COGNITIVE MULTITASKING – TOWARDS AUGMENTED INTELLIGENCE

EDITED BY: Yew Soon Ong, Liang Feng, Huajin Tang and Will Neil Browne
PUBLISHED IN: Frontiers in Neuroscience, Frontiers in Psychology,
Frontiers in Neurorobotics and Frontiers in Computational Neuroscience

frontiers Research Topics

frontiers

ISSN 1664-8714
ISBN 978-2-88966-560-0
DOI 10.3389/978-2-88966-560-0

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

COGNITIVE MULTITASKING – TOWARDS AUGMENTED INTELLIGENCE

Topic Editors:
Yew Soon Ong, Nanyang Technological University, Singapore
Liang Feng, Chongqing University, China
Huajin Tang, Zhejiang University, China
Will Neil Browne, Victoria University of Wellington, New Zealand

Citation: Ong, Y. S., Feng, L., Tang, H., Browne, W. N., eds. (2021). Cognitive Multitasking – Towards Augmented Intelligence. Lausanne: Frontiers Media SA. doi: 10.3389/978-2-88966-560-0

Table of Contents

frontiers in Neuroscience

EDITORIAL
published: 22 January 2021
doi: 10.3389/fnins.2021.619090

Editorial: Cognitive Multitasking – Towards Augmented Intelligence

Liang Feng[1]*, *Yew Soon Ong*[2], *Huajin Tang*[3] and *Will Neil Browne*[4]

[1] College of Computer Science, Chongqing University, Chongqing, China, [2] School of Computer Science and Engineering, Nanyang Technological University, Singapore, Singapore, [3] College of Computer Science and Technology, Zhejiang University, Hangzhou, China, [4] School of Engineering and Computer Science, Victoria University of Wellington, Wellington, New Zealand

Keywords: cognitive multitasking, multitask learning, transfer optimization, evolutionary multitasking, guest editorial

Editorial on the Research Topic

Cognitive Multitasking – Towards Augmented Intelligence

The original inspiration of artificial intelligence (AI) was to build autonomous systems that were capable of demonstrating humanlike behaviors. However, modern AI systems have begun to far exceed humanly achievable performance levels in areas such as image processing, complex optimization, and unmanned systems, due to the present-day data deluge, accompanied by subtle algorithmic enhancements in machine learning algorithms. This is occurring across a variety of domains, where prominent examples include IBM Watson winning Jeopardy! and Google DeepMind's AlphaGo beating the world's leading Go player. However, the AI future need not be limited to a human imitating standpoint. Instead, it may be more beneficial to build AI systems that are able to excel at that which humans have not evolved to do or to even consider. Humans have not evolved to process multiple distinct situations within short timespans (i.e., in the order of a few seconds) – as interleaving more than one task usually entails a considerable switching cost during which the brain must readjust from one task to the other.

Machines, on the other hand, are largely free from any such switching bottlenecks. Thus, machines can move more fluidly between tasks. Furthermore, when related tasks are bundled together, it may also be possible to seamlessly transfer or share the learned knowledge among them. As a result, while an AI attempts to solve some complex task, several other simpler ones may be unconsciously solved. Moreover, knowledge learned unconsciously in one task may be harnessed for intentional use in another application.

This special issue aims to explore deeply the issues faced in cognitive multitasking. Emphasis is placed on computational models and algorithms, as well as new hardware advances, that shall enable machines to be developed as consummate multitask problem-solvers. Following a rigorous peer review process, 11 papers have been accepted to be included in the special issue.

The first paper, "Multi-Task Learning Based Network Embedding" by Wang et al. presents a multi-task learning-based network embedding approach for network representation learning. The first task is designed to preserve the high-order proximity between pairwise nodes, while the second task is to preserve the low-order proximity in the one-hop area of each node. Comprehensive empirical studies on multi-label classification, link prediction, and visualization in five real-world networks, including social network, citation network, and language network, have been conducted to evaluate the performance of the proposed method over existing state-of-the-art approaches.

In the second paper entitled "High Cognitive Flexibility Learners Perform Better in Probabilistic Rule Learning," Feng et al. analyze how cognitive flexibility of human being, as assessed by the number-letter task, is associated with the learning process of a probabilistic rule task. This paper concludes that further research should be conducted to explore the internal process of learning differences between high and low flexibility learners by using other technologies across multiple modes.

Edited and reviewed by:
Mehdi Khamassi,
Centre National de la Recherche Scientifique (CNRS), France

****Correspondence:***
Liang Feng
liangf@cqu.edu.cn

Specialty section:
This article was submitted to Decision Neuroscience, a section of the journal Frontiers in Neuroscience

Received: *19 October 2020*
Accepted: *04 January 2021*
Published: *22 January 2021*

Citation:
Feng L, Ong YS, Tang H and Browne WN (2021) Editorial: Cognitive Multitasking – Towards Augmented Intelligence. Front. Neurosci. 15:619090. doi: 10.3389/fnins.2021.619090

To improve the convergence speed, a two-level transfer learning method has been proposed by Ma et al. in their paper "A Two-Level Transfer Learning Algorithm for Evolutionary Multitasking." The proposed method intends to use the correlation and similarity among the paired tasks to improve the efficiency and effectiveness of a multifactorial evolutionary algorithm.

The forth paper, "A Preliminary Study of Knowledge Transfer in Multi-Classification Using Gene Expression Programming" by Wei and Zhong. embarks a preliminary study on evolutionary multitasking optimization with gene expression programming for multi-classification. Experimental studies on 10 high-dimensional datasets show that knowledge transfer among separate binary classifiers under the proposed multitasking method can enhance multi-classification performance when compared to existing approaches.

To learn good representations of node in graphs or network, Xie et al. proposed a multi-task representation learning architecture coupled with the task of supervised node classification for graph classification and an end-to-end multi-task network representation learning framework with multi-task loss function for network embedding, in "A Multi-Task Representation Learning Architecture for Enhanced Graph Classification" and "Multi-Task Network Representation Learning," respectively.

In the seventh paper entitled "Droplet-Transmitted Infection Risk Ranking Based on Close Proximity Interaction," to identify people who are potentially-infected by droplet-transmitted diseases, Guo et al. present a multi-tasking framework to model the principle of Close Proximity Interaction and thus infer the infection risk of individuals. Experimental studies in different scenarios, including indoor office, bus station and bus compartment, hospital, show that the proposed method can achieve consistent results when compared to manual analysis very efficiently.

The eighth paper, "A Privacy-Preserving Multi-Task Learning Framework for Face Detection, Landmark Localization, Pose Estimation, and Gender Recognition" by Zhang et al. introduce a privacy-preserving multi-task learning approach to address the privacy issue existing in the training data for face processing tasks. The proposed method utilizes the differential private stochastic gradient descent algorithm to optimize the end-to-end multi-task model and weighs the loss functions of multiple tasks to improve learning efficiency and prediction accuracy.

To improve the performance of multi-task optimization, Xu et al. present new transfer sparks in fireworks algorithm for multitasking. For each task to be optimized, transfer sparks are generated with adaptive length and promising direction vector to transfer useful genetic information between different tasks while the optimization progresses online. The efficacy of the proposed method has been validated on the multi-task optimization benchmarks against existing state-of-the-art evolutionary multitasking approaches.

In the 10th paper entitled "Electroencephalographic Workload Indicators During Teleoperation of an Unmanned Aerial Vehicle Shepherding a Swarm of Unmanned Ground Vehicles in Contested Environments," Rojas et al. try to identify the electroencephalographic (EEG) indicators that can be used for the objective assessment of cognitive workload in a multitasking setting and as a foundational step toward a human-autonomy augmented cognition system.

Last but not the least, Howard et al. in their paper "BrainOS: A Novel Artificial Brain-Alike Automatic Machine Learning Framework," explores some of the principles of the brain that seem to be responsible for its autonomous, problem-adaptive nature. The presented BrainOS is an automatic approach for selecting the appropriate model based on three factors, which are (a) input at hand, (b) prior experience, which is a history of results of prior problem solving attempts), and (c) world knowledge that represented in the symbolic way and used as a means to explain its approach. Preliminary studies of BrainOS show that it can deal with complex problems, such as natural language processing.

As can be observed, in the 11 accepted papers, the second and tenth paper focus on psychology and neuroscience in cognitive multitasking, while the other papers concentrate on the multi-task optimization and learning algorithm designs. The human brain possesses the most remarkable ability to perform multiple tasks with apparent simultaneity, and leverages the experiences in solving one task to help the decision making in another task. These accepted papers have first illustrated the explorations of intelligent systems and algorithms that mimic beyond the human brain in efficient multitasking. Secondly, it can also be observed in these papers that the rapid increase in the variety, volume and complexity of real-world problems, the opportunity, tendency, and (even) the need to multitask is unprecedented.

The guest editors would like to thank all the authors who submitted their work to the special issue, and all reviewers for their hard work in completing timely and constructive reviews. Special thanks also go to the Editor-in-Chiefs and members of the editorial team for their support during the editing process of this Special Issue.

AUTHOR CONTRIBUTIONS

LF was the corresponding guest editor. All the authors are the guest editors of this special issue.

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.

frontiers in Neuroscience

ORIGINAL RESEARCH
published: 09 January 2020
doi: 10.3389/fnins.2019.01395

A Multi-Task Representation Learning Architecture for Enhanced Graph Classification

Yu Xie[1], *Maoguo Gong*[1]*, *Yuan Gao*[1], *A. K. Qin*[2] *and Xiaolong Fan*[1]

[1] *Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education, School of Electronic Engineering, Xidian University, Xi'an, China,* [2] *Department of Computer Science and Software Engineering, Swinburne University of Technology, Melbourne, VIC, Australia*

Composed of nodes and edges, graph structured data are organized in the non-Euclidean geometric space and ubiquitous especially in chemical compounds, proteins, etc. They usually contain rich structure information, and how to effectively extract inherent features of them is of great significance on the determination of function or traits in medicine and biology. Recently, there is a growing interest in learning graph-level representations for graph classification. Existing graph classification strategies based on graph neural networks broadly follow a single-task learning framework and manage to learn graph-level representations through aggregating node-level representations. However, they lack the efficient utilization of labels of nodes in a graph. In this paper, we propose a novel multi-task representation learning architecture coupled with the task of supervised node classification for enhanced graph classification. Specifically, the node classification task enforces node-level representations to take full advantage of node labels available in the graph and the graph classification task allows for learning graph-level representations in an end-to-end manner. Experimental results on multiple benchmark datasets demonstrate that the proposed architecture performs significantly better than various single-task graph neural network methods for graph classification.

Keywords: multi-task learning, representation learning, graph classification, node classification, graph neural network

OPEN ACCESS

Edited by:
Huajin Tang,
Zhejiang University, China

Reviewed by:
Zexuan Zhu,
Shenzhen University, China
Jinghui Zhong,
South China University of Technology, China

***Correspondence:**
Maoguo Gong
gong@ieee.org

Specialty section:
This article was submitted to Decision Neuroscience, a section of the journal Frontiers in Neuroscience

Received: *30 August 2019*
Accepted: *10 December 2019*
Published: *09 January 2020*

Citation:
Xie Y, Gong M, Gao Y, Qin AK and Fan X (2020) A Multi-Task Representation Learning Architecture for Enhanced Graph Classification. Front. Neurosci. 13:1395. doi: 10.3389/fnins.2019.01395

1. INTRODUCTION

Learning with graph-structured data, such as chemical compounds or proteins, requires effective representations of their internal structure (Hamilton et al., 2017b), as the structural changes usually have an impact on the traits they express. Nodes with different properties and unique connections make up a variety of graphs, and one of the graph learning tasks is to predict the labels for graphs. Specifically, nodes represent entities and edges represent relationships between them, and the category of a graph is always correlated with the graph structure and node labels in real world. Therefore, models capable of capturing node features and graph structure have been shown to achieve superior performances on classification tasks (Rossi et al., 2012).

In recent years, there has been a surge of interest in Graph Neural Networks (GNNs) (Cao et al., 2016; Monti et al., 2017; Schlichtkrull et al., 2018; Zou and Lerman, 2019) for learning representations of graphs and nodes. The general approach with GNNs broadly follows a recursive neighborhood aggregation scheme by passing, transforming and aggregating feature vectors of

nodes across the graph (Gilmer et al., 2017; Hamilton et al., 2017a; Xu et al., 2018). Empirically, these GNNs have achieved outstanding performance in many tasks such as graph classification and node classification. However, a major limitation of these GNN architectures is that they only focus on a specific task and their design is based on heuristics or experimental trial-and-error, and there is little theoretical understanding of the properties. As a result, GNNs' representational capacity and generalization ability are limited (Xu et al., 2019).

In real-world applications, the graph classification task is always correlated with the node classification task, and effective node representations are conducive to learning graph features with the same aggregation scheme (Petar et al., 2018). For example, a graph classification task is to predict the carcinogenicity of proteins, for which categories of nodes that represent different amino acids are of crucial importance. Nevertheless, previous related deep graph embedding methods treat real problems as several single tasks, while ignoring the rich correlation information between these related tasks. They do not follow human's cognitive laws of new things that people often apply the knowledge they have acquired by learning related tasks, whereas working on a single task from scratch is inefficient and increases the risk of overfitting. Moreover, they usually require multiple training steps that are difficult to optimize for each task (Tran, 2018).

To address the aforementioned challenges, we present a multi-task representation learning (MTRL) framework for both graph classification and node classification, schematically depicted in **Figure 1**. The MTRL framework is capable of learning representations of latent node embeddings and graph embeddings from local graph topology, and the shared representations between different tasks enable our model to generalize better on each task. A densely connected neural network is trained end-to-end to learn embeddings for nodes and graphs from the adjacency vector or feature vector, in which the READOUT function aggregates node representations from the final iteration to generate the entire graph's representation. The weighted sum of losses of graph classification and node classification is utilized in the back propagation of the multi-task learning process, thus graph-level features and fine-grained node features can be captured synchronously, and the generalization ability of models is improved through collaborative training. Specifically, our contributions in this paper are as follows:

- We propose a novel multi-task representation learning architecture and extend it further for different models designed specifically for graph classification. Compared with single-task learning models, our approach shows better performance in different tasks.
- Our architecture is efficiently trained end-to-end for the joint and simultaneous multi-task learning of supervised graph classification and node classification in a single stage.
- We conduct empirical evaluation of our architecture on five challenging benchmark graph-structured datasets, and the experimental results demonstrate significant improvement over state-of-the-art baselines.

The full text is structured as follows. After a basic introduction, the related backgrounds and algorithms about GNNs are shown in section 2. In section 3, we give a clear definition of the graph classification and the node classification, then the MTRL architecture is developed. Section 4 provides the experimental results of two classification tasks. Finally, in section 6 we conclude with a discussion of our architecture and summarize the future work.

2. RELATED WORK

Representation learning (Bengio et al., 2013) has been widely utilized in various fields such as computer vision (Du and Wang, 2015; Butepage et al., 2017) and natural language processing (Janner et al., 2018). With the rapid development of biology, chemistry, and medical science, the microscopic structure of molecular compounds as proteins and genes are paid more attention. This kind of graph-structured data attracts the interests of researchers in graph classification, and various methods are presented to learn graph representations.

Recently, a wide variety of GNN models have been proposed, including approaches inspired by convolutional neural networks (Defferrard et al., 2016; Kipf and Welling, 2016; Lei et al., 2017), recursive neural networks (Scarselli et al., 2008) and recurrent neural networks (Li et al., 2016). These methods have been applied to various tasks, such as graph classification (Dai et al., 2016; Zhang et al., 2018) and node classification (Kipf and Welling, 2016; Hamilton et al., 2017a). Instead of using hand-crafted features suited for specific tasks, deep learning techniques enable models to automatically learn features and representations for each node. In the context of graph classification, which is our main task, the major challenge is going from node embeddings to the representation of the entire graph. Most methods (Duvenaud et al., 2015; Li et al., 2016; Gilmer et al., 2017) have the limitation that they simply pool all the node embeddings in a single layer and do not learn the hierarchical representations, so they are unable to capture the natural structures of large graphs. Some recent approaches have focused on alleviating this problem by adopting novel aggregation approaches.

A latest research (Xu et al., 2019) developed theoretical foundations for reasoning about the expressive power of GNNs and presented a Graph Isomorphism Network (GIN) under the neighborhood aggregation framework. They proved that GNNs are at most as powerful as the Weisfeiler-Lehman (WL) test in distinguishing graph structures, and showed the discriminative power of GIN is equal to that of the WL test. They developed a "deep multisets" theory, which parameterizes universal multiset functions with the neural network, and a multiset is a generalized concept of a set that allows elements in it have multiple instances. Besides, multi-layer perceptrons (MLPs) are utilized in the model so that different graph structures could be discriminated through aggregation, combination and READOUT strategy. GIN updates node representations as:

$$h_v^{(l)} = MLP^{(l)}\big((1+\epsilon^{(l)})\cdot h_v^{(l-1)} + \sum\nolimits_{u\in\mathcal{N}(v)} h_u^{(l-1)}\big). \quad (1)$$

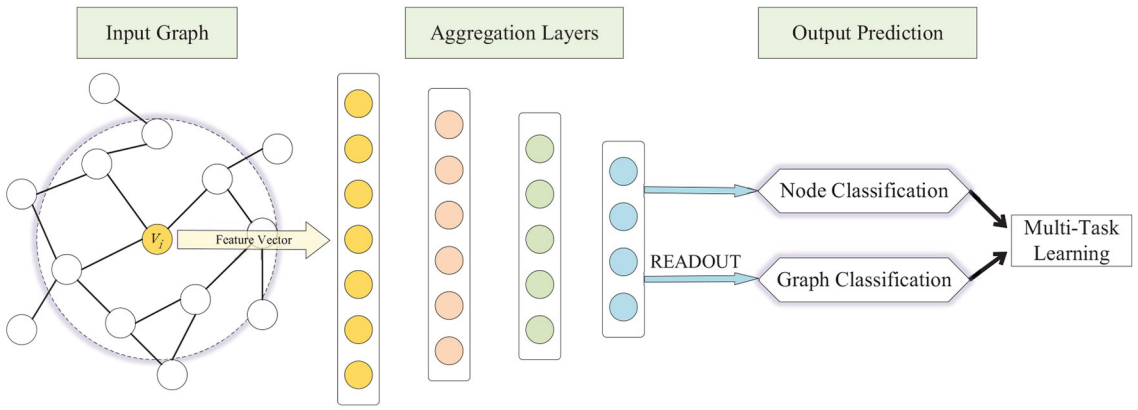

FIGURE 1 | Schematic depiction of the Multi-Task Representation Learning (MTRL) architecture.

They applied the sum aggregator that adds all neighbors of the current node, and set the combination method as $(1+\epsilon^{(l)})$ in lth layer, so that all nodes can be effectively integrated and mapped to the next layer. As a theoretical framework, GIN outperforms popular GNN variants, while some other researchers focus on coarsening the input graph inspired by the pooling method in convolutional neural networks.

DIFFPOOL (Ying et al., 2018) is a differentiable graph pooling module that can be adapted to various GNN architectures in a hierarchical and end-to-end fashion. DIFFPOOL learns a cluster assignment for nodes at each layer, which then forms the coarsened input for the next layer, and it is able to extract the complex hierarchical structure of graphs. Given the input adjacency matrix and node embedding matrix, the DIFFPOOL layer coarsens the input graph and generates a coarsened adjacency matrix as well as a new embedding matrix for each node or clusters in the coarsened graph. In particular, they applied the two following equations:

$$X^{(l+1)} = S^{(l)^T} Z^{(l)} \in \mathbb{R}^{n_{l+1} \times d}, \tag{2}$$

$$A^{(l+1)} = S^{(l)^T} A^{(l)} S^{(l)} \in \mathbb{R}^{n_{l+1} \times n_{l+1}}, \tag{3}$$

where $A^{(l)}$ represents the adjacency matrix at this layer. $Z^{(l)}$ and $X^{(l)}$ denote the input node embedding matrix and the cluster embedding matrix respectively. $S^{(l)}$ is the probabilistic assignment matrix that assigns each node at layer l to a specific cluster in the next coarsened layer $l+1$. Each row of $S^{(l)}$ corresponds to a node or cluster at layer l, and each column corresponds to a target cluster at layer $l+1$. The assignment matrix is generated from the pooling GNN using input cluster features $X^{(l)}$ and the cluster adjacency matrix $A^{(l)}$:

$$S^{(l)} = softmax\big(GNN_{l,pool}(A^{(l)}, X^{(l)})\big), \tag{4}$$

where the softmax function is utilized in a row-wise fashion. The output dimension of $GNN_{l,pool}$ is pre-defined as the hyperparameter of the model, which corresponds to the maximum number of clusters in each layer. Besides, the embedding GNN is a standard GNN module applied to $A^{(l)}$ and $X^{(l)}$:

$$Z^{(l)} = GNN_{l,embed}(A^{(l)}, X^{(l)}). \tag{5}$$

The adjacency matrix between the cluster nodes $A^{(l)}$ from Equation (3) and the pooled features for clusters $X^{(l)}$ from Equation (2) are passed through a standard GNN to obtain new embeddings $Z^{(l)}$ for the cluster nodes. GIN and DIFFPOOL can learn to discriminate and capture the meaningful structure of graphs in terms of aggregation and pooling, respectively, and they are powerful in the graph classification task.

In many real-world applications, such as network analysis and molecule classification, the input data is observed with a fraction of labeled graphs and labeled nodes. Thus it is desirable for the model to predict the labels of graphs and nodes simultaneously in a multi-task learning setting. Multi-task learning (MTL) refers to the paradigm of learning several related tasks together, which has been broadly used in natural language processing (Chen et al., 2018; Schulz et al., 2018; Sanh et al., 2019), computer vision (Choi et al., 2018; Kendall et al., 2018; Liu et al., 2019) and genomics (Yang et al., 2018). To be specific, SaEF-AKT (Huang et al., 2019) introduces a general similarity measure and an adaptive knowledge transfer mechanism to assist the knowledge transfer among tasks. EMT (Evolutionary multitasking) via autoencoding (Feng et al., 2018) allows the incorporation of multiple search mechanisms with different biases in the EMT paradigm. MTL is inspired by human learning activities where people could transfer the knowledge learned from the previous problems to facilitate learning a new task. Similar to human learning, the knowledge contained in a problem can be leveraged by related problems in the multi-task machine learning process. A main assumption of MTL is that there is an optimal shared parameter space for all problems, which is regularized by a specific loss, manually defined relationships or other automatic methods that estimate the latent structure of relationships among problems. Due to the shared processes that give rise to strong dependencies of multiple

tasks, the MTL approach is able to explore and leverage the commonalities among related tasks in the learning process.

3. METHODOLOGY

The key idea of the MTRL architecture is that it enables the graph classification and node classification tasks to be performed simultaneously. Along the way, it helps to improve the generalization ability of the model and avoid falling into the local minimum. In this section, we outline the MTRL structure and demonstrate how it works on the GIN and DIFFPOOL models. Before introducing the architecture, we start by discussing the statement of the problem.

3.1. Problem Statement

The input to the MTRL architecture is a set of labeled graphs $\mathcal{D} = \{(G_1, y_1), (G_2, y_2), ...\}$, where $y_i \in \mathcal{Y}$ is the label associated with graph $G_i \in \mathcal{G}$, and $G = (A, F, V)$ denotes a graph with an adjacency matrix $A \in \{0, 1\}^{n \times n}$ and node feature vectors $F \in \mathbb{R}^{n \times d}$, assuming each node $v \in V$ has d features. There are two tasks of interest: (1) *Graph classification*, where graph labels y_G are given and the goal is to learn a representation vector r_G that helps predict the label of the graph, $y_G = g(r_G)$; (2) *Node classification*, where each node v has a corresponding label y_v and we aim to learn a representation vector r_v such that $v's$ label could be predicted as $y_v = h(r_v)$. The main symbols are listed in **Table 1**.

3.2. Multi-Task Representation Learning

In this work, we build upon the MTRL architecture to learn useful representations for graph classification and node classification in an end-to-end fashion. The graph classification is set as the primary task while the node classification as the secondary task, and the performance of the model could be improved by sharing the training information in the primary task and the auxiliary related task. Since these two classification tasks are related, it is intuitive to assume that they share a common feature representation based on the original features, which do not have enough expressive power for multiple tasks. A more powerful representations could be learned for both tasks by the MTRL architecture and it will bring improvement on the performance.

Follow the GNN structure, the architecture adopts a neighborhood aggregation and combination strategy, where the representation of a node is iteratively updated by aggregating its neighbors' representations and combining its representation of the previous layer. Especially, after k iterations of aggregation and combination, representations of each node is able to capture the structural information within its k-hop graph neighborhood. For node classification, the node representation of the final layer is utilized for prediction. For graph classification, there should be a READOUT method that aggregates all node representations of the final iteration to generate the graph representation.

Based on the normal GNN models for graph classification, the MTRL architecture adds an additional softmax layer for node classification. Given an input graph G, the parameters of the model are trained to minimize the cross-entropy of the predicted and true distributions,

$$\mathcal{L}_v = -\sum\nolimits_{v \in V} \sum\nolimits_{c \in C} y_v^c \cdot log(\hat{y}_v^c) \tag{6}$$

where y_v^c is the ground-truth label; $\hat{y}_v^c$ is prediction probabilities, and C indicates node classes. The loss of graph classification $\mathcal{L}_G$ is similar to Equation (6).

During the multi-task learning process, the related information is exchanged and supplemented by a shared representation at a shallow level, and the accuracy of node classification and graph classification are optimized simultaneously. The node classification task enforces node-level representations to take full advantage of node labels available in the graph and the graph classification task allows for learning graph-level representations in an end-to-end manner. More precisely, we achieve multi-task learning on graphs by designing a joint loss function that combines the two masked categorical cross-entropy losses for supervised graph classification and node classification:

$$\mathcal{L}_{MTRL} = \mathcal{L}_G + \alpha \cdot \mathcal{L}_v \tag{7}$$

where α is used for the integration of the loss so that the scale of all losses is close. Noted that when α is 0, the architecture is equal to a single-task graph classification model. Besides, how we extract node representations is crucial to the discrimination task. In particular, we consider two state-of-the-art models that employ the above MTRL architecture.

3.2.1. Multi-Task GIN

The original GIN applies five GNN layers and all MLPs have two layers. It utilizes information from all depths of the model to consider all structural information in Equation (8), because features from deep layers are key to achieving better discriminative performance while features from shallow layers could generalize better.

$$r_G = CONCAT\big(READOUT(\{r_v^{(l)} | v \in G\}) \,|\, l = 0, 1, ..., L\big). \tag{8}$$

The READOUT is set as a simple permutation invariant function such as summation. Similarly, to obtain both global and refined representations of nodes, we achieve node features extraction that concatenated across all layers as follows, and then the softmax

TABLE 1 | Main symbols and descriptions in the paper.

Notations	Descriptions
G	Input labeled graph
A	Adjacency matrix
F	Feature information matrix
n	Number of nodes in a graph
d	Dimension of node features
r_G	Graph embedding representation
r_v	Node embedding representation

activation function is used to produce a probability distribution over node labels.

$$r_v = CONCAT(r_v^{(l)} \mid l = 0, 1, ..., L). \quad (9)$$

In the multi-task GIN (MT-GIN), all parameters in the network except for two softmax layers are shared. Considering that different tasks may have various sample noises in all directions with different patterns, the hard parameter sharing method could offset some noises through learning from multiple tasks, which will result in better performance on each task.

3.2.2. Multi-Task DIFFPOOL

Different from GIN, DIFFPOOL applies a more sophisticated graph-level pooling READOUT function. The GNN model used for DIFFPOOL is built on top of the GRAPHSAGE (Hamilton et al., 2017a) architecture as it has superior performance compared with the standard graph convolutional network. It sets a DIFFPOOL layer after two GRAPHSAGE layers, then three layers of graph convolutions are performed before the final READOUT layer. Since the DIFFPOOL layer will reduce the number of nodes by 90%, which makes it impossible for the node classification task, we extract the features matrix from the GRAPHSAGE layer before the DIFFPOOL layer and utilize each row in the matrix as the node representation, which is shown in **Figure 2**.

For this reason, in the multi-task DIFFPOOL (MT-DIFFPOOL), only parameters in the first two GRAPHSAGE layers are shared. The backpropagation of the graph classification loss starts from the last layer of the network, and the vanishing gradient problem leads to slower learning in the first few layers, thus their parameters may be dominated by the node classification task. These GRAPHSAGE layers before the pooling layer aim to learn efficient node representations, therefore the node classification task could facilitate capturing enhanced node features.

3.3. Complexity Analysis

Although applying multi-task framework requires additional computation of the node classification loss, we observed that the MT-GIN and the MT-DIFFPOOL do not incur substantial additional running time compared with GIN and DIFFPOOL in practice. Specifically, for the DIFFPOOL model, the computing cost is concentrated on GRAPHSAGE layers and the computation of an assignment matrix in DIFFPOOL layers, whereas the node classification loss is calculated in the first GRAPHSAGE layer, and it introduces only a few additional computation. Suppose K is the number of layers. n is the total number of nodes. m is the total number of edges. r is the number of neighbors being sampled for each node, and d is the dimensions of the node hidden features remain constant. The time complexity of a GRAPHSAGE layer is $O(r^K nd^2)$, and that of the DIFFPOOL algorithm could be denoted as $O(n^2)$. Similarly, the time complexity of GIN is $O(m)$, and our MTRL framework has the same time complexity as them respectively.

4. EXPERIMENTS

In this section, two state-of-the-art models employed with the proposed multi-task learning architecture are compared with the single-task learning ones. We evaluate the algorithms on an unsupervised learning task: visualization, and two supervised

TABLE 2 | Statistics of datasets used in our experiments.

Datasets	MUTAG	PTC	ENZYMES	PROTEINS	NCI1
Num. of Graphs	188	344	600	1113	4110
Avg. Number of Nodes	14.29	25.56	32.63	39.06	29.87
Avg. Number of Edges	14.69	25.96	62.14	72.82	32.30
Node Attr. (Dim.)	–	–	+(18)	+(1)	–
Num. of Graph Classes	2	2	6	2	2
Num. of Node Classes	7	19	3	3	37

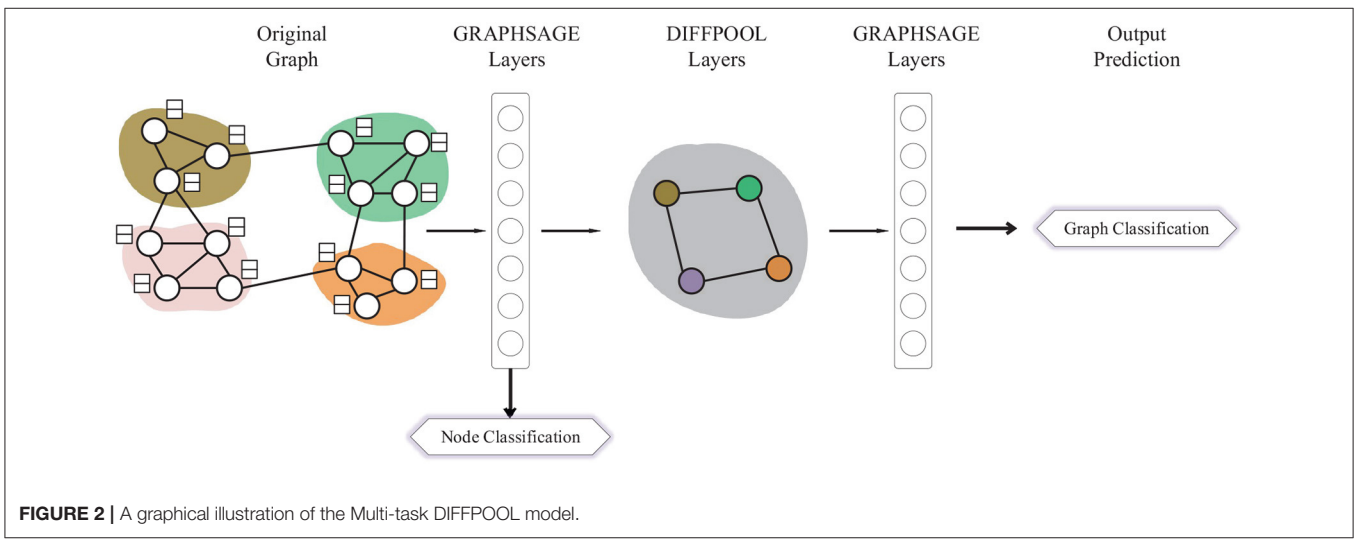

FIGURE 2 | A graphical illustration of the Multi-task DIFFPOOL model.

learning tasks: graph classification and node classification. Before we analyze the effect of the presented framework, we first introduce the datasets and model configurations.

4.1. Datasets

We use five bioinformatics graph classification benchmarks. For the ENZYMES dataset, the nodes have feature vectors, while for the other datasets, we set the adjacency matrix as input features since that have no features. The statistics of datasets are summarized in **Table 2**, and details of datasets are as following:

MUTAG (Debnath et al., 1991) is a dataset of 188 mutagenic aromatic and heteroaromatic nitro compounds, and the classification is based on whether or not they have a mutagenic effect on the Gram-negative bacterium Salmonella typhimurium.

PTC (Predictive ToxicologyChallenge) dataset (Toivonen et al., 2003) contains 344 chemical compounds tested for carcinogenicity in mice and rats. The classification task is to predict the carcinogenicity of the chemical compounds.

ENZYMES (Borgwardt et al., 2005) is a dataset of protein tertiary structures consisting of 600 enzymes from the BRENDA enzyme database (Schomburg et al., 2004). In this case, the task is to correctly assign each enzyme to one of the six EC top-level classes.

PROTEINS (Dobson and Doig, 2003) is similar to ENZYMES, where nodes are secondary structure elements. If two nodes are neighbors in the amino acid sequence or 3D space, there will be an edge between them. Each node has a discrete type attribute (helix, sheet or turn). Different from ENZYMES, it comes with the task of classifying into enzymes and non-enzymes.

NCI1 (Wale et al., 2008) represents a balanced subset of chemical compounds screened for activity against non-small cell lung cancer. This dataset contains more than 4,000 chemical compounds, each of which has a class label between positive and negative. Each chemical compound is represented as an undirected graph where nodes, edges and node labels correspond to atoms, chemical bonds, and atom types respectively.

4.2. Model Configurations

In our experiments, we evaluate the MTRL framework on GIN and DIFFPOOL model. Following (Yanardag and Vishwanathan, 2015; Niepert et al., 2016), we report the average of validation accuracy across the 10 folds within the cross-validation. For DIFFPOOL and MT-DIFFPOOL, the mean variant is used in GRAPHSAGE layers, and the l_2 normalization is added to the node embeddings at each layer to make the training more stable. For GIN and MT-GIN, ϵ in Equation (1) is fixed to 0, since this variant is proved to have strong empirical performance (Xu et al., 2019). Batch normalization (Ioffe and Szegedy, 2015) is applied for each layer in the two models. All models are trained for 350 epochs and 10 iterations for each epoch. We use the Adam optimizer (Kingma and Ba, 2015) with the initial learning

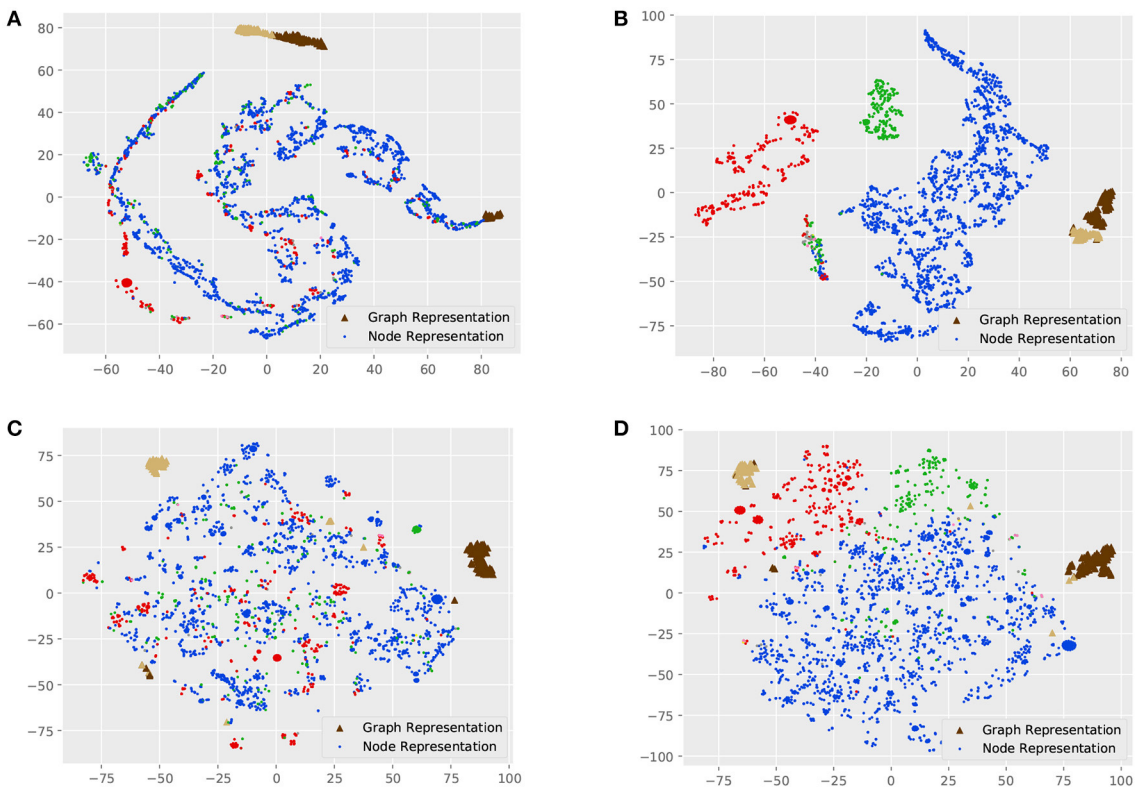

FIGURE 3 | Visualization of the MUTAG dataset. Each point represents a node in the dataset, and triangles of different colors represent graphs of different classes. **(A)** GIN, **(B)** MT-GIN, **(C)** DIFFPOOL, **(D)** MT-DIFFPOOL.

rate 0.01 and decay it by 0.5 every 50 epochs. Besides, the hyperparameter we tune is the weight of the node classification task $\alpha \in \{0, 0.5, 0.75, 1.25, 1.5, 2\}$.

