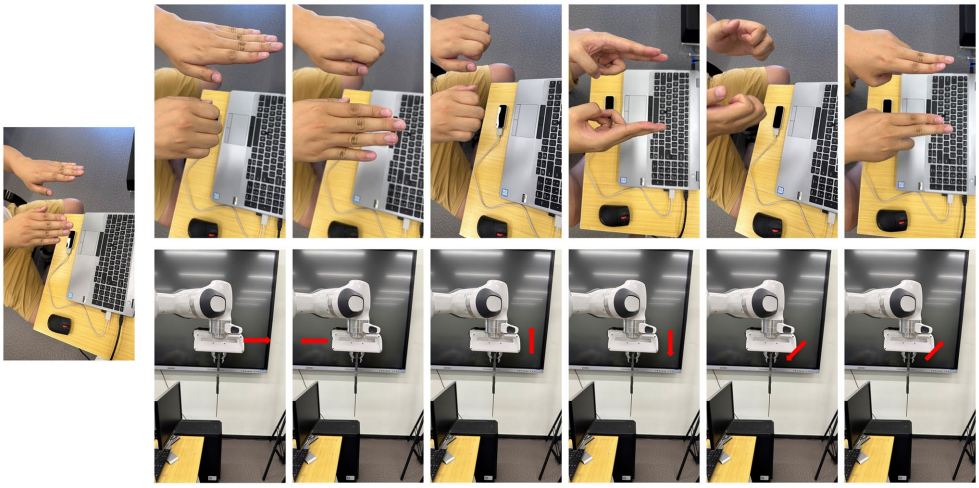


FIGURE 6
The hand gesture and corresponding action.

3.1. LeapMotion hand recognition accuracy

To integrate our gesture recognition module into a robot, we first conducted an accuracy test on our module. We recruited 10 volunteers and evaluated the success rate of six gestures on average. Each volunteer performed each gesture five times, resulting in a time range of 2 min 10 s to 3 min 15 s for the task to be completed. These findings demonstrate that our gesture-based operation method is not only user-friendly but also effective, even for beginners. The actions we designed are reasonable and efficient.

3.2. Aruco recognition accuracy

Different from gestures, as shown in Table 2, using Aruco to operate can represent 12 types of instructions for fine-tuning the position of surgical instruments. Specifically, the difference lies in

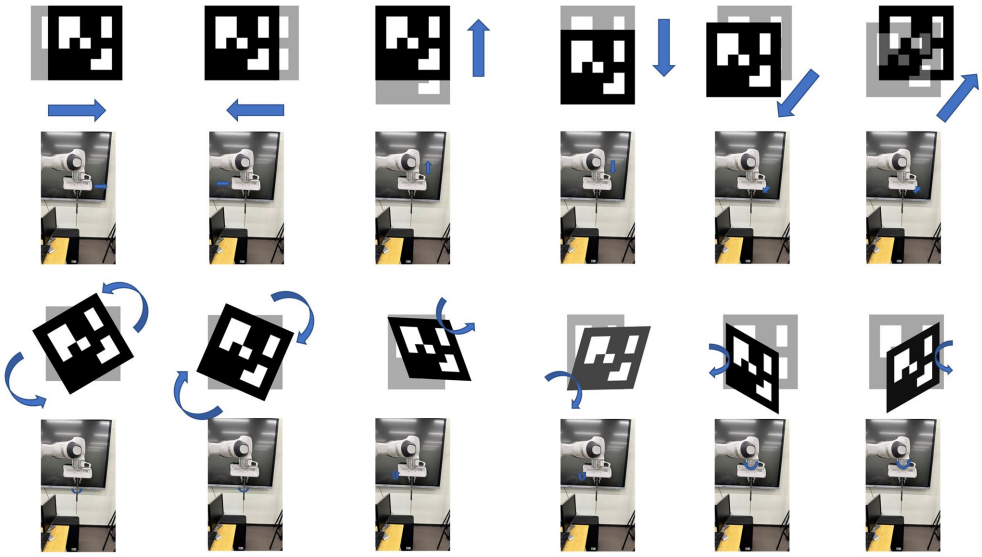


FIGURE 7
The Aruco gesture and corresponding action.

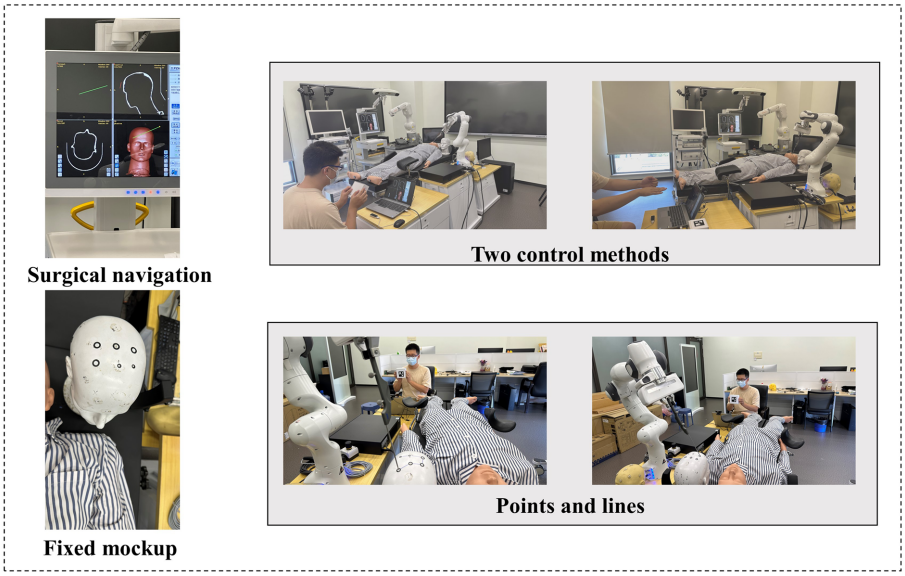


FIGURE 8
The flow of phantom experiment.

two aspects. Firstly, there are six more rotation instructions than gesture operations. Secondly, the operation using Aruco will be much more precise. For fine-tuning after adjusting the instrument using hand gestures. We had 10 volunteers attempt to trigger 12 different commands, and their success rates are as follows. In addition, the same target experiment was carried out, but this time volunteers were allowed to use a combination of both modes of operation. The result was a significant reduction in time, with an average of 25 s shaved off. The main reason for the improved operating efficiency is the simplicity and convenience of the newly designed control method.

3.3. Phantom experiment

In this experimental part, under the guidance of the surgical navigation system, we used LeapMotion and Aruco tags to control the surgical robot to hold the puncture instrument and reach the preset point. And with the help of the surgical system we can calculate the error in position (Table 3).

3.3.1. Experimental settings

In this experiment, the volunteers will move according to the predetermined route on the target on the skull model. In this process,

TABLE 2 Accuracy of Aruco recognition with predefined categories.

Category	1	2	3	4	5	6
Accuracy	97%	94%	95%	96%	97%	98%
Category	7	8	9	10	11	12
Accuracy	98%	99%	99%	96%	98%	98%

TABLE 3 Error of the needle insertion in phantom experiment.

Number of experiments	Position error (mm)
1	1.25
2	1.20
3	1.06
4	1.00
5	1.15

the volunteers will first use gestures to control the surgical instruments to move faster, and then use Aruco to control them when they reach the target point. We analyzed the planned surgical path and the actual surgical path, and obtained the error of the path.

3.3.2. Experimental result

The five experiments in Table 2 show the average alignment error of the needle tip, and the experimental data show that the average alignment error is 1.13 mm.

4. Discussion

In this research, we propose a novel pose recognition framework and integrate it into a robot, successfully completing related operations in various task sets. Compared to traditional manual operation, remote teleoperation based on human body pose estimation is highly feasible and can be seamlessly integrated into existing robots, enabling remote-operated robotics under various conditions. This is particularly evident in the medical task of puncture operation, where remote operation minimizes the risk of germ spread, reduces the time required for disinfection, and enhances the efficiency of surgery.

Although the current system design has shown good performance in various tasks, there are still some limitations to be addressed. In the trajectory tracking task, the robot did not move precisely along the set trajectory, resulting in slight trajectory fluctuations. This may have contributed to the poor network communication effect, causing the robot to receive the attitude estimation signal from the operator with a delay. As a result, the operator had to continuously adapt their posture to operate the robot, leading to an over-correction phenomenon. However, these limitations can be overcome by leveraging 5G technology and dedicated network lines. Additionally, in the robot-assisted puncture surgery task, although the robot completed the task flawlessly under remote operation by the operator, the operator lacked the tactile feedback and force feedback that are essential in a real surgical setting. As a result, the operator had to rely more heavily on the visual information provided by the navigation system throughout the procedure.

The current system design allows for remote teleoperation through the operator's posture, with the assistance of relevant visual information. Experimental results demonstrate that the proposed system can effectively control surgical instruments for both large-scale movements and fine-tuning in a non-contact scenario.

5. Conclusion

In this work, a novel human-computer interaction method based on LeapMotion and Aruco is proposed and applied in contactless robotic surgery. This approach allows for a more hygienic and cost-effective surgical experience compared to traditional methods of grasping the robotic arm with the surgeon's hand. By leveraging the guidance of surgical navigation systems, the position of surgical instruments can be accurately and quickly adjusted, streamlining the surgical process. Our proposed method has been proven effective and robust through previous experiments, and holds significant practical potential in clinical settings. Moving forward, the upgrade of sensors and optimization of algorithms can further expand the auxiliary functions of surgical robots, providing stronger support for the surgical system.

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 authors.

Author contributions

XZ: Writing – original draft, Methodology. JW: Methodology, Software, Writing – original draft. XD: Project administration, Software, Writing – original draft. SS: Writing – review & editing. XC: Funding acquisition, Writing – review & editing.

Funding

The author(s) declare financial support was received for the research, authorship, and/or publication of this article. This work was supported by the National Natural Science Foundation of China (grant no. 62076070), the Local Cooperation Project of Artificial Intelligence Medical Hospital in Xuhui District, Shanghai (grant no. 2021-008), and the Ningxia Hui Autonomous Region Key R&D Plan, Ningxia (grant no. 2023BEG02035).

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

Ali, S., Mumtaz, W., and Maqsood, A. (2023). "EEG based thought-to-text translation via deep learning" in *2023 7th International Multi-Topic ICT Conference (IMTIC)*. IEEE. 1–8.

Chang, C.-C., and Lin, C.-J. (2011). LIBSVM: a library for support vector machines. *ACM Transac. Intellig. Syst. Technol.* 2, 1–27. doi: 10.1145/1961189.1961199

Cao, H., Chen, G., Li, Z., Feng, Q., Lin, J., and Knoll, A. (2002a). Efficient grasp detection network with Gaussian-based grasp representation for robotic manipulation. *IEEE/ASME Transactions on Mechatronics*. doi: 10.1109/TMECH.2022.3224314

Cao, H., Chen, G., Li, Z., Hu, Y., and Knoll, A. (2002b). NeuroGrasp: multimodal neural network with Euler region regression for neuromorphic vision-based grasp pose estimation. *IEEE Transactions on Instrumentation and Measurement*. doi: 10.1109/TIM.2022.3179469

Chen, W., Lyu, M., Ding, X., Wang, J., and Zhang, J. (2023). Electromyography-controlled lower extremity exoskeleton to provide wearers flexibility in walking. *Biomed. Signal Process. Control* 79:104096. doi: 10.1016/j.bspc.2022.104096

Chen, C., Yu, Y., Sheng, X., Meng, J., and Zhu, X. (2023). Mapping individual motor unit activity to continuous three-DoF wrist torques: perspectives for myoelectric control. *IEEE Trans. Neural Syst. Rehabil. Eng.* 31, 1807–1815. doi: 10.1109/TNSRE.2023.3260209

Cho, Y., Lee, A., Park, J., Ko, B., and Kim, N. (2018). Enhancement of gesture recognition for contactless interface using a personalized classifier in the operating room. *Comput. Methods Prog. Biomed.* 161, 39–44. doi: 10.1016/j.cmpb.2018.04.003

Coughlan, J. M., Biggs, B., Rivière, M.-A., and Shen, H. (2020). An audio-based 3d spatial guidance ar system for blind users. *Comput. Help People Spec. Needs* 12376, 475–484. doi: 10.1007/978-3-030-58796-3_55

De Rossi, G., Minelli, M., Roin, S., Falezza, F., Sozzi, A., Ferraguti, F., et al. (2021). A first evaluation of a multi-modal learning system to control surgical assistant robots via action segmentation. *IEEE Transac. Med. Robot. Bionics* 3, 714–724. doi: 10.1109/TMRB.2021.3082210

Díaz, I., Gil, J. J., and Louredo, M. (2014). A haptic pedal for surgery assistance. *Comput. Methods Prog. Biomed.* 116, 97–104. doi: 10.1016/j.cmpb.2013.10.010

Dwivedi, A., Gorjup, G., Kwon, Y., and Liarokapis, M. (2019). "Combining electromyography and fiducial marker based tracking for intuitive telemanipulation with a robot arm hand system" in *2019 28th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*. IEEE, 1–6.