5. RESULTS

5.1. Visualization

Visualizations are indispensable for analyzing high-dimensional data, which is able to intuitively reveal the intrinsic structure of data. Graphs and nodes of a smaller dataset, MUTAG, are represented as representation vectors with different models, and these vectors are further mapped into a two-dimensional space using t-SNE (Maaten and Hinton, 2008).

Figure 3 shows the visualization of graph and node representations. For MT-GIN and MT-DIFFPOOL, the hyperparameter α is fixed to 1. There are obvious differences between GIN and DIFFPOOL, as GIN could distinguish the graph representations from the node representations, while graph representations of different classes learned by DIFFPOOL are further away. All models are able to learn distinguishable graph representations, whereas GIN has a part of outliers on the right side and the same thing happens with DIFFPOOL in the lower left corner. In contrast, MT-GIN and MT-DIFFPOOL achieve more compact clusters. These models differ greatly in the performance of node representation learning. The node visualization results of GIN and DIFFPOOL are not meaningful, in which nodes with different tags are clustered together. Models with the MTRL framework achieve superior performance on node visualization, and both MT-GIN and MT-DIFFPOOL form clear boundaries among three main classes of nodes. Intuitively, this experiment demonstrates that the MTRL framework could help learn more meaningful and robust representations.

5.2. Training Set Performance

We validate the performance of our architecture and baselines by comparing their training accuracies, and we measure the effect of the key parameter α. An attributed dataset – ENZYMES and a large dataset – NCI1 are taken as examples. **Figures 4**, **5** show training curves of MT-GIN and MT-DIFFPOOL with different α, noted that the multi-task architecture is equal to a single-task graph classification model when α is 0. In our experiments, the multi-task learning model has a relatively rapid convergence rate, and they brings gain in fitting training compared to fixing α to 0 as in MT-GIN (MIN-0) and MT-DIFFPOOL (DIFFPOOL-0). It should be noted that the node classification accuracy of the MIN-0 and DIFFPOOL-0 tends to decline as iteration increases on ENZYMES, as latent representations of nodes are learned to fit the graph classification task. In particular, the training accuracy aligns with the models' representation power, and the multi-task learning models with different α tend to have higher

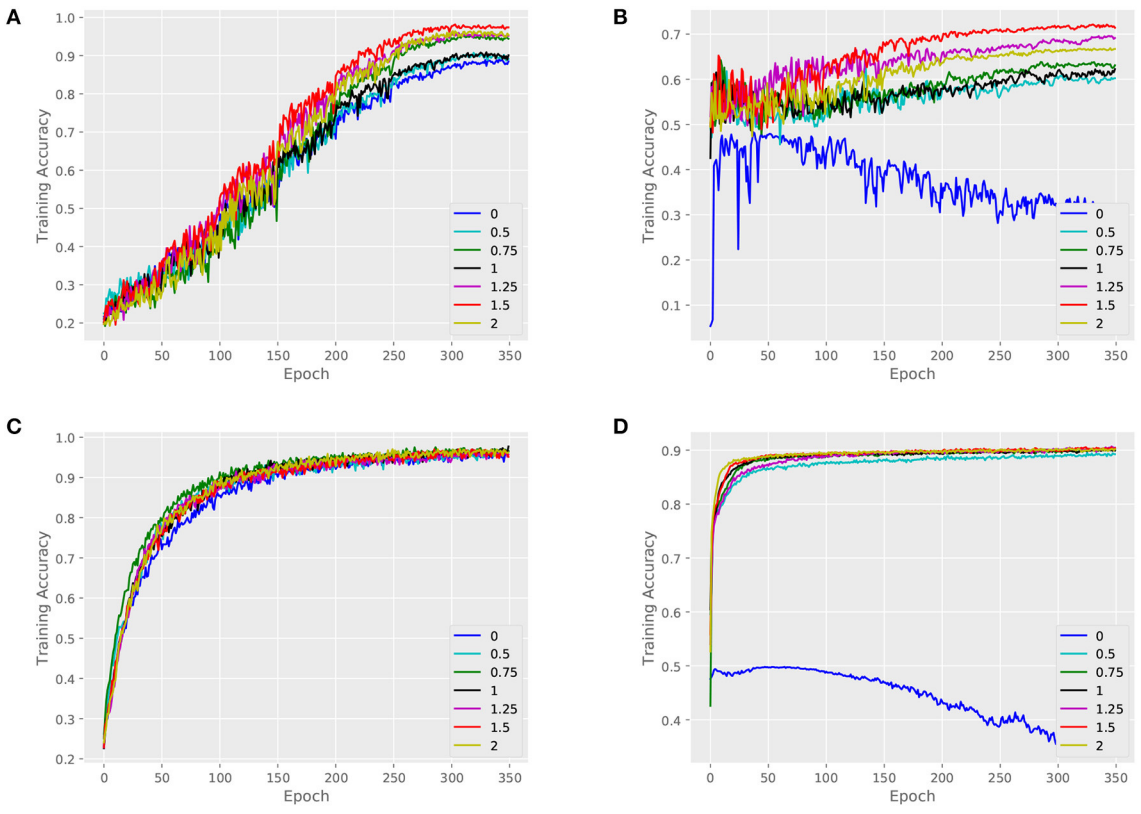

FIGURE 4 | Training set performance of different models on the ENZYMES dataset. **(A)** Training loss for graphs of MT-GIN. **(B)** Training loss for nodes of MT-GIN. **(C)** Training loss for graphs of MT-DIFFPOOL. **(D)** Training loss for nodes of MT-DIFFPOOL.

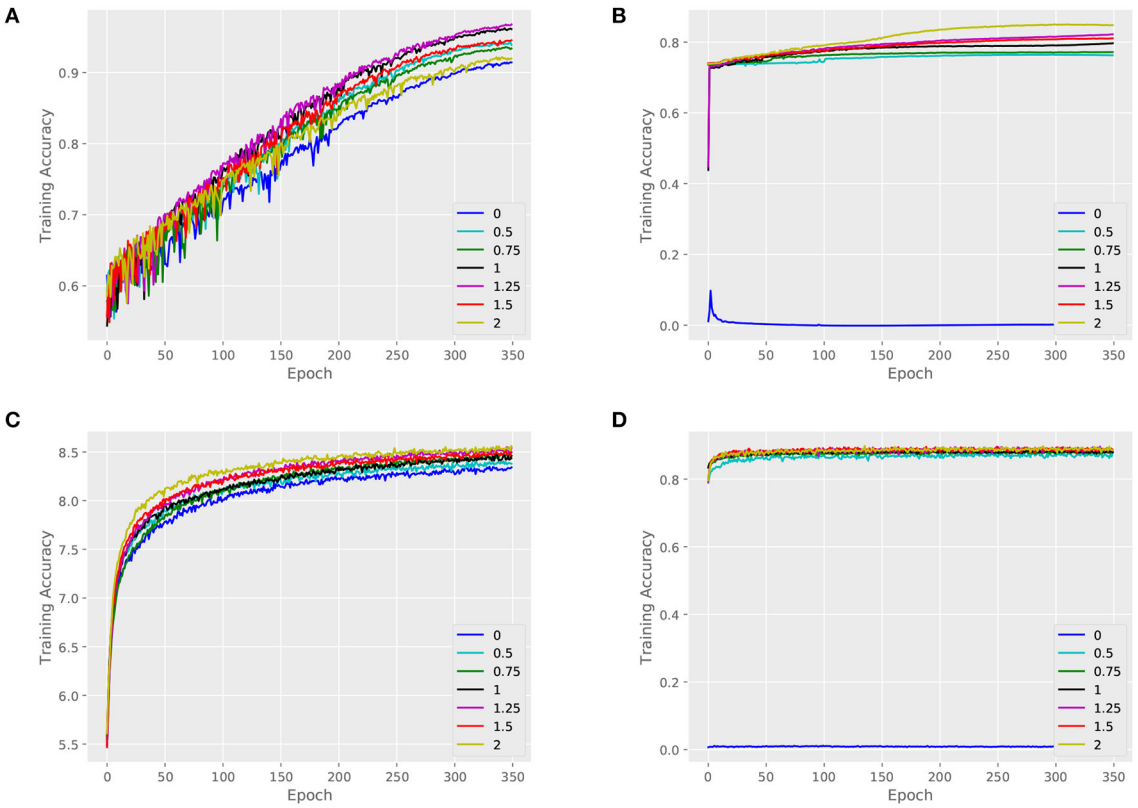

FIGURE 5 | Training set performance of different models on the NCI1 dataset. **(A)** Training loss for graphs of MT-GIN. **(B)** Training loss for nodes of MT-GIN. **(C)** Training loss for graphs of MT-DIFFPOOL. **(D)** Training loss for nodes of MT-DIFFPOOL.

TABLE 3 | Graph classification accuracy (%) of the MTRL architecture as well as the state-of-the-art baselines.

Datasets	MUTAG	PTC	ENZYMES	PROTEINS	NCI1
GIN	89.55	69.71	65.67	73.29	77.12
MT-GIN	**91.63**	**72.65**	**69.55**	**75.48**	**82.59**
DIFFPOOL	87.21	65.04	62.68	72.08	68.91
MT-DIFFPOOL	**87.36**	**70.52**	**64.90**	**76.18**	**71.26**

The best results are shown in bold.

TABLE 4 | Node classification accuracy (%) of the MTRL architecture as well as the state-of-the-art baselines.

Datasets	MUTAG	PTC	ENZYMES	PROTEINS	NCI1
GIN	28.21	18.60	27.33	26.49	2.08
MT-GIN	**94.35**	**91.02**	**71.23**	**61.85**	**80.48**
DIFFPOOL	19.76	3.11	31.87	29.27	1.22
MT-DIFFPOOL	**97.20**	**88.33**	**82.71**	**73.02**	**83.99**

The best results are shown in bold.

training accuracies than the single-task learning ones. Moreover, the weight of node classification loss is not always positively correlated with the training accuracy for graphs or nodes, thus the hyperparameter α is important and should be well tuned.

5.2.1. Test Set Performance

Next, we compare test accuracies. We fix the training ratio to 90% and display the average accuracy of graph classification and node classification, as shown in **Tables 3**, **4**. The MTRL architecture consistently outperforms the original GNN models, and it is able to efficiently capture graph structure and node features. By means of node classification task that accurately extracts node attributes, the MTRL architecture can achieve better performance in graph classification.

For graph classification, both MT-GIN and MT-DIFFPOOL outperform the original models on all datasets. The MUTAG dataset is relatively small with simple structure thus the improvement is not obvious. Specifically, even if node adjacency vectors are provided as input features, it still reaches higher accuracy on PTC and NCI1 dataset. The experimental results demonstrate that models' generalization performance is improved as the potential information contained in multiple tasks is leveraged.

For node classification, it is observed that the MTRL architecture shows significant improvement on five protein datasets, since the results of single-task GNN models are hardly better than random guesses, and their accuracy is relative to the number of nodes in each class. The training accuracy of node

classification is very close to the test accuracy on ENZYMES and NCI1, which means the learning of graph-level structure is able to prevent the overfitting of fine-grained node-level features from a macroscopical view.

6. CONCLUSION

In this paper, we develop a novel multi-task representation learning architecture coupled with the task of supervised node classification for enhanced graph classification. Along the way, we extend the architecture to two state-of-the-art GNN models, thus the model could perform node classification during the process of graph classification. We conduct extensive experiments on multiple benchmark datasets, and the experimental results demonstrate that the proposed architecture performs significantly better than various superior GNN methods for graph classification as well as node classification.

Moreover, we will explore the following directions in the future:

(1) The MTRL architecture could simultaneously optimize graph classification and node classification task, and we will make it scalable for other graph applications such as unsupervised link prediction or community detection.

(2) We have analyzed the effect of the weight parameter α, and we plan to explore a self-adaptive parameter or structure that could balance losses of each task. Moreover, it would also be interesting to investigate soft parameter sharing or regularization-based sharing.

DATA AVAILABILITY STATEMENT

The datasets for this study can be found in the TU Dortmund at https://ls11-www.cs.tu-dortmund.de/staff/morris/graphkerneldatasets.

AUTHOR CONTRIBUTIONS

YX and MG conceptualized the problem and the technical framework. MG and YG developed the algorithms and supervised the experiments and exported the data. YX, AQ, and XF implemented the multi-task representation learning architecture simulation. MG managed the project. All authors wrote the manuscript, discussed the experimental results and commented on the manuscript.

FUNDING

This work was supported by the National key research and development program of China (Grant no. 2017YFB0802200) and the Key research and development program of Shaanxi Province (Grant no. 2018ZDXM-GY-045).

ACKNOWLEDGMENTS

We would like to thank Bin Yu for thoughtful comments on the manuscript and language revision. We are grateful to all study participants for their time and effort.

REFERENCES

Bengio, Y., Courville, A., and Vincent, P. (2013). Representation learning: a review and new perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.* 35, 1798–1828. doi: 10.1109/TPAMI.2013.50

Borgwardt, K. M., Ong, C. S., Schönauer, S., Vishwanathan, S. V., Smola, A. J., and Kriegel, H. P. (2005). Protein function prediction via graph kernels. *Bioinformatics* 21, i47–i56. doi: 10.1093/bioinformatics/bti1007

Butepage, J., Black, M. J., Kragic, D., and Kjellstrom, H. (2017). "Deep representation learning for human motion prediction and classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Honolulu, HI), 6158–6166. doi: 10.1109/CVPR.2017.173

Cao, S., Lu, W., and Xu, Q. (2016). "Deep neural networks for learning graph representations," in *Proceedings of the 30th AAAI Conference on Artificial Intelligence* (Phoenix), 1145–1152.

Chen, J., Qiu, X., Liu, P., and Huang, X. (2018). "Meta multi-task learning for sequence modeling," in *Proceedings of the 32nd AAAI Conference on Artificial Intelligence* (New Orleans, LA), 5070–5077.

Choi, Y., Choi, M., Kim, M., Ha, J., Kim, S., and Choo, J. (2018). "Stargan: unified generative adversarial networks for multi-domain image-to-image translation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Salt Lake City, UT), 8789–8797.

Dai, H., Dai, B., and Song, L. (2016). "Discriminative embeddings of latent variable models for structured data," in *Proceedings of the 33rd International Conference on Machine Learning* (New York, NY), 2702–2711.

Debnath, A. K., Lopez de Compadre, R. L., Debnath, G., Shusterman, A. J., and Hansch, C. (1991). Structure-activity relationship of mutagenic aromatic and heteroaromatic nitro compounds. correlation with molecular orbital energies and hydrophobicity. *J. Med. Chem.* 34, 786–797.

Defferrard, M., Bresson, X., and Vandergheynst, P. (2016). "Convolutional neural networks on graphs with fast localized spectral filtering," in *Advances in Neural Information Processing Systems* (Barcelona) 3844–3852.

Dobson, P. D., and Doig, A. J. (2003). Distinguishing enzyme structures from non-enzymes without alignments. *J. Mol. Biol.* 330, 771–783. doi: 10.1016/S0022-2836(03)00628-4

Du, X., and Wang, J. J. (2015). Support image set machine: jointly learning representation and classifier for image set classification. *Knowl Based Syst.* 78, 51–58. doi: 10.1016/j.knosys.2015.01.016

Duvenaud, D. K., Maclaurin, D., Iparraguirre, J., Bombarell, R., Hirzel, T., Aspuru-Guzik, A., et al. (2015). "Convolutional networks on graphs for learning molecular fingerprints," in *Advances in Neural Information Processing Systems* (Montreal, QC), 2224–2232.

Feng, L., Zhou, L., Zhong, J., Gupta, A., Ong, Y.-S., Tan, K.-C., et al. (2018). Evolutionary multitasking via explicit autoencoding. *IEEE Trans. Cybernet.* 49, 3457–3470. doi: 10.1109/TCYB.2018.2845361

Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O., and Dahl, G. E. (2017). "Neural message passing for quantum chemistry," in *Proceedings of the 34th International Conference on Machine Learning* (Sydney, NSW), 1263–1272.

Hamilton, W., Ying, Z., and Leskovec, J. (2017a). "Inductive representation learning on large graphs," in *Advances in Neural Information Processing Systems* (Long Beach, CA), 1024–1034.

Hamilton, W. L., Ying, R., and Leskovec, J. (2017b). Representation learning on graphs: methods and applications. *IEEE Data Eng. Bull.* 40, 52–74.

Huang, S., Zhong, J., and Yu, W. (2019). Surrogate-assisted evolutionary framework with adaptive knowledge transfer for multi-task optimization. *IEEE Trans. Emerg. Top. Comput.* doi: 10.1109/TETC.2019.2945775. [Epub ahead of print].

Ioffe, S. and Szegedy, C. (2015). "Batch normalization: accelerating deep network training by reducing internal covariate shift," in *Proceedings of the 32nd International Conference on Machine Learning* (Lille), 448–456.

Janner, M., Narasimhan, K., and Barzilay, R. (2018). Representation learning for grounded spatial reasoning. *Trans. Assoc. for Comput. Linguist.* 6, 49–61. doi: 10.1162/tacl-a-00004

Kendall, A., Gal, Y., and Cipolla, R. (2018). "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Salt Lake City, UT), 7482–7491.

Kingma, D. P., and Ba, J. (2015). "Adam: a method for stochastic optimization," in *Proceedings of the 3rd International Conference on Learning Representations*.

Kipf, T. N., and Welling, M. (2016). "Semi-supervised classification with graph convolutional networks," in *Proceedings of the 4th International Conference on Learning Representations*.

Lei, T., Jin, W., Barzilay, R., and Jaakkola, T. (2017). "Deriving neural architectures from sequence and graph kernels," in *Proceedings of the 34th International Conference on Machine Learning* (Sydney, NSW), 2024–2033.

Li, Y., Tarlow, D., Brockschmidt, M., and Zemel, R. (2016). "Gated graph sequence neural networks," in *Proceedings of the 4th International Conference on Learning Representations* (New York, NY).

Liu, S., Johns, E., and Davison, A. J. (2019). "End-to-end multi-task learning with attention," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (San Juan), 1871–1880.

Maaten, L. V. D., and Hinton, G. (2008). Visualizing data using t-sne. *J. Mach. Learn. Res.* 9, 2579–2605.

Monti, F., Bronstein, M., and Bresson, X. (2017). "Geometric matrix completion with recurrent multi-graph neural networks," in *Advances in Neural Information Processing Systems* (Long Beach, CA), 3697–3707.

Niepert, M., Ahmed, M., and Kutzkov, K. (2016). "Learning convolutional neural networks for graphs," in *Proceedings of the 33rd International Conference on Machine Learning* (New York, NY), 2014–2023.

Petar, V., Guillem, C., Arantxa, C., Adriana, R., Pietro, L., and Yoshua, B. (2018). "Graph attention networks," in *Proceedings of the 6th International Conference on Learning Representations* (Vancouver, BC).

Rossi, R. A., McDowell, L. K., Aha, D. W., and Neville, J. (2012). Transforming graph data for statistical relational learning. *J. Artif. Intell. Res.* 45, 363–441. doi: 10.1613/jair.3659

Sanh, V., Wolf, T., and Ruder, S. (2019). "A hierarchical multi-task approach for learning embeddings from semantic tasks," in *Proceedings of the 33rd AAAI Conference on Artificial Intelligence* (Honolulu, HI).

Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M., and Monfardini, G. (2008). The graph neural network model. *IEEE Trans. Neural Netw.* 20, 61–80. doi: 10.1109/TNN.2008.2005605

Schlichtkrull, M., Kipf, T. N., Bloem, P., Van Den Berg, R., Titov, I., and Welling, M. (2018). "Modeling relational data with graph convolutional networks," in *Proceedings of the 15th European Semantic Web Conference* (Heraklion), 593–607.

Schomburg, I., Chang, A., Ebeling, C., Gremse, M., Heldt, C., Huhn, G., et al. (2004). Brenda, the enzyme database: updates and major new developments. *Nucleic Acids Res.* 32, D431–D433. doi: 10.1093/nar/gkh081

Schulz, C., Eger, S., Daxenberger, J., Kahse, T., and Gurevych, I. (2018). "Multi-task learning for argumentation mining in low-resource settings," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (New Orleans), 35–41.

Toivonen, H., Srinivasan, A., King, R. D., Kramer, S., and Helma, C. (2003). Statistical evaluation of the predictive toxicology challenge 2000–2001. *Bioinformatics* 19, 1183–1193. doi: 10.1093/bioinformatics/btg130

Tran, P. V. (2018). "Learning to make predictions on graphs with autoencoders," in *Proceedings of the 5th IEEE International Conference on Data Science and Advanced Analytics* (Turin), 237–245.

Wale, N., Watson, I. A., and Karypis, G. (2008). Comparison of descriptor spaces for chemical compound retrieval and classification. *Knowl. Inf. Syst.* 14, 347–375. doi: 10.1007/s10115-007-0103-5

Xu, K., Hu, W., Leskovec, J., and Jegelka, S. (2019). "How powerful are graph neural networks?" in *Proceedings of the 7th International Conference on Learning Representations*.

Xu, K., Li, C., Tian, Y., Sonobe, T., Kawarabayashi, K., and Jegelka, S. (2018). "Representation learning on graphs with jumping knowledge networks," in *Proceedings of the 35th International Conference on Machine Learning* (New Orleans, LA), 5453–5462.

Yanardag, P., and Vishwanathan, S. (2015). "Deep graph kernels," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (New York, NY), 1365–1374.

Yang, M., Simm, J., Lam, C. C., Zakeri, P., van Westen, G. J. P., Moreau, Y., et al. (2018). Linking drug target and pathway activation for effective therapy using multi-task learning. *Sci. Rep.* 8:8322. doi: 10.1038/s41598-018-25947-y

Ying, Z., You, J., Morris, C., Ren, X., Hamilton, W., and Leskovec, J. (2018). "Hierarchical graph representation learning with differentiable pooling," in *Advances in Neural Information Processing Systems* (Montreal, QC), 4800–4810.

Zhang, M., Cui, Z., Neumann, M., and Chen, Y. (2018). "An end-to-end deep learning architecture for graph classification," in *Proceedings of the 32nd AAAI Conference on Artificial Intelligence* (New Orleans, LA), 4438–4445.

Zou, D., and Lerman, G. (2019). Graph convolutional neural networks via scattering. *Appl. Comput. Harmon. Anal.* doi: 10.1016/j.acha.2019.06.003

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.

frontiers in Neuroscience

ORIGINAL RESEARCH
published: 14 January 2020
doi: 10.3389/fnins.2019.01408

A Two-Level Transfer Learning Algorithm for Evolutionary Multitasking

Xiaoliang Ma[1,2,3†], Qunjian Chen[1,2,3†], Yanan Yu[1,2,3], Yiwen Sun[4], Lijia Ma[1,2,3] and Zexuan Zhu[1,2,3]*

[1] College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China, [2] Guangdong Laboratory of Artificial Intelligence and Digital Economy (SZ), Shenzhen University, Shenzhen, China, [3] National Engineering Laboratory for Big Data System Computing Technology, Shenzhen University, Shenzhen, China, [4] School of Medicine, Shenzhen University, Shenzhen, China

Different from conventional single-task optimization, the recently proposed multitasking optimization (MTO) simultaneously deals with multiple optimization tasks with different types of decision variables. MTO explores the underlying similarity and complementarity among the component tasks to improve the optimization process. The well-known multifactorial evolutionary algorithm (MFEA) has been successfully introduced to solve MTO problems based on transfer learning. However, it uses a simple and random inter-task transfer learning strategy, thereby resulting in slow convergence. To deal with this issue, this paper presents a two-level transfer learning (TLTL) algorithm, in which the upper-level implements inter-task transfer learning via chromosome crossover and elite individual learning, and the lower-level introduces intra-task transfer learning based on information transfer of decision variables for an across-dimension optimization. The proposed algorithm fully uses the correlation and similarity among the component tasks to improve the efficiency and effectiveness of MTO. Experimental studies demonstrate the proposed algorithm has outstanding ability of global search and fast convergence rate.

Keywords: evolutionary multitasking, multifactorial optimization, transfer learning, memetic algorithm, knowledge transfer

OPEN ACCESS

Edited by:
Huajin Tang,
Zhejiang University, China

Reviewed by:
Jinghui Zhong,
South China University of Technology, China
Zhou Wu,
Chongqing University, China

****Correspondence:***
Zexuan Zhu
zhuzx@szu.edu.cn

† These authors have contributed equally to this work

Specialty section:
This article was submitted to Decision Neuroscience, a section of the journal Frontiers in Neuroscience

Received: *25 August 2019*
Accepted: *12 December 2019*
Published: *14 January 2020*

Citation:
Ma X, Chen Q, Yu Y, Sun Y, Ma L and Zhu Z (2020) A Two-Level Transfer Learning Algorithm for Evolutionary Multitasking. Front. Neurosci. 13:1408. doi: 10.3389/fnins.2019.01408

INTRODUCTION

In recent years, the development of evolutionary computation has attracted extensive attention. Based on the Darwinian theorem of "Survival of the Fittest" (Dawkins, 2006; Ma et al., 2014a), the population-based evolutionary algorithms (EAs) have been successfully used to solve a wide range of optimization problems (Deb, 2001; Qi et al., 2014; Ma et al., 2018). Multitasking optimization (MTO) problems have emerged as a new interest in the area of evolutionary computation (Da et al., 2016; Gupta et al., 2016a; Ong and Gupta, 2016; Yuan et al., 2016). Inspired by the ability of human beings to process multiple tasks at the same time, MTO aims at dealing with different optimization tasks simultaneously within a single solution framework. MTO introduces implicit transfer learning across different optimization tasks to improve the solving of each task (Gupta and Ong, 2016; Gupta et al., 2016b). If the component tasks in an MTO problem possess some commonalities and similarities, sharing knowledge among these optimization tasks is helpful to solve the whole MTO problems (Bali et al., 2017; Yuan et al., 2017).

Transfer learning is a new machine learning method that has caught increasing attention in recent years (Pan and Yang, 2010; Tan et al., 2017). It focuses on solving the target problem by applying the existing knowledge learned from other related problems (Gupta et al., 2018). In general, the more commonalities and similarities are shared between the source problem and target problem, the more effectively the transfer learning work for them. Multifactorial evolutionary algorithm (MFEA) is the first work to introduce transfer learning into the domain of evolutionary computation to deal with MTO problem (Gupta and Ong, 2016). In MFEA, the knowledge is implicitly transferred through chromosomal crossover (Gupta and Ong, 2016). As a general framework, MFEA uses a simple inter-task transfer learning by assortative mating and vertical cultural transmission with randomness, which tends to suffer from excessive diversity thereby leading to a slow convergence speed (Hou et al., 2017).

To deal with the aforementioned issues of MFEA, this paper proposes a two-level transfer learning (TLTL) framework in MTO. The upper level performs inter-task knowledge transfer via crossover and exploits the knowledge of the elite individuals to reduce the randomness, which is expected to enhance the search efficiency. The lower level is an intra-task knowledge transfer for transmitting information from one dimension to other dimensions within the same optimization task. The two levels cooperate with each other in a mutually beneficial fashion. The experimental results on various MTO problems show that the proposed algorithm is capable of obtaining high-quality solutions compared with the state-of-the-art evolutionary MTO algorithms.

In the rest of this paper, section "Background and Related Work" introduces the background of MTO and MFEA as well as the related work of transfer learning in evolutionary computation. The proposed TLTL algorithm is described in section "Method." Section "Experimental Methodology" presents the MTO test problems. The comparison results between the proposed algorithm and the state-of-the-art evolutionary multitasking algorithms are shown in section "Results." Finally, section "Discussion and Conclusion" concludes this work and points out some potential future research directions.

BACKGROUND AND RELATED WORK

This section introduces the basics of MTO and MFEA, and the related work of Evolutionary MTO.

Multitasking Optimization

The main motivation of MTO is to exploit the inter-task synergy to improve the problem solving. The advantage of MTO over the counterpart single-task optimization in some specific problems has been demonstrated in the literature (Xie et al., 2016; Feng et al., 2017; Ramon and Ong, 2017; Wen and Ting, 2017; Zhou et al., 2017).

Without loss of generality, we consider a scenario in which K distinct minimization tasks are solved simultaneously. The j-th task is labeled T_j, and its objective function is defined as $F_j(x) : X_j \rightarrow R$. In such setting, MTO aims at searching the space of all optimization tasks concurrently for $\left\{x_1^*, \ldots, x_k^*\right\} = \text{argmin}\{F_1(x_1), \ \ldots, \ F_K(x_k)\}$, where each x_j^* is a feasible solution in decision space X_j. To compare solution individuals in the MFEA, it is necessary to assign new fitness for each population member p_i based on a set of properties as follows (Gupta and Ong, 2016).

Definition 1 (Factorial Cost)

The *factorial cost* of an individual is defined as $\alpha_{ij} = \gamma\delta_{ij} + F_{ij}$, where F_{ij} and δ_{ij} are the objective value and the total constraint violation of individual p_i on optimization task T_j, respectively. The coefficient γ is a large penalizing multiplier.

Definition 2 (Factorial Rank)

For an optimization task T_j, the population individuals are sorted in ascending order with respect to the *factorial cost*. The *factorial rank* r_{ij} of an individual p_i on optimization task T_j is the index value of p_i in the sort list.

Definition 3 (Skill Factor)

The *skill factor* τ_i of an individual p_i is the component task on which p_i performs the best $\tau_i = \text{argmin}\{r_{ij}\}$.

Definition 4 (Scalar Fitness)

The *scalar fitness* of an individual p_i in a multitasking environment is calculated by $\beta_i = \max\{1/r_{i1}, \ \ldots, \ 1/r_{iK}\}$.

Multifactorial Evolutionary Algorithm

This subsection briefly introduces MFEA (Gupta and Ong, 2016), which is the first evolutionary MTO algorithm inspired by the work (Cloninger et al., 1979). MFEA evaluates a population of N individuals in a unified search space. Each individual in the initial population is pre-assigned a dominant task randomly. In the process of evolution, each individual is only evaluated with respect to one task to reduce the computing resource consumption. MFEA uses typical crossover and mutation operators of classical EAs to the population. Elite individuals for each task in the current generation are selected to form the next generation.

The knowledge transfer in MFEA is implemented through assortative mating and vertical cultural transmission (Gupta and Ong, 2016). If two parent individuals assigned to different skill factor are selected for reproduction, the dominant tasks, and genetic material of offspring inherit from their parent individuals randomly. MFEA uses a simple inter-task transfer learning and has strong randomness.

Evolutionary Multitasking Optimization

Transfer learning is one active research field of machine learning, where the related knowledge in source domain is used to help the learning of the target domain. Many transfer learning techniques have been proposed to enable EAs to solve MTO problems. For example, the cross-domain MFEA, i.e., MFEA, solves multi-task optimization problems using implicit transfer learning in crossover operation. Wen and Ting (2017) proposed a utility detection of information sharing and a resource

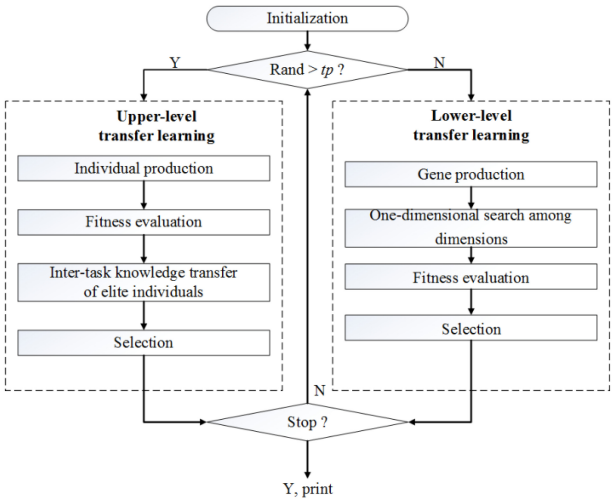

FIGURE 1 | The general flowchart of proposed algorithm.

redistribution method to reduce resource waste of MFEA. Yuan et al. (2017) presented a permutation-based MFEA (P-MFEA) for multi-tasking vehicle routing problems. Unlike the original MFEA using a random-key representation, P-MFEA adopts a more effective permutation-based unified representation. Zhou et al. (2017) suggested a novel MFEA for combinatorial MTO problems. They developed two new mechanisms to improve search efficiency and decrease the computational complexity, respectively. Xie et al. (2016) enhanced the MFEA based on particle swarm optimization (PSO). Feng et al. (2017) developed a MFEA with PSO and differential evolution (DE). Bali et al. (2017) put forward a linearized domain adaptation strategy to deal with the issue of the negative knowledge transfer between uncorrelated tasks. Ramon and Ong (2017) presented a multi-task evolutionary algorithm for search-based software test data generation. Their work is the first attempt to demonstrate the feasibility of MFEA for solving real-world problems with more than two tasks. Da et al. (2016) advanced a benchmark problem set and a performance index for single-objective MTO. Yuan et al. (2016) designed a benchmark problem set for multi-objective MTO that can facilitate the development and comparison of MTO algorithms. Hou et al. (2017) proposed an evolutionary transfer reinforcement learning framework for multi-agent intelligent system, which can adapt to the dynamic environment. Tan et al. (2017) introduced an adaptive knowledge reuse framework across expensive multi-objective optimization problems. Multi-problem surrogates were proposed to reuse knowledge gained from distinct but related problem-solving experiences. Gupta et al. (2018) discussed the recent studies on global black-box optimization via knowledge transfer across different problems, including sequential transfer, multitasking, and multiform optimization. For a general survey of transfer learning, the reader is referred to Pan and Yang (2010).

METHOD

This section introduces the TLTLA algorithm for MTO. The upper level is an inter-task knowledge learning, which uses the inter-task commonalities and similarities to improve the efficiency of cross-task optimization. The lower level transfer learning focuses on intra-task knowledge learning, which transmits the information from one dimension to other dimensions to accelerate the convergence. The general flowchart of the proposed algorithm is shown in **Figure 1**.

At the beginning of TLTLA, the individuals in the population are initialized with a unified coding scheme. Let *tp* indicate the inter-task transfer learning probability. If a generated random value is greater than *tp*, the algorithm goes through four steps to complete the inter-task transfer learning process. The parent population produces offspring population by crossover operator

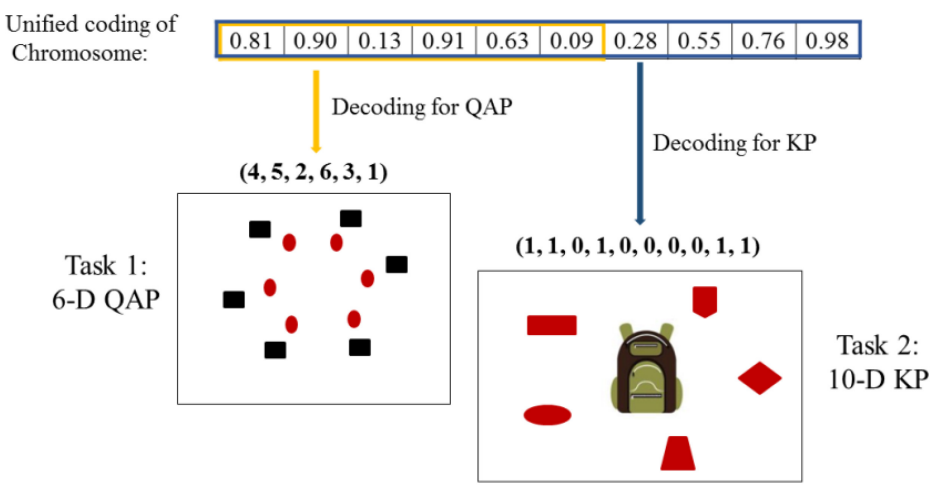

FIGURE 2 | The unified coding and different decoding in multi-tasking optimization with quadratic assignment problem (QAP) and knapsack problem (KP).