Garrido-Jurado, S., Muñoz-Salinas, R., Madrid-Cuevas, F. J., and Marín-Jiménez, M. J. (2014). Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recogn.* 47, 2280–2292. doi: 10.1016/j.patcog.2014.01.005

Gifari, M. W., Naghibi, H., Stramigioli, S., and Abayazid, M. (2019). A review on recent advances in soft surgical robots for endoscopic applications. *Int. J. Med. Robot. Comput. Assist. Surg.* 15:e2010. doi: 10.1002/rcs.2010

Jacob, M.G., Li, Y.-T., and Wachs, J.P. (2011). "A gesture driven robotic scrub nurse" in *2011 IEEE International Conference on Systems, Man, and Cybernetics*. IEEE, 2039–2044.

Kam, H.C., Yu, Y.K., and Wong, K.H. (2018). "An improvement on aruco marker for pose tracking using kalman filter" in *2018 19th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)*. IEEE, 65–69

Kang, S.-W., Jeong, J. J., Yun, J.-S., Sung, T. Y., Lee, S. C., Lee, Y. S., et al. (2009). Robot-assisted endoscopic surgery for thyroid cancer: experience with the first 100 patients. *Surg. Endosc.* 23, 2399–2406. doi: 10.1007/s00464-009-0366-x

Long, Y., Wu, J.Y., Lu, B., Jin, Y., Unberath, M., Liu, Y.-H., et al. (2021). "Relational graph learning on visual and kinematics embeddings for accurate gesture recognition in robotic surgery" in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 13346–13353.

Mutegeki, R., and Han, D.S. (2020). "A CNN-LSTM approach to human activity recognition" in *2020 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*. IEEE, 362–366.

Nagyné Elek, R., and Haidegger, T. (2019). Robot-assisted minimally invasive surgical skill assessment—manual and automated platforms. *Acta Polytech. Hungar.* 16, 141–169. doi: 10.12700/APH.16.8.2019.8.9

Ogunleye, A., and Wang, Q.-G. (2019). XGBoost model for chronic kidney disease diagnosis. *IEEE/ACM Trans. Comput. Biol. Bioinform.* 17, 2131–2140. doi: 10.1109/TCBB.2019.2911071

Ohmura, Y., Nakagawa, M., Suzuki, H., Kotani, K., and Teramoto, A. (2018). Feasibility and usefulness of a joystick-guided robotic scope holder (Soloassist) in laparoscopic surgery. *Visceral Med.* 34, 37–44. doi: 10.1159/000485524

Van Amsterdam, B., Funke, I., Edwards, E., Speidel, S., Collins, J., Sridhar, A., et al. (2022). Gesture recognition in robotic surgery with multimodal attention. *IEEE Trans. Med. Imaging* 41, 1677–1687. doi: 10.1109/TMI.2022.3147640

Wang, J., Zhang, X., Chen, X., and Song, Z. (2023). A touch-free human-robot collaborative surgical navigation robotic system based on hand gesture recognition. *Front. Neurosci.* 17:1200576. doi: 10.3389/fnins.2023.1200576

frontiers | Frontiers in Neuroscience

TYPE Original Research
PUBLISHED 05 March 2024
DOI 10.3389/fnins.2024.1297671

Check for updates

OPEN ACCESS

EDITED BY
Manning Wang,
Fudan University, China

REVIEWED BY
Jinxing Liang,
Wuhan Textile University, China
Kexue Fu,
Shandong Academy of Sciences, China

*CORRESPONDENCE
Qiang Wang
✉ caption_wang@21cn.com

RECEIVED 20 September 2023
ACCEPTED 02 February 2024
PUBLISHED 05 March 2024

CITATION
Wang X, Wang Q, Zhang L, Qu Y, Yi F, Yu J, Liu Q, Xia R, Xu Z and Tong S (2024) DCENet-based low-light image enhancement improved by spiking encoding and convLSTM. *Front. Neurosci.* 18:1297671. doi: 10.3389/fnins.2024.1297671

COPYRIGHT
© 2024 Wang, Wang, Zhang, Qu, Yi, Yu, Liu, Xia, Xu and Tong. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

DCENet-based low-light image enhancement improved by spiking encoding and convLSTM

Xinghao Wang, Qiang Wang*, Lei Zhang, Yi Qu, Fan Yi, Jiayang Yu, Qiuhan Liu, Ruicong Xia, Ziling Xu and Sirong Tong

Equipment Management and Unmanned Aerial Vehicle Engineering School, Air Force Engineering University, Xi'an, China

The direct utilization of low-light images hinders downstream visual tasks. Traditional low-light image enhancement (LLIE) methods, such as Retinex-based networks, require image pairs. A spiking-coding methodology called intensity-to-latency has been used to gradually acquire the structural characteristics of an image. convLSTM has been used to connect the features. This study introduces a simplified DCENet to achieve unsupervised LLIE as well as the spiking coding mode of a spiking neural network. It also applies the comprehensive coding features of convLSTM to improve the subjective and objective effects of LLIE. In the ablation experiment for the proposed structure, the convLSTM structure was replaced by a convolutional neural network, and the classical CBAM attention was introduced for comparison. Five objective evaluation metrics were compared with nine LLIE methods that currently exhibit strong comprehensive performance, with PSNR, SSIM, MSE, UQI, and VIFP exceeding the second place at 4.4% (0.8%), 3.9% (17.2%), 0% (15%), 0.1% (0.2%), and 4.3% (0.9%) on the LOL and SCIE datasets. Further experiments of the user study in five non-reference datasets were conducted to subjectively evaluate the effects depicted in the images. These experiments verified the remarkable performance of the proposed method.

KEYWORDS

intensity-to-latency, spiking encoding, low-light enhancement, unpaired image, deep learning

1 Introduction

The lack of illumination leads to the loss of image information, which severely affects the execution of visual tasks, e.g., face recognition, object detection, dataset preparation, and autonomous driving (Li J. et al., 2021; Liu et al., 2021; Tang et al., 2022; Guo et al., 2023). Capturing images in low-light conditions poses a challenge owing to the limited aperture size, demand for instantaneous processing, and limited memory resources. To mitigate the issues of structuring and the high expense of research and development associated with hardware, refining images in low-light settings through minimalistic software algorithms aligns better with predictable requirements.

In low-light image enhancement (LLIE), the first effective methods were based on histogram equalization, the Retinex model, gamma transform, and fusion. Fusion-based methods achieve better performance in terms of image indicators, such as brightness and color, through exposure-splicing fusion methods. This method is typically synthesized by collecting images under different exposure conditions (Wang et al., 2016). Another method

fuses the illumination map of night and day to enhance the image (Rao et al., 2010); however, such processing generally renders a poor visual effect.

The method based on the Retinex model divides the low illumination image into reflection and illumination components or adds a noise component by constructing a suboptimal problem. The estimated reflection component is considered the result of enhancement. Previous attempts to improve Retinex replaced the logarithmic solution with a typical enhanced Lagrange solver to enhance the image with a long image processing time. However, the variational optimization algorithm has a high computational cost. Moreover, it introduces unnecessary pseudo-details in the image.

The adaptive GAMMA transform can improve an image's contrast; however, most algorithms of this class still cause local overexposure or underexposure in the enhanced result. As most images are captured in non-uniform lighting conditions, Chen et al. (2022) proposed a naturalness- and information-preserving method for processing them. The MEMBHE algorithm (Dar and Mittal, 2020) improved the functionality of the transform through histogram equalization after multiple exposure smoothing. Nevertheless, it overconsumes memory and requires arduous incremental updates.

Several methods for achieving LLIE with deep learning (DL) have been researched. Among them, supervised learning, a mature and informative DL method typically constructed by an end-to-end network, was the first to be applied to an LLIE field. Low-light net (LLNet) (Lore et al., 2017) was the first end-to-end LLIE network established by constructing a deep auto-encoder structure. MBLLEN (Lv et al., 2018) uses three subnetworks to extract rich image features of different levels and introduces a regional loss function into the network loss function to employ different loss weights for high- and low-light regions. In the same vein, Li et al. (2021) determined that enhancing the low-frequency layer of a low-light image with noise was easier than directly enhancing the whole image. Progressive recursive networks (Cai et al., 2018) were used to perform staging, which is a more efficient method for preserving image details and removing noise. In that method, each subnetwork could better achieve its own function, which was eventually enhanced by gradually improving the quality of the image.

Ke et al. (2020) established an SCIE multiexposure dataset (Ke et al., 2020) consisting of low-contrast images with different exposure levels and their corresponding high-quality reference images. Furthermore, they introduced the high- and low-frequency components of images as prediction targets. A double-exposure fusion algorithm (Ying et al., 2017) was proposed to design the weight matrix of image fusion using an illuminance-estimation technique. Then, a camera response model was introduced to synthesize the multiexposure images. Low- and high-exposure images can also be used to estimate the perceptual gain, signal strength, signal structure, and mean intensity. Perceptual gain suits an underexposed image. The feature fusion and recalibration module (FFRM) (Singh et al., 2024) was proposed to recalibrate and merge the features to provide an enhanced output image. Intrinsic image decomposition (Zhang and Ma, 2023) can be applied to the fusion of multiexposure to generate HDR images.

Retinex was combined with DL for enhanced performance (Chen et al., 2018; Zhang Y. et al., 2019; Tang et al., 2023). The attention mechanism was combined with the Retinex model to construct DL networks for enhancement (Chen et al., 2022). A decomposition network (Liu et al., 2023) was developed with a self-supervised fine-tuning strategy that achieved promising performance without manual hyperparameter tuning. Different sensitivities relate to different regions. The low-rank regularized Retinex model (Bao et al., 2022) can represent the image as low-rank decomposition, preserve the image details and high-frequency information, and improve the visual quality of the image. A plug-and-play framework for image enhancement and noise removal based on the Retinex theory (Wu et al., 2023) was introduced. Inspired by guided filtering and using synthetic data for network training, Li et al. (2018) designed a lightweight network architecture based on the Retinex theory. By including the unsettling V channel image component in the HSV color space, the component was converted to a reflection component using a DL network (Jiang Z. et al., 2021). Owing to their significant worth, their Retinex and DL-based methods were applied in image dehazing and underwater image enhancement (Xu et al., 2022; Shen et al., 2023).

The development of LLIE in DL is not limited. Creative thinking models, such as those based on unsupervised learning, represented by the unsupervised learning method (Zhu et al., 2020; Li et al., 2021), generative network architecture (Jiang Z. et al., 2021), and normalizing flow (Wang et al., 2022), show the immense research potential of LLIE. The strategy network learns the local exposure sequentially using reinforcement learning for a segmented subimage (Rong et al., 2018). In the generated adversarial network architecture, global–local discriminators (Jiang Z. et al., 2021) were used to ensure that the enhanced results resemble real normal light images. With the strong capability of image generation, diffusion models were applied to LLIE. For example, the pyramid diffusion model (Zhou et al., 2023) was constructed to solve the RGB shift. Moreover, the inference speed of the diffusion model was accelerated. As a scientific structure for image feature extraction, transformers have become some of the most prevalent network structures in vision processing. The regional distributions have been effectively managed, and the histogram loss has been designed in a stage transformer-guided network (Jiang et al., 2023). Half-wavelet attention block and hierarchical M-Net were utilized to improve computation consumption and reserve context information, aided by the DAU block and discrete wavelet transformation (Fan et al., 2022).

Spiking neural networks (SNNs) are frequently employed in numerous pixel-level classification tasks (Martinez-Seras et al., 2023), such as object detection (Zhang et al., 2023b), image segmentation (Zhang et al., 2023a), and anomaly detection (Yusob et al., 2018). Research centered on SNNs includes methods for neural network learning, data coding, and hardware platforms. The learning approaches for SNNs can be divided into supervised and unsupervised learning, which are represented by spike-timing-dependent plasticity (STDP). Spiking encoding, which involves utilizing discrete pulsed signals to convey information, is a method of signal transmission. Neuroscience computing has access to specialized offline or online application-specific integrated circuit platforms, as well as neuromorphic computing cores that can support various learning rules and neuronal models. Nonetheless, spiking neural network research continues to confront significant barriers. The training of the transformed SNN still relies on the backpropagation algorithm of artificial neural networks (ANNs). As

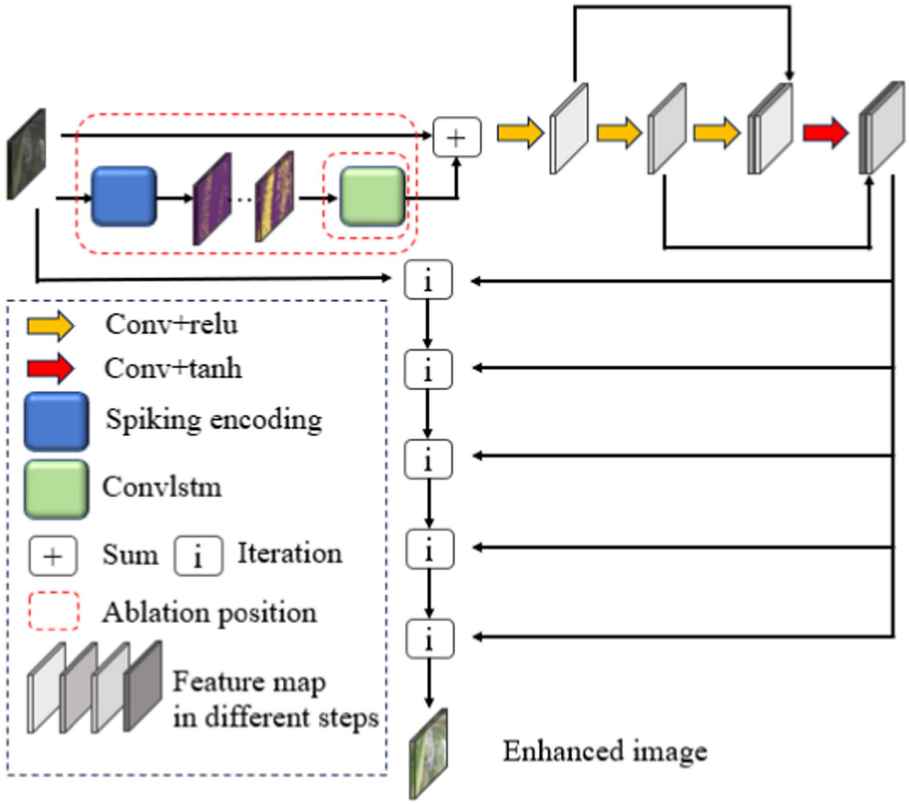


FIGURE 1
DCENet structure with spiking encoding and convLSTM.

the performance difference between the SNN and the core ANN is small, the former cannot provide significant advantages. Moreover, generative tasks, such as LLIE, image patching, multimodal image generation, and network deployment, present significant challenges. As a new neural network structure, the SNN's internal algorithm can be implemented in LLIE.