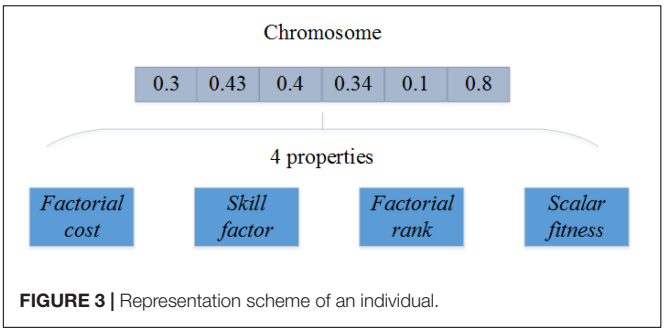

FIGURE 3 | Representation scheme of an individual.

and mutate operator. In chromosome crossover, part of the knowledge transfer is realized with the random inheritance of culture and gene from parent to children. However, this pattern is accompanied by strong randomness. To deal with this issue, this paper suggests knowledge transfer of inter-task elite individuals. Finally, the individuals with high fitness are selected into the next generation. If the generated random value is less than *tp*, the algorithm performs a local search based on intra-task knowledge transfer. According to the individual fitness and the elite selection operator, the algorithm executes 1-dimensional search using information from other dimensions. Detailed description of the above two processes are provided in the following subsections.

Encoding and Decoding

To facilitate the knowledge transfer in the multitasking environment, Gupta et al. (2016b) suggested using the unified individual coding scheme. Let K denote the number of distinct component tasks in the multitasking environment, the search space dimension of the i-th task is denoted as D_i. Through the unified processing, the number of decision variables of every chromosome is set to $D_{MTO} = \max\{D_i\}$. Each decision variable in a chromosome is normalized in the range [0, 1] as shown in **Figure 2**. Conversely, in the phase of decoding, each chromosome can be decoded into a task-specific solution representation. For the i-th task T_i, we extract D_i decision variables from the chromosome, and decoded these decision variables into a feasible solution for the optimization tasks T_i. In general, the extracted part is the first D_i decision variables of the chromosome.

Initialization

In the initialization, a population p_0 of N individuals is generated randomly by using a unified coding scheme. Every individual is encoded in a chromosome and associated with a set of properties including *factorial cost*, *skill factor*, *factorial rank*, and *scalar fitness*. The four properties have been described in section "Background and Related Work." Representation scheme of an individual is shown in **Figure 3**.

In such a setting, considering K optimization tasks in the initial multitasking environment, we assign the equal computation resource to each component task. In other words, the subpopulation of each component task is composed by N/K individuals in the evolutionary process.

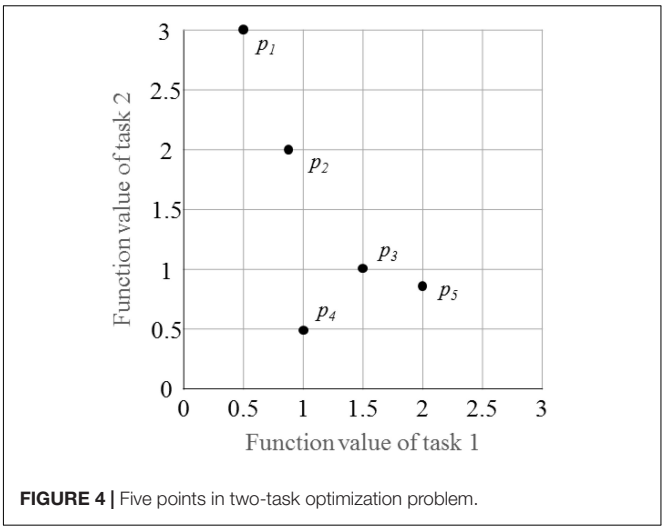

FIGURE 4 | Five points in two-task optimization problem.

Fitness Evaluation

In a multitasking environment, an individual may optimize one or multiple optimization tasks. Herein, a generic way is used to calculate the fitness of each individual (Gupta and Ong, 2016). **Figure 4** and **Table 1** illustrate the fitness assignment of the individuals in a two-task optimization problem.

As shown in **Figure 4**, five individuals and their corresponding fitness function values on different tasks are given. According to the definitions of four properties described in the section "Background and Related Work," the corresponding values are shown in **Table 1**. For example, individual p_2 has factorial costs 0.8 and 2 on component tasks T_1 and T_2, respectively. After sorting all individuals based on their factorial costs in ascending order, the factorial ranks of individual p_2 on tasks T_1 and T_2 are 2 and 4, respectively. Thus, the final scalar fitness and skill factor of individual p_2 are $1/2 = \max\{1/2,\ 1/4\}$ and T_1, respectively.

Inter-Task Knowledge Transfer

This subsection describes the inter-task transfer learning in **Algorithm 1**, which enables the discovery and transfer of existing genetic material from one component task to another. Individuals in the multitasking environment may have different cultural backgrounds, i.e., different skill factors. When the cultural background of an individual is changed, the individual

TABLE 1 | The results of calculating individual fitness.

Individual	Factorial cost		Factorial rank		Skill factor	Scalar fitness
	α_{i1}	α_{i2}	r_{i1}	r_{i2}	τ_i	β_i
p_1	0.5	3	1	5	T_1	1
p_2	**0.8**	**2**	**2**	**4**	**T_1**	**1/2**
p_3	1.5	1	4	3	T_2	1/3
p_4	1	0.5	3	1	T_2	1
p_5	2	0.8	5	2	T_2	1/2

Bold values in the case used to explain the concept of individual fitness evaluation in multitasking environment.

is transferred from one task to another (Gupta and Ong, 2016). One of the drawbacks in MFEA is the strong randomness in its inter-task knowledge transfer. To deal with this issue, an elite individual transfer is proposed in this subsection.

Algorithm 1: Inter-task transfer learning.
Require:
Pt, the current population;
rmp, the balance factor between crossover and mutation;
N, the population size;
K, the number of component tasks.

1. **for** $i = 1$ to $N/2$ **do**
2. Randomly choose parents (pa, pb) from P_t
3. **if** ($\tau a == \tau b$) or ($rand < rmp$)
4. (ca, cb) = crossover on (pa, pb)
5. ca and cb randomly inherits τa or τb
6. **else**
7. ca = mutation in (pa) and cb = mutation on (pb)
8. ca inherits (τa) and cb inherits (τb)
9. **end if**
10. **end for**
11. **for** $i = 1$ to N **do**
12. Evaluate ci on task τi
13. **end for**
14. Compute *factorial rank* for all individuals
15. Record elite individuals (*factorial rank* == 1) as $Bt = \{b1,\ldots,bK\}$ and set
16. **for** $i = 1$ **to** K
17. Evaluate bi on task τ_r, where $r = rand(K)$ and $r != i$
18. Put the evaluated individualinto
19. **end for**
20. $R_t = C_t \cup P_t \cup B_t^r$
21. Compute scalar fitness for all individuals
22. Select N elite individuals from R_t to P_{t+1}
23. Set $t = t+1$

There are two ways of inter-task individual transfer in **Algorithm 1**. One is implicit genetic transfer through chromosomal crossover as shown in line 5 (Gupta and Ong, 2016). If two parent individuals with different cultural backgrounds undergo crossover, their offspring can inherit from one of them (Cavallisforza and Feldman, 1973; Gupta and Ong, 2016). The other is the elite individual transfer among tasks, which interchanges the skill factor of the best individuals among tasks in lines 17. If multiple optimization tasks are of commonality and similarities, a good solution to one task is also expected to have a good performance on other tasks. To reduce resource consumption, this operation is applied to the best individuals only.

Individual Production

In inter-task transfer learning, the proposed algorithm uses the simulated binary crossover (SBX) (Deb and Agrawal, 1994; Ma et al., 2016b) operator and the polynomial mutation (Ma et al., 2016a) operator to produce the offspring population.

In lines 2–9 of **Algorithm 1**, assortative mating and vertical cultural transmission are performed in the parent pool. Specifically, two randomly selected parent individuals undergo crossover or mutation based on the balance factor *rmp*. In the crossover operation, the mating of parent individuals with different skill factor may lead to the emergence of genetic transfer (Cavallisforza and Feldman, 1973; Feldman and Laland, 1996). Each child imitates the skill factor from one of the two parent individuals randomly. The random inheritance mechanism can be considered as an inter-task knowledge transfer, which shares relevant information for promoting population evolution.

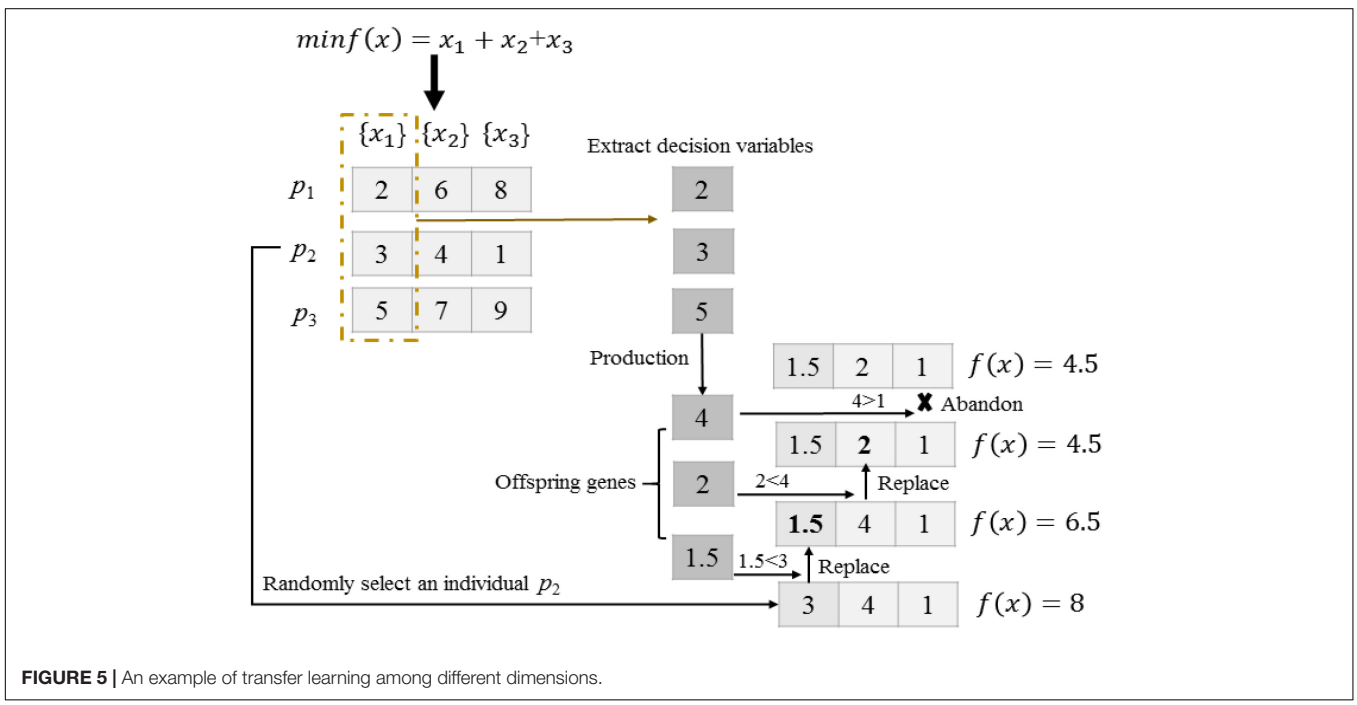

FIGURE 5 | An example of transfer learning among different dimensions.

TABLE 2 | Nine bi-tasking benchmark problems.

Two-task problem	Intersection and similarity	Task	R_s	D_T	Global optimal	Landscape
1	CI+HS	Griewank	1.0000	50	$(0, 0, \ldots, 0) \in [-100, 100]^{50}$	Multimodal+Non-separable
		Rastrigin		50	$(0, 0, \ldots, 0) \in [-50, 50]^{50}$	Multimodal+Non-separable
2	CI+MS	Ackley	0.2261	50	$(0, 0, \ldots, 0) \in [-50, 50]^{50}$	Multimodal+Non-separable
		Rastrigin		50	$(0, 0, \ldots, 0) \in [-50, 50]^{50}$	Multimodal+Non-separable
3	CI+LS	Ackley	0.0002	50	$(42.0969, \ldots, 42.0969) \in [-50, 50]^{50}$	Multimodal+Non-separable
		Schwefel		50	$(420.9687, \ldots, 420.9687) \in [-500, 500]^{50}$	Multimodal+Separable
4	PI+HS	Rastrigin	0.8670	50	$(0, 0, \ldots, 0) \in [-50, 50]^{50}$	Multimodal+Non-separable
		Sphere		50	$(0, \ldots, 0, 20, \ldots, 20) \in [-100, 100]^{50}$	Unimodal+Separable
5	PI+MS	Ackley	0.2154	50	$(0, \ldots, 0, 1, \ldots, 1) \in [-50, 50]^{50}$	Multimodal+Non-separable
		Rosenbrock		50	$(1, 1, \ldots, 1) \in [-50, 50]^{50}$	Multimodal+Non-separable
6	PI+LS	Ackley	0.0725	50	$(0, 0, \ldots, 0) \in [-50, 50]^{50}$	Multimodal+Non-separable
		Weierstrass		25	$(0, 0, \ldots, 0) \in [-0.5, 0.5]^{25}$	Multimodal+Non-separable
7	NI+HS	Rosenbrock	0.9434	50	$(1, 1, \ldots, 1) \in [-50, 50]^{50}$	Multimodal+Non-separable
		Rastrigin		50	$(0, 0, \ldots, 0) \in [-50, 50]^{50}$	Multimodal+Non-separable
8	NI+MS	Griewank	0.3669	50	$(10, 10, \ldots, 10) \in [-100, 100]^{50}$	Multimodal+Non-separable
		Weierstrass		50	$(0, 0, \ldots, 0) \in [-0.5, 0.5]^{50}$	Multimodal+Non-separable
9	NI+LS	Rastrigin	0.0016	50	$(0, 0, \ldots, 0) \in [-50, 50]^{50}$	Multimodal+Non-separable
		Schwefel		50	$(420.9687, \ldots, 420.9687) \in [-500, 500]^{50}$	Multimodal+Separable

Inter-Task Knowledge Transfer of Elite Individuals

Due to the strong randomness of assortative mating and vertical cultural transmission, population evolution has some limitations in the global search and convergence. In lines 15–19 of **Algorithm 1**, an elite individual transfer is introduced to alleviate this issue.

In each generation, the best individual of each component task (i.e., the factorial rank of this individual is 1) is recorded in line 15. Considering the commonalities and similarities among different tasks, a new skill factor for each best individual is assigned and evaluated with respect to the new task. The inter-task knowledge transfer of elite individuals is shown in line 17. If multiple optimization tasks are of strong commonalities and similarities, a good solution of one task is also expected to have good performance on the other tasks.

Evaluation and Selection

As shown in line 20, the combined population R_t consists of parent population P_t, offspring population C_t, and learned individuals B_t^r. An elitist selection operator is used and the individuals with higher scalar fitness are selected into the next generation in line 22.

Intra-Task Knowledge Transfer

Besides, inter-task transfer learning, the proposed algorithm is also characterized with intra-task transfer learning as shown in **Algorithm 2**. The intra-task transfer learning transmits the knowledge from one dimension to other dimensions within the same task. The proposed cross-dimensional one-dimensional search complements well with SBX and is expected to prevent the algorithm from getting trapped in local optima.

Algorithm 2: Intra-task transfer learning.
Require:
P_t, the current population;
S, the number of variables in unified individual coding.

1. **for** $i = 1$ to S **do**
2. Randomly select an individual p_r from P_t
3. $Off\ (1, S)$ = differential evolution on $\{x_i\}$
4. **for** $j = 1$ to S **do**
5. $d_j = (p_r\ (1), \ldots, p_r(j\text{-}1), Off\ (j), p_r\ (j+1), \ldots, p_r\ (S))$
6. Evaluate d_j on task τ_{p_r}
7. **if** d_j is better than p_r
8. $p_r(j) = Off\ (j)$
9. **end if**
10. **end for**
11. **end for**

One-Dimensional Mutation

At the beginning of **Algorithm 2**, an individual is randomly selected from the current population in line 2. In line 3, S offspring genes $[Off(1), \ldots, Off(S)]$ are generated by DE mutation operator (Qin and Suganthan, 2005; Ma et al., 2014b,c), with the parent genes coming from the i-th dimension variable x_i of the population.

One-Dimensional Search Among Dimensions

As shown in lines 4–10 of **Algorithm 2**, S offspring are iteratively used to compare with the S variables of the selected individual p_r as shown in **Figure 5**. Three individuals with the same dominant task are given in the search space. Firstly, we randomly select an individual p_2 from the current population. Secondly, three decision variables 2, 3, and 5 are extracted in the 1st dimension of individuals p_1, p_2, and p_3, respectively. Thirdly, three extracted decision variables undergo DE to generate three offspring genes 4, 2 and 1.5. Finally, the cross-dimensional search for individual p_2 is performed to find out improved solutions. Offspring genes 1.5 and 2 replace

TABLE 3 | The mean and standard deviation of function values obtained by TLTLA and MFEA on nine tri-tasking optimization problems.

Problem	TLTLA			MFEA		
	T_1	T_2	T_3	T_1	T_2	T_3
CI+HS++Ackley (50D)	**0.00E+00** (0)	**0.00E+00** (0)	**1.83E-13** (4.72E-13)	3.36E-01 (0.0650)	2.00E+02 (43.5807)	2.87E+00 (0.5167)
CI+MS++Schwefel (50D)	**2.75E-12** (9.29E-12)	**1.74E+01** (55.6038)	**2.96E+02** (1.32E+03)	5.26E+00 (0.8443)	2.68E+02 (58.3610)	3.77E+03 (497.5763)
CI+LS++Weierstrass (25D)	**9.20E-12** (2.09E-11)	**6.36E-04** (1.11E-19)	**6.64E-01** (1.2568)	2.02E+01 (0.0738)	3.91E+03 (583.5658)	2.03E+01 (2.1087)
PI+HS++Ackley (50D)	**2.20E+01** (46.7283)	**7.85E-04** (0.0030)	**1.45E+00** (0.9410)	2.78E+02 (66.1748)	1.25E+01 (1.7731)	5.24E+00 (1.0121)
PI+MS++Schwefel (50D)	**1.05E+00** (1.0191)	**2.06E+01** (23.1907)	**2.96E+02** (1.32E+03)	3.76E+00 (0.5517)	8.96E+02 (206.5210)	3.94E+03 (413.8822)
PI+LS+ Rastrigin (50D)	**1.74E-12** (7.67E-12)	**1.98E-18** (7.90E-34)	**0.00E+00** (0)	4.91E+00 (1.0324)	5.42E+00 (1.1193)	2.45E+02 (41.2149)
NI+HS++Ackley (50D)	**3.54E+01** (20.1767)	**0.00E+00** (0)	**2.68E-14** (3.43E-14)	5.98E+02 (213.2004)	2.06E+02 (46.6145)	3.60E+00 (0.8252)
NI+MS++Rastrigin (50D)	**3.00E-12** (1.33E-11)	**1.04E-02** (0.0321)	**1.98E+01** (49.8744)	4.74E-01 (0.0784)	2.01E+01 (2.8085)	5.59E+02 (132.9283)
NI+LS++Griewank (50D)	**0.00E+00** (0)	**6.36E-04** (1.11E-19)	**0.00E+00** (0)	2.07E+02 (57.5701)	3.81E+03 (518.0790)	4.58E-01 (0.0671)

the parent genes 3 and 4, respectively, as they obtain better fitness. On the contrary, offspring gene 4 is abandoned as it attains no improvement.

Evaluation and Selection

The evaluation and selection of a temporary individual d_j constructed by the one-dimensional search are shown in lines 8–11. To reduce the number of function evaluations, the temporary individual d_j is evaluated only on task τ_{p_r}. In line 7, if the new constructed individual d_j is better than p_r in terms of fitness value, p_r is updated by d_j in line 8.

EXPERIMENTAL METHODOLOGY

The proposed TLTLA is compared with the state-of-the-art evolutionary MTO algorithms, i.e., MFDE (Feng et al., 2017), MFEA (Gupta and Ong, 2016), and SOEA (Gupta and Ong, 2016). The benchmark MTO problems (Da et al., 2016) are used to test the algorithms. All test problem are bi-tasking optimization problems. To verify the effectiveness of the compared algorithms, component tasks in MTO problems possess different types of correlation in Da et al. (2016). To demonstrate the scalability of the proposed algorithm on more complex problems, we also construct nine tri-tasking optimization problems in this study.

Optimization Functions

This section introduces seven elemental single-objective continuous optimization functions (Da et al., 2016) used to construct the MTO test problems. The specific definitions of these seven functions are shown as follows. In particular, the dimensionality of the search space is denoted as D.

(1) *Sphere:*

$$F_1(x) = \sum_{i=1}^{D} x_i^2, x \in [-100, 100]^D$$

(2) *Rosenbrock:*

$$F_2(x) = \sum_{i=1}^{D-1} (100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2), x \in [-50, 50]^D$$

(3) *Ackley:*

$$F_3(x) = -20\exp(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2}) -$$

$$\exp(\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi x_i)) + 20 + e, x \in [-50, 50]^D$$

TABLE 4 | The mean and standard deviation of function values obtained by four compared algorithms on nine bi-tasking optimization problems.

Problem	Task	TLTLA	MFDE	MFEA	SOEA
CI+HS	T_1	**0.00E + 00** (0)	1.00E − 03 (3.05E − 03)	3.73E − 01 (0.0617)	9.08E − 01 (0.0585)
	T_2	**0.00E + 00** (0)	2.61E + 00 (7.96)	1.95E + 02 (34.4953)	4.10E + 02 (49.0439)
CI+MS	T_1	**1.20**E − 14 (2.47E − 14)	1.00E − 03 (0.003)	4.39E + 00 (0.4481)	5.32E + 00 (1.2338)
	T_2	**0.00E + 00** (0)	3.00E − 03 (0.012)	2.27E + 02 (52.2778)	4.41E + 02 (65.0750)
CI+LS	T_1	**3.41E − 14** (1.21E − 14)	2.12E + 01 (0.04)	2.02E + 01 (0.0798)	2.12E + 01 (0.2010)
	T_2	**6.36E − 04** (1.11E − 19)	1.84E + 04 (1578.16)	3.70E + 03 (429.1093)	4.18E + 03 (657.2786)
PI+HS	T_1	**2.88E + 01** (62.1998)	7.83E + 01 (15.37)	6.14E + 02 (131.0438)	4.45E + 02 (57.2891)
	T_2	**9.63E − 08** (3.86E − 07)	2.20E − 05 (2.90E − 05)	1.01E + 01 (2.4734)	8.40E + 01 (17.1924)
PI+MS	T_1	1.02E + 00 (1.1088)	**1.00E − 03** (0.001)	3.49E + 00 (0.6289)	5.07E + 00 (0.4417)
	T_2	**2.65E + 01** (24.5602)	6.03E + 01 (20.53)	7.02E + 02 (267.8668)	2.40E + 04 (10487.2597)
PI+LS	T_1	**1.60E − 12** (4.90E − 12)	4.60E − 01 (0.58)	2.00E + 01 (0.1302)	5.05E + 00 (0.6299)
	T_2	**1.59E − 14** (6.32E − 14)	2.20E − 01 (0.47)	1.93E + 01 (1.7291)	1.32E + 01 (2.3771)
NI+HS	T_1	**3.52E + 01** (20.8321)	8.93E + 01 (48.60)	1.01E + 03 (346.1264)	2.43E + 04 (5842.0394)
	T_2	**2.54E + 00** (11.3913)	2.05E + 01 (15.41)	2.87E + 02 (92.4182)	4.48E + 02 (61.1642)
NI+MS	T_1	**5.55E − 17** (2.23E − 16)	2.03E − 03 (0.0042)	4.20E − 01 (0.0654)	9.08E − 01 (0.0702)
	T_2	**1.35E − 03** (0.0030)	2.97E + 00 (1.08)	2.71E + 01 (2.6883)	3.70E + 01 (3.4558)
NI+LS	T_1	**3.85E + 01** (89.1612)	9.62E + 01 (20.02)	6.51E + 02 (98.6871)	4.37E + 02 (62.6339)
	T_2	**6.36E − 04** (7.31E − 10)	3.94E + 03 (730.99)	3.62E + 03 (325.0275)	4.14E + 03 (524.4335)

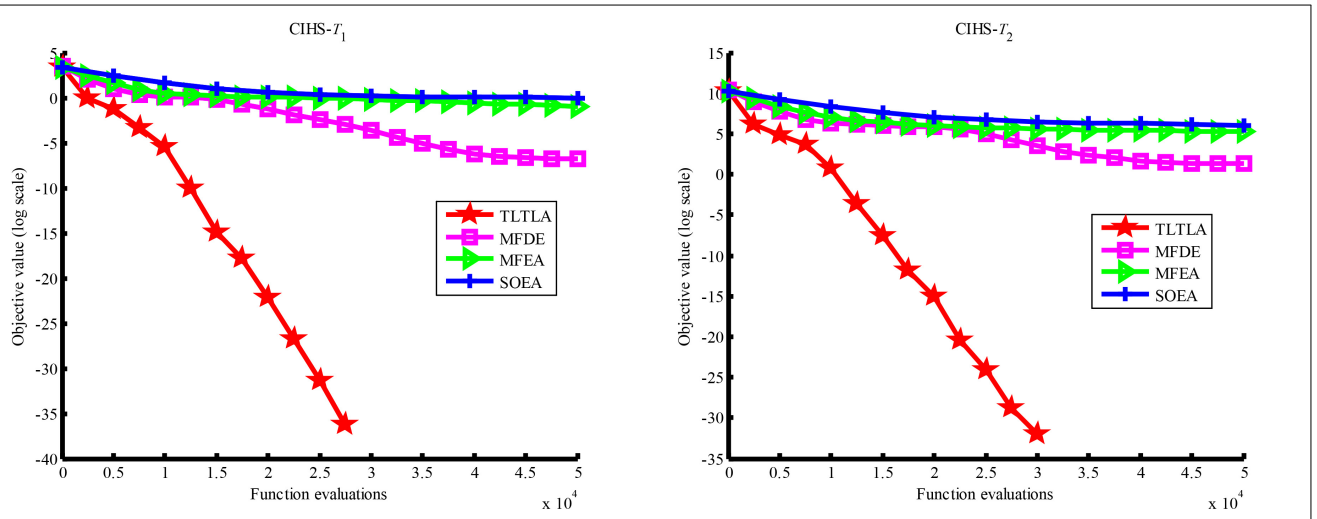

FIGURE 6 | Convergence trends of tasks in CI+HS.

(4) *Rastrigin*:

$$F_4(x) = \sum_{i=1}^{D}(x_i^2 - 10\cos(2\pi x_i) + 10), x \in [-50, 50]^D$$

(5) *Schwefel*:

$$F_5(x) = 418.9829 \times D - \sum_{i=1}^{D} x_i \sin(|x_i|^{\frac{1}{2}}), x \in [-500, 500]^D$$

(6) *Griewank*:

$$F_6(x) = 1 + \frac{1}{4000}\sum_{i=1}^{D} x_i^2 - \prod_{i=1}^{D}\cos(\frac{x_i}{\sqrt{i}}), x \in [-100, 100]^D$$

(7) *Weierstrass*:

$$F_7(x) = \sum_{i=1}^{D}(\sum_{k=0}^{k_{\max}}[a^k \cos(2\pi b^k(x_i + 0.5))]) - D\sum_{k=0}^{k_{\max}}$$

$$[a^k \cos(2\pi b^k \cdot 0.5)]a = 0.5, b = 3, k_{\max} = 20, x \in [-0.5, 0.5]^D$$

Two Multitasking Optimization Problem Sets

The nine bi-tasking optimization problems were first proposed in Da et al. (2016), based on which nine tri-tasking optimization problems are constructed in this paper. The properties of the bi-tasking optimization problems are summarized in **Table 2**,

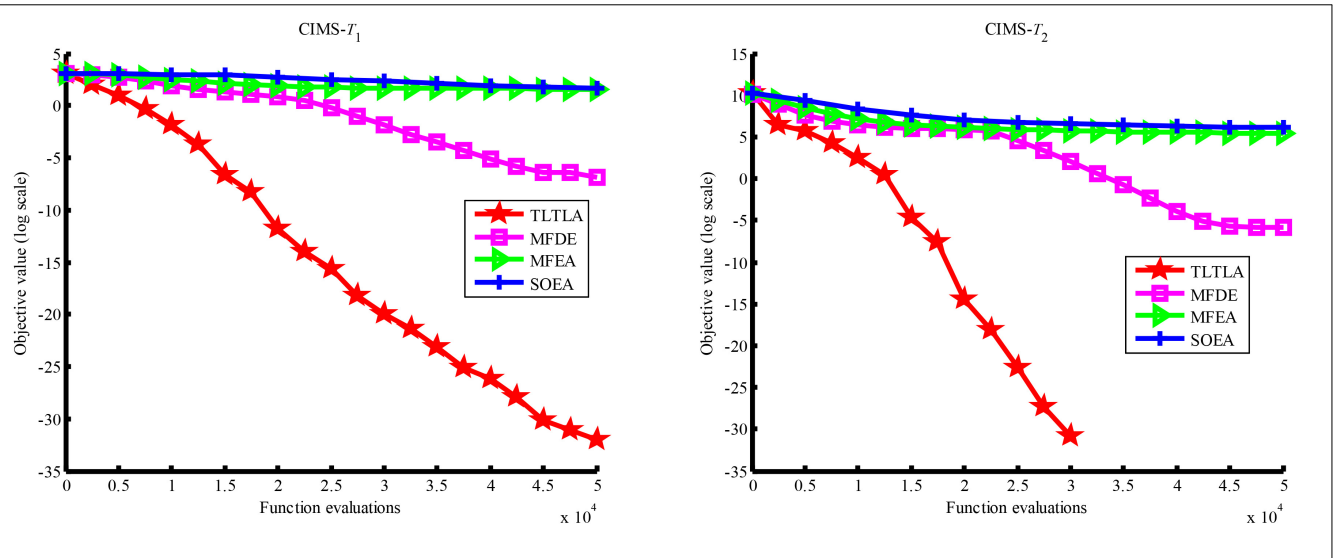

FIGURE 7 | Convergence trends of tasks in CI+MS.

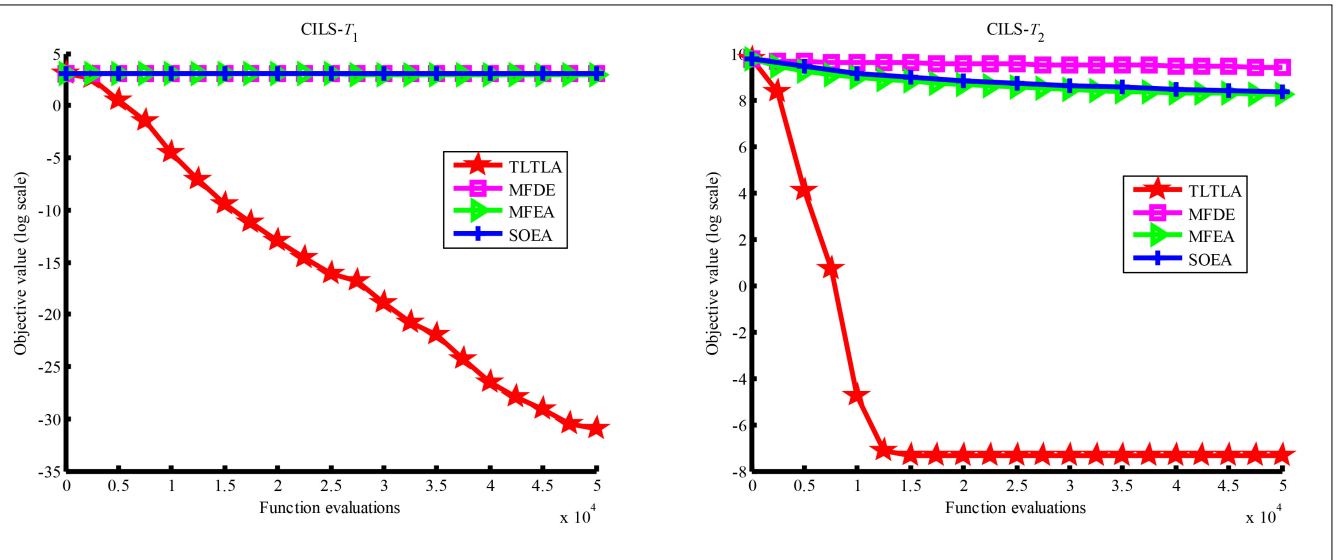

FIGURE 8 | Convergence trends of tasks in CI+LS.

which clearly shows the commonalities and similarities among component tasks.

For the global optimal solutions of the two component tasks, complete intersection (CI) indicates that the global optima of the two optimization tasks are identical on all variables in the unified search space. No intersection (NI) means that the global optima of the two optimization tasks are different on all variables in the unified search space. Partial intersection (PI) suggests that the global optima of the two tasks are the same on a subset of variables in the unified search space.

The similarity (R_s) of a pair of optimization tasks are divided into three categories (Da et al., 2016). According to the Spearmans rank correlation similarity metric [40], $R_s < 0.2$ indicates low similarity (LS), $0.2 < R_s < 0.8$ means medium similarity (MS), and $R_s > 0.8$ denotes high similarity (HS).

In addition to the above nine bi-tasking optimization problems, this paper attempts to solve tri-tasking optimization problems. Nine constructed tri-tasking optimization problems are shown in **Table 3**.

RESULTS

Experimental Results on Bi-Task Optimization Problems

On the nine bi-tasking optimization problems, the population size is set to $N = 100$ for TLTLA, MFDE, MFEA, and SOEA. The maximum number of function evaluations is set to be 50,000 for SOEA and 100,000 for TLTLA, MFDE, and MFEA. Since SOEA is a single-tasking algorithm, it has to be run twice on bi-tasking

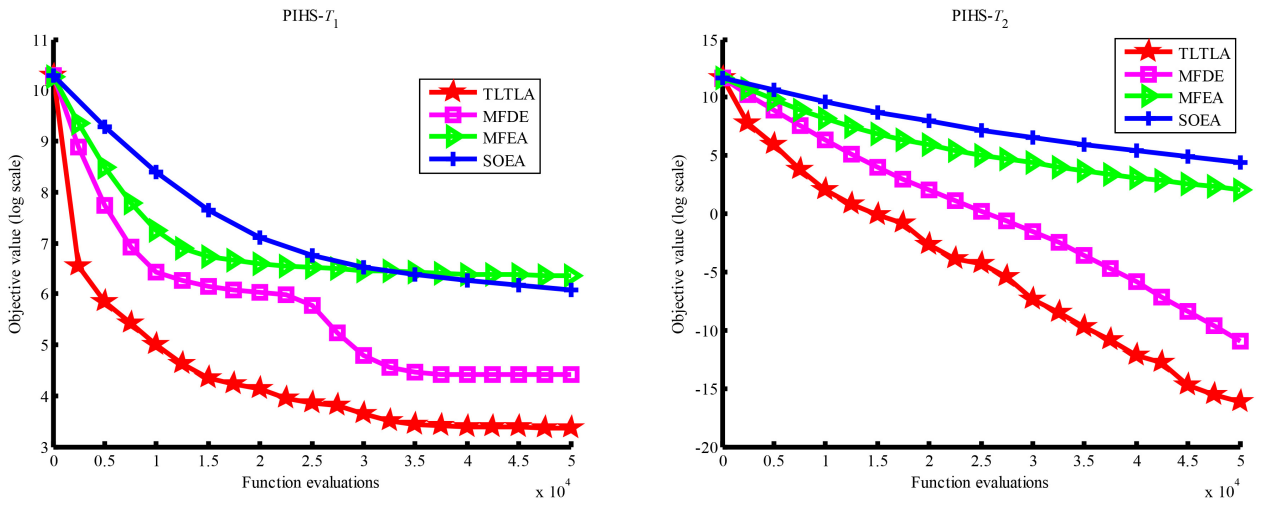

FIGURE 9 | Convergence trends of tasks in PI+HS.

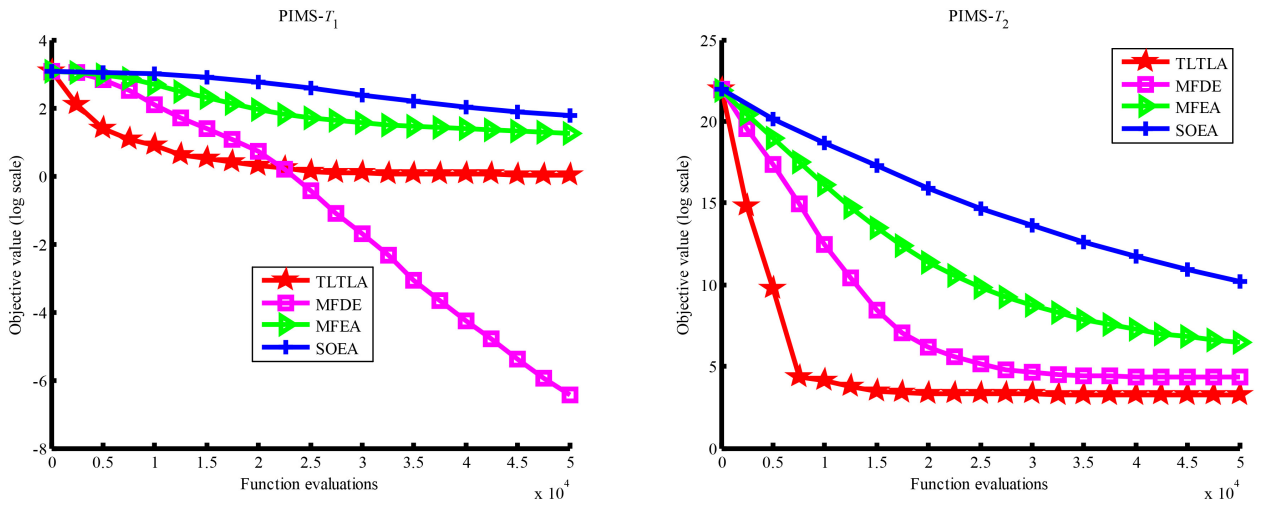

FIGURE 10 | Convergence trends of tasks in PI+MS.

problems. As such, SOEA consumes the same computational budget with other algorithms. All compared algorithms are performed in 20 independent runs on each MTO problem. The balance factor between crossover and mutation is set to $rmp = 0.3$ in TLTLA, MFDE, and MFEA.