The main contributions of this study are as follows: (1) According to the progressive output results with the specified number concluding the embodiment of the image structure characteristics, the application of the SNN in a spiking encoding method for LLIE has distinct advantages in extracting structural features from images (the intensity-to-latency encoding outputs multiple feature maps with structure and specified steps); and (2) a convLSTM structure that can better absorb the features from multiple feature maps. Based on unlabeled, unsupervised, and unpaired image training via simplified DCENet, the proposed structure is improved by spiking encoding and the convLSTM module. The research introduces spiking encoding, which concludes the image's backbone information to describe the hierarchical information. The rest of the paper follows this structure: Section 2 describes the proposed enhancement method. Section 3 describes the user study and ablation experiment carried out in the study and compares the performance of the proposed method with the state-of-the-art network structure based on seven objective indicators. Section 4 concludes the study and discusses the potential applications.

2 Proposed method

2.1 DCENet structure

The DCENet structure, as the primary structure used for unsupervised enhancement, divides the LLIE into a high-order iterative process, i.e., the input dark light image is finally enhanced through several iterations of the same operation. Figure 1 depicts the overall enhancement process and part of the ablation study, which can also have a description in literal form. The input passes through the spiking encoding module and ConvLSTM described in subheadings 2.2 and 2.3, respectively, and then through the convolution module containing skip links. The sum module in Figure 1 means a direct overlay between the ConvLSTM's final output and the input dark light images. The resulting features select the feature graph of a certain channel in order and combine the matrix of the same size in the length and width scale of the output and input images with the initial input tensor according to Equation (1). The matrix is used as the input for the next iteration, and the feature graph of the next channel is selected as needed for the next iteration operation.

Compared with the mathematical relationship represented by the previous gamma transform, the DCENet structure changes the training coefficient of the second term of the right-hand side of Equation (1) into a training coefficient matrix with the same dimensions as those of the

input image. This can restrain the problem of over-enhancement or under-enhancement of the image to a certain extent. Finally, the normal brightness area in the image is maintained, and the low illumination area is restored. A is the output of the network, which can be divided into several pieces denoted by A_n. Based on the number of iterations n, the final output enhancement result is x_n.

$$x_n = x_{n-1} + A_n\left(x_{n-1}^2 - x_{n-1}\right) \tag{1}$$

2.2 Spiking encoding method

In this study, spiking encoding from the overall DCENet structure equals the intensity-to-latency transform (Mozafari and Ganjtabesh, 2019), as illustrated in Figure 2. First, the intensity-to-latency transform requires an initial parameter, i.e., time step S. Then, the grayscale image, which corresponds to a matrix with shape (H,W), is reshaped to a vector with $H\times W$ dimensions by $R(\cdot)$ as illustrated in Equation (2). We named this original vector V. For the next step, the vector was arranged in descending order. This procedure generated two vectors with the same dimensions: the first vector is the descending order vector V_d, while the second one is the index vector V_i, corresponding to the index in V and this relation is represented by Equation (3).

$$V\left[1,H*W\right] = R\left(I\left[H,W\right]\right) \tag{2}$$

$$V_d, V_i = D\left(V\right) \tag{3}$$

where K is the number of non-zero elements in an original vector V. The split parameter θ is set in Equation (4). V_d and V_i are split into small vector pieces; θ decides the shape of these pieces. The small vector piece returns to the dimension $H\times W$, which is called the spiking encoding vector T_m in Equation (5), with the complementary element filled with 0. The small label m ranges from 0 to the time step S. The start time step T_0 is composed of the value in the first split piece, and the value in T_0 is rearranged to the original position in V according to V_i. The second time step T_1, which is based on T_0, adds the second small piece, and the value in the second piece is adjusted to the original position in the same way. Thus, the intensity-to-latency transformation is complete. The sequence of outputs T_m is reshaped to similar dimensions as those of the input image, which are denoted by E_m, with the dimensions of (S,H,W). This procedure is formulated as Equation (6). $R^*(\cdot)$ means the reverse calculation manipulation of $R(\cdot)$. (Considering the length of the paper, its time step in the figure is set to 6.)

$$\theta = \begin{cases} \dfrac{K}{S}, \dfrac{K}{S} > 2 \\ 2, \dfrac{K}{S} \le 2 \end{cases} \tag{4}$$

$$T_m = split\left(V_d, \theta\right) \tag{5}$$

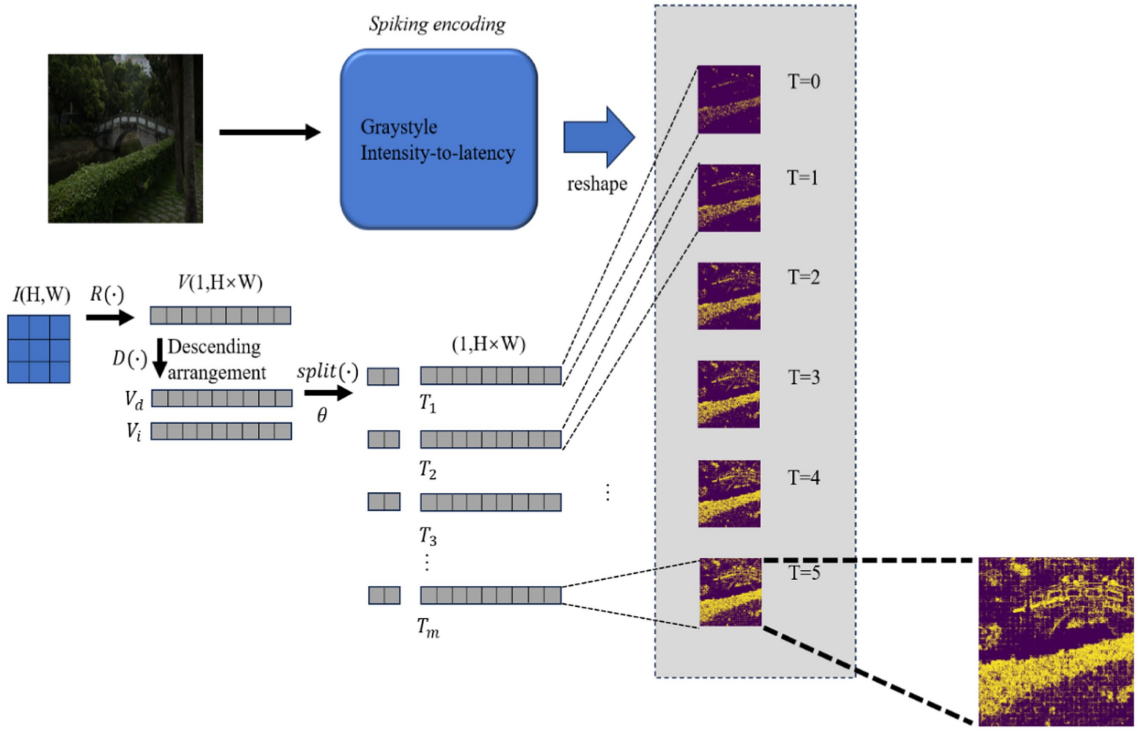


FIGURE 2
Intensity-to-latency encoding data-flow schematic diagram.

$$E_m = R^*\left(T_m, V_i\right) \tag{5}$$

2.3 ConvLSTM

The features extracted by the intensity-to-latency transform have certain similarities and differences. These features will constitute an image sequence with fluent features. The convLSTM structure is applicable in this scenario. ConvLSTM is proposed for precipitation nowcasting (Shi et al., 2015), the backbone of which is the recurrent neural network (RNN) for spatiotemporal prediction with convolutional structures. This design is convenient for video and image sequence-related tasks. ConvLSTM is similar to LSTM, which is also called FC-LSTM, and its block structure is illustrated in Figure 3.

The convLSTM computation method is based on LSTM's gate relationship. The distinction between its different layers is the input and output dimensions. The core of convLSTM is the convLSTM cell, which represents one convLSTM layer. ConvLSTM cell is an RNN-like structure; therefore, a specific hidden-layer parameter called *hidden state* is required. In every convLSTM layer, the hidden state is initialized with a zero element of dimensions $(C1,H,W)$. One of the input sequence time step tensors I, which is another input of convLSTM and the outputs from spiking encoding, with dimensions (C,H,W), were concatenated with δ. It outputs a combined tensor with dimensions $(C+C1,H,W)$, corresponding to the concatenate calculation represented by *concat* in Equation (6), which needs two different variables. The convLSTM cell accepts this combined tensor and outputs the tensor with dimensions $(4\times C1,H,W)$. The outputs were divided into four tensors with dimensions $(C1,H,W)$ for the outputs of different gates: input, forget, and output gates, and a new δ for the subsequent layer and input time step. This divided single step is represented by the *split*. The calculation procedure is summarized in Equations 6–10 and Figure 2.

$$mid_1 = conv\left(concat\left(\delta_{t-1}, I_{t-1}\right), \dim = 1\right) \tag{6}$$

$$c_i, c_f, c_o, c_g = split\left(mid_1, hidden, \dim = 1\right) \tag{7}$$

$$a_t = S\left(c_a\right), a = i, f, o; g_t = A\left(c_g\right) \tag{8}$$

$$\eta_t = f_t \circ \eta_{t-1} + i_t \circ g_t \tag{9}$$

$$\delta_t = o_t \circ A\left(\eta_t\right) \tag{10}$$

In the convLSTM structure, δ and η, which are output by one convLSTM cell, pass to the next cell at a certain time. This time

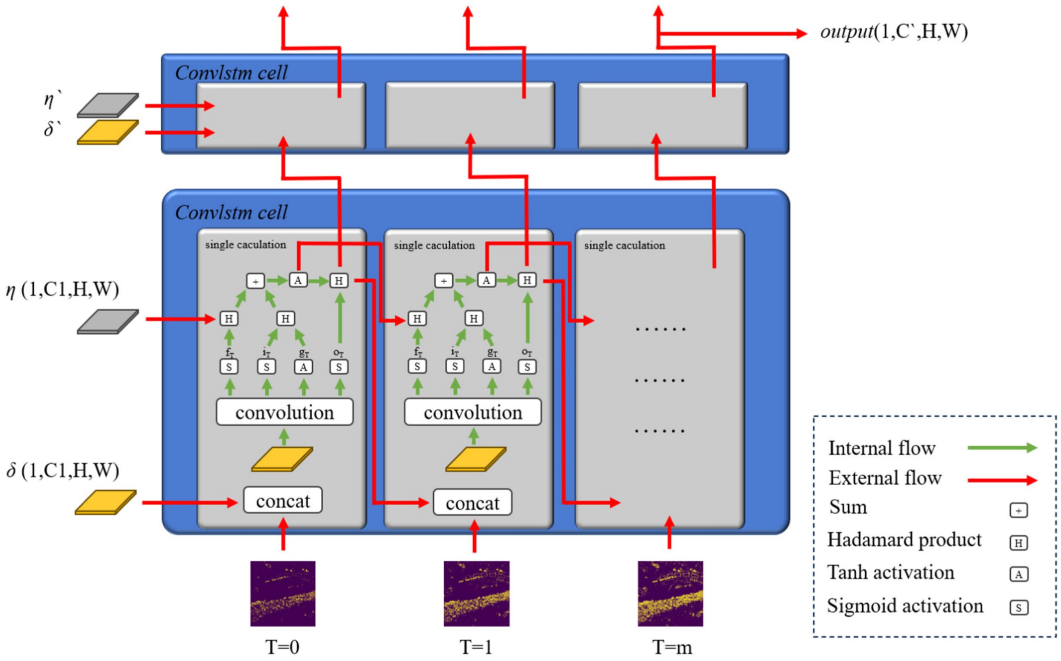


FIGURE 3
ConvLSTM block data-flow schematic diagram.

corresponds to the next time step in the same layer. This RNN-like network structure will preserve the main features from the previous time step image feature. The δ is also output to the convLSTM cell, combined with the new hidden states δ` and $\eta$` in the next layer in the same time step. The overall output of the convLSTM module is the tensor with dimensions (1,C`,H,W), which is labeled *output* in Figure 2. The time step dimension is eliminated with the convLSTM module and $S(\cdot)$, sigmoid activation function, and $A(\cdot)$, tanh activation function.