Table 4 presents the mean and standard deviation of function values obtained by the four compared algorithms on nine bi-tasking optimization problems. The best mean function value on each task is highlighted in bold. Compared with MFEA, MFDE and SOEA, TLTLA obtains much better performance. TLTLA obtains the best results in 17 out of 18 independent optimization tasks, except the task T_1 of the PI+MS problem. To study the search efficiency of TLTLA, MFDE, MFEA, and SOEA, **Figures 6–14** show the convergence trends of all compared algorithms on the representative optimization tasks. In terms of convergence rate, TLTLA obtains a better overall performance than MFDE, MFEA, and SOEA on most of optimization tasks.

On the MTO problems with the high inter-task similarity or complementarity, such as CI+HS, CI+MS, CI+LS, PI+HS, and NI+HS, as shown in **Tables 2**, **4**, TLTLA performs much better than MFEA, MFDE and SOEA in terms of solution quality. In particular, TLTLA obtains the corresponding global optimum 0 on tasks $T_1$and T_2 of CI+HS and task T_2 of CI+MS. Three MTO algorithms, i.e., TLTLA, MFEA, and MFDE, work better than the traditional single-task optimization algorithm SOEA thanks to the use of inter-task knowledge transfer. However, the knowledge transfer in MFEA and MFDE is of strong randomness. TLTLA handles this issue by the inter-task elite individual transfer and intra-task cross-dimensional search. The inter-task elite individual transfer is more suitable for MTO

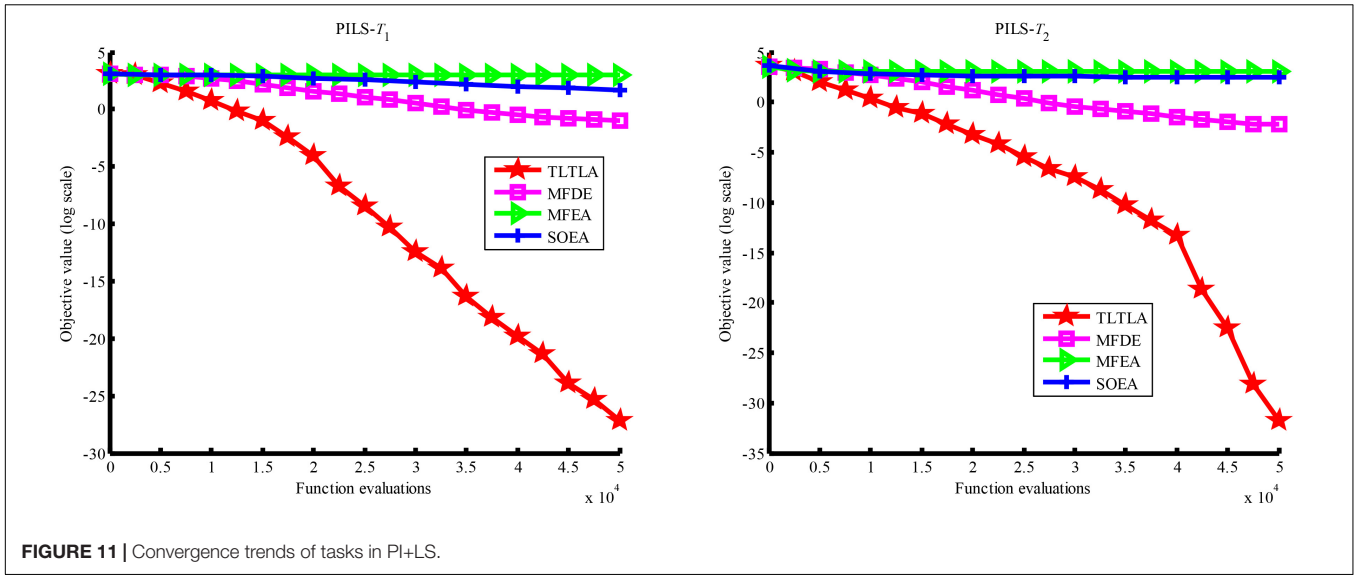

FIGURE 11 | Convergence trends of tasks in PI+LS.

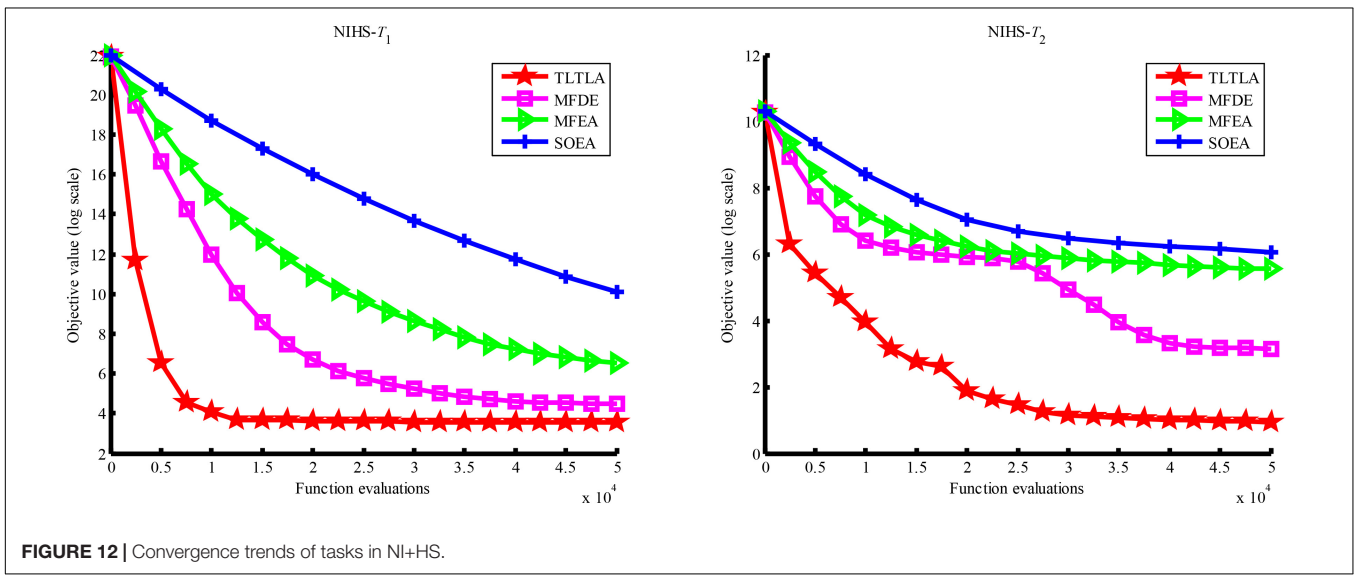

FIGURE 12 | Convergence trends of tasks in NI+HS.

problems with CI, i.e., the global optima of two component optimization tasks are identical in the unified search space. The intra-task transfer learning can improve the population diversity and complement well with SBX.

On some MTO problems, the component tasks have different number and/or different kinds of decision variables, such as PI+LS problem. Let one of the component tasks be α-dimensional and the other be β-dimensional (supposing $\alpha < \beta$). Therefore, all the individuals in the unified search space are encoded by β decision variables. Using cross-dimensional search, TLTLA is able to utilize the information of the extra $\beta - \alpha$ decision variables to optimize the α-dimensional component task, which is ignored by the other compared algorithms. This may be the reason TLTLA performs the best on PI+LS problem.

On separable and non-separable optimization tasks, as shown in **Tables 2**, **4**, TLTLA performs well on all separable optimization tasks but not on the non-separable Rosenbrock function. The reason is that Rosenbrock function is fully non-separable problem making the cross-dimensional search of intra-task knowledge transfer inefficient.

Experimental Results on Tri-Tasking Optimization Problems

To study the scalability of the proposed algorithm in solving more complex tri-tasking optimization problems, we construct nine tri-tasking optimization problems based on the bi-tasking problems (Da et al., 2016). Specifically, nine tri-tasking optimization problems are constructed by adding an additional task into a bi-tasking optimization problem proposed in Da et al. (2016). All compared algorithms are performed in 20 independent runs on each tri-tasking problem. TLTLA is compared with MFEA. Both algorithms are extended to handle tri-tasking problems. The balance factor between crossover and

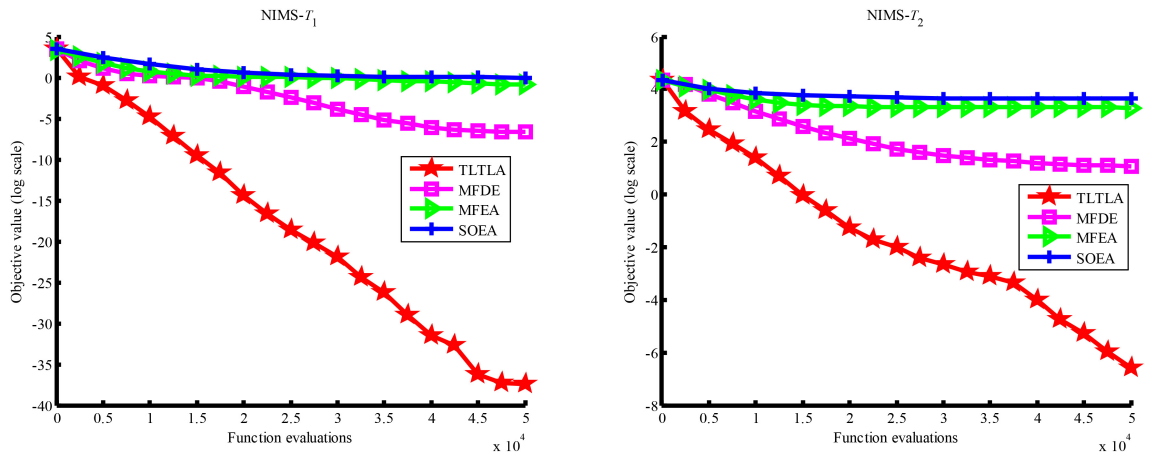

FIGURE 13 | Convergence trends of tasks in NI+MS.

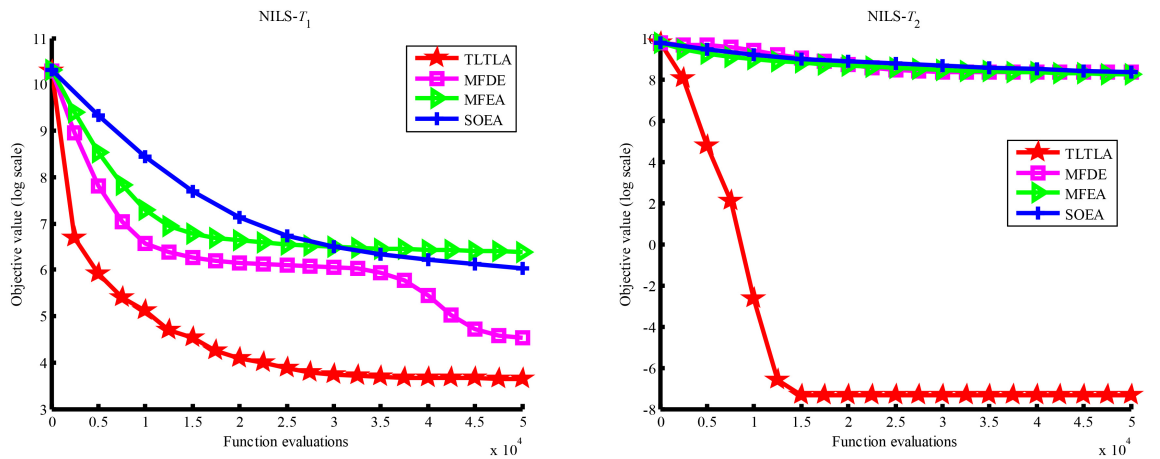

FIGURE 14 | Convergence trends of tasks in NI+LS.

mutation is set to rmp = 0.3 for all compared algorithms. The population size is set to N = 150 for all compared algorithms. The maximum number of function evaluations is set to 150,000 for all compared algorithms. It is important to note that the experimental settings assign an equal amount of computing resources for each component optimization task in bi-tasking and tri-task optimization problems.

Table 3 reports the mean and standard deviation of the function values obtained by TLTLA and MFEA on nine tri-tasking optimization problems. The best mean function value on each task is highlighted in bold. As can be summarized in **Table 3**, TLTLA performs significantly better than MFEA in dealing with the tri-tasking problems. The experimental results in **Tables 3**, **4** demonstrate the high scalability of the proposed algorithm. When the number of component tasks is increased, TLTLA can still obtain solutions of high quality. In particular, on task T_2 of NI+HS+Ackley and task T_1 of NI+LS+Griewank, the proposed algorithm gets more improvements in solving tri-tasking problem than the corresponding bi-tasking problem. The reason is that the corresponding global optimum 0 of the added Griewank task is found, which indicates that TLTLA can utilize the population diversity in the multitasking environment to escape from the local optima.

The Effectiveness Analysis of Two Proposed Knowledge Transfers

In this section, we empirically study the effectiveness of the two proposed knowledge transfer methods, including inter-task and intra-task knowledge transfers. Two variants of TLTLA, namely TLTLA-U and TLTLA-L are designed to compared with TLTLA. The former is the same as TLTLA without using the intra-task knowledge transfer, the latter is TLTLA without using the inter-task knowledge transfer. MFEA is also involved in the comparison as the baseline. **Table 5** shows the mean and standard deviation of the function values obtained by each compared algorithm on nine bi-tasking optimization problems. The best mean function value on each task is highlighted in bold. The sums of rankings of the four compared algorithms are also presented.

TABLE 5 | The mean and standard deviation of function values between the algorithms TLTLA, TLTLA-U, TLTLA-L, and MFEA.

Problem	Task	TLTLA	Rank	TLTLA-U	Rank	TLTLA-L	Rank	MFEA	Rank
CIHS	T_1	**0.00E+00** (0)	**1**	3.38E-01 (0.0701)	3	7.93E-02 (0.0311)	2	3.73E-01 (0.0617)	4
	T_2	**0.00E+00** (0)	**1**	1.75E+02 (51.3951)	2	5.49E+02 (39.1071)	4	1.95E+02 (34.4953)	3
CIMS	T_1	**1.20E-14** (2.47E-14)	**1**	5.35E+00 (0.9860)	3	2.10E+01 (0.1022)	4	4.39E+00 (0.4481)	2
	T_2	**0.00E+00** (0)	**1**	2.33E+02 (60.9264)	3	5.44E+02 (49.9483)	4	2.27E+02 (52.2778)	2
CILS	T_1	**3.41E-14** (1.21E-14)	**1**	2.01E+01 (0.0431)	2	2.11E+01 (0.0457)	4	2.02E+01 (0.0798)	3
	T_2	**6.36E-04** (1.11E-19)	**1**	3.65E+03 (435.9930)	3	1.91E+00 (1.4314)	2	3.70E+03 (429.1093)	4
PIHS	T_1	**2.88E+01** (62.1998)	**1**	6.80E+02 (165.2077)	4	5.44E+02 (39.4790)	2	6.14E+02 (131.0438)	3
	T_2	**9.63E-08** (3.86E-07)	**1**	7.07E+00 (1.6748)	2	8.99E+00 (4.8763)	3	1.01E+01 (2.4734)	4
PIMS	T_1	1.02E+00 (1.1088)	**1**	3.27E+00 (0.4646)	2	2.09E+01 (0.0578)	4	3.49E+00 (0.6289)	3
	T_2	**2.65E+01** (24.5602)	**1**	6.43E+02 (580.1922)	3	2.60E+02 (46.9642)	2	7.02E+02 (267.8668)	4
PILS	T_1	**1.60E-12** (4.90E-12)	**1**	1.99E+01 (0.1446)	2	2.10E+01 (0.1169)	4	2.00E+01 (0.1302)	3
	T_2	**1.59E-14** (6.32E-14)	**1**	2.08E+01 (3.0661)	3	2.26E+01 (1.8860)	4	1.93E+01 (1.7291)	2
NIHS	T_1	**3.52E+01** (20.8321)	**1**	1.06E+03 (1.20E+03)	4	2.72E+02 (40.9484)	2	1.01E+03 (346.1264)	3
	T_2	**2.54E+00** (11.3913)	**1**	2.58E+02 (90.7596)	2	5.28E+02 (38.6019)	4	2.87E+02 (92.4182)	3
NIMS	T_1	**5.55E-17** (2.23E-16)	**1**	3.76E-01 (0.0754)	3	6.74E-02 (0.0172)	2	4.20E-01 (0.0654)	4
	T_2	**1.35E-03** (0.0030)	**1**	2.76E+01 (2.6969)	3	5.55E+01 (2.3183)	4	2.71E+01 (2.6883)	2
NILS	T_1	**3.85E+01** (89.1612)	**1**	6.52E+02 (120.3008)	4	5.42E+02 (34.9702)	2	6.51E+02 (98.6871)	3
	T_2	**6.36E-04** (7.31E-10)	**1**	3.70E+03 (613.1705)	4	1.93E+00 (1.6964)	2	3.62E+03 (325.0275)	3
SUM			**18**		52		55		55

In **Table 5**, using only one knowledge transfer method, TLTLA-U and TLTLA-L achieve similar overall performance to MFEA. However, combining two proposed knowledge transfers, TLTLA performs much better than MFEA, TLTLA-U, and TLTLA-L on nine test problems, which indicates that the inter-task and the intra-task knowledge transfer procedures cooperate with each other in a mutually beneficial fashion. Therefore, the inter-task and intra-task transfer learning components are indispensable for the proposed algorithm.

DISCUSSION AND CONCLUSION

In this paper, a novel evolutionary MTO algorithm with TLTL is introduced. Particularly, the upper level transfer learning uses the commonalities and similarities among tasks to improve the efficiency and effectiveness of genetic transfer. The lower level transfer learning focuses on the intra-task knowledge learning, which transmits the beneficial information from one dimension to other dimensions. The intra-task knowledge learning can effectively use decision variables information from other dimensions to improve the exploration ability of the proposed algorithm. The experimental results on two-task and three-task optimization problems show the superior performance and high scalability of the proposed TLTLA.

Evolutionary MTO is a recent paradigm introducing the transfer learning of machine learning into the evolutionary computation (Zar, 1972; Noman and Iba, 2005; Chen et al., 2011; Zhu et al., 2011, 2015a,b,c, 2016, 2017; Gupta and Ong, 2016; Hou et al., 2017). There remain many open challenging problems. For instance, how to avoid the negative transfer? Most evolutionary MTO algorithms were proposed based on the inter-task similarity and commonality. However, on problems with few inter-task similarity and commonality, these algorithms may have worse performance than those with no transfer learning. To deal with this issue, introducing similarity measurement between two tasks could be a good choice. Moreover, how to extend the existing transfer learning based optimization algorithms to solve large-scale multitask problems in real applications remains a challenging problem.

DATA AVAILABILITY STATEMENT

The code of the proposed algorithm for this study is available on request to the corresponding author.

AUTHOR CONTRIBUTIONS

QC and XM performed the experiments, analyzed the data, and wrote the manuscript with supervision from ZZ, YS, and YY. LM contributed substantially to manuscript revision, editing for language quality, and gave suggestions on experimental studies. All authors provided the critical feedback, edited, and finalized the manuscript.

FUNDING

This work was supported in part by the National Natural Science Foundation of China, under grants 61976143, 61471246, 61603259, 61803629, 61575125, 61975135, and 61871272, the International Cooperation and Exchanges NSFC, under grant 61911530218, the Guangdong NSFC, under grant 2019A1515010869, the Guangdong Special Support Program of

Top-notch Young Professionals, under grants 2014TQ01X273 and 2015TQ01R453, the Shenzhen Fundamental Research Program, under grant JCYJ20170302154328155, the Scientific Research Foundation of Shenzhen University for Newly-introduced Teachers, under grant 2019048, and the Zhejiang Lab's International Talent Fund for Young Professionals. This work was supported by the National Engineering Laboratory for Big Data System Computing Technology.

REFERENCES

Bali, K., Gupta, A., Feng, L., Ong, Y. S., and Tan, P. S. (2017). "Linearized domain adaptation in evolutionary multitasking," in *Proceedings of the 2017 IEEE Congress on Evolutionary Computation*, San Sebastian, 1295–1302.

Cavallisforza, L. L., and Feldman, M. W. (1973). Cultural versus biological inheritance: phenotypic transmission from parents to children. (A theory of the effect of parental phenotypes on children's phenotypes). *Am. J. Hum. Genet.* 25, 618–627.

Chen, X., Ong, Y. S., Lim, M. H., and Tan, K. C. (2011). A multi-facet survey on memetic computation. *IEEE Trans. Evol. Comput.* 15, 591–607. doi: 10.1109/tevc.2011.2132725

Cloninger, C. R., Rice, J., and Reich, T. (1979). Multifactorial inheritance with cultural transmission and assortative mating. II. a general model of combined polygenic and cultural inheritance. *Am. J. Hum. Genet.* 31, 176–198.

Da, B., Ong, Y. S., Feng, L., Qin, A. K., Gupta, A., Zhu, Z., et al. (2016). *Evolutionary Multitasking for Single-Objective Continuous Optimization: Benchmark Problems, Performance Metric, and Baseline Results. Technical Report*. Singapore: Nanyang Technological University.

Dawkins, R. (2006). The selfish gene. *Q. Rev. Biol.* 110, 781–804.

Deb, K. (2001). *Multi-Objective Optimization Using Evolutionary Algorithms*. New York, NY: Wiley.

Deb, K., and Agrawal, R. B. (1994). Simulated binary crossover for continuous search space. *Compl. Syst.* 9, 115–148.

Feldman, M. W., and Laland, K. N. (1996). Gene-culture coevolutionary theory. *Trends Ecol. Evol.* 11, 453–467.

Feng, L., Zhou, W., Zhou, L., Jiang, S., Zhong, J., Da, B., et al. (2017). "An empirical study of multifactorial PSO and multifactorial DE," in *Proceedings of the IEEE Congress on Evolutionary Computation*, San Sebastian, 921–928.

Gupta, A., Mandziuk, J., and Ong, Y. S. (2016a). Evolutionary multitasking in bi-level optimization. *Compl. Intell. Syst.* 1, 83–95. doi: 10.1007/s40747-016-0011-y

Gupta, A., Ong, Y. S., and Feng, L. (2016b). Multifactorial evolution: toward evolutionary multitasking. *IEEE Trans. Evol. Comput.* 20, 343–357. doi: 10.1109/tevc.2015.2458037

Gupta, A., and Ong, Y. S. (2016). "Genetic transfer or population diversification? Deciphering the secret ingredients of evolutionary multitask optimization," in *Proceedings of the 2016 IEEE Symposium Series on Computational Intelligence*, Athens, 1–7.

Gupta, A., Ong, Y. S., and Feng, L. (2018). Insights on transfer optimization: because experience is the best teacher. *IEEE Trans. Emerg. Top. Comput. Intell.* 2, 51–64. doi: 10.1109/tetci.2017.2769104

Hou, Y., Ong, Y. S., Feng, L., and Zurada, J. M. (2017). Evolutionary transfer reinforcement learning framework for multi-agent system. *IEEE Trans. Evol. Comput.* 21, 601–615.

Ma, X., Liu, F., Qi, Y., Gong, M., Yin, M., Li, L., et al. (2014a). MOEA/D with opposition-based learning for multiobjective optimization problem. *Neurocomputing* 146, 48–64. doi: 10.1016/j.neucom.2014.04.068

Ma, X., Liu, F., Qi, Y., Li, L., Jiao, L., Liu, M., et al. (2014b). MOEA/D with Baldwinian learning inspired by the regularity property of continuous multiobjective problem. *Neurocomputing* 145, 336–352. doi: 10.1016/j.neucom.2014.05.025

Ma, X., Qi, Y., Li, L., Liu, F., Jiao, L., and Wu, J. (2014c). MOEA/D with uniform decomposition measurements for many-objective problems. *Soft Comput.* 18, 2541–2564. doi: 10.1007/s00500-014-1234-8

Ma, X., Liu, F., Qi, Y., Li, L., Jiao, L., Deng, X., et al. (2016a). MOEA/D with biased weight adjustment inspired by user-preference and its application on multi-objective reservoir flood control problem. *Soft Comput.* 20, 4999–5023. doi: 10.1007/s00500-015-1789-z

Ma, X., Liu, F., Qi, Y., Wang, X., Li, L., Jiao, L., et al. (2016b). A multiobjective evolutionary algorithm based on decision variable analyses for multiobjective optimization problems with large-scale variables. *IEEE Trans. Evol. Comput.* 20, 275–298. doi: 10.1109/tevc.2015.2455812

Ma, X., Zhang, Q., Yang, J., and Zhu, Z. (2018). On Tchebycheff decomposition approaches for multi-objective evolutionary optimization. *IEEE Trans. Evol. Comput* 22, 226–244. doi: 10.1109/tevc.2017.2704118

Noman, N., and Iba, H. (2005). "Enhancing differential evolution performance with local search for high dimensional function optimization," in *Proceedings of the 7th Annual Conference on Genetic and Evolutionary Computation*, New York, NY, 967–974.

Ong, Y. S., and Gupta, A. (2016). Evolutionary multitasking: a computer science view of cognitive multitasking. *Cogn. Comput.* 8, 125–142. doi: 10.1007/s12559-016-9395-7

Pan, S. J., and Yang, Q. (2010). A survey on transfer learning. *IEEE Trans Knowl. Data Eng.* 22, 1345–1359.

Qi, Y., Ma, X., Liu, F., Jiao, L., Sun, J., and Wu, J. (2014). MOEA/D with adaptive weight adjustment. *Evol. Comput.* 22, 231–264. doi: 10.1162/EVCO_a_00109

Qin, A. K., and Suganthan, P. N. (2005). "Self-adaptive differential evolution algorithm for numerical optimization," in *Proceedings of the 2015 IEEE Congress on Evolutionary Computation*, Edinburgh, 1785–1791.

Ramon, S., and Ong, Y. S. (2017). "Concurrently searching branches in software tests generation through multitask evolution," in *Proceedings of the 2016 IEEE Symposium Series on Computational Intelligence*, Athens, 1–8.

Tan, W. M., Ong, Y. S., Gupta, A., and Goh, C. K. (2017). Multi-problem surrogates: transfer evolutionary multiobjective optimization of computationally expensive problems. *IEEE Trans. Evol. Comput.* 23:99. doi: 10.1109/TEVC.2017.2783441

Wen, Y. W., and Ting, C. K. (2017). "Parting ways and reallocating resources in evolutionary multitasking," in *Proceedings of the 2017 IEEE Congress on Evolutionary Computation*, San Sebastian, 2404–2411.

Xie, T., Gong, M., Tang, Z., Lei, Y., Liu, J., and Wang, Z. (2016). "Enhancing evolutionary multifactorial optimization based on particle swarm optimization," in *Proceedings of the 2016 IEEE Congress on Evolutionary Computation*, Vancouver, BC, 1658–1665.

Yuan, Y., Ong, Y. S., Feng, L., Qin, A. K., Gupta, A., Da, B., et al. (2016). *Evolutionary Nultitasking for Multiobjective Continuous Optimization: Benchmark Problems, Performance Metrics and Baseline Results. Technical Report*. Singapore: Nanyang Technological University.

Yuan, Y., Ong, Y. S., Gupta, A., Tan, P. S., and Xu, H. (2017). "Evolutionary multitasking in permutation-based combinatorial optimization problems: Realization with TSP, QAP, LOP, and JSP," in *Proceedings of the IEEE Region 10 Conference*, Singapore, 3157–3164.

Zar, J. H. (1972). Significance testing of the Spearman rank correlation coefficient. *Publ. Am. Stat. Assoc.* 67, 578–580. doi: 10.1080/01621459.1972.10481251

Zhou, L., Feng, L., Zhong, J., Ong, Y. S., Zhu, Z., and Sha, E. (2017). "Evolutionary multitasking in combinatorial search spaces: a case study in capacitated vehicle routing problem," in *Proceedings of the 2016 IEEE Symposium Series on Computational Intelligence*, (Athens), 1–8.

Zhu, Z., Jia, S., He, S., Sun, Y., Ji, Z., and Shen, L. (2015a). Three-dimensional Gabor feature extraction for hyperspectral imagery classification using a memetic framework. *Inf. Sci.* 298, 274–287. doi: 10.1016/j.ins.2014.11.045

Zhu, Z., Wang, F. S., and Sun, Y. (2015b). Global path planning of mobile robots using a memetic algorithm. *Int. J. Syst. Sci.* 46, 1982–1993. doi: 10.1080/00207721.2013.843735

Zhu, Z., Xiao, J., Li, J., Wang, Q. F., and Zhang, Q. (2015c). Global path planning of wheeled robots using multi-objective memetic algorithms. *Integ. Comput. Aid. Eng.* 22, 387–404. doi: 10.3233/ica-150498

Zhu, Z., Ong, Y. S., and Dash, M. (2017). Wrapper-filter feature selection algorithm using a memetic framework. *IEEE Trans. Syst. Man Cybernet. Part B* 37, 70–76. doi: 10.1109/tsmcb.2006.883267

Zhu, Z., Xiao, J., He, S., Ji, Z., and Sun, Y. (2016). A multi-objective memetic algorithm based on locality-sensitive hashing for one-to-many-to-one dynamic pickup-and-delivery problem. *Inform. Sci.* 329, 73–89. doi: 10.1016/j.ins.2015.09.006

Zhu, Z., Zhou, J., Ji, Z., and Shi, Y. H. (2011). DNA sequence compression using adaptive particle swarm optimization-based memetic algorithm. *IEEE Trans. Evol. Comput.* 15, 643–658. doi: 10.1109/tevc.2011.2160399

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.

frontiers in Neurorobotics

HYPOTHESIS AND THEORY
published: 14 January 2020
doi: 10.3389/fnbot.2019.00112

A Privacy-Preserving Multi-Task Learning Framework for Face Detection, Landmark Localization, Pose Estimation, and Gender Recognition

Chen Zhang[1], *Xiongwei Hu*[1], *Yu Xie*[1], *Maoguo Gong*[2] *and Bin Yu*[1]*

[1] School of Computer Science and Technology, Xidian University, Xi'an, China, [2] Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education, School of Electronic Engineering, Xidian University, Xi'an, China

Recently, multi-task learning (MTL) has been extensively studied for various face processing tasks, including face detection, landmark localization, pose estimation, and gender recognition. This approach endeavors to train a better model by exploiting the synergy among the related tasks. However, the raw face dataset used for training often contains sensitive and private information, which can be maliciously recovered by carefully analyzing the model and outputs. To address this problem, we propose a novel privacy-preserving multi-task learning approach that utilizes the differential private stochastic gradient descent algorithm to optimize the end-to-end multi-task model and weighs the loss functions of multiple tasks to improve learning efficiency and prediction accuracy. Specifically, calibrated noise is added to the gradient of loss functions to preserve the privacy of the training data during model training. Furthermore, we exploit the homoscedastic uncertainty to balance different learning tasks. The experiments demonstrate that the proposed approach yields differential privacy guarantees without decreasing the accuracy of HyperFace under a desirable privacy budget.

Keywords: multi-task learning, privacy preserving, differential private stochastic gradient descent, balance different learning tasks, differential privacy guarantees

OPEN ACCESS

Edited by:
Liang Feng,
Chongqing University, China

Reviewed by:
Zexuan Zhu,
Shenzhen University, China
Xiaofen Lu,
Southern University of Science and Technology, China

***Correspondence:**
Bin Yu
yubin@mail.xidian.edu.cn

Received: *10 September 2019*
Accepted: *11 December 2019*
Published: *14 January 2020*

Citation:
Zhang C, Hu X, Xie Y, Gong M and Yu B (2020) A Privacy-Preserving Multi-Task Learning Framework for Face Detection, Landmark Localization, Pose Estimation, and Gender Recognition. Front. Neurorobot. 13:112. doi: 10.3389/fnbot.2019.00112

1. INTRODUCTION

Recently, neurorobotics has made great progress in a wide range of scientific fields, including locomotion and motor control, learning and memory systems, action selection and value systems, and many more. All of these models need to consider the problem of simultaneously solving multiple related tasks, which is the prevalent idea behind multi-task learning (MTL). MTL focuses on learning several tasks simultaneously by transferring knowledge among these tasks. In training machine learning models, the required datasets may contain private and sensitive information. Privacy is considered the private sphere of an individual or group that secludes information about themselves from the public environment and ought to be preserved adequately. These datasets for machine learning tasks enable faster commercial or scientific progress, but privacy-preservation has become an urgent issue that needs to be addressed. In early works, some privacy-preserving techniques, including k-anonymity (Sweeney, 2002), l-diversity (Machanavajjhala et al., 2006), and t-closeness (Li et al., 2007), that anonymize the data before analyzing it, were proposed. Even though curators can apply several simple anonymization techniques, sensitive personal information still has a high probability of being disclosed (Wang et al., 2010). As an essential and robust

privacy model, differential privacy can successfully resist most privacy attacks and provide a provable privacy guarantee (Dwork, 2011a; McMahan et al., 2017; Wang et al., 2018; Erlingsson et al., 2019). Moreover, differentially private MTL was introduced by Gupta et al. (2016), where the authors proposed a differentially private algorithm using a noisy task relation matrix and developed an attribute-wise noise addition scheme that significantly improves the utility of their proposed method. However, those algorithms significantly increase the time complexity of MTL, making it difficult to perform the iterative calculation in training models.

MTL is widely used in a broad range of practical applications, including face detection (Ranjan et al., 2017; Ahn et al., 2018; Chen et al., 2018; Zhao et al., 2019), federated MTL (Smith et al., 2017; Corinzia and Buhmann, 2019; Sattler et al., 2019), speech recognition (Huang et al., 2013; Kim et al., 2017; Liu et al., 2017; Subramanian et al., 2018), and other applications (Doersch and Zisserman, 2017; Han et al., 2017; Liu et al., 2017, 2019; Hessel et al., 2019). Ranjan et al. (2017) presented an algorithm for simultaneous face detection, landmark localization, pose estimation, and gender recognition. The proposed method, called HyperFace, exploits the synergy among the tasks to boost their individual performance. Their work demonstrates that HyperFace is able to capture both global and local information regarding faces and performs significantly better than many competitive algorithms for each of these four tasks. However, multi-task models without privacy preservation may impair the privacy of users during the training process of models. Therefore, enforcing privacy preservation on private datasets is a challenge that needs to be addressed. Existing privacy preservation methods have successfully integrated differential privacy into iterative training processes like stochastic gradient descent (Abadi et al., 2016; Papernot et al., 2016; McMahan et al., 2017; Wu et al., 2017; Bun et al., 2018; Wang et al., 2018). These differentially private frameworks preserve private and sensitive data within an acceptable performance range in single-task models. However, up until now, there have been few studies on privacy preservation in MTL. Another major challenge is that a reasonable trade-off of multi-task losses can make the noise level more balanced among individual tasks. Previous methods (Sermanet et al., 2013; Eigen and Fergus, 2015; Kokkinos, 2017) always manually adjust weights or just initialize weights and often become trapped in a local optimum.

As mentioned above, MTL has made great progress in a wide range of practical applications. However, an important challenge is how to preserve private and sensitive information contained in training datasets. In practice, existing privacy preservation methods have been successfully applied to many single-task models, but they are rarely applied to multi-task models. In this paper, we integrate the rigorous differential privacy mechanism with a multi-task framework named HyperFace through training five related tasks within a desirable privacy budget. We adopt the differential private stochastic gradient descent algorithm to optimize the end-to-end multi-task model. Specifically, Gaussian noise is added to the gradient of loss functions for preserving the privacy of the training data during the training process of the model. Furthermore, we exploit the homoscedastic uncertainty to weigh loss functions of multiple tasks, which can improve learning efficiency and prediction accuracy. Our main contributions are summarized as follows:

1. We propose a novel privacy-preserving multi-task learning framework that provides differential privacy guarantees on HyperFace.
2. The loss functions of multiple tasks are adjusted by utilizing the homoscedastic uncertainty, which makes the model more balanced within the privacy budget on individual tasks.
3. We evaluate our approach on face detection, landmark localization, pose estimation, and gender recognition. The extensive experiments demonstrate that data privacy can be preserved without decreasing accuracy.

The rest of the paper is organized as follows. The next section reviews differential privacy and multi-task learning. Section 3 describes the proposed approach in detail. Section 4 analyzes the experimental results of our approach, and section 5 concludes the paper.

2. RELATED WORK

In this section, we briefly review differential privacy and multi-task learning.