2.4 The loss items of the DCENet structure

Four loss items, namely spatial consistency loss, color constancy loss, exposure control loss, and illumination smoothness loss, were considered for the convergence of the network. The loss function used by the network is represented by Equation (12). The spatial consistency loss item was calculated by Equation (13). The purpose of setting the spatial consistency loss item was to maintain the difference between the original image and the adjacent area of a pixel in the enhanced image as small as possible. The $\bar{X}$ represents the tensor X after channel averaging and average pooling for every 4×4 area. K is the number of pixels after average pooling in one feature map channel. These pixels are separated by a distance of 1, which corresponds to a point assemble called $R(i)$. This difference logic will enhance the pixel neighborhood within the same spatial structure. By introducing the sum item, the pixel neighborhood consistency can be promoted to the spatial position consistency of the whole image. The setting of this loss item will maintain the spatial consistency of the image before and after enhancement.

$$L = L_{spa} + L_{\exp} + W_{col}L_{col} + W_{tv}L_{tv} \tag{12}$$

$$L_{spa} = \frac{1}{K}\sum_{i=1}^{K}\sum_{j\in \mathrm{R}(i)}\left(\left|\bar{E}_i - \bar{E}_j\right| - \left|\bar{O}_i - \bar{O}_j\right|\right)^2 \tag{13}$$

To ensure the overall improvement in brightness, the exposure loss was established as Equation (14). The average value of pixels in the pixel block corresponding to the gray-level image of the output-enhanced image should meet certain size requirements, and the reference average value was set to 0.7. $\bar{E}_m{}'$ represents the m^{th} pixel value after image channel mean processing and pooling for the enhancement of the final result. The pooling operation may have different parameters. Hence, quotes were added to distinguish it from the spatial consistency loss term. The number of pixels after pooling was set to M.

$$L_{\exp} = \frac{1}{M}\sum_{m=1}^{M}\left(\bar{E}_m{}' - 0.7\right)^2 \tag{14}$$

The value of one color channel of the image should not significantly exceed that of the other channels. Hence, the loss of color was set to a constant value represented by L_{col} represented by Equation (15). This loss should go through all pairings in the three color channels. To better satisfy this condition, the spatial average of the enhanced image is calculated, and a three-channel difference loss term was constructed to satisfy this conclusion. ($c1$,$c2$) traverses all pairwise combinations in the three RGB color channels. $\bar{E}^{c1}$ and $\bar{E}^{c2}$ represent the enhancement result's mean value of one RGB channel.

$$L_{col} = \sum_{\forall(c1,c2)\in c}\left(\bar{E}^{c1} - \bar{E}^{c2}\right)^2, c = \left\{(R,G),(G,B),(B,R)\right\} \tag{15}$$

Different from the final enhanced image result, A is the network output. In Equation (16), N, which equals to $H\times W$ dimensioned by A_n, represents the shape of the input. d represents the gradient of A; for instance, A_{iy}^{d} relates to the longitudinal gradient of A in the i^{th} iteration. The illumination smooth loss L_{illu} was established here.

$$L_{illu} = \frac{1}{N}\sum_{i=0}^{n}\sum_{d}\left(A_{ix}^{d} + A_{iy}^{d}\right) \tag{16}$$

Considering that the brightness change between adjacent pixels is not significant, the gradient term was introduced to the network output to ensure a monotonic relationship between adjacent pixels. No texture was introduced in the network output. Instead, it was introduced from the original image through the relationship. As a common loss term for LLIE, the estimation of the illumination smooth loss term is similar to the calculation of light smoothness loss in Zhang Y. et al. (2019).

3 Experiments and evaluation

3.1 Experimental setup

The hardware part adopts an 11 GB GTX 1080 Ti. The software is PyTorch framework 1.10.0 v. The spiking encoding convLSTM-augmented LLIE model was constructed using the Python 3.7 library of PyTorch and trained using datasets consisting of unpaired images. The optimization process of the proposed network employed the ADAM optimizer with default parameters and a fixed learning rate of 1×10^{-4}. The weights W_{col} and W_{tv} were set to 0.5 and 20, respectively. These parameters remained constant in all experiments.

The datasets, i.e., LLIE fields, were divided into referenced and unreferenced image datasets. Typical referenced image datasets include LOL, SCIE, and MIT-Adobe FiveK, while unreferenced datasets include VV, NPE, and LIME. The LOL dataset has a considerably different degree of underexposure from the rest, which is suitable for the comparison of the overall performance of LLIE algorithms. The SCIE dataset is a multiexposure image sequence dataset with rich illumination information, which is highly suitable for algorithm debugging. Hence, we selected the LOL and SCIE datasets for the experiments. We retained the original training and test dataset distributions for the LOL dataset. In each image sequence of the SCIE dataset, the first image was chosen as the low-light image to be enhanced, whereas the most suitable one was chosen as the high-light reference image among the third, fourth, and fifth images. We used a user study to evaluate five common unreferenced datasets, namely VV, NPE, LIME, DICM, and MEF. We hypothesized that the key performance of LLIE should lie in the size of the space occupied by its running process, which can influence the integration of related

TABLE 1 A calculation of the objective image evaluation index.

Image evaluation index	Mathematical expression	Range	Trend for better
PSNR	$10\log_{10}\frac{MAX(H)^2}{MSE}$	[0,+∞)	To positive infinity
SSIM	$\frac{1}{3}\sum_{channel}\frac{\left[2\bar{R}\bar{H}+K1\times MAX(H)^2\right]+\left[2\hat{R}\hat{H}+K2\times MAX(H)^2\right]}{\left[\bar{R}^2+\bar{H}^2+K1\times MAX(H)^2\right]\left[\hat{R}^2+\hat{H}^2+K2\times MAX(H)^2\right]}$	[0,1]	Closer to 1
MSE	$\frac{1}{H\times W}\sum_{(a,b)\in I}(R_{ab}-H_{ab})$	[0,+∞)	Smaller
UQI	$\frac{4\bar{R}\bar{H}\hat{R}\hat{H}}{\left(\bar{R}^2+\bar{H}^2\right)\left(\hat{R}^2+\hat{H}^2\right)}$	[−1,1]	Closer to 1
VIF	$\sum_{gauss}\sum_{(a,b)\in I}\log_{10}\left[1+g^2\frac{H^2-\bar{H}^2}{svsq+2}\right]$	[0, +∞]	Closer to 1

TABLE 2 The performance comparison of the ablation study for different substructures (red bold for the best, black bold for the second best).

Detection methods	datasets	PSNR	SSIM	MSE	UQI	VIF
SimpleDCE	LOL	17.1200	0.5969	0.0309	**0.7694**	0.8564
	SCIE	**15.4942**	**0.6036**	**0.0337**	**0.8047**	0.5617
Spiking + CNN	LOL	**17.2140**	**0.5999**	**0.0291**	0.7689	**0.9394**
	SCIE	15.1792	0.6007	0.0353	0.7945	**0.5856**
Proposed (Spiking+convLSTM)	LOL	**18.3374**	**0.5974**	**0.0227**	**0.8369**	**1.1019**
	SCIE	**16.7519**	**0.7289**	**0.0221**	**0. 8,413**	0.5358
Proposed + CBAM	LOL	16.9693	0.5968	0.0315	0.7653	0.8539
	SCIE	15.3300	0.6026	0.0347	0.8010	**0.5635**

tiny systems. This feature represents the application's ability to integrate with other functions and algorithms of the testing process and of the model itself.

There are five assessment indices for image objective evaluation, namely peak signal-to-noise ratio (PSNR), structural similarity index measure (SSIM), mean square error (MSE), universal image quality index (UQI) (Wang and Bovik, 2002), and visual information fidelity (VIF) (Sheikh and Bovik, 2006). The calculated evaluation indices are listed in Table 1. In this table, *h,r* corresponds to *H*'s and *R*'s results of the Laplace filter. The *nonzero* (*) function realizes 0 to 1. $\sum_{\text{gauss}}$ represents the summation of the results of different Gaussian filter parameters. Gaussian filtering was used for *H* and *R*. The number *n* depicts the Gaussian filtering times. For instance, $\widehat{R^2_{n+1}}$ indicates that the square calculation was performed first, followed by Gaussian filtering. $\widehat{R_{n+1}}^2$ indicates that Gaussian filtering was performed first, followed by square calculation. $\bar{x}$ represents the uniform filter for *x*. PSNR and MSE are non-negative. Test images with reference images were calculated to get the PSNR value. The larger the PSNR, the less the image noise and the better the image quality, and SSIM reflects structural similarity. It is typically used to measure whether the image backbone of the image recovered by the LLIE has also been restored. The SSIM ranges from 0 to 1; only when two sets of identical image data converge will the SSIM reach 1. The indicator, UQI, reflects the measure of the degree of linear correlation, the closeness of the mean luminance, and the similarity of contrast between the enhanced result and the reference image. VIF combines a natural image statistical model, an image distortion model, and a human vision system model. Compared to the PSNR, SSIM, and other indicators, because the numerator of the VIF index calculation formula is the information fidelity criterion (IFC), VIF has a higher consistency with subjective vision. The higher its value, the better the image quality.

3.2 Ablation study

As the proposed method is based on the DCENet structure, the change in the enhancement properties after introducing the spiking+convLSTM structure must be considered. The study demonstrates the influence of each loss term of the loss function on the enhancement results under different loss combinations. In the ablation experiment, different loss combinations were used for retraining. The necessity of each loss item was retested using the proposed DCENet-based method to prevent the negative effects of spiking encoding and convLSTM.

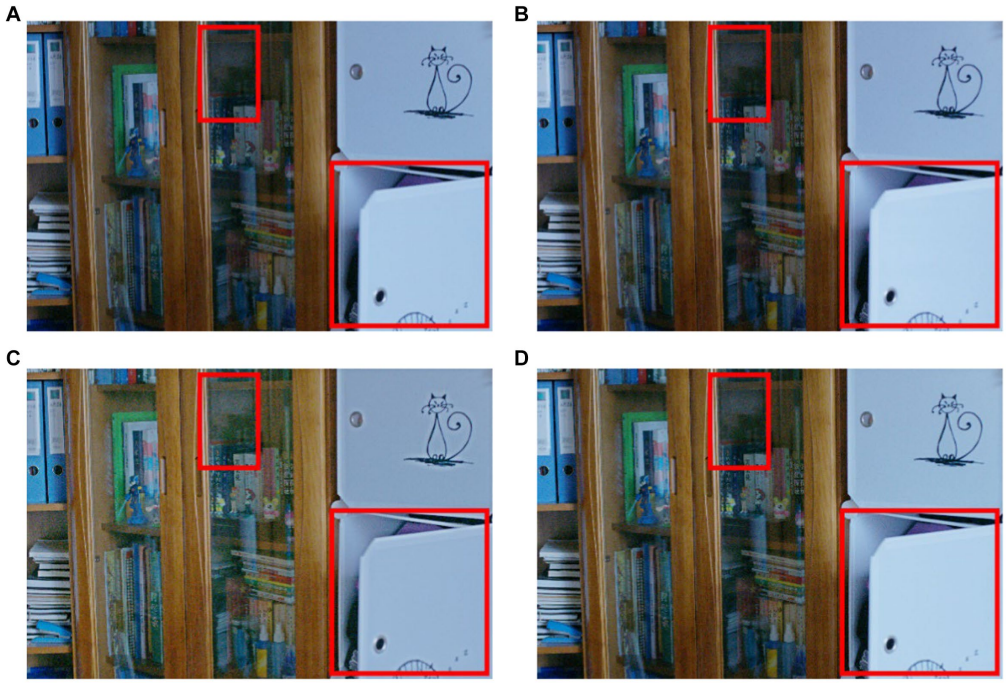


FIGURE 4
Ablation study by three substructures (obvious areas for specific differences) **(A)** CBAM **(B)** spikingCNN result **(C)** the simple dce structure **(D)** the result obtained by proposed structure.