2.1. Differential Privacy

Differential privacy is a new and promising model presented by Dwork et al. (2006b) in 2006. It provides strong privacy guarantees by requiring the indistinguishability of whether or not an individual's data exists in a dataset (McSherry and Talwar, 2007; Dwork, 2011b; Dwork and Roth, 2014; McMahan et al., 2017; Wang et al., 2018; Erlingsson et al., 2019). We regard a dataset as d or d' on the basis of whether the individual is present or not. A differential privacy mechanism provides indistinguishability guarantees with respect to the pair (d, d'); the datasets d and d' are referred to as adjacent datasets. The definition of (ε, δ)-differential privacy is provided as follow:

DEFINITION 1. *A randomized mechanism $\mathcal{M} : \mathcal{D} \to \mathcal{R}$ satisfies (ε, δ)-differential privacy if, for any two adjacent datasets $d, d' \in \mathcal{D}$ and for any subset of outputs $\mathcal{Y} \subseteq \mathcal{R}$, it holds that*

$$Pr[\mathcal{M}(X) \in \mathcal{Y}] \leq e^{\varepsilon} Pr[\mathcal{M}(X') \in \mathcal{Y}] + \delta$$

The parameter ε denotes the privacy budget, which controls the privacy level of $\mathcal{M}$. For a small ε, the probability distributions of the output results of $\mathcal{M}$ on d and d' are extremely similar, and it is difficult for attackers to distinguish the two datasets. In addition, the parameter δ, which provides a possibility to violate ε-differential privacy, does not exist in the original definition of ε-differential privacy (Dwork et al., 2006a).

There are several common noise perturbation mechanisms for differential privacy that mask the original datasets or intermediate results during the training process of models: the Laplace mechanism, the exponential mechanism, and the Gaussian mechanism. Phan et al. (2017) developed a novel mechanism that injects Laplace noise into the computation of Layer-Wise Relevance Propagation (LRP) to preserve differential privacy in deep learning. Chaudhuri et al. (2011, 2013) adopted

the exponential mechanism as a privacy-preserving tuning method by training classifiers with different parameters on disjoint subsets of the data and then randomizing the selection of which classifier to release. In Yin and Liu (2017), numerical evaluations of the Gaussian cumulative density function are used to obtain the optimal variance to improve the utility of output perturbation Gaussian mechanisms for differential privacy.

To add less noise, the gradient computation of loss functions samples Gaussian noise instead of Laplacian noise, since the tail of the Gaussian distribution diminishes far more rapidly than that of the Laplacian distribution. A general paradigm for approximating the deterministic real-valued function $f : \mathcal{M} \rightarrow \mathbb{R}$ with a differential privacy mechanism is via additive noise calibrated to f's sensitivity S_f, which is defined as the maximum of the absolute distance $|f(d) - f(d')|$ where d and d' are adjacent datasets. For instance, the Gaussian noise mechanism is defined by

$$\mathcal{M}(d) \triangleq f(d) + \mathcal{N}(0, S_f^2 \cdot \sigma^2)$$

where $\mathcal{N}(0, S_f^2 \cdot \sigma^2)$ is the normal (Gaussian) distribution with mean 0 and standard deviation $S_f\sigma$.

2.2. Multi-Task Learning

MTL is an interesting and promising area in machine learning that aims to improve the performance of multiple related learning tasks by transferring useful information among them. Based on an assumption that all of the tasks, or at least a subset of them, are related, jointly learning multiple tasks is empirically and theoretically found to lead to better performance than learning them independently. Recently, MTL is becoming increasingly popular in many applications, such as recommendation, natural language processing, and face detection. Yin and Liu (2017) proposed a pose-directed multi-task convolutional neural network (CNN), and most importantly, an energy-based weight analysis method to explore how CNN-based multi-task learning works. However, multi-task learning algorithms may cause the leakage of information from different models across different tasks. Specifically, an attacker can participate in the multi-task learning process through one task, thereby acquiring model information of another task. To address this problem, Liu et al. (2018) developed a provable privacy-preserving MTL protocol that incorporates a homomorphic encryption technique to achieve the best security guarantee. Xie et al. (2017) proposed a novel privacy-preserving distributed multi-task learning framework for asynchronous updates and privacy preservation. Previous methods always apply privacy preservation to the parameters of models. In this paper, we combine HyperFace with a differential privacy mechanism for preserving the privacy of original datasets.

3. METHODOLOGY

This section presents our approach of differentially private learning on HyperFace, which provides a (ε, δ)-differential privacy guarantee for HyperFace. Section 3.1 summarizes the definition of the problem that needs to be resolved and the notations used, section 3.2 introduces the details of the framework, while section 3.3 discusses and analyzes the method.

TABLE 1 | Notations and symbols.

Notations	Descriptions
(ε, δ)	Privacy budget
$\mathcal{L}(\cdot)$	General loss function with parameters
$g_t(x_i), \bar{g}_t(x_i)$	Gradient and bounded gradient of the i^{th} example in a subset of examples L_t
$\hat{g}_t$	Noisy gradient of a subset of examples
$\|\cdot\|_2, S$	ℓ_2 norm of the gradient of an example
$\mathcal{N}(\cdot)$	Normal Gaussian distribution
η_t	Learning rate of a subset of examples
$loss_*$	Corresponding loss functions of different tasks

3.1. Review of the Problem and Notations

HyperFace is a prevalent multi-task model for simultaneously learning the related tasks of face detection, landmark localization, pose estimation, and gender recognition. In this model, the synergy between related tasks is utilized to improve the performance of the individual tasks. There are two main problems for preserving privacy and boosting model performance in Hyperface. In practice, facial datasets used to train Hyperface contain a large amount of private and sensitive information. Training data without a strong privacy guarantee can be maliciously recovered by carefully analyzing the model and outputs. Another problem is that the performance of a multi-task model is highly dependent on appropriate weights among the loss of each task. However, HyperFace simply initializes these weights, which may cause the model to become trapped in a local optimum rather than reaching the global optimum. The notations and symbols used throughout the paper are summarized in **Table 1**.

3.2. Our Approach

In this paper, we present a novel approach called Differentially Private Learning on HyperFace (DPLH) to preserve the privacy of original facial datasets that contain landmark coordinates, pose estimations, gender information, and much more. To collect the faces with private attributes that need to be protected, we need to crop all faces from each given image in facial datasets. When optimizing the loss function of each task with the stochastic gradient descent algorithm, we allocate a reasonable privacy budget across each of the gradient updates on examples and analyze the privacy cost of the trained model. To trade off the privacy and utility of the Hyperface multi-task model, we utilize the synergy between related tasks to adjust the weights of each loss function.

3.2.1. Pre-training

There are two pre-training steps that need to be performed before the model update on Hyperface by applying the Gaussian mechanism: regional candidate selection and initializing the weights of HyperFace.

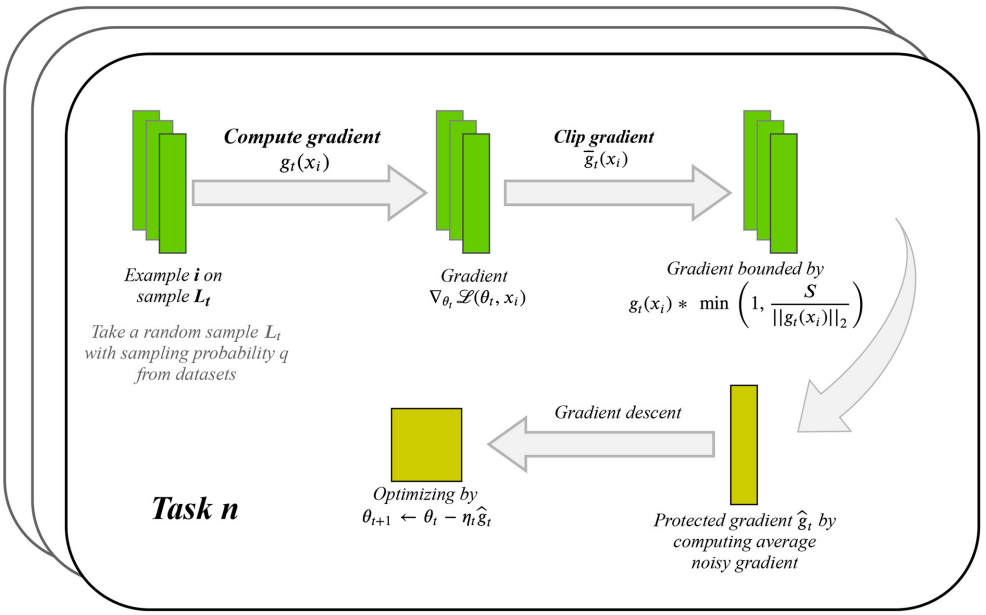

FIGURE 1 | Per iterative computation process for preserving privacy on each learning task.

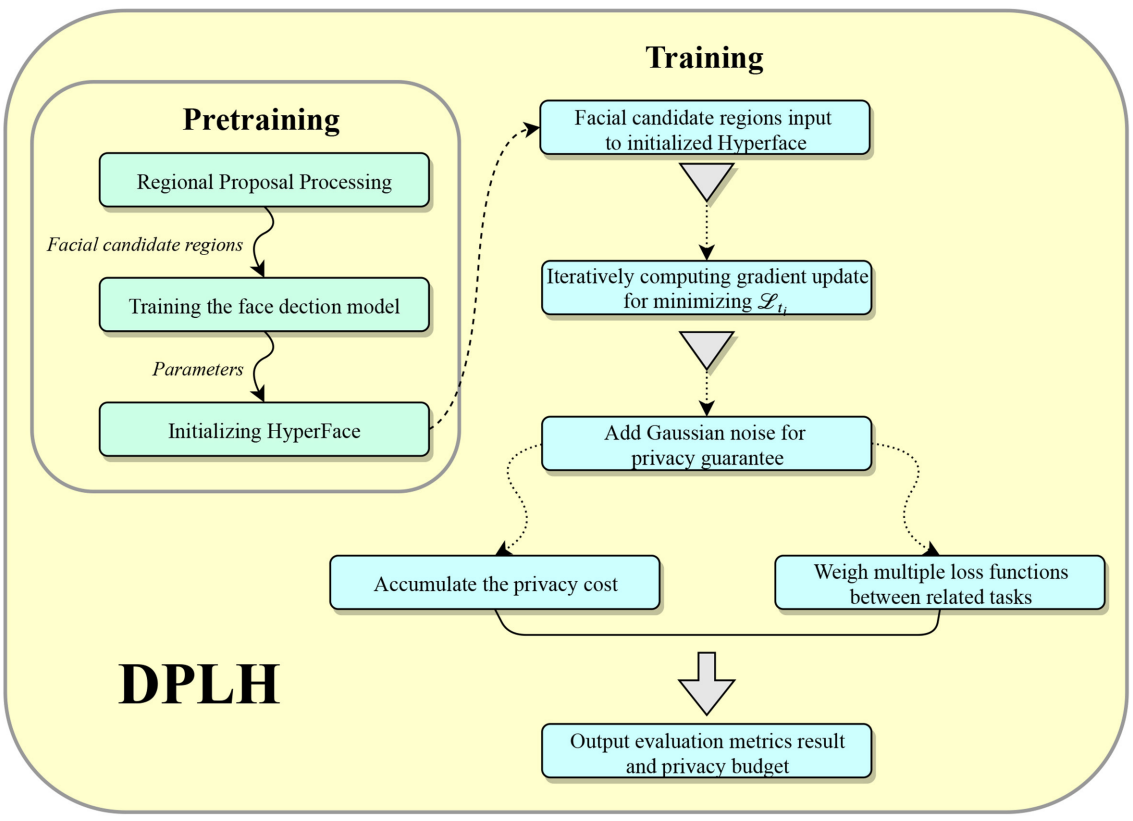

FIGURE 2 | The architecture of DPLH.

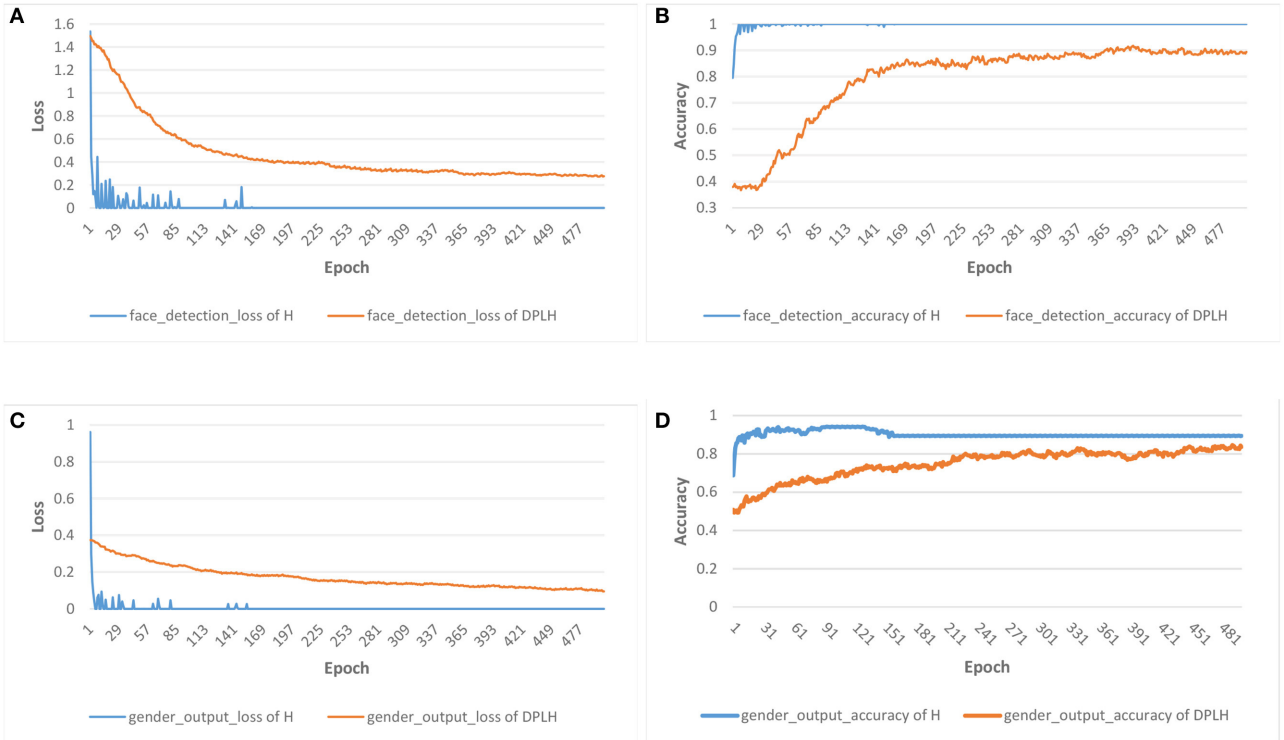

FIGURE 3 | Results for the loss and accuracy of face detection and gender classification on HyperFace (H) vs. DPLH. **(A)** Loss of face detection. **(B)** Accuracy of face detection. **(C)** Loss of gender classification. **(D)** Accuracy of gender classification.

Facial datasets usually involve a large amount of private information that is potentially distributed over the images. In order to apply the differential privacy mechanism to these facial data, the given images are selectively cropped to generate positive candidate regions with faces and negative candidate regions without faces by regional candidate selection. We filter out candidate regions as positive and negative by computing the Intersection over Union (IOU) overlap. The candidate regions are considered as positive with an IOU overlap of more than 0.5, and negative candidate regions have an IOU overlap of <0.35. Subsequently, these selected candidate regions are scaled to 227 * 227 pixels as the input of the model. In addition, the ground truths, such as landmark localization and the visibility factor corresponding to these candidate regions, need to be adjusted as well since they are relative to the original images rather than the selected regions.

Initializing the weights of network is helpful for finding global optimal solutions or avoiding becoming trapped in poor local optimal solutions. A good initialization facilitates gradient propagation in deep networks and avoids the problems of a vanishing gradient or gradient exploding. In this paper, we pre-train a single-task model, whose parameters are initialized to the default, for face detection with an input of the candidate regions generated by regional selection. Then, the parameters of this single task are used to initialize HyperFace for better convergence performance.

3.2.2. Training

Training data may not be effectively protected by only adding noise to the final parameters that result from the training process. Generally, there are few useful and exact characterizations of the dependence of these parameters on the training data. Moreover, adding excessive noise to the parameters may destroy the utility of the learning model. In the worst case, excessive noise will degrade the model performance, and a small amount of noise may not provide a strong privacy guarantee. Hence, we propose a novel approach for HyperFace to preserve the privacy of training data and control the influence of training data in the stochastic gradient descent computation.

In the training process of our DPLH model, we iteratively compute the gradient update from training data and then apply the Gaussian mechanism for differential privacy to the gradient update. **Figure 1** shows the per iterative computation process for protecting privacy while learning each task. Suppose the training datasets with N examples consist of selected candidate regions with adjusted ground truth. Given a sampling probability q, clipping threshold S, and noise multiplier z, our approach focuses on minimizing each task loss function $\mathcal{L}(\theta^j)$ with parameter $\theta^j (1 \leq j \leq 5)$ in the training process by using a stochastic gradient descent optimizing algorithm. At each step of stochastic gradient descent, we select a subset of the examples $L_t \subseteq [1, ..., N]$ by choosing each example with probability q. We compute the gradient $\nabla_{\theta^j} \mathcal{L}(\theta^j, x_i)$ as $g_t(x_i)$ with each example $i \in L_t$, clip

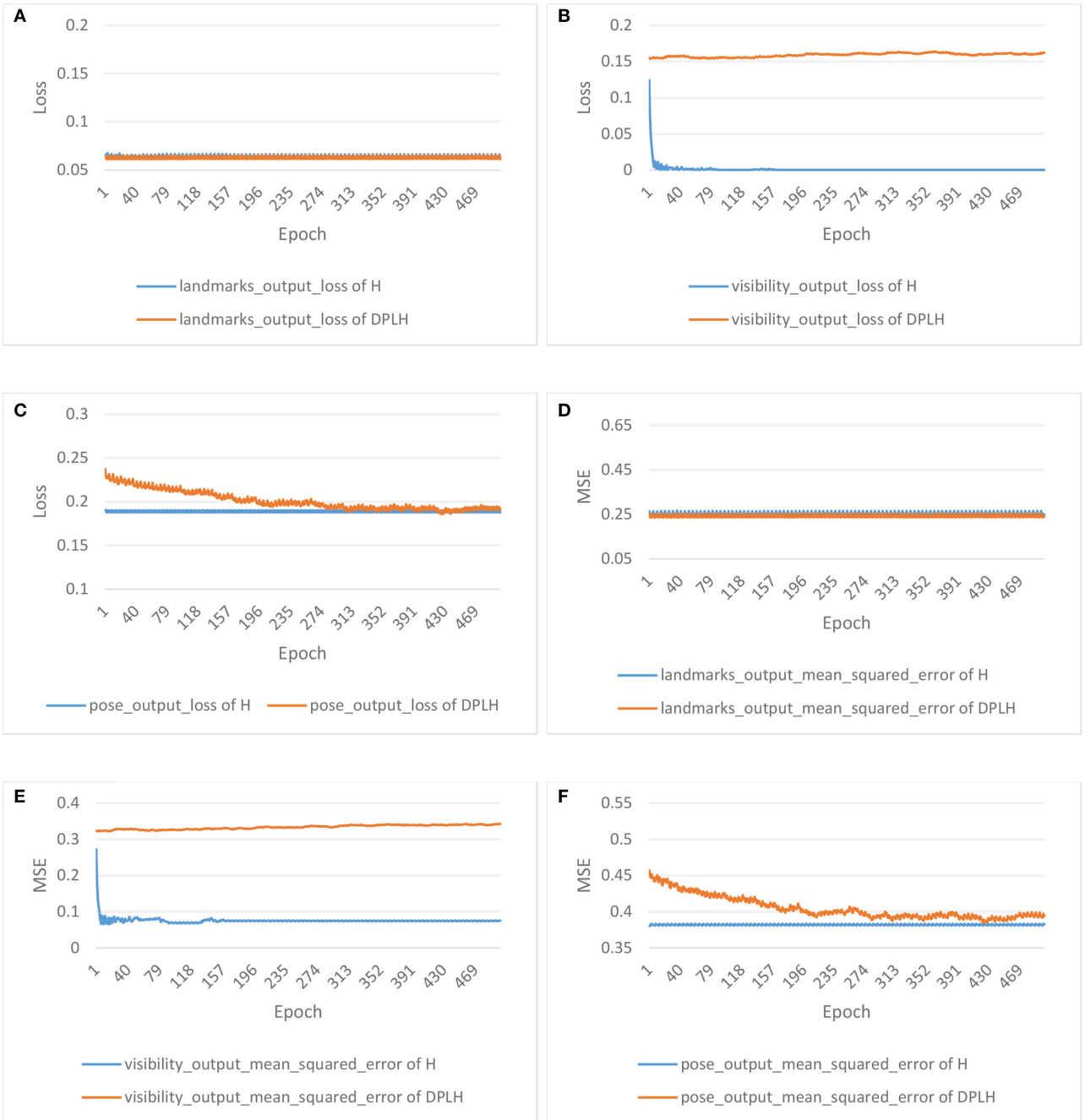

FIGURE 4 | Results for the loss and mean square error (MSE) of landmark localization, landmark visibility, and pose estimation on HyperFace(H) vs. DPLH. **(A)** Loss of landmark localization. **(B)** Loss of landmark visibility. **(C)** Loss of pose estimation. **(D)** MSE of landmark localization. **(E)** MSE of landmark visibility. **(F)** MSE of pose estimation.

each gradient to have maximum ℓ_2 norm S using $\bar{g}_t(x_i) = g_t(x_i) * min(1, \frac{S}{||g_t(x_i)||_2})$, then add noise to them and compute the average of the noisy gradients by $\hat{g}_t = \frac{1}{qN}(\sum_i \bar{g}_t(x_i) + \mathcal{N}(0, \sigma^2 I))$. Subsequently, we take a step in the opposite direction of this average noisy gradient, like $\theta_{t+1} = \theta_t - \eta_t \hat{g}_t$. In addition to outputting the model, we estimate the privacy budget of an iterative Gaussian noise mechanism by privacy accounting. We describe our approach in more detail below.

Loss functions. In order to better measure the performance of the model, different loss functions and evaluation metrics are used for the training tasks of face detection, landmark localization, landmark visibility, pose estimation, and gender classification.

Face detection. We use regional candidate selection to generate positive candidate regions with faces ($l = 1$) and negative candidate regions without faces ($l = 0$) in given images. We can train the face detection task with loss function $loss_D$, given as follows

$$loss_D = (1 - l) * log(1 - p)^{-1} + l * log(p)^{-1}, \quad (1)$$

where p is the prediction probability of a candidate region with a face.

Landmark localization. We consider the category of candidate regions and the visibility factor of landmark points when computing the loss function of landmark localization. There is no loss corresponding to invisible landmark points or negative candidate regions. We compute the loss of landmark location by

$$loss_L = \frac{l}{2N_l} \sum_{i=1}^{N_l} v_i((\hat{a}_i - a_i)^2 + (\hat{b}_i - b_i)^2), \quad (2)$$

where $(\hat{a}_i, \hat{b}_i)$ is the i^{th} predicted landmark location. If the i^{th} landmark is visible in the positive candidate region, the visibility factor v_i is 1; otherwise, it is 0. N_l is the total number of landmark points in a candidate region.

Landmark visibility. This task is learned with positive regions to estimate the presence of the predicted landmark. The loss function is shown in (3)

$$loss_V = \frac{l}{N_l} \sum_{i=1} N_l(\hat{v}_i - v_i)^2, \quad (3)$$

where $\hat{v}_i$ is the predicted visibility of the i^{th} landmark.

Pose estimation. The head pose annotation contains roll, pitch, and yaw expressed as (p_1, p_2, p_3) in ground truth. We compute the loss of pose estimation for a positive candidate region by

$$loss_P = \frac{l}{3}((\hat{p}_1 - p_1)^2 + (\hat{p}_2 - p_2)^2 + (\hat{p}_3 - p_3)^2), \quad (4)$$

where $(\hat{p}_1, \hat{p}_2, \hat{p}_3)$ are the pose estimations.

Gender classification. Predicting gender is a two-class problem similar to face detection. Computing the loss of the gender prediction for a positive candidate region is defined as

$$loss_G = l(1 - g) * log(1 - p_g)^{-1} + lg * log(p_g)^{-1}, \quad (5)$$

where $g = 0$ if the gender is male, or else $g = 1$. p_g is the predicted probability of male.

Trading off loss. The simple approach to combining losses among learning tasks is to directly perform a linear weighted sum of the losses for each individual task, as shown in (6)

$$loss_{all} = \sum_{i=1}^{5} \lambda_{t_i} loss_{t_i}, \quad (6)$$

where t_i is the i^{th} element from the set of tasks $T = \{D, L, V, P, G\}$ and parameter λ_{t_i} is the weight of each task. However, the naive method of tuning weights manually makes it difficult to balance the performance of individual tasks. We aim to better balance the process of iteratively computing average noisy gradient for each task by using homoscedastic task uncertainty to trade off multiple loss functions. Homoscedastic task uncertainty, which captures the relative confidence between tasks, is a quantity that remains constant for all input data and varies between different tasks, reflecting the uncertainty inherent to the regression or classification task. Homoscedastic uncertainty can be used as a basis for weighting losses in a multi-task learning problem. The positive scalar σ added to the total loss function relates to the uncertainty of the tasks as measured in terms of entropy. The total loss function with the homoscedastic task uncertainty is finally provided by

$$\mathcal{L}(\lambda_{t_i}, \sigma_1, \sigma_2, \ldots, \sigma_i) = \sum_{i=1}^{5} \frac{1}{2\sigma_i^2} \mathcal{L}_{t_i}(\lambda_{t_i}) + log\sigma_i^2 \quad (7)$$

Privacy accounting. For our DPLH model, we attach importance to computing the overall privacy cost of training. When iteratively computing the average noisy gradient for each task, the composability of differential privacy allows the privacy accountant to accumulate the privacy cost corresponding to all of the gradients. To make the testing process more transparent and to ensure our model provides a (ε, δ)-differential privacy guarantee, we encapsulate the key differential privacy mechanism into the privacy accountant and positively tune the hyperparameters to achieve different levels of privacy protection.

3.2.3. Architecture of DPLH

In this section, we describe the flow of processing training data in our proposed method, as illustrated in **Figure 2**.

As shown in **Figure 2**, the model input is composed of candidate regions with a specific size of (227, 227) generated by the regional candidate selection. Positive candidate regions have full ground truth of face detection, landmark coordinates, landmark visibility factors, pose estimation, and gender information. In contrast, negative candidate regions without faces have the ground truth of face detection, and other ground truths are set to none. These data with ground truth are used to adjust the weights and bias of each layer in the network. In pre-training, we train a single-task model for face detection, and the learned parameters from this network are used to initialize Hyperface. Thereby, we use the candidate regions with adjusted ground truth as input to train the privacy-preserving model. We iteratively compute the gradient update from training data and then apply the Gaussian mechanism for differential privacy to the gradient update, and the privacy cost of iterative calculation is accumulated and accounted. We balance the loss functions of related tasks to ensure better performance for applying the differential privacy mechanism on each task and output a modest small loss. In the end, we will get an output of the evaluation metric results and the privacy budget.

TABLE 2 | *T*-test results on the performance of multiple tasks at different epochs.

Epochs	Task	50	100	150	200	250	300	350	400	450	500
P-value (%)	$loss_D$	0.068	0.054	0.039	0.033	0.032	0.031	0.031	0.031	0.030	0.030
	$loss_L$	0.185	0.123	0.105	0.089	0.075	0.073	0.073	0.072	0.072	0.072
	$loss_V$	0.314	0.285	0.212	0.183	0.179	0.178	0.178	0.178	0.178	0.178
	$loss_P$	0.351	0.317	0.297	0.283	0.279	0.279	0.279	0.279	0.279	0.279
	$loss_G$	0.063	0.057	0.049	0.039	0.038	0.038	0.038	0.038	0.038	0.038
	Acc_D	0.093	0.078	0.061	0.057	0.056	0.054	0.054	0.054	0.054	0.054
	Acc_G	0.045	0.031	0.027	0.025	0.024	0.024	0.024	0.024	0.024	0.024
	MSE_L	0.194	0.172	0.151	0.143	0.139	0.138	0.138	0.138	0.138	0.138
	MSE_V	0.112	0.089	0.073	0.067	0.065	0.065	0.065	0.065	0.065	0.065
	MSE_P	0.185	0.169	0.154	0.147	0.145	0.145	0.145	0.145	0.145	0.145

3.3. Discussion

The proposed approach, DPLH, aims to preserve private and sensitive information in training datasets. The main idea is to iteratively compute the HyperFace model update from optimizing loss functions and then apply the Gaussian mechanism for differential privacy to the gradient update before incorporating it into the model. In principle, this method can theoretically provide the (ε, δ)-differential privacy guarantee and can prevent private and sensitive data from being maliciously recovered. Furthermore, we use a privacy accountant to estimate the privacy cost of the training process and use different loss functions and evaluation metrics for the training tasks of face detection, landmark localization, landmark visibility, pose estimation, and gender classification. In the end, the losses of each task have reasonable, small values, and the evaluation metrics of each loss function will reflect good performance.

4. EXPERIMENT

In this section, we evaluate our approach on the AFLW dataset (Martin Koestinger and Bischof, 2011) and report the results of each task for different noise levels. Section 4.1 introduces the details of the experimental setup and the training dataset. Sections 4.2 and 4.3 show the results and analysis.

4.1. Dataset and Experimental Setup

We train our model by using the AFLW dataset, which contains more than 25,000 faces in almost 22,000 real-world images with full poses, gender variations, and some more private information. It provides 21 landmark point coordinates per face, along with the face bounding-box, face pose (yaw, pitch, and roll), and gender information. These data cannot be directly used as inputs to the model. We need to prepare the input of the model for evaluating face detection, landmark localization, landmark visibility, pose estimation, and gender classification.

The input does not come from the original dataset, AFLW, but rather comprises candidate regions generated by the regional candidate selection method. The proposed method introduced in section 3 is used for cropping essential regions from images and adjusting privacy-related facial features. For each image from the AFLW dataset, we use the Selective Search (Van de Sande et al., 2011) algorithm to generate candidate regions for faces and then filter out positive samples and negative samples by computing the Intersection over Union (IOU) overlap. The equation of IOU is

$$IOU = \frac{A_{overlap}}{A_{union}} \tag{8}$$

where $A_{overlap}$ is the area of overlap between the selected candidate region and the ground truth bounding-box, and A_{union} is the area of union encompassed by both of them. Positive candidate regions are selected from regions that have an IOU overlap of more than 0.5 with the ground truth bounding box. The candidate regions with an IOU overlap of $<$0.35 are considered as negative candidate regions, and other candidate regions are neglected. Subsequently, we scale these selected candidate regions uniformly to 227 * 227 pixels to match the input size of our model. Note that the faces in the images have full pose variations, resulting in some of the landmark points being invisible. We use a visibility factor to annotate visible landmarks provided by the AFLW dataset (Martin Koestinger and Bischof, 2011). However, the given ground truth fiducial coordinates and corresponding visibility factors are relative to the original images. Training the model directly by using the raw information can have a negative impact on the quality of the model. Hence, the landmark points are shifted and scaled to the selected candidate regions using (9)

$$(a_i, b_i) = (\frac{c_i - c}{w} * w', \frac{d_i - d}{h} * h') \tag{9}$$

where $(c_i, d_i)'$s are the given ground truth fiducial coordinates, and $(a_i, b_i)'$s are the ground truth fiducial coordinates of adjusted candidate regions. These regions can be characterized by $\{c, d, w, h\}$, where (c, d) are the upper left coordinates of a region and w, h are the width and height of the region, respectively. In the end, some of the visible landmark are modified to be invisible, because positive candidate regions may not contain all $(a_i, b_i)'$s. The landmark

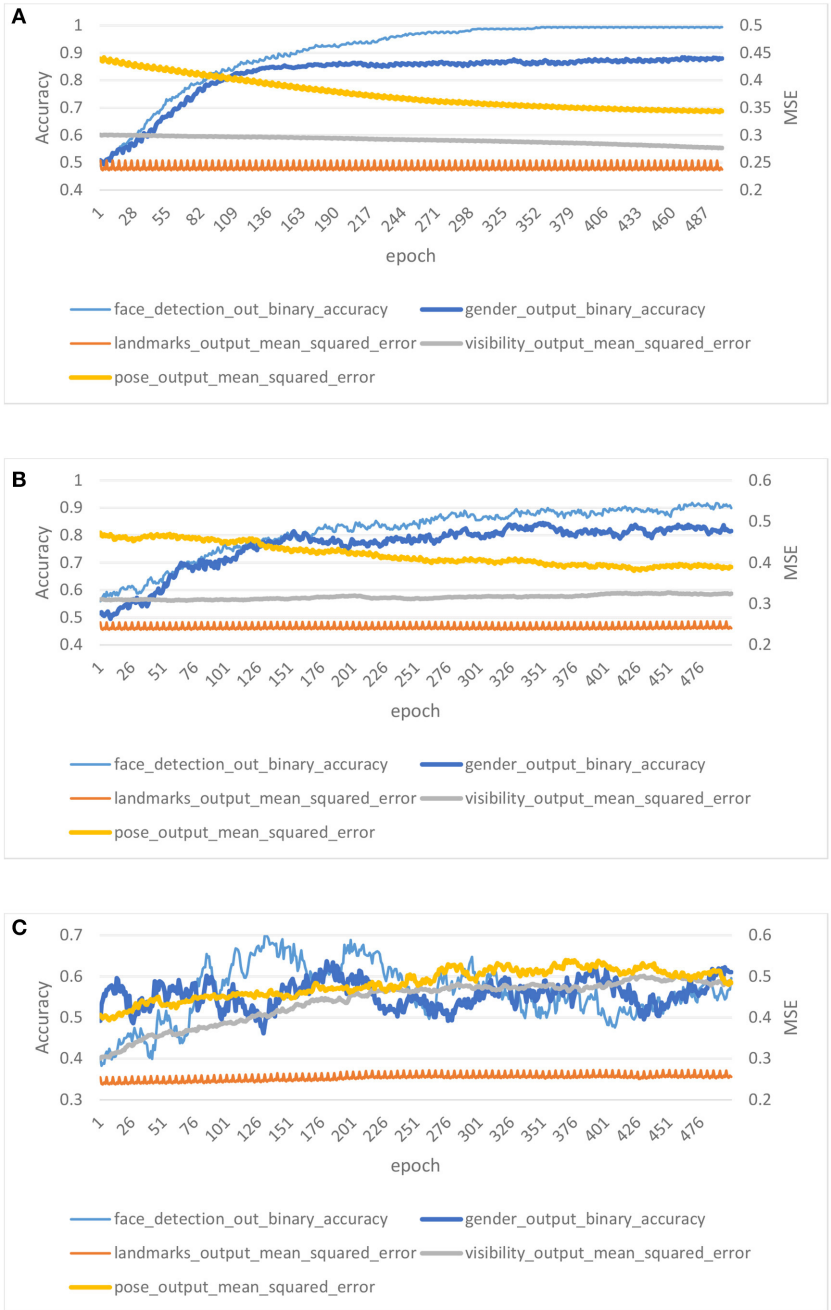

FIGURE 5 | Results for low noise level training, modest noise level training, and high noise level training with privacy budget $(10, 10^{-5})$, $(5, 10^{-5})$, $(0.7, 10^{-5})$. **(A)** Low noise level. **(B)** Modest noise level. **(C)** High noise level.

points of negative candidate regions are set to invisible by default.

In our experiments, we obtain more than 40,000 candidate regions. We take 70% of them to train models and the rest for evaluating model performance. Moreover, we set some hyperparameters to fixed values for the next experiments. The sampling probability is given by $q = L/N_c$, where N_c is the total number of inputs and L is the number of samples involved in a batch. We fix the clipping threshold $S = 0.5$, the number of epochs $E = 500$, batch size $L = 32$, input size $N_c = 40,000$, and the learning rate $\eta = 0.00015$.

4.2. Results of Model Training

In this experiment, we compare the results of our model DPLH training and HyperFace training. In order to better

evaluate the performance of each task, we choose the accuracy metrics for face detection and gender classification and the mean square error metrics for landmark localization, landmark visibility, and pose estimation. We allocate the privacy budget (ε, δ) as $(5, 10^{-5})$ to DPLH to provide a privacy guarantee.

Figure 3 shows the results for the loss and accuracy of face detection and gender classification on the two models. A declining trend of losses is depicted in **Figure 3A** for face detection and **Figure 3C** for gender classification. As the epochs increase in number, the losses of HyperFace on these tasks decline faster, and the losses of DPLH decrease gently. After convergence, the two models consume 500 epochs to reduce the loss to desirably small values. Moreover, the losses from training HyperFace converge to a smaller level than the differentially private losses. Additionally, **Figures 3B,D** illustrate the growth trend of accuracy for face detection and gender classification, respectively. The accuracy from training HyperFace consuming the same number of epochs rises fastest, and, in addition, the metrics evaluating the two models both converge to high levels. **Figure 4** shows the results for the loss and mean square error (MSE) of landmark localization, landmark visibility, and pose estimation on the two models. Similar to **Figure 3**, the losses of the three tasks on training the two models converge to desirably small levels. The MSE curves decline to small values, converging to a nearby level, respectively on their tasks.

These figures indicate that the final results for loss, accuracy, and mean square error converge to a desirable level. From the perspective of three metrics, the two models can almost achieve approximate results on respective tasks, which demonstrates that our approach decreases model performance and utility very little compared with HyperFace. Our approach achieves 90 and 86% accuracy on face detection and gender classification, respectively, compared with 99 and 90% accuracy on HyperFace. For landmark localization, landmark visibility, and pose estimation, our approach achieves 0.255, 0.25, and 0.27 mean square error, respectively, compared with 0.245, 0.2, and 0.24 on HyperFace. The final results indicate that our approach can provide a differential privacy guarantee with desirable performance of the system. We conduct a t-test on the performance of multiple tasks with different epochs. For p-value ≤ 0.05, the performance of the DPLH method approximates to that of Hyperface without privacy preservation. As shown in **Table 2**, the extremely small p-value indicates that the DPLH method provides a differential privacy guarantee and achieves performance that is similar to that of the Hyperface method.

4.3. Results for Training With Different Noise Levels

In this experiment, we consider the effect of different noise levels on the performance of DPLH. We compare three noise levels for the training characteristics of HyperFace integrated with differential privacy. We set a privacy budget $\varepsilon = 0.7$ to train the DPLH with a number of epochs $E = 500$, which represents high noise level training. Besides, we consume a fixed $\varepsilon = 5$ privacy budget per epoch to train HyperFace with a modest noise level. Moreover, low noise level training is performed on HyperFace with a privacy budget $\varepsilon = 10$ per epoch. In addition, that we fix $\delta = 10^{-5}$ per figure.

Figure 5 shows the results on different privacy budgets (ε, δ). In each plot, we show the evaluation of accuracy for two tasks (face detection and gender classification) and the mean square error for three tasks (landmark localization, landmark visibility, and pose estimation). **Figures 5A,B** illustrate low noise level training and modest noise level training, respectively. The accuracy of the two noise levels rises gently, and the accuracy of low noise level training is higher than that of modest noise level training after convergence. On the evaluation of MSE, the two noise level trainings converge to a desirable level. In contrast, **Figure 5C** illustrates high noise level training performance on DPLH. The accuracy of high noise level training converges to lower values, and the MSE shows a unstable decline trend. We achieve desirable performance for $(10, 10^{-5})$, $(5, 10^{-5})$ differential privacy, respectively, since the accuracy converges to a high level and the MSE converges to a low level. However, $(0.7, 10^{-5})$-differential privacy training brings too much noise to the model, resulting in unstable performance. The final results indicate that acceptable noise level training on HyperFace can provide a differential privacy guarantee and stable performance, while an excessive noise level may destroy the performance and utility of the model, making privacy preservation irrelevant.

5. CONCLUSION

In this paper, we propose a novel method called differentially private learning on HyperFace that provides a differential privacy guarantee and desirable performance for simultaneously learning face detection, landmark localization, pose estimation, and gender classification. We demonstrate the utility and effectiveness of our model for training all four tasks on the datasets. In the future, we will carry out further studies on selecting the most appropriate noise level automatically to provide a differential privacy guarantee and excellent performance.

DATA AVAILABILITY STATEMENT

The datasets analyzed in this manuscript are not publicly available. Requests to access the datasets should be directed to huhsiungwei@gmail.com.

AUTHOR CONTRIBUTIONS

YX and XH conceptualized the problem and the technical framework. MG and CZ developed the algorithms, supervised the experiments, and exported the data. YX, XH, and

BY implemented the privacy-preserving multi-task learning architecture simulation. BY managed the project. All of the authors wrote the manuscript, discussed the experimental results, and commented on the manuscript.

FUNDING

This work was supported by the Key Research and Development Program of Shaanxi Province (Grant no. 2019ZDLGY17-01, 2019GY-042).

ACKNOWLEDGMENTS

We would like to thank Xinyi Niu and Chunyi Li for their thoughtful comments on the manuscript and language revision. We were grateful to all of the study participants for their time and effort.

REFERENCES

Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., Talwar, K., et al. (2016). "Deep learning with differential privacy," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security* (Vienna: ACM), 308–318.

Ahn, B., Choi, D.-G., Park, J., and Kweon, I. S. (2018). Real-time head pose estimation using multi-task deep neural network. *Robot. Auton. Syst.* 103, 1–12. doi: 10.1016/j.robot.2018.01.005

Bun, M., Dwork, C., Rothblum, G. N., and Steinke, T. (2018). "Composable and versatile privacy via truncated CDP," in *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing* (Los Angeles, CA: ACM), 74–86.

Chaudhuri, K., Monteleoni, C., and Sarwate, A. D. (2011). Differentially private empirical risk minimization. *J. Mach. Learn. Res.* 12, 1069–1109.

Chaudhuri, K., Sarwate, A. D., and Sinha, K. (2013). A near-optimal algorithm for differentially-private principal components. *J. Mach. Learn. Res.* 14, 2905–2943.

Chen, J.-C., Lin, W.-A., Zheng, J., and Chellappa, R. (2018). "A real-time multi-task single shot face detector," in *2018 25th IEEE International Conference on Image Processing (ICIP)* (Athens: IEEE), 176–180.

Corinzia, L., and Buhmann, J. M. (2019). Variational federated multi-task learning. *arXiv* 1906.06268.

Doersch, C., and Zisserman, A. (2017). "Multi-task self-supervised visual learning," in *Proceedings of the IEEE International Conference on Computer Vision* (Venice), 2051–2060.

Dwork, C. (2011a). Differential privacy. *Encycl. Cryptogr. Secur.* 338–340. doi: 10.1007/978-1-4419-5906-5_752

Dwork, C. (2011b). A firm foundation for private data analysis. *Commun. ACM* 54, 86–95. doi: 10.1145/1866739.1866758

Dwork, C., Kenthapadi, K., McSherry, F., Mironov, I., and Naor, M. (2006a). "Our data, ourselves: privacy via distributed noise generation," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques* (St. Petersburg: Springer), 486–503.

Dwork, C., McSherry, F., Nissim, K., and Smith, A. (2006b). "Calibrating noise to sensitivity in private data analysis," in *Theory of Cryptography Conference* (New York, NY: Springer), 265–284.

Dwork, C., and Roth, A. (2014). The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.* 9, 211–407. doi: 10.1561/0400000042

Eigen, D., and Fergus, R. (2015). "Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture," in *Proceedings of the IEEE International Conference on Computer Vision* (Santiago), 2650–2658.

Erlingsson, Ú., Feldman, V., Mironov, I., Raghunathan, A., Talwar, K., and Thakurta, A. (2019). "Amplification by shuffling: from local to central differential privacy via anonymity," in *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms* (San Diego, CA: SIAM), 2468–2479.

Gupta, S. K., Rana, S., and Venkatesh, S. (2016). "Differentially private multi-task learning," in *Pacific-Asia Workshop on Intelligence and Security Informatics* (Springer), 101–113.

Han, H., Jain, A. K., Wang, F., Shan, S., and Chen, X. (2017). Heterogeneous face attribute estimation: a deep multi-task learning approach. *IEEE Trans. Pattern Anal. Mach. Intell.* 40, 2597–2609. doi: 10.1109/TPAMI.2017.2738004

Hessel, M., Soyer, H., Espeholt, L., Czarnecki, W., Schmitt, S., and van Hasselt, H. (2019). "Multi-task deep reinforcement learning with Popart," in *Proceedings of the AAAI Conference on Artificial Intelligence* (Hawaii), Vol. 33, 3796–3803.

Huang, J.-T., Li, J., Yu, D., Deng, L., and Gong, Y. (2013). "Cross-language knowledge transfer using multilingual deep neural network with shared hidden layers," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing* (Vancouver, BC: IEEE), 7304–7308.

Kim, S., Hori, T., and Watanabe, S. (2017). "Joint ctc-attention based end-to-end speech recognition using multi-task learning," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing* (New Orleans, LA: IEEE), 4835–4839.

Kokkinos, I. (2017). "Ubernet: training a universal convolutional neural network for low-, mid-, and high-level vision using diverse datasets and limited memory," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Hawaii), 6129–6138.

Li, N., Li, T., and Venkatasubramanian, S. (2007). "t-Closeness: privacy beyond k-anonymity and l-diversity," in *Proceedings of 23rd International Conference on Data Engineering* (Istanbul: IEEE), 106–115.

Liu, K., Uplavikar, N., Jiang, W., and Fu, Y. (2018). "Privacy-preserving multi-task learning," in *IEEE International Conference on Data Mining* (Singapore), 1128–1133.

Liu, P., Qiu, X., and Huang, X. (2017). Adversarial multi-task learning for text classification. *arXiv* 1704.05742. doi: 10.18653/v1/P17-1001

Liu, S., Johns, E., and Davison, A. J. (2019). "End-to-end multi-task learning with attention," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Long Beach, CA), 1871–1880.

Machanavajjhala, A., Gehrke, J., Kifer, D., and Venkitasubramaniam, M. (2006). "l-Diversity: Privacy beyond k-anonymity," in *Proceedings of 22nd International Conference on Data Engineering (ICDE'06)* (Atlanta, GA: IEEE), 24–24.

Martin Koestinger, Paul Wohlhart, P. M. R., and Bischof, H. (2011). "Annotated facial landmarks in the wild: a large-scale, real-world database for facial landmark localization," in *Proceedings of 1st IEEE International Workshop on Benchmarking Facial Image Analysis Technologies* (Barcelona).

McMahan, H. B., Ramage, D., Talwar, K., and Zhang, L. (2017). Learning differentially private recurrent language models. *arXiv* 1710.06963.

McSherry, F., and Talwar, K. (2007). Mechanism design via differential privacy. *FOCS* 7, 94–103. doi: 10.1109/FOCS.2007.66

Papernot, N., Abadi, M., Erlingsson, U., Goodfellow, I., and Talwar, K. (2016). Semi-supervised knowledge transfer for deep learning from private training data. *arXiv* 1610.05755.

Phan, N., Wu, X., Hu, H., and Dou, D. (2017). "Adaptive laplace mechanism: differential privacy preservation in deep learning," in *2017 IEEE International Conference on Data Mining (ICDM)* (New Orleans, LA: IEEE), 385–394.

Ranjan, R., Patel, V. M., and Chellappa, R. (2017). Hyperface: a deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* 41, 121–135. doi: 10.1109/TPAMI.2017.2781233

Sattler, F., Müller, K.-R., and Samek, W. (2019). Clustered federated learning: model-agnostic distributed multi-task optimization under privacy constraints. *arXiv* 1910.01991.

Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., and LeCun, Y. (2013). Overfeat: integrated recognition, localization and detection using convolutional networks. *arXiv* 1312.6229.

Smith, V., Chiang, C.-K., Sanjabi, M., and Talwalkar, A. S. (2017). "Federated multi-task learning," in *Advances in Neural Information Processing Systems* (Long Beach, CA), 4424–4434.

Subramanian, S., Trischler, A., Bengio, Y., and Pal, C. J. (2018). Learning general purpose distributed sentence representations via large scale multi-task learning. *arXiv* 1804.00079.

Sweeney, L. (2002). k-Anonymity: a model for protecting privacy. *Int. J. Uncertainty Fuzziness Knowl. Based Syst.* 10, 557–570. doi: 10.1142/S0218488502001648

Van de Sande, K. E., Uijlings, J. R., Gevers, T., and Smeulders, A. W. (2011). "Segmentation as selective search for object recognition," in *Proceedings of 11th IEEE International Conference on Computer Vision*, Vol. 1 (Barcelona), 7.

Wang, K., Chen, R., Fung, B., and Yu, P. (2010). Privacy-preserving data publishing: a survey on recent developments. *ACM Comput. Surveys* 42:14. doi: 10.1145/1749603.1749605

Wang, Y.-X., Balle, B., and Kasiviswanathan, S. (2018). Subsampled rényi differential privacy and analytical moments accountant. *arXiv* 1808. 00087.

Wu, X., Li, F., Kumar, A., Chaudhuri, K., Jha, S., and Naughton, J. (2017). "Bolt-on differential privacy for scalable stochastic gradient descent-based analytics," in *Proceedings of the 2017 ACM International Conference on Management of Data* (Chicago, IL: ACM), 1307–1322.

Xie, L., Baytas, I. M., Lin, K., and Zhou, J. (2017). "Privacy-preserving distributed multi-task learning with asynchronous updates," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Halifax, NS: ACM), 1195–1204.

Yin, X., and Liu, X. (2017). Multi-task convolutional neural network for pose-invariant face recognition. *IEEE Trans. Image Process.* 27, 964–975. doi: 10.1109/TIP.2017.2765830

Zhao, Y., Tang, F., Dong, W., Huang, F., and Zhang, X. (2019). Joint face alignment and segmentation via deep multi-task learning. *Multimed. Tools Appl.* 78, 13131–13148. doi: 10.1007/s11042-018-5609-1

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.

frontiers in Neuroscience

ORIGINAL RESEARCH
published: 14 January 2020
doi: 10.3389/fnins.2019.01387

Multi-Task Learning Based Network Embedding

Shanfeng Wang[1], *Qixiang Wang*[2] *and Maoguo Gong*[2]*

[1] School of Cyber Engineering, Xidian University, Xi'an, China, [2] Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education, School of Electronic Engineering, Xidian University, Xi'an, China

The goal of network representation learning, also called network embedding, is to encode the network structure information into a continuous low-dimensionality embedding space where geometric relationships among the vectors can reflect the relationships of nodes in the original network. The existing network representation learning methods are always single-task learning, in which case these methods focus on preserving the proximity of nodes from one aspect. However, the proximity of nodes is dependent on both the local and global structure, resulting in a limitation on the node embeddings learned by these methods. In order to solve this problem, in this paper, we propose a novel method, Multi-Task Learning-Based Network Embedding, termed MLNE. There are two tasks in this method so as to preserve the proximity of nodes. The aim of the first task is to preserve the high-order proximity between pairwise nodes in the whole network. The second task is to preserve the low-order proximity in the one-hop area of each node. By jointly learning these tasks in the supervised deep learning model, our method can obtain node embeddings that can sufficiently reflect the roles that nodes play in networks. In order to demonstrate the efficacy of our MLNE method over existing state-of-the-art methods, we conduct experiments on multi-label classification, link prediction, and visualization in five real-world networks. The experimental results show that our method performs competitively.

Keywords: network representation learning, multi-task learning, network embedding, high-order proximity, low-order proximity

OPEN ACCESS

Edited by:
Liang Feng,
Chongqing University, China

Reviewed by:
Jinghui Zhong,
South China University of Technology, China
Yaqing Hou,
Dalian University of Technology (DUT), China

***Correspondence:**
Maoguo Gong
gong@ieee.org

Specialty section:
This article was submitted to Decision Neuroscience, a section of the journal Frontiers in Neuroscience

Received: *28 July 2019*
Accepted: *10 December 2019*
Published: *14 January 2020*

Citation:
Wang S, Wang Q and Gong M (2020) Multi-Task Learning Based Network Embedding. Front. Neurosci. 13:1387. doi: 10.3389/fnins.2019.01387

1. INTRODUCTION

A network is an important way of representing the relationships between objects, for example, in social networks, state grids, and citation networks (Gong et al., 2017). With the increasing complexity of a network, it is more valuable to explore it as a carrier of information. There are some meaningful applications in network analysis, such as node classification (Tsoumakas and Katakis, 2007), link prediction (Lü and Zhou, 2011), community detection (Fortunato, 2010), and recommender systems (Lü et al., 2012). Traditional network representation methods, such as an adjacency matrix, pose several challenges (Peng et al., 2019). First, network analysis methods based on traditional forms of representation usually have high computational complexity. Second, traditional network representation methods make it difficult to design parallel and distributed algorithms. These two challenges make these methods hard to use for large-scale network analysis. Moreover, there is a limitation when machine learning is applied in network analysis due to high dimensionality and sparsity. Thus, determining how to properly construct a meaningful representation of the structure information extracted from networks is promising research.

Network representation learning (NRL), also called network embedding (Hamilton et al., 2017; Goyal and Ferrara, 2018), has been proposed for encoding network information into a continuous low-dimensionality feature space. From the perspective of network topology, those nodes that have similar structures should have similar representation vectors. For example, those nodes within the same community in a network have similar proximity structures, and thus they should be closer in embedding space. Due to the learned representations, the relationships between nodes and the roles that nodes play in networks can be efficiently analyzed. Many network analysis tasks can be dealt with based on the distances in the embedding space, so that the computational complexity is low and parallel algorithms can be adopted for network analysis problems. Moreover, many machine-learning algorithms have been used for network analysis, benefiting from network embedding. Not only that, but those representations can be applied in other application tasks (Herman et al., 2000; Hu et al., 2016; Wang et al., 2017; Wei et al., 2017; Shi et al., 2018).

Recently, an increasing number of methods have been proposed for network representation learning (Chen et al., 2018; Peng et al., 2019; Zhang et al., in press). These methods can mainly be classified into three categories (Peng et al., 2019). The first is matrix factorization-based methods (Qiu et al., 2018; Liu et al., 2019b), which are directly inspired by the dimension-reduction technique. One of the best-known methods is Laplacian Eigenmaps (Belkin and Niyogi, 2002), which generate a network representation through factorizing the Laplacian of the network adjacency matrix. GraRep (Cao et al., 2015) builds a k-step relationship information matrix so as to sufficiently capture the pairwise node proximity. According to the matrix, it adopts SVD to generate different representations and finally concatenates all of them to form a global representation. Qiu et al. exploited sparse matrix factorization for large-scale network embedding (Qiu et al., 2019). The second category is random walk-based methods. DeepWalk (Perozzi et al., 2014) was the first method to introduce random walk into network representation learning. It uses a sampling method called unbiased random walk to generate discrete sequences of nodes, in which case sequences and nodes are abstracted as sentences and words. It also introduces the skip-gram (Mikolov et al., 2013), the best-known model in natural language processing (NLP), to learn representations for nodes from those sequences. Node2vec (Argerich et al., 2016) was proposed to develop a novel sampling method named biased random walk, which is based on breadth-first search (BFS) and depth-first search (DFS), resulting in more flexibility in the exploration of networks. The third category is deep learning-based methods. Wang et al. proposed a structural deep network embedding method named SDNE (Wang et al., 2016). Cao et al. proposed a deep neural network for learning graph representations (DNGR) (Cao et al., 2016). Both SDNE and DNGR follow the encoder-decoder framework, where the encoder maps a high-dimensionality feature vector into a lower-dimensionality representation and the decoder reconstructs the original feature vector from that. They build a proximity matrix in which an element represents the pairwise node proximity and apply an autoencoder model to learn representations from that matrix. SDNE directly adopts a network adjacency matrix as the proximity matrix and combines the autoencoder loss function with the Laplacian Eigenmaps loss function. DNGR introduces the pointwise mutual information (PMI) matrix as the proximity matrix, which is mostly used to evaluate the similarity among words in NLP. Network embedding methods are not limited to the above three categories (Tang et al., 2015; Donnat et al., 2018).

Existing network embedding algorithms have achieved promising performance, but these methods all focus on single-task learning, resulting in a lack of diversity in representations. A good representation of a node should depend on its position and structure in the local community and global network. For example, a node may be the centroid of the local community and also play a role as a bridge between communities in the global network (Musiał and Juszczyszyn, 2009). To learn network representation from both the local and global network structure information, we resort to multi-task learning (MTL) for help in exploring and exploiting global and local network representation learning.

In this paper, we propose a multi-task learning-based network embedding called MLNE. In MLNE, there are three components: a shared encoder, decoder, and classifier. We adopt *positive pointwise mutual information* (PPMI), which is a commonly used method to measure the similarity between discrete objects, to build the global proximity matrix. In order to build the matrix, we introduce random surfing to gather graph information. The shared encoder and decoder form a standard autoencoder to learn the latent representation from the global proximity matrix in an unsupervised manner. The shared encoder encodes the global feature information into a low-dimensionality node embedding, and the decoder decodes that information from the learned embeddings. Another task is to preserve the local features. The key idea behind this task is that the learned embedding from the shared encoder contains graph information such as the structure of local graph neighborhoods, so that the one-hop area of nodes can be reconstructed from that learned embedding. Due to the network sparsity, the direct neighborhoods should make more contributions to nodes, and thus it is worth designing a specific task to optimize embedding with respect to first-order proximity. The task is carried out by the shared encoder and the specific classifier, which predicts whether there is an edge between pairwise nodes. As a result, the learned embeddings can preserve both the local and global structural information. In addition, we design a regularizer to make those nodes that are direct neighborhoods for each other much closer in Euclidean space and vice versa, resulting in good clustering. Empirically, we conduct experiments on five real-world network datasets and three tasks: node classification, link prediction, and visualization. The experimental results show that our model has competitive performance against baselines.

The rest of this paper is organized as follows. In section II, preliminaries are given. Section III introduces the proposed algorithm in detail. In section IV, we briefly compare our algorithm with other related network embedding methods and analyze the experimental results. In the last section, this paper is concluded.

2. PRELIMINARIES

In this section, we discuss the preliminaries of network representation learning in detail. First, we briefly introduce the notation and formulate the problem. Second, detailed descriptions of positive pointwise mutual information and random surfing are presented. An introduction to multi-task learning is then given.

2.1. Notations and Definitions

A network can be formally modeled as a graph $G = (V, E)$, where V is the set of nodes and E is the set of edges. $v \in V$ represents a node in the graph, and $(v_i, v_j) \in E$ represents an edge between v_i and v. The adjacency matrix is defined as $A \in \mathbb{R}^{|V|\times|V|}$. Network representation learning aims to build an embedding matrix $Z \in \mathbb{R}^{|V|\times d}$, where $d \ll |V|$ and each row $z \in \mathbb{R}^d$ represents a vector representation of a node.

2.2. PPMI and Random Surfing

Pointwise mutual information (PMI) is a measure to quantify the correlation between two discrete objects. PMI has commonly been applied in the field of NLP such as in the measurement of the similarity between words. PMI can be defined as follows:

$$PMI(w, c) = \log\left(\frac{\#(w, c) \cdot D}{\#(w) \cdot \#(c)}\right) \tag{1}$$

where $\#(\cdot)$ means the number of occurrences of an object and $D = \sum_w \sum_c \#(w, c)$.

It is found that when the statistics of co-occurrence count between two objects $\#(w, c)$ is 0, the measure will result in $\log(0) = -\infty$. An alternative measure called positive pointwise mutual information (PPMI) is proposed to address this problem. PPMI can be defined as follows:

$$PPMI(w, c) = \max(0, PMI(w, c)) \tag{2}$$

Cao et al. firstly introduced PPMI into NRL to generate node representation (Cao et al., 2016). In order to build PPMI matrix, they designed a random surfing model to extract structure information of network and directly generate the probabilistic co-occurrence matrix without sampling process. The key idea behind the model is that the visited probability from source node to target node can be iteratively calculated by a transition matrix.

Let the P_k be the k-th step visited probability matrix in which each element $P_k(i, j)$ represents the probability from source node v_i to node v_j after k times transitions. The P_0 initially is set as A. The P_k can be defined as follows:

$$P_k = \gamma \cdot P_{k-1} \cdot T + (1 - \gamma) \cdot P_0 \tag{3}$$

where T is the transition matrix, γ is the probability that the model will continue simulation, and $1 - \gamma$ is the restart probability. The element in T is the probability that node v_i will reach node v_j. If $A_{i,j} = 1$, $T(i, j) = 1/deg(i)$, otherwise $T(i, j) = 0$.

According to (3), a set of visited probability matrices can be defined as $P = \{P_0, P_1, ..., P_K\}$. The probabilistic matrix can be constructed as follows:

$$r = \sum_{i=k}^{K} P_i. \tag{4}$$

where K is the number of samplings.

2.3. Multi-Task Learning

Traditional machine learning methods aim to optimize for a specific metric. To realize the goal of a task, a model is trained by fine-tuning parameters. By training the model, we can get a satisfying result, but some information that helps to improve the performance will be ignored. This information can be mined from related tasks. To utilize the information effectively, a new approach, named multi-task learning (MTL) (Ruder, 2017; Thung and Wee, 2018), is proposed. In MTL, multiple related tasks are learnt jointly, and useful information is shared among related tasks. In MTL, each task can benefit from other tasks, and then we can get a better result by training several tasks. Multi-task learning has been widely used in several fields, such as natural language processing (Liu et al., 2019a), image processing (Du et al., 2018), computer vision (Zhang et al., 2018), and recommendation (Wang et al., 2018).

There are two commonly used approaches to carrying out MTL in deep learning. The first is hard parameter sharing of hidden layers. In this approach, different tasks share the hidden layers, and the output layers are different. The second is soft parameter sharing of hidden layers. In this approach, different tasks have similar parameters, and the output layers are also different. **Figure 1** shows these two approaches to MTL in deep learning.

3. THE FRAMEWORK

In this section, we first give the detailed description of the framework of our proposed approach, MLNE. Next, our multi-task learning model based on deep learning is described in detail.

3.1. An Overview of the Framework

In this work, we leverage multi-task learning to learn robust and meaningful node representations. **Figure 2** shows the framework of our proposed model, MLNE, in which there are two phases: building the proximity matrix and embedding nodes. In the first phase, we extract information on the local and global structures to build a proximity matrix where each element represents the similarity between nodes. In the second phase, our model jointly optimizes two tasks so as to learn node representations in which there are two tasks, preserving the global and local network structures. The framework of the proposed algorithm is given as Algorithm 1.

3.2. Multi-Task Learning Model

Deep learning is introduced into multi-task learning model so as to learn complex structural information. In our proposed model, there are multiple layers with non-linear activation functions,

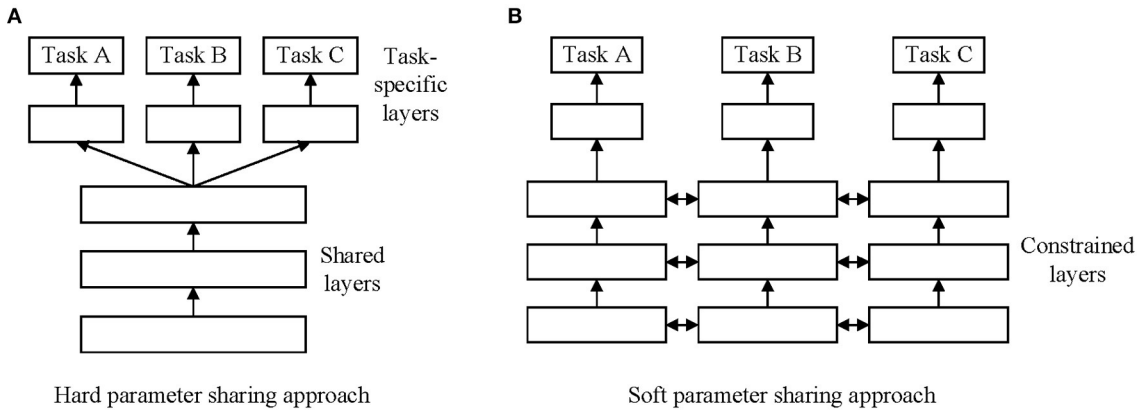

FIGURE 1 | Two approaches for multi-task learning in deep neural networks. **(A)** Hard parameter sharing approach. **(B)** Soft parameter sharing approach.

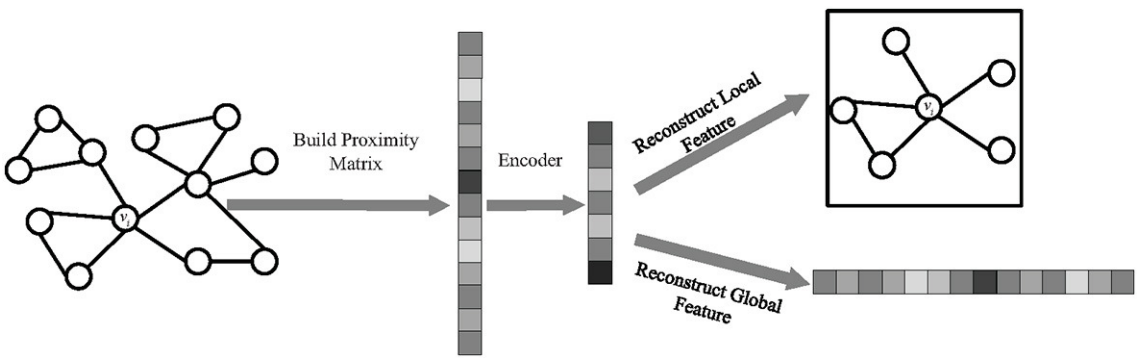

FIGURE 2 | The framework of MLNE.

Algorithm 1: Framework of the proposed MLNE

Input: Input Graph: $G = (V, E)$; Adjacency matrix: A; Number of samplings: K; Probability of resampling: γ; Weighted parameters of the loss function: α, β, η; Number of dimensions of representation vectors: R.

Output:

Representation vectors of nodes: Φ.

1: Initialize matrix of node representations $\Phi \in \mathbb{R}^{|V| \times |d|}$.
2: Construct the global proximity matrix S_{global};
3: Local proximity $S_{local} \leftarrow A$;
4: Initialize the parameters of the network: θ;
5: Input S_{global} into the neural network and train the network model by optimizing the objective function (Equation 10) by stochastic gradient descent.

such as *sigmoid* and *relu*, so as to build non-linear projections. At a high level, our model consists of three components as shown in **Figure 3**: a shared encoder network, a decoder network, and a classifier network.

The first task includes the shared encoder and the specific decoder and can be seen as a standard autoencoder model. The encoder maps the high-dimensionality structural information into a lower-dimensionality embedding space, $s_i \rightarrow z_i$, and the decoder reconstructs the structural information from the learned embeddings, $z_i \rightarrow \bar{s}_i$. In order to preserve the global structural information, the PPMI matrix is adopted as the global proximity matrix S, and a random surfing model is used to build the PPMI. The loss function can be defined as follows:

$$L_{global} = \sum_{i=1}^{n} ||s_i - \bar{s}_i||_2 = ||S - \bar{S}||_2. \tag{5}$$

The second task includes the shared encoder and the specific classifier. The task is to preserve the local structural information, so the classifier is used to reconstruct the structure of the one-hop area of the nodes. On the other words, the classifier decodes the local structural information from the learned node embeddings based on the shared encoder so as to predict the direct neighborhoods of nodes. Thus, the adjacency matrix A is adopted as the classifier's expected output Y. The second task can be seen as a multi-label classifier task, and the loss function can be defined as follows:

$$L_{local} = -\sum_{i=1}^{n} y_i \log \bar{y}_i + (1 - y_i) \log (1 - \bar{y}_i) \tag{6}$$

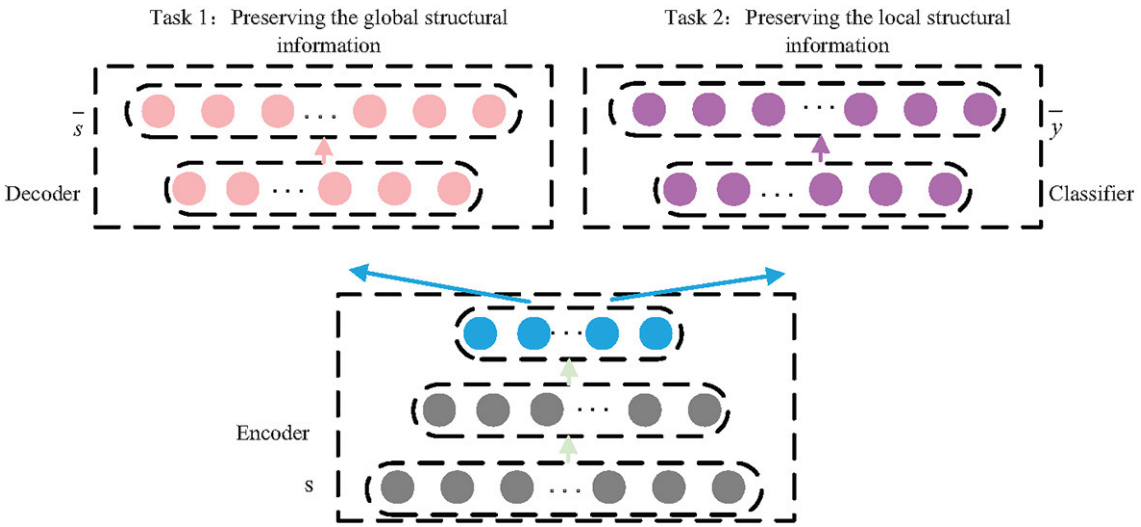

FIGURE 3 | The architecture of MLNE.

where $\bar{y}_i$ is the output of the classifier.

Furthermore, mini-batch batch gradient decent (MBGD) is used to optimize the parameters of the model. As shown in **Figure 4**, the sampled batch with a fixed number of nodes can be regarded as a sampled sub-graph. As a result of this, a regularizer component is formulated to optimize those nodes in Euclidean space so as to make nodes with edges linked closer together and nodes without edges linked farther apart.

The size of the batch is defined as M, and the adjacency matrix of the sampled sub-graph can be defined as $A_{sub}^{M\times M} \in A^{|A|\times|A|}$, where each element represents the relationship between nodes. We let $Z_{sub}^{|V|\times d} \in Z^{|V|\times d}$ be the corresponding sub-embedding matrix. The regularizer attempts to minimize the following contrastive loss:

$$L_{reg} = \frac{1}{2\times M\times M}\sum_{i=1}^{M}\sum_{j=1}^{M} A_{sub}(i,j)d_{i,j}^2 + (1-A_{sub}(i,j))\max(m-d_{i,j},0)^2 \quad (7)$$

where m is the margin and $d_{i,j}$ is the Euclidean distance between the i-th and j-th representations, $d_{i,j} = ||Z_{sub}(i) - Z_{sub}(j)||_2$.

The Euclidean distance matrix D, where each element represents the measure between representations, can be defined as follows:

$$D = H + H^T - 2G. \quad (8)$$

where Z_{sub} is Gram matrix of G and H is the Diagonal matrix of G.

The revised regularizer is shown as follows:

$$L_{reg} = \frac{1}{2\times M\times M}||A_{sub}\bigodot D + (1-A_{sub})\bigodot \max(m-D,0)||_2. \quad (9)$$

where $\bigodot$ is the Hadamard product.

In order to preserve the local and global structural information, we design a multi-task learning model and jointly optimize (Equations 5, 6, and 9). The objective function can be defined as follows:

$$L = \alpha L_{global} + \beta L_{local} + \eta L_{reg}. \quad (10)$$

where α, β, and η are the corresponding weights of each task and the regularizer.

4. EXPERIMENTS

In this section, we evaluate our proposed model, MLNE, on five real-world network datasets and three tasks, namely node classification, link prediction, and visualization. The experimental results demonstrate that MLNE has competitive performance.

4.1. Datasets

There are five real-world networks in our experiments, including a social network, citation networks, and a language network. They are listed as follows:

- *Cora* (McCallum et al., 2000) is a citation network with 2,708 nodes and 5,429 edges, where the nodes represent the scientific publications and the edges represent the citation relationship between publications. The nodes are split into seven classes according to scientific field.
- *DBLP* (Tang et al., 2008) is another citation network composed of 13,184 publications from five classes and with 95,955 edges.
- *20-NEWSGROUP* (Lang, 1995) is a language network that contains 20,000 newsgroup documents with 20 different labels. The tf-idf vectors of each word are adopted as the representations of documents, and cosine similarity is used to measure the similarity between documents. We select 592 documents from three classes, *com.graphics*, *rec.sport.baseball*,

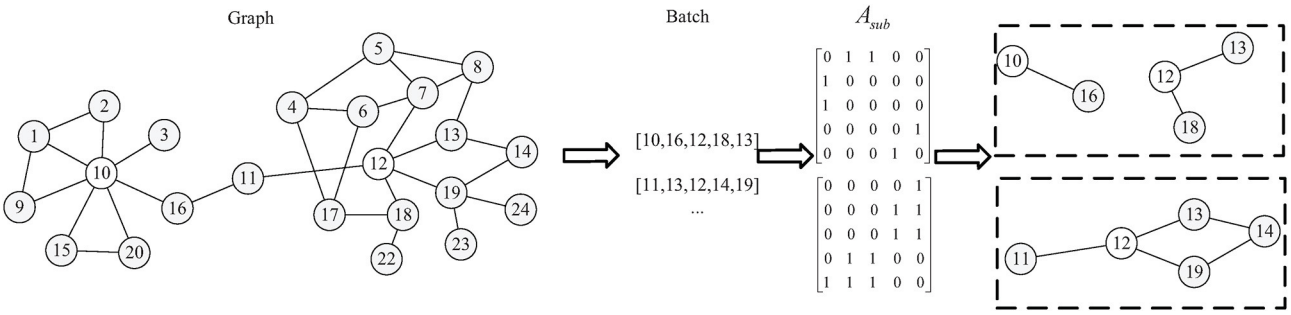

FIGURE 4 | Illustration of the regularizer.

and *talk.politics.gums* respectively, to build a network in which the nodes represent the documents and cosine similarity is the weight of each edge.

- *Blogcatalog* (Tang and Liu, 2009) is a social network in which nodes represent the authors and edges represent the relationships between users. According to user interests, there are 39 different categories, and each user is labeled with at least one category. The network contains 10,312 nodes and 333,983 edges.
- *Pubmed* (Sen et al., 2008) is a citation network collected from the PubMed database in which nodes represent scientific publications and all the nodes are classified into three classes. This network consists of 19,717 nodes and 44,338 edges.