TABLE 3 **Different loss function assemblies of ablation study in the LOL and SCIE datasets (red bold for the best, black bold for the second best).**

Lcol	Ltv	Lspa	Lexp	Datasets	PSNR	SSIM	MSE	UQI	VIF
☑	☑			LOL	10.7278	0.3607	0.0940	0.3944	0.2599
				SCIE	9.7752	0.3467	0.1326	0.2811	0.2252
☑		☑		LOL	10.8758	0.3745	0.0908	0.4175	0.2800
				SCIE	10.2279	0.3660	0.1241	0.3243	0.2342
☑			☑	LOL	12.4312	0.4727	0.0670	0.7956	**0.5626**
				SCIE	13.1966	0.4463	0.0602	0.7650	0.7316
		☑	☑	LOL	11.0469	0.4633	0.0844	0.7478	0.4725
				SCIE	11.3607	0.4384	0.0793	0.7144	0.6979
	☑		☑	LOL	8.0262	0.3444	0.1657	0.5044	0.4472
				SCIE	8.6462	0.3565	0.1435	0.4935	**0.8410**
	☑	☑		LOL	10.9274	0.3682	0.0909	0.4029	0.2717
				SCIE	9.8145	0.3491	0.1317	0.2844	0.2273
☑		☑	☑	LOL	**14.2867**	0.5006	0.0470	**0.8284**	0.4368
				SCIE	**14.3752**	0.4610	0.0437	0.7912	**0.8286**
	☑	☑	☑	LOL	8.1473	0.3512	0.1607	0.5078	0.4504
				SCIE	8.6934	0.3591	0.1417	0.4963	0.7412
☑	☑	☑		LOL	10.7671	0.3631	0.0932	0.3984	0.2608
				SCIE	9.8145	0.3491	0.1317	0.2844	0.2273
☑	☑		☑	LOL	13.0447	**0.5859**	**0.0693**	0.8047	0.1875
				SCIE	14.9476	**0.5691**	**0.0335**	**0.8125**	0.7470
☑	☑	☑	☑	LOL	**18.3374**	**0.5974**	**0.0227**	**0.8369**	**1.1019**
				SCIE	**16.7519**	**0.7289**	**0.0221**	**0.8413**	0.5358

Another ablation experiment should also be considered, which focuses on spiking encoding and convLSTM itself. Thus, three ablation study experiments, whose network is made up of the only light DCENet structure, the structure with the CBAM attention mechanism, or the CNN structure that replaces the convLSTM, have been considered for comprehensively verifying the proposed structure's necessity. In the two ablation studies, the training parameter did not change. The SCIE dataset was applied for specific calculations.

Only light DCENet structure: Without the proposed spiking encoding and convLSTM structure, the enhancement is only realized by DCENet.

Structure with CBAM attention mechanism: Based on the only-light DCENet structure, the CBAM attention mechanism is set after the first layer.

CNN structure that replaces convLSTM: The enhancement was running using a CNN structure instead of convLSTM. The dimensions of the spiking encoding image sequence were trimmed, and the image sequence was superimposed to form a feature map.

The ablation study about the importance of spiking encoding and convLSTM is summarized in Table 2. Compared with the basic DCENet structure, spiking combined with the CNN structure revealed that the integration of spiking encoding alone improved the

FIGURE 5 (Continued)

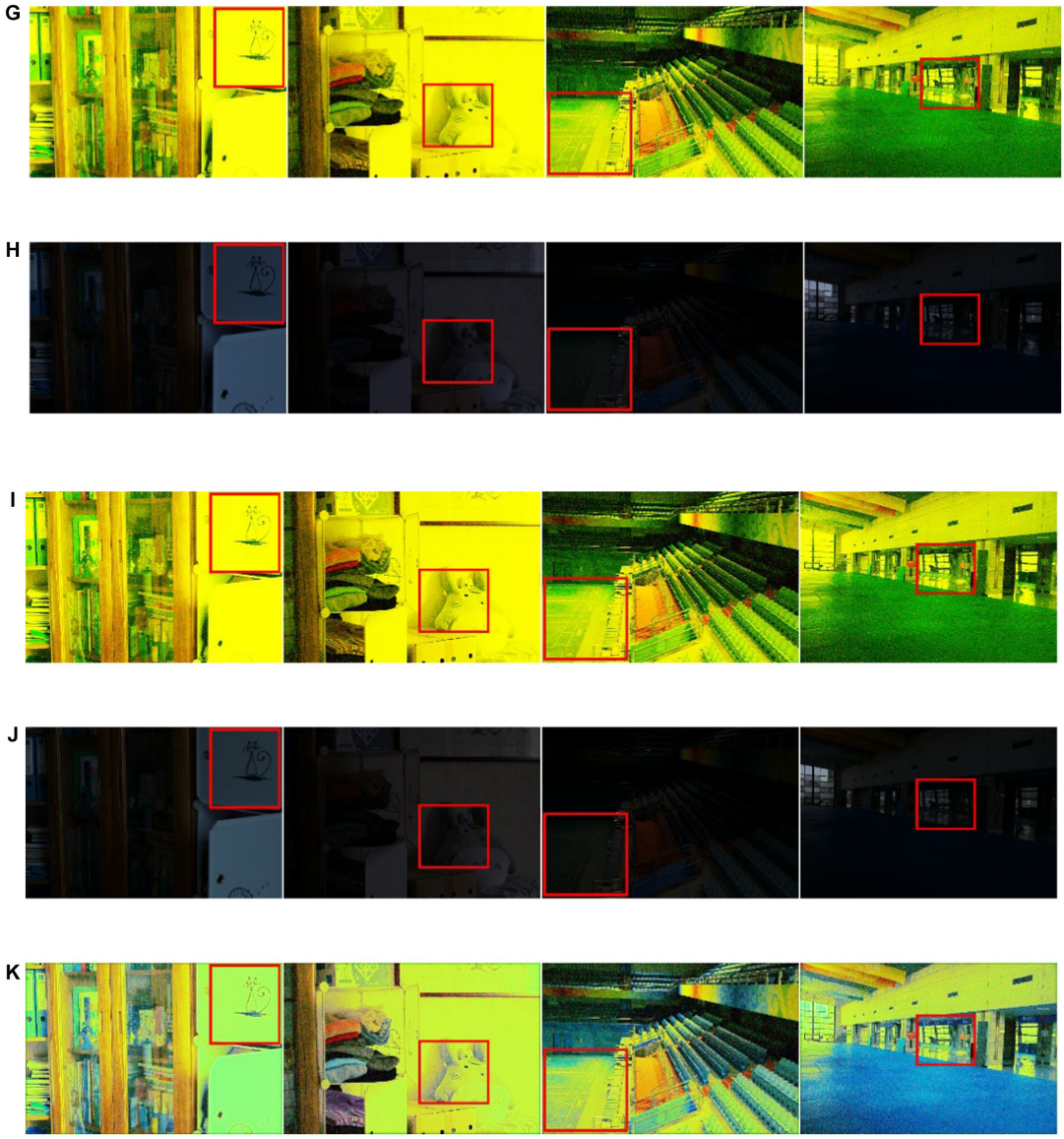


FIGURE 5
The loss function assembly in LOLdataset **(A)** original low-light image **(B)** color constancy loss and exposure loss **(C)** illumination smooth loss, exposure loss **(D)** color constancy loss, spaital consistency loss, **(E)** spaital consistency loss, exposure loss, **(F)** illumination smooth loss, color constancy loss, **(G)** illumination smooth loss, spaital consistency loss **(H)** illumination loss, exposure loss and color constancy loss **(I)** color constancy loss, exposure loss and spatial consistency loss **(J)** illumination loss, color constancy loss and spatial consistency loss **(K)** illumination loss, exposure loss and spatial consistency loss.

performance. Specifically, the VIF parameter showed significant increments of 9.7 and 4.3% on the LOL and SCIE datasets, respectively. However, when considering other indicators, the objective evaluations of the LOL and SCIE datasets demonstrated a contrary trend. This suggests that the combination of spiking and CNN methods may not be beneficial for enhancing model generalization stability. To improve upon this, the study adopted the classic CBAM attention mechanism as a representative approach for introducing attention mechanisms. Data suggest that incorporating attention mechanisms alone reduced the number of essential evaluation criteria, such as PSNR, SSIM, and UQI. Additionally, the combination of convLSTM and spiking encoding not only elevated the evaluation index on the SCIE dataset but also surpassed the effect of the convolutional network combination. In addition, we identified only minor differences in the subjective effects of the methods under the ablation experiments. These effects are presented in Figure 4.

Ablation experiments assess the impact of different loss function terms on the image enhancement quality. The proposed approach employed four loss function terms. Their pairwise and three-way combinations and the corresponding image evaluation index parameters are listed in Table 3. Of the six paired combination parameters, color constant loss and exposure loss substantially enhanced image quality, followed by spatial consistency loss and exposure loss. Consequently, we infer that exposure loss is the most crucial loss item, followed by color constant loss and spatial consistency loss, which exert the least impact on light smoothness loss.

TABLE 4 The performance comparison of the ablation study in the SICE dataset (red bold for the best, black bold for the second best).

Detection methods	PSNR	SSIM	MSE	UQI	VIF
RRDNet	12.4675	0.5469	0.0594	0.5713	0.3487
zero-DCE	16.0794	**0.6618**	0.0365	0.8208	**0.6829**
DRBN	15.8745	0.4667	0.0290	0.8008	0.2575
EXCNet	16.0427	0.6006	0.0334	0.7626	0.3675
LightenNet	11.2150	0.3216	0.0810	0.7228	0.4736
DSLR	15.1002	0.5996	0.0318	0.7839	0.3348
BIMEF	15.7917	0.5904	0.0326	0.7936	0.3678
LLFLOW	15.0300	0.5830	0.0364	0.7918	0.4452
Enlighten anything	14.9228	0.6193	0.0443	**0.8393**	0.4487
Proposed (Spiking+convLSTM)	**16.7519**	**0.7289**	**0.0221**	**0. 8,413**	0.5358
EnlightenGAN	**16.6135**	0.6219	**0.0260**	0.8306	0.5310
BFSA	12.0203	0.4419	0.0721	0.5401	0.5002
Bread	16.0787	0.6209	0.0292	0.7962	**0.6295**

TABLE 5 The performance comparison of the ablation study in the LOL dataset (red bold for the best, black bold for the second best).

Detection methods	PSNR	SSIM	MSE	UQI	VIF
RRDNet	13.1360	0.5598	0.0695	0.5286	0.4951
zero-DCE	17.5592	0.5750	**0.0228**	**0.8355**	**1.0560**
DRBN	17.2850	0.5174	0.0238	0.7222	0.5668
EXCNet	14.8137	0.5539	0.0643	0.7010	0.4894
LightenNet	10.5513	0.1243	0.1183	0.6142	0.1467
DSLR	16.1505	0.6273	0.0389	0.6932	0.5073
BIMEF	17.0586	0.5565	0.0254	0.7792	0.4692
LLFLOW	15.5407	0.5625	0.3552	0.7353	0.5331
Enlighten Anything	16.8056	0.5646	0.0280	0.8678	0.5288
Proposed (Spiking+convLSTM)	**18.3374**	0.5974	**0.0227**	**0.8369**	**1.1019**
EnlightenGAN	17.2322	**0.6945**	0.0287	0.8189	0.7589
BFSA	11.0324	0.4429	0.1031	0.3894	0.3573
Bread	**17.6990**	**0.6530**	0.0273	0.7888	0.8823

The study revealed a consistent trend among the four pairs of three-way combination parameters. The method that incorporated exposure loss, color constant loss, and spatial consistency loss outperformed all others in the overall index. However, in terms of UQI, the method combined with exposure loss, color constant loss, and illumination smooth loss performed similarly to the rest. Notably, all four loss functions operated simultaneously. In other words, the index value corresponding to the method proposed in Table 1 is still the best. However, in both the LOL and SCIE datasets, UQI and VIF were marginally inferior to the composite approach of exposure loss, color constant loss, and spatial consistency loss. This highlights the indispensability of using four loss functions. Figures 5, 6 illustrate the influence of each loss function on the image enhancement effect. As observed, exposure loss directly controls image enhancement, while color constant loss mainly controls image distortion after enhancement.

3.3 Performance comparison

After gaining an understanding of the proposed LLIE method, we conclude that the LLFLOW (Wang et al., 2022), BIMEF (Ying et al., 2017), RRDNet (Zhu et al., 2020), zero-DCE (Guo et al., 2020a), DRBN (Yang et al., 2020), EXCNet (Zhang Y. et al., 2019), Lightennet (Zhang et al., 2019), Enlighten Anything (Zhou et al., 2023), EnlightenGAN, DSLR (Ignatov et al., 2017), BREAD (Hu and Guo,

2022), and BFSA (Long et al., 2023) algorithms have strong robustness and potential applications. The proposed method was compared with two referenced datasets in the LLIE field based on five performance indicators. Figure 7 directly demonstrates the enhancement effect. Tables 4, 5 list the performance index values of the proposed method and several of the most popular enhancement methods in the two reference image datasets. The proposed method yielded the best values, with the PSNR, SSIM, MSE, UQI, and VIFP exceeding the second place at 4.4% (0.8%), 3.9% (17.2%), 0% (15%), 0.1% (0.2%), and 4.3% (0.9%), respectively. The numbers inside parentheses represent the increase in the SCIE dataset. In addition, we also measured the parameters related to the actual application characteristics of the resulting algorithm. In the actual application of the image algorithm, the hardware space occupied by the model and the space occupied by the test process warrant attention.

Enlighten Anything performs similarly to the EXCNet method. However, it has a good research starting point, which is combined with the large segmentation pretrained model algorithm (Kirillov et al., 2023). Although the LightenNet method meets the characteristics of lightweight, it yields several poor indices. The

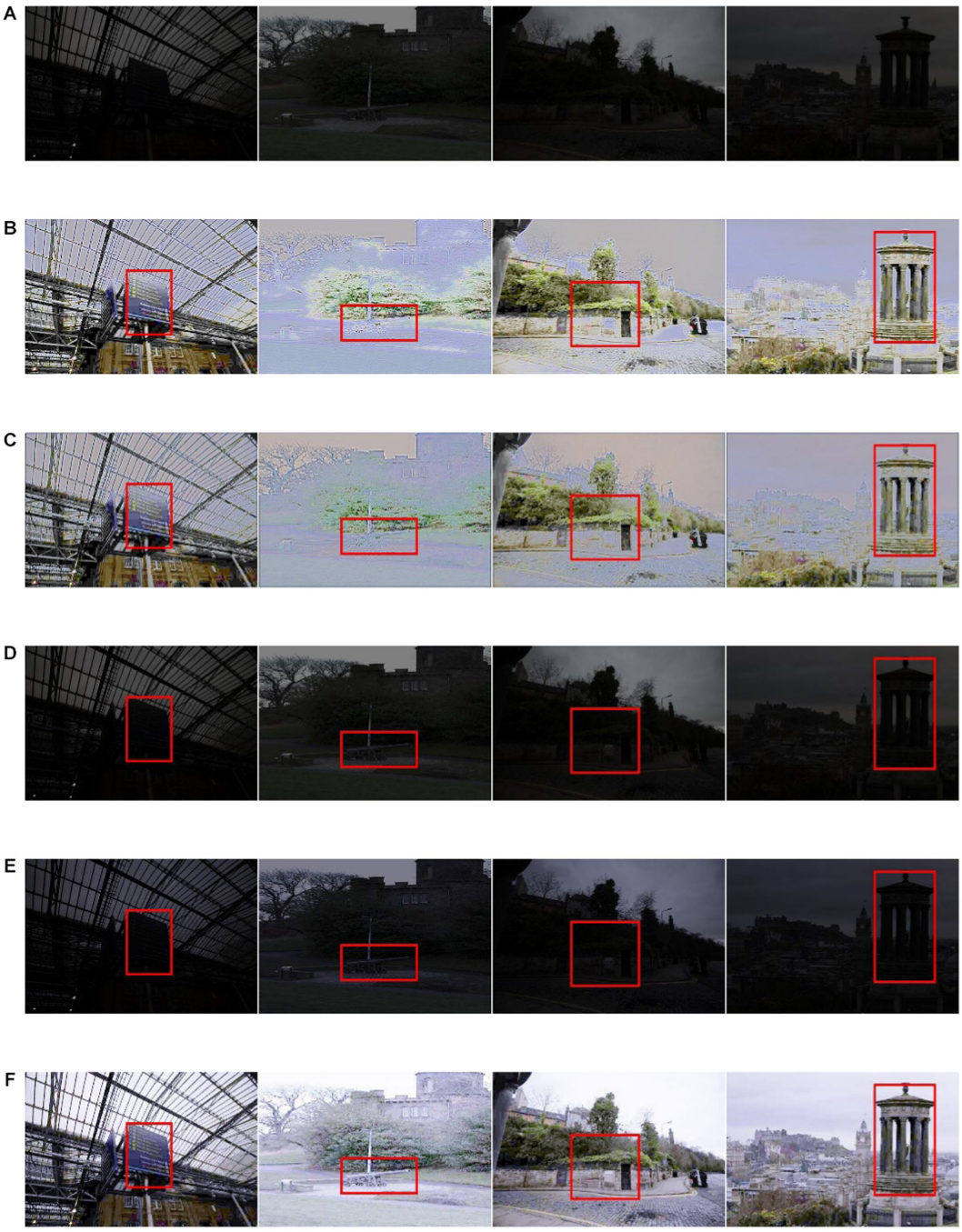


FIGURE 6 (Continued)

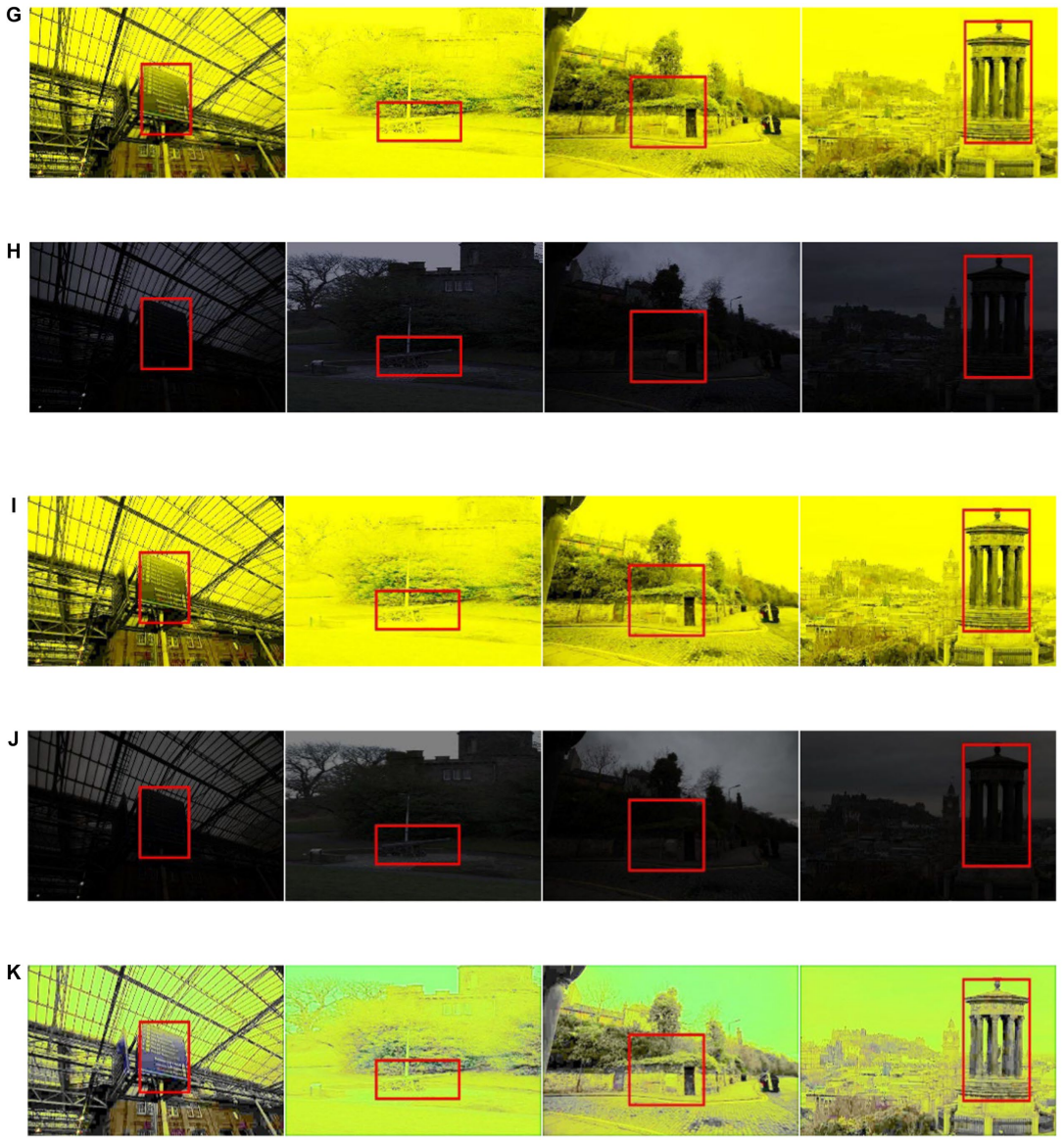


FIGURE 6
The loss function assembly in SCIE dataset **(A)** original low-light image **(B)** color constancy loss and exposure loss **(C)** illumination smooth loss, exposure loss **(D)** color constancy loss, spaital consistency loss, **(E)** spaital consistency loss, exposure loss, **(F)** illumination smooth loss, color constancy loss, **(G)** illumination smooth loss, spaital consistency loss **(H)** illumination loss, exposure loss and color constancy loss **(I)** color constancy loss, exposure loss and spatial consistency loss **(J)** illumination loss, color constancy loss and spatial consistency loss **(K)** illumination loss, exposure loss and spatial consistency loss.

performance of EXCNet methods is relatively moderate compared to other state-of-the-art methods. The model occupies a relatively large space. In the performance evaluation, the proposed methods, zero-DCE and EnlightenGAN, ranked the highest. The primary role of LLIE methods is to assist with enhancing the realization of other algorithmic functions. Generally, the model and testing process should occupy less space for better integration with other product features. As indicated in Table 6, the space occupied by the proposed algorithm in the test process ranks second, which is only larger than the poorly performing RRDNet, while the space occupied by the model itself reaches 151 KB, which is more than half of the space occupied by the second place.

3.4 User study

Certain LLIE-related datasets have no reference images corresponding to normal light, only images under dark lighting conditions, and therefore it was difficult to use objective evaluation indicators, such as PSNR, to evaluate image quality. To make the performance comparison clearer, more intuitive, and more efficient for these non-reference image datasets, a user study was performed to assess the human perception of the proposed method. The images tested by the user study included various image contents in different environments, including animals, exterior scenes, and buildings. Based on the user

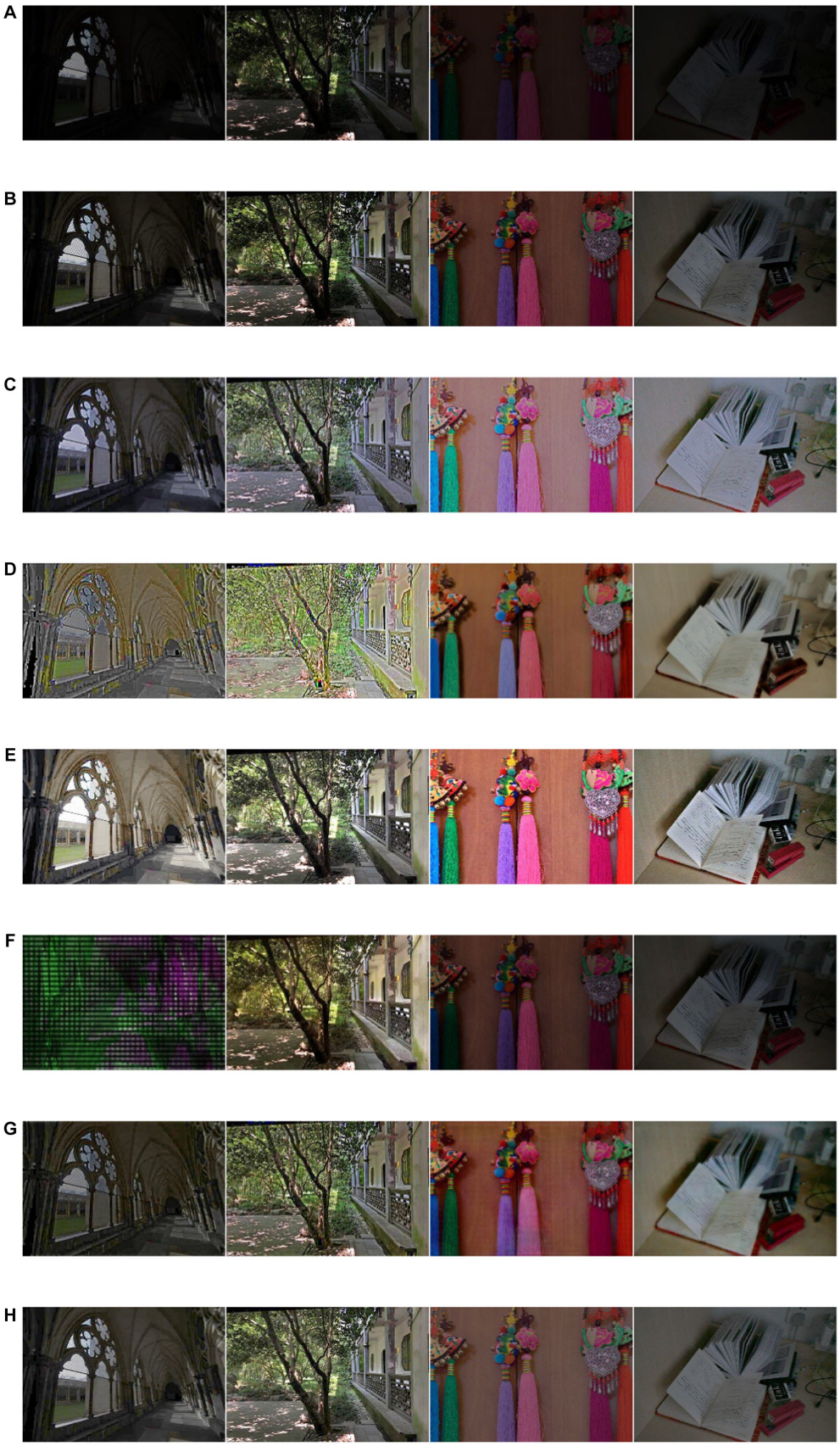


FIGURE 7 (Continued)