4.2. Baseline Algorithms

We consider the following three baseline algorithms.

- *DeepWalk* adopts random walk to sample paths composed of discrete nodes, and the Skip-gram model, which has achieved great success in word embedding, is used to generate node representations from the sampled paths.
- *node2vec* optimizes the DeepWalk through jointly combining the BFS and DFS. There are two hyperparameters, p and q, that lead the sampling such that the network structure can be deeply exploited.
- *GraRep* builds k different node representations by SVD and connects them so as to generate a global node representation.

4.3. Parameter Setting

As mentioned in Perozzi et al. (2014); Grover and Leskovec (2016), we set walk length $l = 80$, number of walks $n = 10$, and window size $w = 10$ for random walk in DeepWalk and node2vec. Specifically, we employ a grid search over return and in-out hyperparameters $p, q \in \{0.25, 0.5, 1, 1.5, 2\}$ by 10-fold cross-validation for node2vec. For GraRep, we set the number of sampling steps $k = 4$ by trial and error.

In our model, the shared encoder contains an input layer and a hidden layer, where the size of the input layer is the same as the size of network V and the size of the hidden layer is the dimensionality of the node representation vector. The decoder and the classifier contain an output layer with the size of $|V|$. The *sigmoid* activation function is used in all layers. For hyperparameters, α, β, and η are set at 1000, 1, and 10, respectively, through using grid search on the validation set. As suggested in Cao et al. (2016), we set $K = 10$ and $\gamma = 0.98$ for random surfing.

For a fair comparison, the dimensionality of node representation vector d is set to 128 for all algorithms, as used in Cao et al. (2015).

4.4. Link Prediction

The link prediction task is to predict whether an edge exists between pairwise nodes in the original network. In order to conduct the task, a portion of the existing edges in the original network is randomly selected to be hidden. The remaining networks are then used as the input of NRL models. Node embeddings can then be learned from the trained models, and the inner product between the representation vectors of pairwise nodes is normalized by the *sigmoid* function. To evaluate the performance of each algorithm over the link prediction task, 10% of the hidden edges are utilized as the positive data. In addition, an equal number of edges not existing in the network is sampled as the negative data. AUC and Macro-F1 are utilized as evaluation metrics.

Table 1 shows the results of link prediction on Cora, 20-NEWSGROUP, and Blogcatalog. We find that MLNE and GraRep perform well but DeepWalk and node2vec have similar and poor performance. MLNE is consistently better than the baselines with respect to Macro-F1. For AUC, GraRep and MLNE perform similarly in most cases and outperform the others, except for in 20-NEWSGROUP, where GraRep is markedly better than MLNE, outperforming it by 22.71%.

4.5. Node Classification

Node classification is an important task in network analysis. Thus, this task is used to evaluate the quality of different learned network representations. In this experiment, *Logistic Regression* (LR) is used as a classifier. A portion of the labeled nodes are randomly selected as the training dataset, and thus the remaining nodes without labels are adopted to test the performance. The training ratio is raised from 10% to 90%. The process is repeated 10 times for all algorithms on five networks. The nodes in

Blogcatalog have at least one label and thus Micro-F1 and Micro-F1 are used as with the evaluation metrics. The experimental results are reported in **Tables 2–5**.

Table 2 shows the results of node classification on Cora. We find that MLNE has good performance. As the training ratio increases, MLNE outperforms the others on Macro-F1 and Micro-F1. When the training ratio is less than 50%, MLNE achieves better performance than DeepWalk and GraRep with a 90% training ratio. For Micro-F1, MLNE has the best performance in most cases. When the training ratio is better than 30%, MLNE achieves 0.39, 2.05, and 6.77% gains over DeepWalk, node2vec, and GraRep, respectively. For Macro-F1, MLNE has performance that is competitive with DeepWalk. As the training ratio increases, MLNE is better than the other baselines.

Table 3 shows the results of node classification on DBLP. The number of different labels in DBLP is lower than in the other networks, and thus the evaluation metrics of all of the algorithms are good. DeepWalk maintains a slight advantage over the others in most cases on Micro-F1 and Macro-F1. GraRep has a poor performance and it is worse than MLNE on those metrics, by 1.02 and 1.23%, respectively.

Table 4 shows the results of node classification on 20-NEWSGROUP. We find that MLNE has the best performance on Micro-F1 and Macro-F1. In fact, MLNE with only 10% training ratio data arrives at a result close to DeepWalk and node2vec when they are given 90% of the data. Compared with DeepWalk, the Micro-F1 values of node2vec, GraRep, and MLNE improve by 12.86, 10.29, and 3.25%. For Macro-F1, MLNE is also better than those baselines, by 13.39, 11.01, and 3.31%. DeepWalk and node2vec have similar performance and are worse than GraRep on these metrics.

Table 5 shows the results of node classification on Blogcatalog. For Micro-F1, when the training ratio is greater than 10%, MLNE is better than DeepWalk, node2vec, and GraRep, by 3.58, 3.57, and 3.68% respectively. Additionally, with a 60% training ratio of data, it beats all of the other algorithms, even when they are given a 90% training ratio. For Macro-F1, the performance of MLNE, DeepWalk, and node2vec proved much more competitive. When the training ratio is less than 60%, MLNE performs better than the baselines. As the training ratio increases from 60 to 90%, node2vec outperforms the others. GraRep has the worst performance on both metrics.

Table 6 shows the results of node classification on Pubmed. For Micro-F1, when the training ratio is equal to 10%, Deepwalk is better than MLNE, and MLNE outperforms the other algorithms. When the training ratio is greater than 10%, the proposed algorithm MLNE outperforms all of the comparison algorithms. For Macro-F1, MLNE performs better than all of the baselines. GraRep also has the worst performance on Micro-F1 and Macro-F2.

TABLE 1 | Macro-F1 and AUC on Cora, 20-NEWSGROUPS, and Blogcatalog for the link prediction task.

Model	Cora		20-NEWSGROUP		Blogcatalog	
	Macro-F1	AUC	Macro-F1	AUC	Macro-F1	AUC
DeepWalk	37.21	86.76	39.25	58.15	44.02	55.30
node2vec	33.33	83.22	33.95	59.88	38.20	55.81
GraRep	64.35	93.24	57.08	78.93	47.01	77.97
MLNE	80.95	93.76	60.24	64.32	68.67	77.47

TABLE 2 | Node classification results on Cora.

	Model	10%	20%	30%	40%	50%	60%	70%	80%	90%
Micro-F1	DeepWalk	76.68	79.55	81.00	82.08	82.85	83.11	83.11	83.58	84.39
	node2vec	77.16	79.40	80.16	81.11	81.51	81.86	81.79	82.08	82.18
	GraRep	75.16	76.64	77.25	77.92	78.06	78.48	77.75	78.28	77.75
	MLNE	75.57	79.34	81.01	82.43	82.99	83.52	83.81	84.11	84.57
Macro-F1	DeepWalk	75.27	78.40	79.91	81.19	82.04	82.24	82.19	82.62	83.41
	node2vec	75.67	78.40	79.20	80.39	80.90	81.36	81.42	81.60	81.92
	GraRep	73.21	74.97	75.52	76.31	76.41	76.74	75.96	76.68	76.15
	MLNE	74.66	78.22	79.89	81.55	82.01	82.61	83.00	83.17	83.73

TABLE 3 | Node classification results on DBLP.

	Model	10%	20%	30%	40%	50%	60%	70%	80%	90%
Micro-F1	DeepWalk	90.96	91.61	91.97	92.28	92.40	92.65	92.61	92.65	92.52
	node2vec	90.89	91.50	91.85	92.07	92.24	92.35	92.43	92.52	92.27
	GraRep	90.51	90.77	90.97	91.11	91.13	91.30	91.35	91.41	91.39
	MLNE	90.34	91.58	92.04	92.16	92.29	92.38	92.41	92.58	92.57
Macro-F1	DeepWalk	90.46	91.20	91.61	91.96	92.12	92.18	92.32	92.40	92.31
	node2vec	90.43	91.12	91.50	91.76	91.596	92.07	92.13	92.24	92.02
	GraRep	89.94	90.22	90.43	90.58	90.61	90.81	90.83	90.90	90.93
	MLNE	89.80	91.15	91.67	91.84	92.00	92.10	92.11	92.31	92.31

TABLE 4 | Node classification results on 20-NEWSGROUP.

	Model	10%	20%	30%	40%	50%	60%	70%	80%	90%
Micro-F1	DeepWalk	61.16	69.64	74.34	75.39	77.33	78.57	79.04	78.31	79.17
	node2vec	65.52	71.30	75.54	77.08	78.55	79.75	80.00	79.41	80.17
	GraRep	69.91	77.36	80.60	82.25	83.58	84.47	84.61	85.71	85.50
	MLNE	79.62	82.05	83.49	84.27	84.43	84.77	85.56	86.05	85.85
Macro-F1	DeepWalk	59.51	69.11	74.28	75.39	77.31	78.56	78.94	78.03	78.95
	node2vec	63.23	70.95	75.52	77.07	78.54	79.71	79.87	79.07	79.70
	GraRep	69.60	77.31	80.60	82.22	83.56	8445	84.49	85.56	85.33
	MLNE	79.63	82.06	83.50	84.25	84.41	84.73	85.44	85.84	85.64

TABLE 5 | Node classification results on Blogcatalog.

	Model	10%	20%	30%	40%	50%	60%	70%	80%	90%
Micro-F1	DeepWalk	33.84	36.71	38.08	38.87	39.46	39.91	40.51	40.76	41.22
	node2vec	33.83	36.49	38.09	38.95	39.67	40.04	40.34	40.96	41.02
	GraRep	36.15	38.05	38.81	39.19	39.51	39.66	39.83	39.89	40.11
	MLNE	35.74	38.85	39.83	40.58	41.15	41.34	41.64	41.80	41.87
Macro-F1	DeepWalk	19.02	22.13	23.79	24.44	25.17	25.61	26.35	26.42	26.75
	node2vec	19.71	22.77	24.63	25.69	26.43	26.80	27.13	27.75	27.70
	GraRep	19.63	22.23	22.63	23.03	23.24	23.45	23.59	23.74	24.26
	MLNE	22.27	24.40	25.39	26.16	26.44	26.56	26.78	26.85	26.94

TABLE 6 | Node classification results on Pubmed.

	Model	10%	20%	30%	40%	50%	60%	70%	80%	90%
Micro-F1	DeepWalk	80.06	80.80	80.89	80.97	81.01	80.55	80.49	80.10	80.83
	node2vec	79.23	80.03	80.16	80.42	80.21	79.91	79.78	79.87	80.83
	GraRep	79.14	79.39	79.39	79.78	79.66	79.56	79.51	78.83	79.82
	MLNE	80.03	81.09	81.44	81.55	81.62	81.71	81.76	81.82	82.35
Macro-F1	DeepWalk	78.69	79.44	79.54	79.69	79.69	79.19	79.06	78.81	79.78
	node2vec	77.70	78.53	78.61	78.96	78.63	78.40	78.16	78.52	79.78
	GraRep	77.70	78.00	77.98	78.53	78.35	78.18	78.18	77.56	78.83
	MLNE	78.73	79.78	80.12	80.24	80.32	80.42	80.44	80.52	81.11

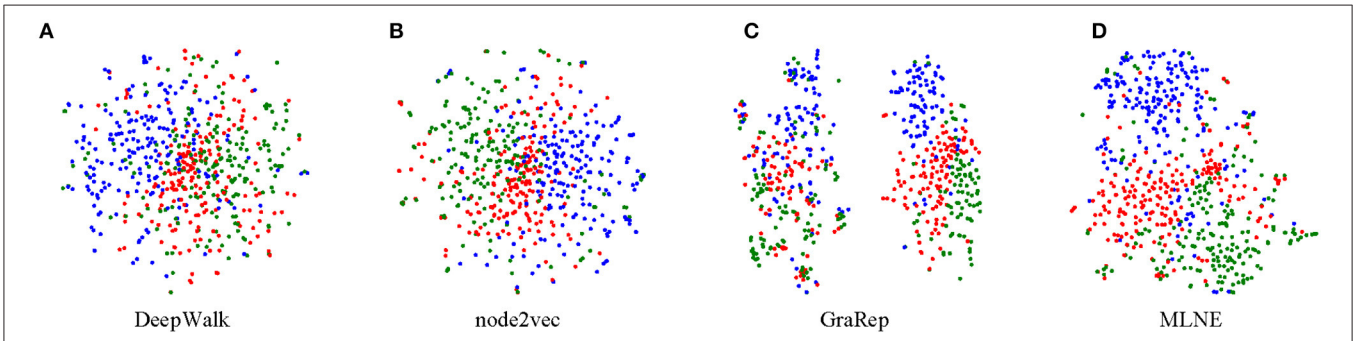

FIGURE 5 | Visualization of 20-NEWSGROUP. Each point represents a document. The color indicates the category of each document. **(A)** is the result of DeepWalk, **(B)** is the result of node2vec, **(C)** is the result of GraRep, and **(D)** is the result of the proposed MLNE.

4.6. Visualization

Visualization is another important task for exploring and analyzing a network. To conduct this task, the size of learned node embeddings is firstly reduced for display; a popular dimensionality reduction technique *t*-SNE is used to visualize the network in two-dimensional space. For documents labeled into

three categories in the 20-NEWSGROUP, three different colors indicate the corresponding points. A good visualization result keeps nodes within the same cluster close and vice versa.

From **Figure 5**, we can see that DeepWalk and node2vec do not perform well because there are no clear boundaries among the groups. For GraRep, there are two clusters where nodes also tend to mix together. Obviously, MLNE slightly outperforms the baselines and learns a good clustering, resulting in much clearer boundaries. The experimental results demonstrate the effectiveness of MLNE in the visualization task.

5. CONCLUSION

In this paper, we propose a multi-task learning-based network embedding named MLNE. In order to jointly preserve the local and global structural information, we design a model based on multi-task learning. The model is composed of three components: a shared encoder, decoder, and classifier. The shared encoder and decoder can be seen as a standard autoencoder that automatically learns representations from the global features. The shared encoder and classifier are used to reconstruct the one-hop area of a node from the learned latent representation. Additionally, a regularization based on mini-batch batch gradient descent is introduced to learn stable and robust representations. Experimental results on node classification, link prediction, and visualization tasks demonstrate the superiority of our proposed MLNE in learning node representations.

In the future, we will extend multi-task learning to heterogeneous information networks and large-scale networks.

DATA AVAILABILITY STATEMENT

The dataset *Cora* for this study can be found at https://linqs.soe.ucsc.edu/node/236. The dataset *DBLP* for this study can be found at http://arnetminer.org/citation. The dataset *20-NEWSGROUP* for this study can be found at http://qwone.com/~jason/20Newsgroups/. The dataset *Blogcatalog* for this study can be found at http://socialcomputing.asu.edu/datasets/BlogCatalog3. The dataset *Pubmed* for this study can be found at https://linqs.soe.ucsc.edu/data.

AUTHOR CONTRIBUTIONS

SW and MG designed the experiments. QW performed the experiments. SW and QW analyzed the data. SW, MG, and QW wrote the paper.

FUNDING

This work was supported by the National Key Research and Development Program of China (Grant no. 2017YFB0802200), the National Natural Science Foundation of China (Grant Nos. 61806153, 61772393, 61603299), the Fundamental Research Funds for the Central Universities (Grant No. JB191501), the National Natural Science Foundation of Shaanxi Province (Grant No. 2019JQ-311), the China Postdoctoral Science Foundation (Grant no. 2018M640961), and the National Program for Support of Top-notch Young Professionals of China.

REFERENCES

Argerich, L., Zaffaroni, J. T., and Cano, M. J. (2016). Hash2vec, feature hashing for word embeddings. *arXiv [preprint] arXiv:1608.08940*.

Belkin, M., and Niyogi, P. (2002). "Laplacian eigenmaps and spectral techniques for embedding and clustering," in *Advances in Neural Information Processing Systems* (Vancouver, BC), 585–591.

Cao, S., Lu, W., and Xu, Q. (2015). "Grarep: learning graph representations with global structural information," in *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management* (New York, NY: ACM), 891–900.

Cao, S., Lu, W., and Xu, Q. (2016). "Deep neural networks for learning graph representationsm," in *Thirtieth AAAI Conference on Artificial Intelligence* (Phoenix, AZ), 1145–1152.

Chen, H., Perozzi, B., Al-Rfou, R., and Skiena, S. (2018). A tutorial on network embeddings. *arXiv [preprint] arXiv:1808.02590*.

Donnat, C., Zitnik, M., Hallac, D., and Leskovec, J. (2018). "Learning structural node embeddings via diffusion wavelets," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (New York, NY: ACM), 1320–1329.

Du, B., Wang, S., Xu, C., Wang, N., Zhang, L., and Tao, D. (2018). Multi-task learning for blind source separation. *IEEE Trans. Image Process.* 27, 4219–4231. doi: 10.1109/TIP.2018.2836324

Fortunato, S. (2010). Community detection in graphs. *Phys. Rep.* 486, 75–174. doi: 10.1016/j.physrep.2009.11.002

Gong, M., Cai, Q., Ma, L., Wang, S., and Lei, Y. (2017). *Computational Intelligence for Network Structure Analytics*. Singapore: Springer.

Goyal, P., and Ferrara, E. (2018). Graph embedding techniques, applications, and performance: a survey. *Knowledge Based Syst.* 151, 78–94. doi: 10.1016/j.knosys.2018.03.022

Grover, A., and Leskovec, J. (2016). "node2vec: scalable feature learning for networks," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (New York, NY: ACM), 855–864.

Hamilton, W. L., Ying, R., and Leskovec, J. (2017). Representation learning on graphs: methods and applications. *arXiv [preprint] arXiv:1709.05584*.

Herman, I., Melançon, G., and Marshall, M. S. (2000). Graph visualization and navigation in information visualization: a survey. *IEEE Trans. Visual. Comput. Graph.* 6, 24–43. doi: 10.1109/2945.841119

Hu, R., Aggarwal, C. C., Shuai, M., and Huai, J. (2016). "An embedding approach to anomaly detection," in *IEEE International Conference on Data Engineering* (Helsinki), 385–396.

Lang, K. (1995). "Newsweeder: learning to filter netnews," in *Machine Learning Proceedings 1995* (Tahoe, CA: Elsevier), 331–339.

Liu, X., He, P., Chen, W., and Gao, J. (2019a). Multi-task deep neural networks for natural language understanding. *arXiv [preprint] arXiv:1901.11504*.

Liu, X., Murata, T., Kim, K.-S., Kotarasu, C., and Zhuang, C. (2019b). "A general view for network embedding as matrix factorization," in *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining* (New York, NY: ACM), 375–383.

Lü, L., Medo, M., Yeung, C. H., Zhang, Y.-C., Zhang, Z.-K., and Zhou, T. (2012). Recommender systems. *Phys. Rep.* 519, 1–49. doi: 10.1016/j.physrep.2012.02.006

Lü, L., and Zhou, T. (2011). Link prediction in complex networks: a survey. *Phys. Stat. Mech. Appl.* 390, 1150–1170. doi: 10.1016/j.physa.2010.11.027

McCallum, A. K., Nigam, K., Rennie, J., and Seymore, K. (2000). Automating the construction of internet portals with machine learning. *Inform. Retriev.* 3, 127–163. doi: 10.1023/A:1009953814988

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv [preprint] arXiv:1301.3781*.

Musiał, K., and Juszczyszyn, K. (2009). "Properties of bridge nodes in social networks," in *International Conference on Computational Collective Intelligence* (Wroclaw: Springer), 357–364.

Peng, C., Xiao, W., Jian, P., and Zhu, W. (2019). A survey on network embedding. *IEEE Trans. Knowledge Data Eng.* 31, 833–852. doi: 10.1109/TKDE.2018.2849727

Perozzi, B., Al-Rfou, R., and Skiena, S. (2014). "Deepwalk: online learning of social representations," in *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (New York, NY: ACM), 701–710.

Qiu, J., Dong, Y., Ma, H., Li, J., Wang, C., Wang, K., et al. (2019). "Netsmf: large-scale network embedding as sparse matrix factorization," in *The World Wide Web Conference* (New York, NY: ACM), 1509–1520.

Qiu, J., Dong, Y., Ma, H., Li, J., Wang, K., and Tang, J. (2018). "Network embedding as matrix factorization: unifying deepwalk, line, pte, and node2vec," in *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining* (New York, NY: ACM), 459–467.

Ruder, S. (2017). An overview of multi-task learning in deep neural networks. *arXiv [preprint] arXiv:1706.05098*.

Sen, P., Namata, G., Bilgic, M., Getoor, L., Galligher, B., and Eliassi-Rad, T. (2008). Collective classification in network data. *AI Magaz.* 29:93. doi: 10.1609/aimag.v29i3.2157

Shi, C., Hu, B., Zhao, W. X., and Philip, S. Y. (2018). Heterogeneous information network embedding for recommendation. *IEEE Trans. Knowledge Data Eng.* 31, 357–370. doi: 10.1109/TKDE.2018.2833443

Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J., and Mei, Q. (2015). "Line: large-scale information network embedding," in *Proceedings of the 24th International Conference on World Wide Web* (Geneva: International World Wide Web Conferences Steering Committee), 1067–1077.

Tang, J., Zhang, J., Yao, L., Li, J., Zhang, L., and Su, Z. (2008). "Arnetminer: extraction and mining of academic social networks," in *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (New York, NY: ACM),990–998.

Tang, L., and Liu, H. (2009). "Relational learning via latent social dimensions," in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (New York, NY: ACM), 817–826.

Thung, K.-H., and Wee, C.-Y. (2018). A brief review on multi-task learning. *Multi. Tools Appl.* 77, 29705–29725. doi: 10.1007/s11042-018-6463-x

Tsoumakas, G., and Katakis, I. (2007). Multi-label classification: an overview. *Int. J. Data Warehous. Mining* 3, 1–13. doi: 10.4018/jdwm.2007070101

Wang, D., Cui, P., and Zhu, W. (2016). "Structural deep network embedding," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (New York, NY: ACM), 1225–1234.

Wang, N., Wang, H., Jia, Y., and Yin, Y. (2018). "Explainable recommendation via multi-task learning in opinionated text data," in *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval* (New York, NY: ACM), 165–174.

Wang, X., Cui, P., Wang, J., Pei, J., Zhu, W., and Yang, S. (2017). "Community preserving network embedding," in *Thirty-First AAAI Conference on Artificial Intelligence* (San Francisco, CA), 203–209.

Wei, X., Xu, L., Cao, B., and Yu, P. S. (2017). "Cross view link prediction by learning noise-resilient representation consensus," in *Proceedings of the 26th International Conference on World Wide Web* (Geneva: International World Wide Web Conferences Steering Committee), 1611–1619.

Zhang, D., Han, J., Yang, L., and Xu, D. (2018). Spftn: a joint learning framework for localizing and segmenting objects in weakly labeled videos. *IEEE Trans. Pattern. Anal. Mach. Intell.* doi: 10.1109/TPAMI.2018.2881114. [Epub ahead of print].

Zhang, D., Yin, J., Zhu, X., and Zhang, C. (in press). Network representation learning: a survey. *IEEE Trans. Big Data*. doi: 10.1109/TBDATA.2018.2850013

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.

frontiers in Neuroscience

ORIGINAL RESEARCH
published: 17 January 2020
doi: 10.3389/fnins.2019.01396

A Preliminary Study of Knowledge Transfer in Multi-Classification Using Gene Expression Programming

Tingyang Wei[1] *and Jinghui Zhong*[1,2]*

[1] School of Computer Science and Engineering, South China University of Technology, Guangzhou, China, [2] Sino-Singapore International Joint Research Institute, Guangzhou, China

Gene Expression Programming (GEP), a variant of Genetic Programming (GP), is a well established technique for automatic generation of computer programs. Due to the flexible representation, GEP has long been concerned as a classification algorithm for various applications. Whereas, GEP cannot be extended to multi-classification directly, and thus is only capable of treating an *M*-classification task as *M* separate binary classifications without considering the inter-relationship among classes. Consequently, GEP-based multi-classifier may suffer from output conflict of various class labels, and the underlying conflict can probably lead to the degraded performance in multi-classification. This paper employs evolutionary multitasking optimization paradigm in an existing GEP-based multi-classification framework, so as to alleviate the output conflict of each separate binary GEP classifier. Therefore, several knowledge transfer strategies are implemented to enable the interation among the population of each separate binary task. Experimental results on 10 high-dimensional datasets indicate that knowledge transfer among separate binary classifiers can enhance multi-classification performance within the same computational budget.

Keywords: gene expression programming, evolutionary multitasking, classification, genetic programming, evolutionary computation

OPEN ACCESS

Edited by:
Huajin Tang,
Zhejiang University, China

Reviewed by:
Yaqing Hou,
Dalian University of Technology (DUT), China
Lei Zhou,
Chongqing University, China
Zhou Wu,
Chongqing University, China

***Correspondence:**
Jinghui Zhong
jinghuizhong@gmail.com

Specialty section:
This article was submitted to Decision Neuroscience, a section of the journal Frontiers in Neuroscience

Received: *29 August 2019*
Accepted: *10 December 2019*
Published: *17 January 2020*

Citation:
Wei T and Zhong J (2020) A Preliminary Study of Knowledge Transfer in Multi-Classification Using Gene Expression Programming. Front. Neurosci. 13:1396. doi: 10.3389/fnins.2019.01396

1. INTRODUCTION

Classification is a fundamental and active research topic in data mining. Various real-world applications involving medical diagnosis, image categorization, credit approval, and etc., are covered by classification techniques. Formally, in a classification task, a classifier is to assign a class label k to the given input data X_i with features $X_i^1, X_i^2, ..., X_i^N$ after being trained by data $X_1, X_2, ..., X_M$, where N and M represent the number of the features and the sample size, respectively. In this paper, we focus on the multi-classification problems in which the number of the candidate values for class labels is larger than two.

Generally, machine learning methods involving Neural Networks (Krizhevsky et al., 2012), Random Forests (Breiman, 2001), Support Vector Machine (Chang and Lin, 2011), and etc., are applied to solve the multi-classification problems. Considering the issue of the curse of dimensionality, many evolutionary algorithms (EA) have been utilized to assist aforementioned machine learning methods to tackle high-dimensional datasets, including Artificial Bee Colony (ABC) (Hancer et al., 2018), Particle Swarm Optimization (PSO) (Xue et al., 2012; Tran et al., 2018), and Genetic Programming (GP) (Chen et al., 2017). To be specific, these population-based algorithms can evolve individuals with a fitness function with respect to the machine learning classifier, and therefore can be conducted in either single-objective or multi-objective fashion. By

searching effective feature subsets and limiting the subset size using EAs, the classifier can be trained in a more efficient way and the classification results can be more interpretable.

Unlike other population-based algorithms that must be implemented with a given machine learning classifier, GP is capable of completing both feature selection and classification independently owing to its tree structure. By converting the tree structure of GP into a string structure, Gene Expression Programming (GEP) (Zhong et al., 2017), a variant of GP, enjoys the same benefit as GP of independent classification ability with additional power of controlling bloat issue by restricted string length (Ferreira, 2002). With the automatic construction capability, GEP-based methods have emerged to show high effectiveness on symbolic regression (Cheng and Zhong, 2018; Huang et al., 2018; Zhong et al., 2018b), time series prediction (Zuo et al., 2004), knowledge discovery (Zhong et al., 2014), and etc.

Although GP and GEP can construct classification rules independently and have been prevailing in a plethora of applications involving spectral image categorization (Rauss et al., 2000), radar imagery recognition (Stanhope and Daida, 1998), medical diagnosis (Gray et al., 1996), credit approval (Sakprasat and Sinclair, 2007), and etc., they cannot be directly applied to multi-classification. To adapt GP and GEP to multi-classification, most researchers are devoted to manually configuring some contrived rules to achieve collision avoidance of class labels, thereby combining the results of multiple binary classifiers. In Muni et al. (2004), a novel evolutionary operator is designed to guide the population, and a meta-heuristic rule is supplied to iteratively remove output collision of different binary classifiers. To avoid output collision, the order, that the varying binary classifiers come into effect for prediction, can also be redesigned according to the accuracy and the reciprocal training samples (Zhou et al., 2003). Moreover, the well-established multi-objective techniques can also enhance the multi-classifiers by maintaining a pareto front of binary classifiers by considering precision, recall, and classification rule size, and employing negative voting to avoid output collision numerically (Nag and Pal, 2015). Notably, any individual in population of aforementioned GP and GEP can only be a binary classifier, hence it is still unnatural to extend these algorithms to multi-classification in despite of explorations in past few years. Furthermore, since nearly all the GP-based and GEP-based multi-classification methods straightforwardly depend on binary classifiers, it is fitness function and combining strategy of binary classifiers that relatively matter in the algorithmic design.

As discussed above, existing GP and GEP methods for multi-classification generally adopt contrived rules to avoid output collision of binary classifiers, and a crucial cause for output collision is the separate training process for each binary classifier, which potentially degrades the performance of multi-classifiers. In fact, intuitively, a classification rule trained by binary classifiers of one class can hopefully be utilized by another class as a rule component that can to some extent boost its own binary classification performance through recognizing the pattern of negative samples. According to the consideration above, this paper takes into account the Evolutionary Multitasking paradigm (Gupta et al., 2015, 2017; Ong and Gupta, 2016; Bali et al., 2019) to facilitate the multi-classification avoiding output collision of binary classifiers by enhancing the knowledge transfer among multiple binary classifiers. Equipped with the capability of latent genetic transfer, Evolutionary Multitasking can resolve many optimization problems simultaneously by enabling the knowledge transfer among different problems through the unified chromosome representation. In control of the synergies of searching space for varying optimization tasks (Gupta et al., 2016a,b; Da et al., 2018; Zhou et al., 2018), Evolutionary Multitasking, which can be easily employed on existing population-based algorithm (Feng et al., 2017; Chen et al., 2018; Liu et al., 2018; Zhong et al., 2019), have shown promising results on a vast number of cases in multi-objective optimization (Gupta et al., 2016c; Feng et al., 2018), symbolic regression (Zhong et al., 2018a), capacitated vehicle routing problems (Zhou et al., 2016), expensive optimization tasks (Min et al., 2017), and can be extended to a large scale version (Chen et al., 2019; Liaw and Ting, 2019) to enable some more scalable applications in the future. The methodology of Evolutionary Multitasking paradigm naturally fits the multi-classification problem, by treating each binary classification problem as an optimization task within certain function evaluations. Notably, concerning the multi-classification as Evolutionary Multitasking problem does not require a design for unified representation as the canonical Multifactorial Evolutionary Algorithm (MFEA) (Gupta et al., 2015) does, since each binary classification task (optimization task) in this scenario shares the same solution representation.

For canonical GP, knowledge transfer especially for Evolutionary Transfer Learning, has been widely investigated in past few years. Generally, two sorts of strategies prevails for knowledge transfer in canonical GP, modularization and initialization (O'Neill et al., 2017). For modularization, fitter canonical GP individuals in source domain can be evaluated and extracted as new function units in the GP population in the target domain (O'Neill et al., 2017), which eliminates the uncommon features between source domain and target domain. For initialization that is a really simple and direct way, GP individuals of higher fitness value in source domain and their subtrees often serve as the initial individuals and favorable components to select (Muller et al., 2019). Initialization techniques also include the knowledge transfer with respect to the feature importance. Using the ranks and fitness value of population in the source domain problems to vote for each feature, relatively fair feature importance can be obtained to guide the evolution of the target domain problems (Ardeh et al., 2019). Whereas, most relevant researchers have focused on the Evolutionary Transfer Learning, where one or several source problems are applied to assist the target problems, rather than the Evolutionary Multitasking, in which various problems are solved simultaneously with the same priority. Moreover, the existing works mainly rely on experiment design related to individual structure of canonical GP, so it is possible that the same strategies may not work in some variants of canonical GP. Therefore, as an important variant of canonical GP, GEP, with a string structure which is distinct from that of GP, should be investigated with

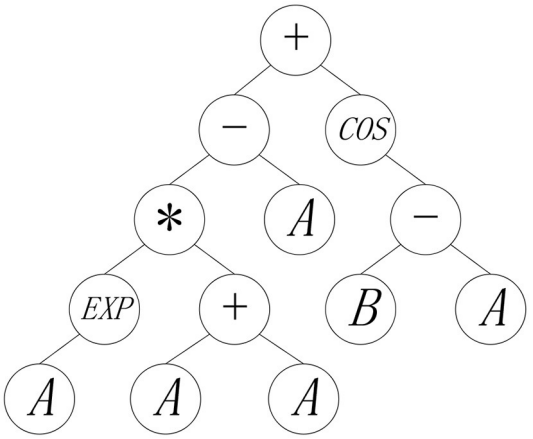

FIGURE 1 | The encoding tree for mathematical expression.

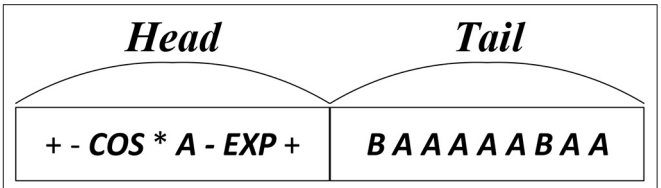

FIGURE 2 | The encoding string for mathematical expression.

some similar knowledge transfer techniques of Evolutionary Multitasking that is more general than Evolutionary Transfer Learning, for more potential promising possibilities. In this paper, GEP methods with different variation operators are employed with corresponding knowledge transfer techniques to show the effectiveness and the limits of the Evolutionary Multitasking methods in multi-classification, based on an existing multi-classification framework designed for GEP.

The rest of this article is organized as follows. Section 2 introduces a GEP-based multi-classification framework that the experimental study is based on. The canonical Evolutionary Multitasking paradigm, MFEA, is described briefly in section 3. The proposed knowledge transfer strategies are presented in section 4, followed by the experimental study in section 5. Eventually, the conclusions are drawn in section 6.

2. GEP MULTI-CLASSIFICATION FRAMEWORK

AccGEP (Zhou et al., 2003) is a well designed GEP-based algorithm for multi-classification. Hence, considering the prevalence and the maturity of this framework, this article will employ AccGEP to serve as the baseline method for the study of knowledge transfer. In this section, the basic concept and the algorithmic details of GEP will be presented, followed by the introduction of AccGEP.

2.1. GP and GEP

As a member of evolutionary algorithms, GP generally considers each solution for optimization problem as an individual of the whole population, in which the evolution of the algorithm is driven by variation operators encompassing mutation operators, crossover operators, and selection operators (Poli et al., 2008) among the individuals, like most meta-heuristic algorithms.

Different from other population-based methods, the representation of each individual of GP is a mathematical expression encoded by a tree, where input variables are represented by leaf nodes, and the function operators like "−" and "*sin*," are represented by intermediate nodes having offspring size of the same value with corresponding operands. For instance, **Figure 1** depicts an individual that is encoded by mathematical expression, $2A\exp(A) - A + \cos(B - A)$, in GP population. For this mathematical expression, given the specific values of A and B, the output of the individual can be decoded in a bottom-up fashion to the root node of the representation tree. In canonical GP, the mutation, crossover, selection variation operators are applied to search for the more effective tree structures, thereby yielding the acceptable individuals with the satisfactory fitness values.

Distinct from canonical GP, GEP owns a string-based structure for each individual. Illustrating the same mathematical expression with the encoding tree of **Figure 1**, the string-based structure of GEP individual can be depicted as **Figure 2**, where the encoding tree is encoded by the string structure in a breadth-first-search traverse way. As illustrated by **Figure 2**, each individual of GEP population is composed of two parts, head part and tail part. In GEP, both the function units and terminal (i.e., variable) units constitute the head part of the string, while no function units but only the terminal units occur in tail part. During the evolution process of GEP, each string-based individual maintains a fixed length for both the head part and the tail part. Precisely speaking, a predefined constraint should be exerted on the length of head part(h) and the length of tail part(l) that:

$$l = h \cdot (u - 1) + 1 \tag{1}$$

where u amounts to the maximum operand of the function unit, so as to guarantee that the encoded mathematical expression is complete (Poli et al., 2008). Furthermore, due to the breath-first-search traverse encoding mechanism, it is possible that some of the nodes saved in the string structure will not be utilized to encode mathematical expression.