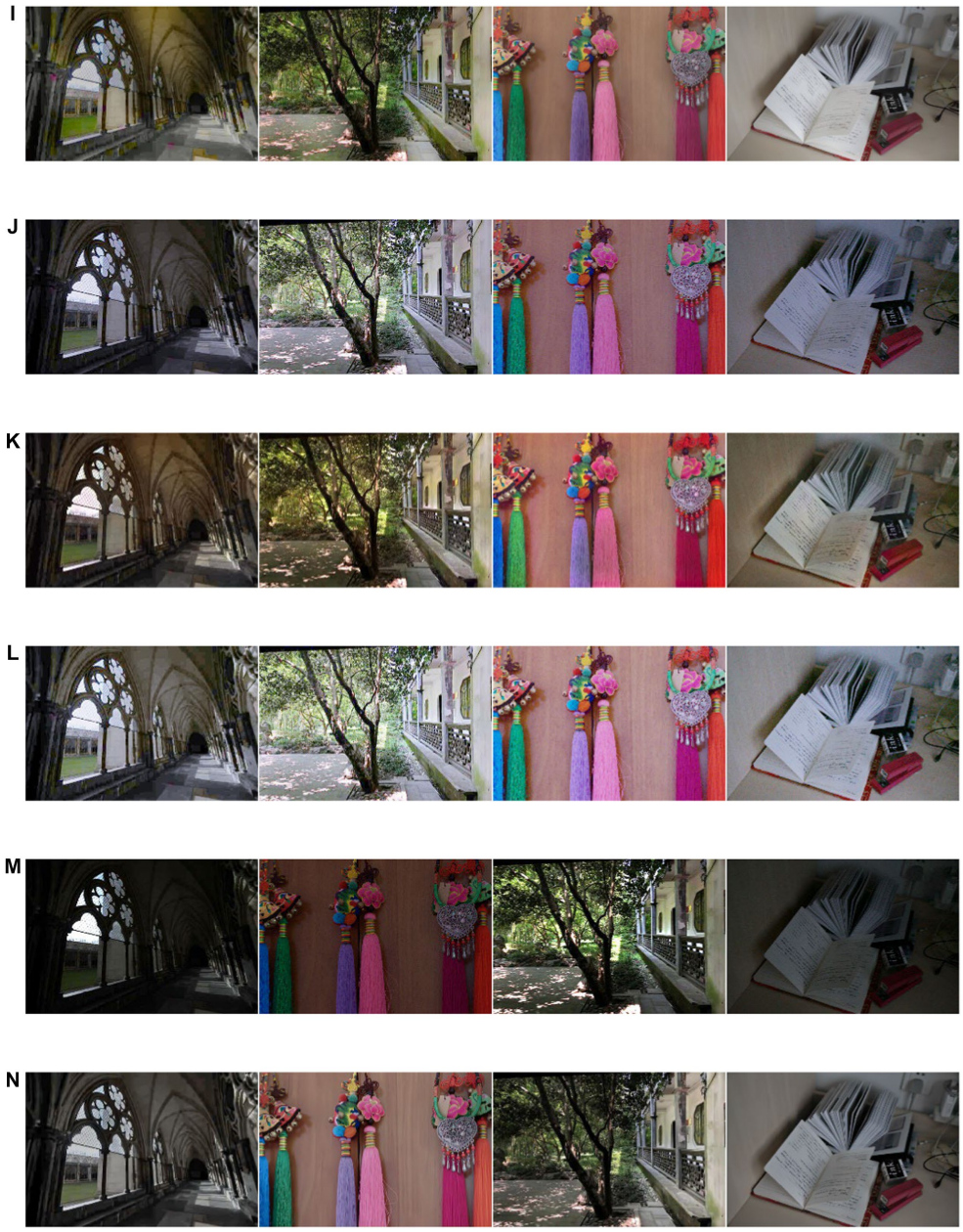


FIGURE 7
The performance comparsions for different combinations of loss function in SCIE dataset **(A)** original low-light image **(B)** RRDNet **(C)** zerodce **(D)** DRBN, **(E)** EXCNet **(F)** Lightennet, **(G)** DSLR, **(H)** BIMEF **(I)** LLFLOW **(J)** Enlighten Anything **(K)** EnlightenGAN **(L)** Proposed method **(M)** Bread **(N)** BFSA.

feedback data, we constructed a radar map with a maximum of 100 score points for each index, which answered the following five questions:

a. Are the details noticeable?
b. Are the colors vivid?
c. Is the result visually realistic?
d. Do the results contain overexposed/underexposed artifacts or over-enhanced/under-enhanced regions?
e. Do the results have unnatural texture and noticeable noise?

A single radar map can clearly compare the performance of different methods in various aspects of an unreferenced image dataset. The larger the area of the radar map, the better the subjective comprehensive evaluation of the method. Each angular direction, which ranges from 70 to 100 on the radar map, represents the user rating score for a specific problem. The five radar plots in Figure 8 illustrate the distributions of scores evaluated on different questions for different LLIE methods, where the bright red lines in the radar map represent the proposed method. We compared the results of the proposed method for the user study with those of the other LLIE

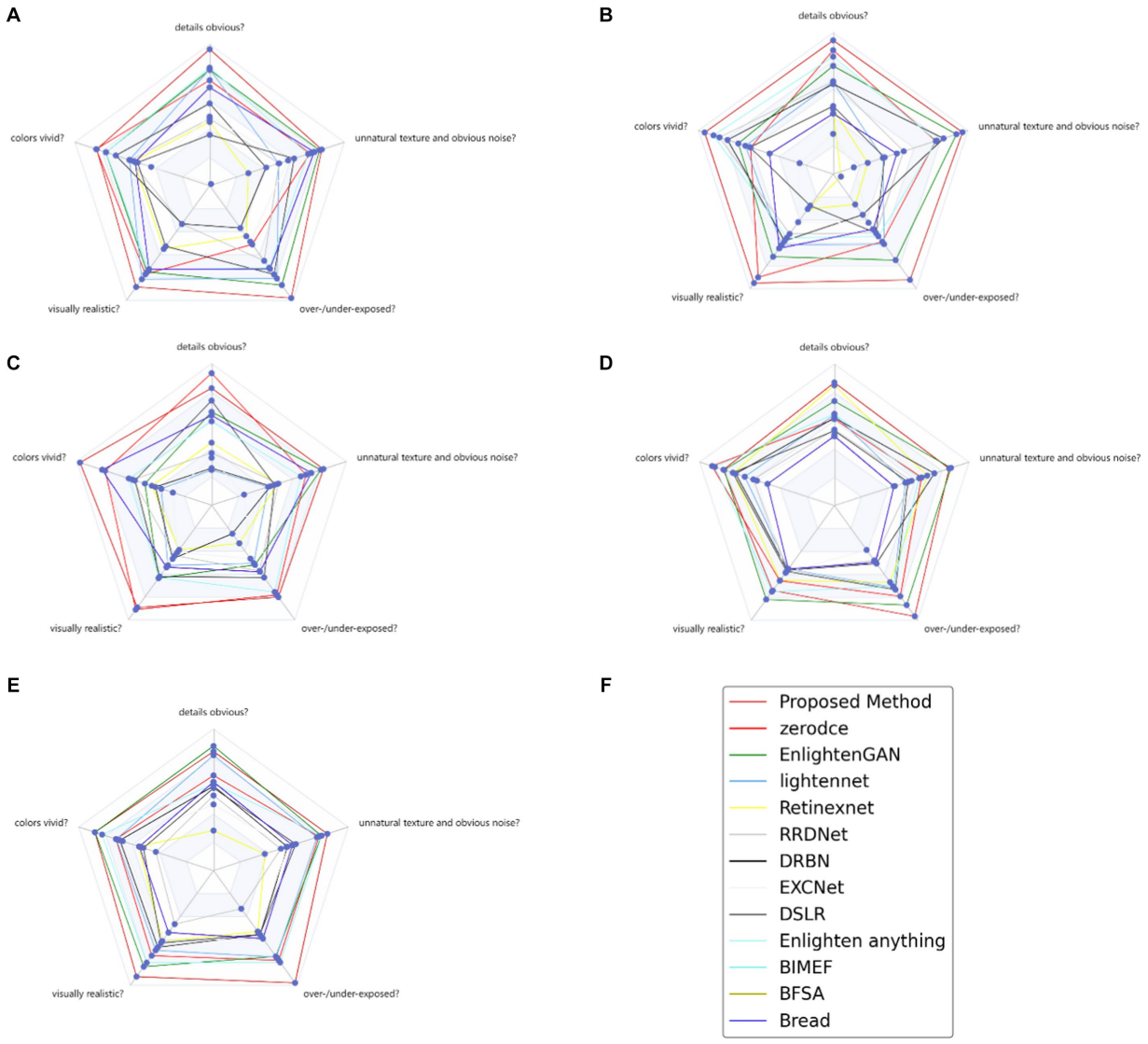


FIGURE 8
User study for 5 non-reference LLIE datasets **(A)** VV **(B)** NPE **(C)** LIME **(D)** DICM **(E)** MEF **(F)** legend of radar map.

TABLE 6 **Memory occupation of the model and testing process (red bold for the best, black bold for the second best).**

Detection methods	Testing memory	Model memory
RRDNet	**1,040,384 (1.04 MB)**	511 KB
zero-DCE	29,149,184 (29.1 MB)	315 KB
DRBN	53,602,358 (53.6 MB)	2.2 MB
EXCNet	22.9 MB (in Tensorflow)	157.2 MB
LightenNet	366.6 MB (in MATLAB)	**108 KB**
DSLR	40,600,532 (40.5 MB)	3.2 MB
BIMEF	125.8 MB (in MATLAB)	/
LLFLOW	239.6 MB	20.9 MB (smallest version)
Enlighten anything	137,830,912 (131.6 MB)	144 MB
Proposed (Spiking+convLSTM)	**22,233,600 (22.23 MB)**	**151 KB**
EnlightenGAN	590,612,992 (590.61 MB)	33,774 KB
Bread	2451.456 MB	6.6 MB
BFSA	3999.744 MB	230.1 MB

methods using a paired *t*-test (Guo et al., 2020). The results revealed that the effect of EnlightenGAN was the least different from that of the proposed method, except for zero-DCE.

4 Discussion

The dark light image enhancement method proposed in this paper has been tested by ablation experiments of different image evaluation indices in different datasets and performance comparison experiments, which have verified its performance superiority. In terms of space proportion, the model in this study is a single model, which does not need to involve a pre-training model or other model framework fusion methods. Enlighten Anything involves the pre-training weight of the SAM model. Compared with EnlightenGAN and related reinforcement learning methods, the new method has relatively low training configuration requirements and difficulty. The limitation of this method is that it is time-consuming at an average of 0.007 s, as determined by the LOL test dataset, which is marginally less than EnlightenGAN. After testing, it was found that the convLSTM structure occupied 0.006 s during testing. However, it still enhances images at 140 fps, which exceeds the real-time demand of 30 fps.

5 Conclusion

Originating from the further introduction of spiking coding mechanisms into DL, a novel network exhibits better performance based on DCENet by spiking encoding and convLSTM. Intensity-to-latency conversion, which is a spiking-coding methodology, can be used to gradually acquire the structural characteristics of an image. The multiple subgraphs generated by this method relate to the time step defined by spiking coding, and convLSTM is suitable for solving the image sequence problem and introducing the relationship information between multiple images into the network structure. Furthermore, the simplified DCENet structure without supervision achieved a superior result in terms of improvement. The performance comparison of this method with nine conventional methods in terms of five metrics was validated. The ablation study proved the necessity of the various parts of the structure, such as network and training losses. The proposed method yielded the best values with PSNR, SSIM, MSE, UQI, and VIFP. The proposed model occupies only 151 KB, which will better meet the algorithm integration and practical application requirements on a small chip.

6 Scope

The dark light enhancement method used in the study is closely related to the bionic neural networks and learning systems section of the special issue. The relationship between dark light enhancement and neural networks is that neural networks can be applied to tasks with dark light enhancement. Dark light enhancement is an image processing technique designed to improve the visibility of images taken in low-light conditions. By learning a large amount of training data, a neural network can automatically learn and extract the features in the image and perform enhancement processing on the image to improve the quality and visibility of the image.

By using a neural network, a dark light-enhanced model can be built, which is capable of receiving an input image and producing the enhanced image as output. The neural network can automatically learn and fit the mapping relationship between the input image and the output image through the connection and weight adjustment between the multiple layers of neurons. By training and optimizing the neural network, it can enhance the dark light image and have better generalization ability for different input images.

Data availability statement

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

Author contributions

XW: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Software, Validation, Visualization, Writing – original draft, Writing – review & editing. QW: Project administration, Resources, Writing – review & editing. LZ: Conceptualization, Writing – original draft, Writing – review & editing. YQ: Software, Writing – review & editing. FY: Conceptualization, Visualization, Writing – review & editing. JY: Supervision, Validation, Writing – review & editing. QL: Formal analysis, Methodology, Writing – review & editing. RX: Investigation, Validation, Writing – review & editing. ZX: Data curation, Formal analysis, Writing – review & editing. ST: Conceptualization, Formal analysis, Writing – review & editing.

Funding

The author(s) declare that no financial support was received for the research, authorship, and/or publication of this article.

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

Bao, C., Guo, Z., Yao, W., Ding, X., and Zhang, D. (2022). A novel low-light enhancement via fractional-order and low-rank regularized retinex model. *Comput. Appl. Math.* 42:7. doi: 10.1007/s40314-022-02140-6

Cai, J., Gu, S., and Zhang, L. (2018). Learning a deep single image contrast enhancer from multi-exposure images. *IEEE Trans. Image Process.* 27, 2049–2062. doi: 10.1109/tip.2018.2794218

Chen, X., Li, J., and Hua, Z. (2022). Retinex low-light image enhancement network based on attention mechanism. *Multim. Tools Appl.* 82, 4235–4255. doi: 10.1007/s11042-022-13411-z

Chen, W., Wang, W., Yang, W., and Liu, J. (2018). Deep Retinex decomposition for low-light enhancement. *arXiv*. doi: 10.48550/arXiv.1808.04560

Dar, K. A., and Mittal, S. (2020). A dynamic fuzzy histogram equalization for high dynamic range images by using multi-scale Retinex algorithm. *SSRN electron. J.* doi: 10.2139/ssrn.3565891

Fan, C.-M., Liu, T.-J., and Liu, K.-H. (2022). Half wavelet attention on m-net+ for low-light image enhancement. *Neural Inf. Process. Syst.*, 3878–3882. doi: 10.1109/ICIP46576.2022.9897503

Guo, C., Li, C., Guo, J., Loy, C. C., Hou, J., Kwong, S., et al. (2020a). Zero-Reference Deep Curve Estimation for Low-Light Image Enhancement. *[online] arXiv.org*. doi: 10.48550/arXiv.2001.06826

Guo, J., Ma, J., Garcia-Fernandez, A. F., Zhang, Y., and Liang, H. (2023). A survey on image enhancement for low-light images. *Heliyon* 9:e14558. doi: 10.1016/j.heliyon.2023.e14558

Guo, X., Wang, Y., Zhou, N., and Zhu, X. (2020b). Optimal weighted two-sample t-test with partially paired data in a unified framework. *Journal of Applied Statistics* 48, 961–976. doi: 10.1080/02664763.2020.1753027

Hu, Q., and Guo, X. (2022). Low-light image enhancement via breaking down the darkness. *Int. J. Comput. Vis.* 131, 48–66. doi: 10.1007/s11263-022-01667-9

Ignatov, A., Kobyshev, N., Timofte, R., Vanhoey, K., and Van Gool, L. (2017). DSLR-quality photos on Mobile devices with deep convolutional networks. *Int. Conf. Comput. Vis.* doi: 10.1109/iccv.2017.355

Jiang, Y., Gong, X., Liu, D., Cheng, Y., Fang, C., Shen, X., et al. (2021). Enlightengan: deep light enhancement without paired supervision. *IEEE Trans. Image Process.* 30, 2340–2349. doi: 10.1109/tip.2021.3051462

Jiang, Z., Li, H., Liu, L., Men, A., and Wang, H. (2021). A switched view of retinex: deep self-regularized low-light image enhancement. *Neurocomputing* 454, 361–372. doi: 10.1016/j.neucom.2021.05.025

Jiang, N., Lin, J., Zhang, T., Zheng, H., and Zhao, T. (2023). Low-light image enhancement via stage-transformer-guided network. *IEEE Trans. Circuits Syst. Video Technol.* 33, 3701–3712. doi: 10.1109/TCSVT.2023.3239511

Ke, X., Xin, Y., Baocai, Y., and Rynson, W.H.L. (2020). *Learning to restore low-light images via decomposition-and-enhancement*. IEEE WA, USA

Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., et al. (2023). Segment Anything. In: *ICCV 2023*. Available at: https://arxiv.org/pdf/2304.02643v1.pdf

Li, J., Feng, X., and Hua, Z. (2021). Low-light image enhancement via progressive-recursive network. *IEEE Trans. Circuits Syst. Video Technol.* 31, 4227–4240. doi: 10.1109/tcsvt.2021.3049940

Li, C., Guo, C., and Chen, C. L. (2021). Learning to enhance low-light image via zero-reference deep curve estimation. *IEEE Trans. Pattern Anal. Mach. Intell.* 1, 4225–4238. doi: 10.1109/tpami.2021.3063604

Li, C., Guo, C., Han, L.-H., Jiang, J., Cheng, M.-M., Gu, J., et al. (2022). Low-light image and video enhancement using deep learning: a survey. *IEEE Trans. Pattern Anal. Mach. Intell.* 44, 9396–9416. doi: 10.1109/tpami.2021.3126387

Li, C., Guo, F., Porikli, J., and Pang, Y. (2018). Lightennet: a convolutional neural network for weakly illuminated image enhancement. *Pattern Recogn. Lett.* 104, 15–22. doi: 10.1016/j.patrec.2018.01.010

Liu, X., Xie, Q., Zhao, Q., Wang, H., and Meng, D. (2023). Low-light image enhancement by retinex-based algorithm unrolling and adjustment. *IEEE Trans. Neural Netw. Learn. Syst.* doi: 10.1109/TNNLS.2023.3289626

Liu, J., Xu, D., Yang, W., Fan, M., and Huang, H. (2021). Benchmarking low-light image enhancement and beyond. *IEEE Trans. Pattern Anal. Mach. Intell.* 129, 1153–1184. doi: 10.1007/s11263-020-01418-8

Long, L., Dian, J., Nan, A., Jinyuan, L., Xin, F., Zhongxuan, L., et al. (2023). Bilevel Fast Scene Adaptation for Low-Light Image Enhancement. *International Journal of Computer Vision*. doi: 10.1007/s11263-023-01900-z

Lore, K. G., Akintayo, A., and Sarkar, S. (2017). LLNet: a deep autoencoder approach to natural low-light image enhancement. *Pattern Recogn.* 61, 650–662. doi: 10.1016/j.patcog.2016.06.008

Lv, F., Lu, F., Wu, J., and Lim, C.. (2018). MBLLEN: Low-light image/video enhancement using CNNs. In British machine vision conference. Newcastle, UK

Martinez-Seras, A., Del Ser, J., Lobo, J. L., Garcia-Bringas, P., and Kasabov, N. (2023). A novel out-of-distribution detection approach for spiking neural networks: design, fusion, performance evaluation and explainability. *Inf. Fusion* 100:101943. doi: 10.1016/j.inffus.2023.101943

Mozafari, M. N. D. A., and Ganjtabesh, M. T. M. (2019). Spyketorch: efficient simulation of convolutional spiking neural networks with at most one spike per neuron. *Front. Neurosci.* 13:625. doi: 10.3389/fnins.2019.00625

Rao, Y., Lin, W., and Chen, L. (2010). Image-based fusion for video enhancement of night-time surveillance. *Opt. Eng.* 49:120501. doi: 10.1117/1.3520553

Rong, Y., Liu, W., Zhang, Y., Zhi, Q., Zhao, D., and Zhang, B. (2018). Deepexposure: learning to expose photos with asynchronously reinforced adversarial learning. *Neural Inf. Process. Syst.* 31, 2149–2159.

Sheikh, H. R., and Bovik, A. C. (2006). Image information and visual quality. *IEEE Trans. Image Process.* 15, 430–444. doi: 10.1109/tip.2005.859378

Shen, Z., Xu, H., Jiang, G., Yu, M., Du, B., Luo, T., et al. (2023). Pseudo-retinex decomposition-based unsupervised underwater image enhancement and beyond. *Digit. Signal Process.* 137:103993. doi: 10.1016/j.dsp.2023.10399

Shi, X., Chen, Z., Wang, H., Yeung, D.-Y., Wong, W. K., and Woo, W. (2015). Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting. *Advances in neural information processing systems*. doi: 10.48550/arxiv.1506.04214

Singh, K., Pandey, A., Agarwal, A., Agarwal, M. K., Shankar, A., and Parihar, A. S. (2024). Frn: fusion and recalibration network for low-light image enhancement. *Multimed. Tools Appl* 83, 12235–12252. doi: 10.1007/s11042-023-15908-7

Tang, L., Ma, J., Zhang, H., and Guo, X. (2022). DRLIE: Flexible Low-Light Image Enhancement via Disentangled Representations. *IEEE transactions on neural networks and learning systems* 1–14. doi: 10.1109/tnnls.2022.3190880

Tang, H., Zhu, H., Tao, H., and Xie, C. (2023). An improved algorithm for low-light image enhancement based on retinexnet. *Appl. Sci.* 12:7268. doi: 10.3390/app12147268

Wang, Z., and Bovik, A. (2002). A universal image quality index. *IEEE Signal Process. Lett.* 9, 81–84. doi: 10.1109/97.995823

Wang, Q., Fu, X., Zhang, X.-P., and Ding, X. (2016). A fusion-based method for single backlit image enhancement. In 2016 IEEE international conference on image processing (ICIP). 4077–4081. Phoenix, AZ, USA

Wang, Y., Wan, R., Yang, W., Li, H., Chau, L.-P., and Kot, A. (2022). Low-light image enhancement with normalizing flow. *Proc. Int. AAAI Conf. Artif. Intell.* 36, 2604–2612. doi: 10.1609/aaai.v36i3.20162

Wu, T., Wu, W., Yang, Y., Fan, F.-L., and Zeng, T. (2023). Retinex image enhancement based on sequential decomposition with a plug-and-play framework. *IEEE Trans. Neural Netw. Learn. Syst.*, 1–14. doi: 10.1109/tnnls.2023.3280037

Xu, S., Zhang, J., Qin, X., Xiao, Y., Qian, J., Bo, L., et al. (2022). Deep retinex decomposition network for underwater image enhancement. *Comput. Electr. Eng.* 100:107822. doi: 10.1016/j.compeleceng.2022.107822

Yang, W., Wang, S., Fang, Y., Wang, Y., and Liu, J. (2020). From fidelity to perceptual quality: a semi-supervised approach for low-light image enhancement. *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit*, 3063–3072. doi: 10.1109/cvpr42600.2020.00313

Ying, Z., Li, G., and Gao, W. (2017). A bio-inspired multi-exposure fusion framework for low-light image enhancement. *arXiv*. doi: 10.48550/arxiv.1711.00591

Yusob, B., Mustaffa, Z., and Sulaiman, J. (2018). Anomaly detection in time series data using spiking neural network. *Adv. Sci. Lett.* 24, 7572–7576. doi: 10.1166/asl.2018.12980

Zhang, H., Fan, X., and Zhang, Y. (2023a). Energy-efficient spiking segmenter for frame and event-based images. *Biomimetics* 8:356. doi: 10.3390/biomimetics8040356

Zhang, H., Li, Y., He, B., Fan, X., Wang, Y., and Zhang, Y. (2023b). Direct training high-performance spiking neural networks for object recognition and detection. *Front. Neurosci.* 17:1229951. doi: 10.3389/fnins.2023.1229951

Zhang, L., Liu, X., Shen, Y., Zhang, S., and Zhao, S. (2019). Zero-Shot Restoration of Back-lit Images Using Deep Internal Learning. Proceedings of the 27th ACM International Conference on Multimedia, Nice, France.

Zhang, H., and Ma, J. (2023). Iid-mef: a multi-exposure fusion network based on intrinsic image decomposition. *Inf. Fusion* 95, 326–340. doi: 10.1016/j.inffus.2023.02.031

Zhang, Y., Zhang, J., and Guo, X. (2019). Kindling the darkness: a practical low-light image enhancer. MM '19: Proceedings of the 27th ACM international conference on multimedia, 1632–1640. doi: 10.1145/3343031.3351069

Zhou, D., Yang, Z., and Yang, Y. (2023). Pyramid diffusion models for low-light image enhancement. *Neural Inf. Process. Syst.* doi: 10.48550/arXiv.2305.10028

Zhu, A., Zhang, L., Shen, Y., Ma, Y., Zhao, S., and Zhou, Y. (2020). "Zero-shot restoration of underexposed images via robust retinex decomposition" in *IEEE Inter. Conf. Multimedia Expo(ICME)*, 1–6. doi: 10.1109/ICME46284.2020.9102962

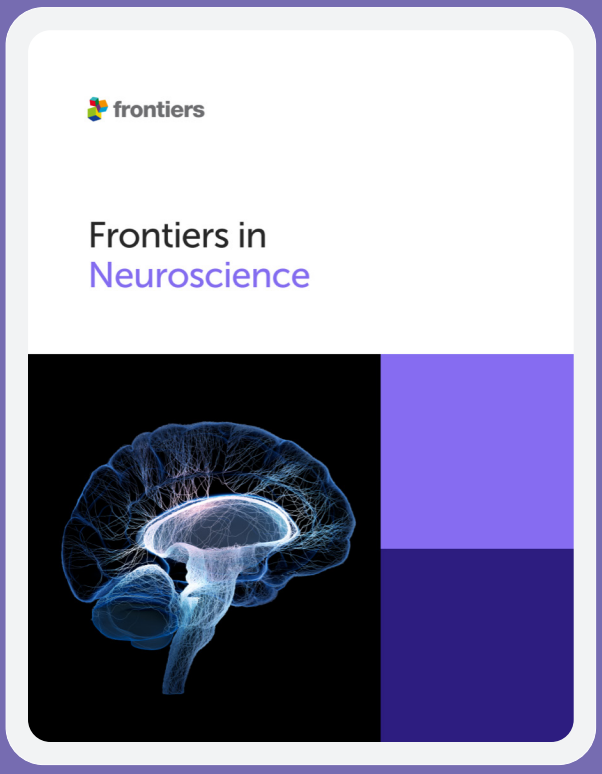
Frontiers in Neuroscience

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