2.2. AccGEP for Multi-Classification

With the capability of constructing mathematical expression, GEP-based algorithms is able to solve regression problems naturally, and can tackle binary classification issues by posing threshold values on regression tasks. For multi-classification problems, like most GEP-based classifiers, AccGEP, employed one-against-all (Aly, 2005) learning method, that is, treating an M-classification problem as M binary classification tasks. In one-against-all strategy, each binary classification problem is adopted

to decide the data samples whether or not belong to a specific class, according to the fittest rule in the GEP population as follow:

$$\begin{cases} X_i \in Class_j, GEP(X_i) > 0 \\ X_i \notin Class_j, GEP(X_i) \leq 0 \end{cases}. \quad (2)$$

Algorithm 1: *Covering Strategy*

Input: E_+ (*set of positive examples*), E_- (*set of negative examples*)
Output: H (*A set of GEP-based rules*)

```
/* Initialization */
H ← ∅
L_min ← +∞ (minimum description length obtained)
L_H ← 0 (current description length)
L_theory ← 0 (theory bits)

/* Learning */
Repeat
    Learn a rule R to cover the positive samples in E+
    E+ ← E+ − {s | s can be covered by R}
    /* Pruning */
    L_theory ← L_theory + number of bits for encoding R
    L_exception(H) ← number of bits for encoding current exceptions
    L_H ← 0.5 · L_theory + L_exception(H)
    If (L_H < L_min) Then
        H ← H ∪ {R}
    Else
        Termination
    /* Update */
    If (L_min > L_H) Then
        L_min ← L_H
Until E+ == ∅
```

To deal with the complex feature spaces in multi-classification (Zhou et al., 2003), AccGEP applied the covering strategy to learn multiple rules for each binary classification problem. As shown in the algorithm 1, for each binary classification issue, AccGEP is designed to exploit a rule set that can cover all the positive data samples, and each rule in the rule set should be learnt by GEP and the according positive sample set with some criteria in each iteration. To be specific, the fitness function of each rule is designed as follow:

$$Fitness(R) = \begin{cases} 0, Pre < 0 \\ Pre \cdot \exp(Rec - 1), Pre \geq 0 \end{cases}. \quad (3)$$

where *Pre*, *Rec* represent the precision and recall in binary classification, respectively. Generally, *Pre* is computed as the ratio of true positive samples and predicted positive samples, while *Rec* is computed as the ratio of true positive samples and all positive samples. However, since the positive sample set, E_+ in algorithm 1, shrinks in each iteration, a new formula for computing *Pre* is presented in AccGEP to better take advantage of the distribution information:

$$Pre = \left(\frac{TP}{TP + FP} - \frac{P}{P + N}\right) \cdot \frac{P + N}{N} \quad (4)$$

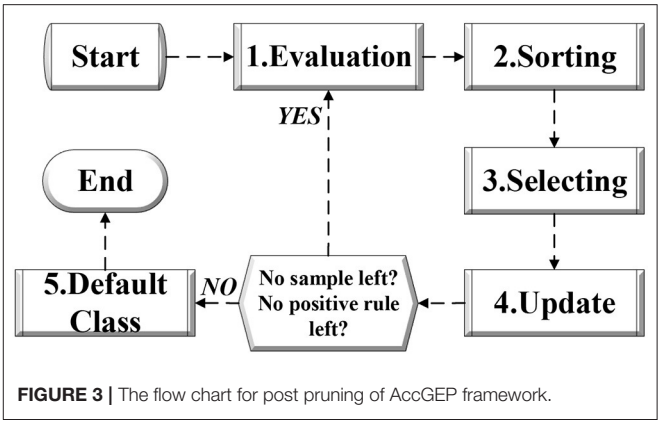

FIGURE 3 | The flow chart for post pruning of AccGEP framework.

where *TP*, *FP*, *P*, *N*, stand for the number of true positive samples, false positive samples, all positive samples in training set, all negative samples in training set of binary classification, correspondingly.

In order to allay the structural risks, the minimum description length principle in information theory is employed as a pruning technique for early stopping. As indicated in algorithm 1, $L(H)$ stands for the description length of the current rule set, H. The learning process is terminated when the description length of rule set no longer declines. Moreover, $L_{exception}$ and L_{theory} amount to the bits for encoding the error of the rule set, and the bits for encoding the rule set itself. The computation formula of the two description length are defined as follow:

$$\begin{cases} L_{exception}(H) = \log_2(C_{TP+FP}^{FP}) + \log_2(C_{TN+FN}^{FN}) \\ L_{theory}(H) = \log_2(N_c) \sum_{i=1}^{s} L(R_i) \end{cases}. \quad (5)$$

where *TP*, *FP*, *TN*, *FN*, N_c, s, $L(R_i)$, represent true positive samples, false positive samples, true negative samples, false negative samples, the number of distinct symbols applied in GEP, the number of the current rules, the valid length of individual for rule R_i, accordingly.

Having obtained multiple decision rules for each binary classification, a post-pruning technique is employed to combine the rules to yield the final results of multi-classification. Generally, as depicted in **Figure 3** the combining strategy consists of steps as follow:

- *Evaluation*: In evaluation process, all the active rules in the rule set should be evaluated according to the fitness function as well as the existing samples in the training set.
- *Sorting*: In sorting process, all the active rules in the rule set should be sorted based on the fitness values.
- *Selecting*: In selecting process, the rule with the highest fitness value is selected, then it is moved into an ordered rule set. For the original rule set, the selected rule is removed.
- *Update*: In updating process, all the samples covered by the selected rule in selecting process will be removed as well.
- *Default Class*: With remaining samples and remaining rules that are able to cover any sample, AccGEP will proceed with the cycle from step 1 to step 4 as illustrated in **Figure 3**. Otherwise, the iteration will terminate and a default class label

is decided, so as to avoid the scenario when all the rules will reject a new example. In general, the default class label will be set as the one that has most samples in the remaining sample set at the end of the algorithm cycle introduced above. Nevertheless, when there is a tie in the sample count in the remaining sample set or the remaining sample set is empty, the default class label can be determined randomly.

Through post-pruning process, AccGEP can attain an ordered rule set as well as a default class label. Subsequently, in the prediction phase for testing data, each testing sample belonging can be determined by the first rule that covers it in the ordered rule set. If a testing sample is rejected by all the rules, then the default class label will be assigned.

3. MULTIFACTORIAL EVOLUTIONARY ALGORITHM

Inspired by the bio-cultural multifactorial inheritance, MFEA (Gupta et al., 2015), a typical Evolutionary Multitasking algorithm, is designed to fully exploit the potential of population-based algorithm to solve several optimization issues simultaneously. By introducing variables including factorial rank r, skill factor τ, scalar fitness ϕ, MFEA can enable the knowledge transfer among varying problems through a unified solution representation. Initially, all the initial solutions in the population should be evaluated across all the target problems. Subsequently, each individual will be assigned with a skill factor τ to indicate the task in which it has the most promising result. At length, the skill factor τ is determined by the factorial rank r of an individual across all the tasks as $\tau = \arg_j \min(r_j)$, and then the scalar fitness ϕ can be computed accordingly by $\phi = \frac{1}{r_\tau}$. In order to improve the algorithm efficiency, in the subsequent evolution process, each individual will be only evaluated for the optimization task of its skill factor. By enabling the associative mating (Gupta et al., 2015), the skill factor of a certain individual can possibly undergo the variation.

With the techniques of assortative mating and selective evaluation for knowledge transfer, MFEA basically can comply with the similar work flow with the conventional Evolutionary Algorithms. In general, the main steps of MFEA can be illustrated as follow:

- *Initialization*: To start with, an initial population, P, is produced in MFEA. Then, all the individuals should be evaluated under all the problems, thereby getting the corresponding τ, ϕ, r.
- *Assortative Mating*: In each generation, the offspring will be generated through the conventional genetic operators including mutation and crossover. In MFEA, a control parameter, *rmp*, is applied to indicate the probability of the crossover between two individuals of different skill factor τ, which is concerned as a process of knowledge transfer. Otherwise, the crossover for parents of the same τ, or the mutation upon a single parent, is implemented.
- *Selective Evaluation*: Having generated an offspring population O, those individuals that undertake the crossover of different skill factor have undetermined τ. Intuitively, the skill factor of an individual should be set randomly based on the values of its parent. For those offsprings that merely undergo the casual crossover or mutation operator, skill factor will simply imitate their parents, known as a cultural transmission process (Gupta et al., 2015). Subsequently, the whole population will only be evaluated according to their best tasks. Aside from the optimization task τ, the fitness value of an individual for other problems should be assigned with ∞, in order that the true factorial ranks r would not be affected.
- *Population Update*: At the end of each generation, the skill factors and the scalar fitness of hybrid population $O \cup P$ should be re-evaluated to maintain only the individuals owning the best scalar fitness.

As discussed in section 1, an essential trigger for output collision in multi-classification is the separate training process of each binary classifier. Intuitively, binary classifier for different class labels might share some common structures and even some influential features. Based on the consideration above, it is believed that the latent genetic transfer attribute in Evolutionary Multitasking can enhance the performance of the existing GEP-based classifier by enabling the interaction among binary classifiers.

4. PROPOSED ALGORITHM

In this section, an Evolutionary-Multitasking-based classification method using GEP (EMC-GEP) is proposed. First, the general framework of the algorithm architecture is given. Then different knowledge transfer strategies for distinct GEP variation operators are discussed in section 4.2.

4.1. Framework

As portrayed by **Figure 4**, the whole algorithm can be divided into four sections. First, the M-classification problem is degraded as M binary classification through One-Against-All learning. Then, each binary classification will be concerned as an optimization task that is tackled by each subpopulation, *POP*, that owns an archive, A. During the iterative evolution, all the subpopulation will undergo the variation operator as well as knowledge transfer. As depicted in **Figure 4**, the evolution process of each subpopulation include three parts: evolution within own subpopulation, knowledge transfer from its own archive, and knowledge transfer from the other archives. Notably, after each evolution iteration, the archive of each subpopulation will be updated as well. After the evolution process, the whole population can obtain various classification rules for each binary classification problem. With these learnt classification rules, the rules combination process (i.e., the post-pruning process depicted in section 2 and **Figure 3**), can combine all the binary rules to yield an ordered rule set (as well as a default class label as explained in section 2), so as to resolve the M-classification problem eventually. Each section will be detailed as follows.

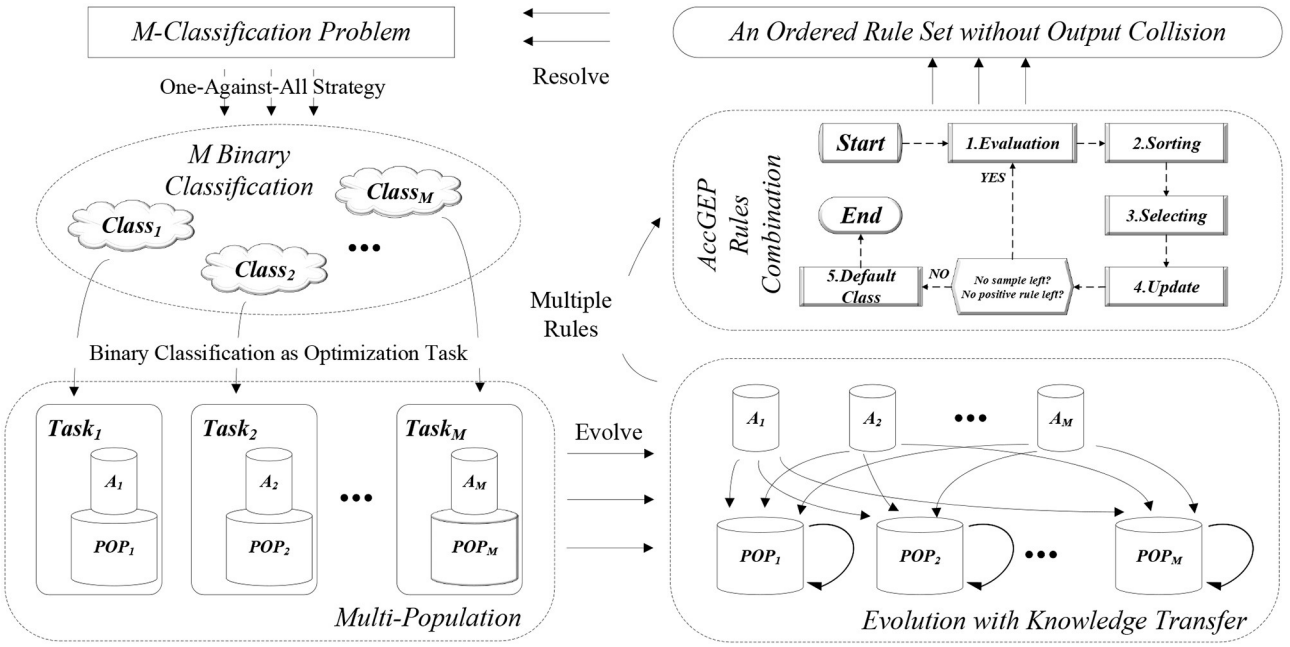

FIGURE 4 | The general flow chart for the EMC-GEP framework.

4.1.1. One-Against-All Strategy

One-Against-All is a casual strategy that treats the *M*-classification as *M* separate binary classification problems. As a variant of Error-Coding Output Codes (ECOC) (Dietterich and Bakiri, 1994), One-Against-All strategy is computationally efficient, compared with other ECOC-based strategies. Based on One-Against-All strategy, EMC-GEP will learn multiple rules for each binary classification problem, which is the same as the covering strategy of AccGEP as algorithm 1, thereby enhancing the robustness and stability of the classification framework.

4.1.2. Paradigm in Multi-Population

In this article, the MFEA paradigm is implemented in a multi-population fashion as illustrated in **Figure 4**, with each subpopulation focusing on one optimization task. Since the canonical paradigm simply evolves the population as a whole encompassing all the target tasks and only one individual is reserved for each task in each iteration, it is possible that the collected information for each task is scant to guide the population to evolve. Moreover, the original framework of MFEA have only one control parameter *rmp* (Gupta et al., 2015) to enable the assortative mating for individuals of the unified representation, but such a framework cannot facilitate more flexible and extensible operation based on the population. Hence, based on discussion of Chen et al. (2018), Gong et al. (2019), and Liu et al. (2018), multi-population mechanism is employed to improve the stability of the MFEA paradigm and to enable more flexible operation on both sub-population and mixed-population (Chen et al., 2018).

For each task, a population, *POP*, is maintained along with an archive, *A*. Population, *POP*, is maintained to enable the flexible population-based operations and variation operators. Archive, *A*, is used to record some successful individuals of the corresponding *POP*, in order that those successful individuals hopefully can transfer their valuable solution components to its own *POP* or other *POP* in later knowledge transfer phase. Similar with the *POP*, each archive *A* will be updated according to the reciprocal subpopulation. Initially, each archive *A* should be initialized randomly. Then, after each evolution iteration, the individuals in population and the individuals in archive will be both sorted. With certain archive replacement probability, *arp*, the individuals in archive will be replaced by some individuals in population. Notably, the archive size is strictly smaller than that of population. Hence, with larger *arp*, the archive tends to resemble the fittest individuals in current population, while with smaller *arp*, more successful individuals in the searching history can be recorded, thereby enhancing the diversity of the archive individuals. Specifically, the archive update mechanism is illustrated as algorithm 2. It is notable that the fittest individual for each population may not be stored in the archive. The rationale behind this idea is intuitive. Generally, the archive is used to update two sorts of populations, its own population and other populations. To update its own population, the fittest individual is not necessarily stored in the archive since the self population expects more randomness and history information from the archive. To update other populations, although the fittest individual may own the most useful information in its own problem context, the synergies between the source archive and target population is uncertain. Therefore, we apply a loosely organized update archive to store both the good individuals and historical individuals to provide a more comprehensive transfer for other populations.

Algorithm 2: *Archive Update*

Input: POP_i, A_i, arp, n(*size of* A_i)
Output: A_i

```
/* Sorting */
Sorting population POP_i according to the fitness value
Sorting archive A_i according to the fitness value

/* Update */
j ← 0
For j < n Do
    If (rand(0, 1) < arp) Then
        A_{i,j} ← POP_{i,j}
        (A_{i,j}, POP_{i,j} amount to the j-th individual in A_i, POP_i)
    j ← j + 1
End For
Return A_i
```

4.1.3. Evolution Process

Distinct from previous works on knowledge transfer in multi-population (Chen et al., 2018; Liu et al., 2018), where the evolutionary operator is directly employed on two different populations, this paper utilizes the archive as the group of representative individuals of each population for knowledge transfer. As depicted in **Figure 4**, the evolution process of EMC-GEP involve three sections, self evolution (i.e., $POP_i \leftarrow POP_i$), self transfer (i.e., $POP_i \leftarrow A_i$), cross transfer (i.e., $POP_i \leftarrow A_j$). The reason why self transfer is adopted in this paper is that, some useful solution components may not be fully exploited and may be forgotten by the subpopulation. Therefore, it is believed that the knowledge transfer from the "former" subpopulation toward the current subpopulation may help as cross knowledge transfer.

Generally, MFEA paradigm employs a probability variable *rmp* to control the mutual knowledge transfer for individuals of distinct skill factors (Gupta et al., 2015). Whereas, in this paper, a step-wise transfer control mechanism is applied to enable a more stable knowledge transfer process like (Da et al., 2018). As illustrated in algorithm 3, the transfer process is launched whenever the iteration count t can be divided by a certain transfer interval δ. Unlike those methods that try to adaptively select a similar task to transfer (Chen et al., 2019), as a preliminary study, this paper simply randomly selects an archive A_j for each subpopulation POP_i, where i may not necessarily differ from j due to the discussion above.

It is notable that, same with covering strategy of AccGEP in algorithm 1, EMC-GEP also learns multiple rules for each binary classification task, and consistently the number of each binary classfication rules for distinct tasks can be different. Hence, it is possible that some binary classification tasks are still searching for the rules to cover the positive samples, while other tasks may already terminate. In this special circumstance, those archives, of which the reciprocal population's learning process has terminated, will still remain for knowledge transfer of those active population, and will undergo no changes during the evolution.

Algorithm 3: *Evolution with Knowledge Transfer*

Input: $POP_1, POP_2, ..., POP_M, A_1, A_2, ..., A_M, \delta$ (*transfer interval*)
Output: *New Population and New Archives*

```
/* Preparation */
t ← 1
Generate initial M population randomly.
Initialize M archive with corresponding population.
/* Evolution */
While ending condition not satisfied Do
    /* Searching */
    For each subpopulation POP_i Do
        If t % δ == 0 Then
            /* Transfer */
            k ← rand(1, M)
            POP_i ← Transfer(POP_i, A_k)
        Else
            Self Evolution
    End For
    /* Updating */
    For each archive A_i Do
        A_i ← Update(POP_i, A_i) as Algorithm 2
    End For
    t ← t + 1
End While
```

4.1.4. Rules Combination Using AccGEP

After the evolution process for those population aiming at varying binary classification tasks, we can obtain a vast number of classification rules, among which multiple rules are utilized for the same binary classification issue. To avoid output conflict, a combination phase is necessary for analyzing these rules. In this paper, EMC-GEP will adopt the same strategy as the post-pruning phase in AccGEP, which has already been specifically explained in **Figure 3** and section 2.2.

4.2. Knowledge Transfer

The knowledge transfer has been investigated in various population-based algorithms, and the investigation mainly concentrated on the chromosome representation (Zhou et al., 2016; Zhong et al., 2018a), and the problem similarity (Da et al., 2018; Chen et al., 2019). However, in this paper, the problem representation for each binary classification problem does not require redesign, and we tend to select the archive randomly to assist the target task. The vital concern of our study is that, most knowledge transfer research highly depends on the data structure, and the efforts on GEP-based method are insufficient to supply a brief understanding of knowledge transfer effect on GEP. Hence, to add to a preliminary insight, this paper tries to employ knowledge transfer operations on different evolutionary operators in GEP.

4.2.1. GEP With Canonical Variation Operator

Originally, the variation operators of GEP include mutation operator and crossover operator (and sometimes rotation operator) based on the string structure in **Figure 2**. Considering

the data structure, the variation operator that really matters in knowledge extraction of the canonical GEP is crossover operator, since crossover operation can extract a continuous segment of an individual, and it is believed that the continuous segment can serve as useful genetic material for some classification problems.

To be specific, in GEP, crossover operator involves two operations, single-point crossover and two-point crossover. For single-point crossover, the crossover operation of GEP individuals resemble the behavior of Genetic Algorithm. Due to the breath-first-search encoding style of GEP, the forward part of crossover can serve as a skeleton of an expression tree. Taking the expression tree in **Figure 1** as an example, the first four operators, "+," "−," "cos," "∗," in a combination as first four nodes in a string-based individual, can construct a basic skeleton of the whole mathematical expression, which can be regarded as a form of transferable knowledge. For two-point crossover, the skeleton of an expression tree can also be extracted in the same way as one-point crossover. Moreover, with more segmented structure, the two-point crossover can hopefully extract the useful structure of an individual more flexible by enabling cutting out the intermediate string section of GEP individual.

To achieve the knowledge transfer through crossover operator, whenever knowledge transfer is launched in algorithm 3, the two parents of a crossover operator should be selected in target population POP_i and the source archive A_k separately. Aside from the selection choice, the crossover operation remains unchanged in other respects.

4.2.2. GEP With DE-Based Variation Operator

Besides the conventional evolutionary operators, some variants of GEP methods can employ DE-based operators by transforming the string construction process into a continuous optimization method, which is highly extensible and has shown promising capability in applications like symbolic regression (Zhong et al., 2015).

In general, individuals in Differential Evolution (DE) (Storn and Price, 1997) should undergo mutation operation, where each element in individual will be replaced, in certain probability, by some random element added to a scaled difference element (Storn and Price, 1997). There are various mutation strategies frequently applied in the literature involving "DE/rand/1," "DE/current-to-best/1," "DE/best/1," In this paper, as in Zhong et al. (2015), "DE/current-to-best/1" is employed as defined follow:

$$v_{i,g} = x_{i,g} + F_i \cdot (x_{best,g} - x_{i,g}) + F_i \cdot (x_{r_1,g} - x_{r_2,g}) \quad (6)$$

where v, x, F, i, g, r_1, r_2, stand for new element, original element, mutation control parameter, individual index, dimension index, the first random index, and the second random index, accordingly. To apply the DE-based operator in GEP, SLGEP (Zhong et al., 2015) can transform the difference operation in equation 6 into a matching binary operator as:

$$\psi(a,b) = \begin{cases} 1, a \neq b \\ 0, a = b \end{cases}. \quad (7)$$

Then subsequently, the mutation operation of DE in equation 6 can be changed into a probability computation process:

$$\phi = 1 - (1 - F \cdot \psi(x_{best,j}, x_{i,j})) * (1 - F \cdot \psi(x_{r_1,j}, x_{r_2,j})) \quad (8)$$

where the probability ϕ is adopted to control mutation operation of a specific node on position j in string structure representation in **Figure 2**. That is, when a random value in [0,1] is smaller than corresponding ϕ, then the node in the reciprocal position should be replaced by a newly sampled node, where the new node is sampled by the frequency record of all the nodes in the population as Zhong et al. (2015). The evolution process in SLGEP can be conluded as algorithm 4.

Algorithm 4: *Evolution Process of SLGEP*

Input: $F, r_1, r_2, x_1, x_2, ..., x_M, CR$ (*replacement probability*), k(mandatory mutation index)
Output: *New Population*

```
For each individual x_i Do
    /* Variation */
    For each dimension x_{i,j} Do
    Compute probability φ based on equation 8
        If (rand_1(0,1) < CR OR j == k) AND rand_2(0,1) < φ Then
            u_{i,j} ← "Frequency-based Assignment" (Zhong et al., 2015)
        Else
            u_{i,j} ← x_{i,j}
    End For
    /* Selection */
    If f(u_i) < f(x_i) Then
        x_i ← u_i
End For
```

To achieve knowledge transfer based on the DE-based operator in SLGEP, similar to the strategy for canonical operator, this paper simply selects the individuals in archive to complete the computation process in the DE-based operator. The core computation part in DE-based operator is equation 8. According to the transfer paradigm Transfer (POP_i, A_k) in algorithm 3, for computation of ϕ, $x_{best,j}, x_{r_1,j}, x_{r_2,j}$ are selected from external archive A_k, and $x_{i,j}$ is selected from population POP_i. Notably, the "Frequency-based Assignment" in the original work is grounded on the frequency record of each sort of node in the whole population, which can also concerned as a form of useful knowledge especially for feature selection. Resembling the feature-wise knowledge transfer in Ardeh et al. (2019), this paper also enables the transfer of the node frequency by applying the frequency record of the archive A_k upon the individual assignment in population POP_i.

5. EXPERIMENTAL STUDY

To verify the assumption that the proposed techniques can hopefully allay the conflict of each binary classification

TABLE 1 | Data information with dimension size, sample size, and class size.

Index	Name	Features	Samples	Classes
1	DLBCL-A	661	141	3
2	DLBCL-B	661	180	3
3	Armstrong-2002	2,063	62	3
4	Lapointe-2004	1,625	69	3
5	Alizadeh-2000	2,116	72	4
6	Wine	13	178	3
7	Lung Cancer	56	32	3
8	Urban Land Cover	148	675	9
9	TOX-171	5,748	171	4
10	GLA-BRA-180	49,151	180	4

problem, the comparative studies on 10 high-dimensional multi-classification datasets for distinct GEP operators with their according transfer strategy are conducted. Aside from the direct comparative results, a relatively detailed discussion is also provided for a deeper insight on the effectiveness of knowledge transfer from various "source archives." The comparison among the proposed method, *K* Nearest Neighbor, and Decision Tree is also provided. For all the experimental studies, the results are yielded by 30 independent trials, and the Wilcoxon sign-rank test (Wilcoxon, 1992) with $\alpha = 0.1$ is performed to check for the significant difference of the experiment results.

5.1. Parameter Settings

Nearly all the fundamental settings of EMC-GEP are based on the original recommended settings of AccGEP in Zhou et al. (2003). In detail, the function set includes {+, -, *, /, **Sqrt**, **IF**}. The terminal set totally depends on the given classification problems, in addition to a list of constants, {1, 2, 3, 5, 7}. As for the algorithmic parameters, the chromosome length, the population size and the maximum iteration are 100, 1,000, and 1,000 respectively. The operator probability is set to 0.02 for mutation, and 0.8 for crossover in which 0.4 for one-point crossover, and 0.4 for two-point crossover.

Furthermore, for the DE-based GEP, SLGEP, Automatically Designed Function (ADF) in Zhong et al. (2015) has been removed to ensure the consistency as the AccGEP framework. The function set, terminal set, chromosome length, population size, maximum iteration should be set as the same settings as AccGEP, as aforesaid. In terms of the DE-based evolutionary operators introduced in section 4.2.2, the mutation factor, *F*, crossover factor, *CR*, and the mandatory index *k*, are all generated randomly according to their corresponding domain.

The original parameters of EMC-GEP only involve the archive replacement parameter, *arp*, as well as the transfer interval, δ. In this study, based on the empirical trials of the authors, *arp* and δ are set to 0.8 and 10 reciprocally for a preliminary study.

5.2. Experiment Data

The experiment datasets in the comparative study are mainly high-dimensional low sample size data as illustrated in **Table 1**, involving those datasets of which the dimension and sample size are both moderately small, thereby embodying the performance of EMC-GEP compared with the original method in diversified circumstances.

Among these datasets, Urban Land Cover (Johnson and Xie, 2013) is a categorization dataset for image information, and Wine (Aeberhard et al., 1992) is a widely used multi-classification dataset. Moreover, we also adopt some bio-information data involving DLBCL-A (Hoshida et al., 2007), DLBCL-B (Hoshida et al., 2007), and Lung Cancer (Hong and Yang, 1991). Complex gene expression data, encompassing Alizadeh-2000 (Alizadeh et al., 2000), Lapointe-2004 (Lapointe et al., 2004), Armstrong-2002 (Armstrong et al., 2001), TOX-171 (Kwon et al., 2012), and GLA-BRA-180 (Sun et al., 2006), are employed as well for a more comprehensive comparison. In this article, for each dataset, 75% of data serves as training data, while 25% of data serves as testing data.

TABLE 2 | Accuracy comparison between AccGEP and EMCGEP under distinct operators.

Data index	AccGEP-GA	EMCGEP -GA	AccGEP -DE	EMCGEP -DE1	EMCGEP -DE2
1	72.9 (4)	71.6 (5)=	75.1 (3)	**77.4 (1)+**	75.6 (2)=
2	74.4 (4)	72.8 (5)=	78.9 (2)	**81.4 (1)+**	78.9 (2)=
3	77.2 (5)	81.1 (3)+	78.9 (4)	**83.9 (1)+**	83.3 (2)+
4	58.2 (4)	46.5 (5)−	61.7 (2)	60.6 (3)=	**62.9 (1)=**
5	53.8 (4)	**60.0 (1)+**	56.3 (3)	55.0 (5)=	**60.0 (1)+**
6	94.9 (2)	86.8 (5)−	90.4 (4)	93.2 (3)+	**95.0 (1)+**
7	48.8 (2)	47.5 (4)=	48.8 (2)	40.0 (5)−	**52.5 (1)+**
8	74.4 (2)	**75.0 (1)=**	72.2 (5)	74.1 (3)+	73.8 (4)+
9	50.0 (4)	48.6 (5)=	55.6 (2)	**56.7 (1)=**	52.6 (3)−
10	58.9 (5)	59.3 (4)=	63.7 (2)	63.1 (3)=	**64.7 (1)=**
Average rank	3.6	3.8	2.9	2.6	1.8

The bold values stand for the best performance across all the methods upon a given dataset.

5.3. Comparison Results

5.3.1. Comparison With AccGEP

As depicted in **Table 2**, five methods are utilized to analyze the 10 datasets to give a brief intuition about the performance of each strategy. For AccGEP-GA, GEP with GA operator (i.e., mutation operator and crossover operator as discussed above) is implemented under AccGEP framework. Accordingly, EMCGEP-GA is based on the AccGEP-GA with additional knowledge transfer for crossover. On the other hand, AccGEP-DE is the implementation of GEP with DE operator (i.e., "Current-to-Best" and "Frequently-based Sampling") under the AccGEP framework. More precisely, EMCGEP-DE1 is based on AccGEP-DE with additional knowledge transfer for "Current-to-Best," while EMCGEP-DE2 is grounded on EMCGEP-DE1 with extensive knowledge transfer for "Frequently-based Sampling."

In **Table 2**, the fundamental data is the accuracy of the multi-classifier in percentage, and the rank number is included in the parenthese to give the relative order for performance of

these methods, to supply a brief intuition of the comparison. As for the significance test, "+," "=," "−," represent our method is significantly better than the original method, has no significant difference with the original method, and is significantly worse than the original method, with Wilcoxon sign-rank test (Wilcoxon, 1992) at $\alpha = 0.1$. To clarify, the test for EMCGEP-GA is conducted to compare with AccGEP-GA, and the tests for EMCGEP-DE1 and EMCGEP-DE2 are conducted to compare with AccGEP-DE. In this way, the effectiveness of knowledge transfer on each component can be clearly investigated.

At length, for knowledge transfer on canonical GEP operators, there is no significant difference between AccGEP-GA and EMCGEP-GA. Even for the average rank among those five algorithms, AccGEP-GA and EMCGEP-GA share the similar rank number. This result can be attributed to the ambiguous structure of GEP. Albeit in GP-based knowledge transfer study, the segments of the expression tree serve as the useful structure to different problems, the knowledge transfer of GEP string segments is in a higher level. Since the active structure is the expression tree, the transfer upon the encoding string tends to be more indirect and more ambiguous. Hence, considering two best results in **Table 2**, although the idea of "abstract knowledge transfer" is intuitively promising, the algorithmic details still require more careful designs. For instance, in each evaluation of GEP individual, a great portion of the string may be the inactive area during decoding, thus the segment-based knowledge transfer somehow may be a cost of time resources, and then it is no wonder why the transfer process cannot enhance the classification accuracy in limited evaluations.

Conversely, the knowledge transfer on DE-based operators basically can attain significantly better results compared with AccGEP-DE. Notably, the average rank of DE-based GEP is apparently better than canonical GEP. Moreover, the average ranks of EMCGEP are also better than the baseline method AccGEP-DE. To elaborate the results, the knowledge transfer upon "Current-to-Best" can possibly lead to the exploration toward the valuable operator in other binary classification of the GEP population, thereby avoiding lasting reliance on mutation operators when stuck in local minima. To be specific, instead of transferring knowledge by the segment structure in EMCGEP-GA, the basic transfer ingredient in EMCGEP-DE is gene, which can more efficiently change the solution structure. Since when the target position in GEP individual is active, then a new injected gene can hopefully change the whole structure of the original individual, which can explore the searching space effectively when the evolution process is stuck in the local minima. Grounded on EMCGEP-DE1, the knowledge transfer on feature, "Frequently-based Sampling," highly depends on the problem dimension. For those datasets with extremely high dimension like gene expression data, data 3 and data 10, transfer on feature to some extent will make no difference due to the complex distribution and the limited evaluations. But according to its average rank (1.8) compared with that of EMCGEP-DE1 (2.6), the feature transfer is still a promising avenue for knowledge transfer mechanism if adopting more detailed rules and employing more well-allocated computational resources.

TABLE 3 | Accuracy comparison with DT, KNN, and EMCGEP under distinct operators.

Data index	Decision tree	*K* Nearest neignbor	EMCGEP-DE1	EMCGEP-DE2
1	76.0 (2)	**87.2 (1)**	77.4 (3)	75.6 (4)
2	75.8 (4)	**83.2 (1)**	81.4 (2)	78.9 (3)
3	80.3 (4)	**88.8 (1)**	83.9 (2)	83.3 (3)
4	**70.8 (1)**	66.3 (2)	60.6 (4)	62.9 (3)
5	74.1 (2)	**85.1 (1)**	55.0 (4)	60.0 (3)
6	93.7 (2)	68.4 (4)	93.2 (3)	**95.0 (1)**
7	45.0 (3)	**61.5 (1)**	40.0 (4)	52.5 (2)
8	**76.9 (1)**	44.4 (4)	74.1 (2)	73.8 (3)
9	56.9 (2)	**62.9 (1)**	56.7 (3)	52.6 (4)
10	58.7 (4)	**71.0 (1)**	63.1 (3)	64.7 (2)
Average rank	2.5	1.7	3.0	2.8

The bold values stand for the best performance across all the methods upon a given dataset.

5.3.2. Comparison With Other Classifiers

In **Table 3**, EMCGEP-DE1, EMCGEP-DE2 are employed to compare with DT and KNN under the given datasets. Specifically, DT and KNN are implemented with scikit-learn (Pedregosa et al., 2011) in python with default settings.

As depicted in **Table 3**, although the Evolutionary Multitasking paradigm can enhance the performance of the existing AccGEP that searches rules based on evolutionary algorithms, the performance of the proposed EMCGEP is still limited compared with DT and KNN. Specifically, the classification results of DT are comparable with those of EMCGEP-DE1 and EMCGEP-DE2. Since DT and EMCGEP are both designed to construct rules according to given data in a non-parametric fashion, the behavior of these methods seem similar. However, KNN generally outperforms the proposed EMCGEP. One of the significant causes can be the intrinsic problem in GP methods, that the rule construction tend to be complex and unstable under high dimension scenario. Although GEP can alleviate the bloating issue of GP to some extent, the "evolutionary" behavior still makes the method unstable and even random. Nevertheless, like in data 6 and data 8, KNN sometimes is also unreliable confronting with data with certain distribution compared with EMCGEP.

5.4. Further Discussion

Grounded on the experiment results above, EMCGEP-DE1 and EMCGEP-DE2 can attain significantly better performance compared with the baseline method. Therefore, to provide a deeper insight into the working mechanism of knowledge transfer upon the "Current-to-Best" operator and the "Frequently-based Assignment" operator, this section tries to offer a more comprehensive and detailed discussion for the factors contributing to the higher-quality solutions.

In **Figures 5**, **6**, four sub-figures are given to illustrate a specific evolution process of a binary classification problem a given class in data 2 (i.e., DLBCL-B). To be specific, the first sub-figure depicts the evolution of the best individual in the binary classification population. The rest sub-figures illustrate

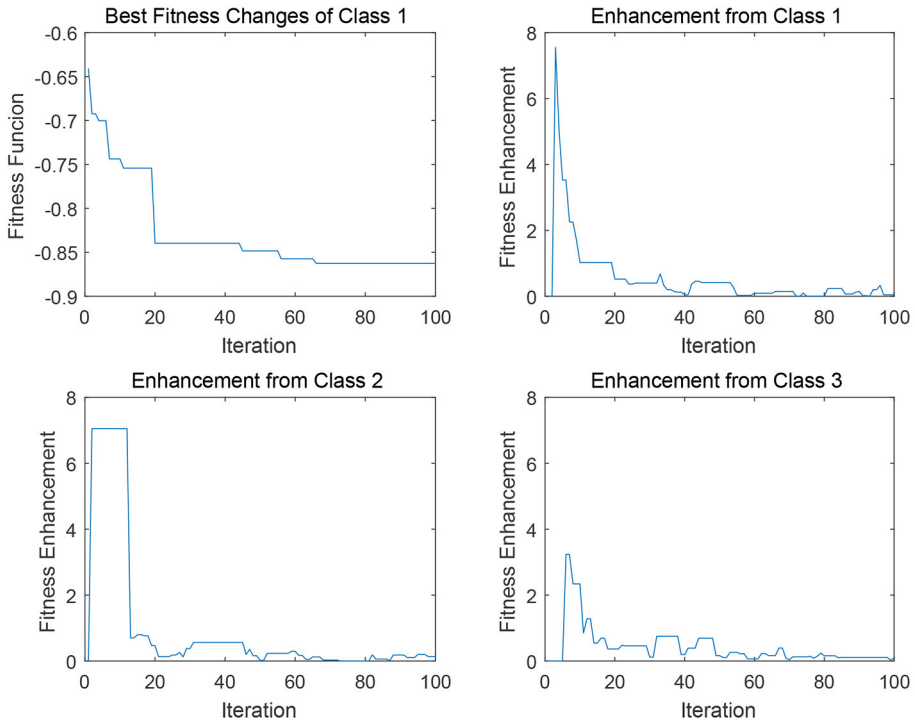

FIGURE 5 | Degrees of assistance to class 1 from various "source domains" in EMC-GEP-DE1.

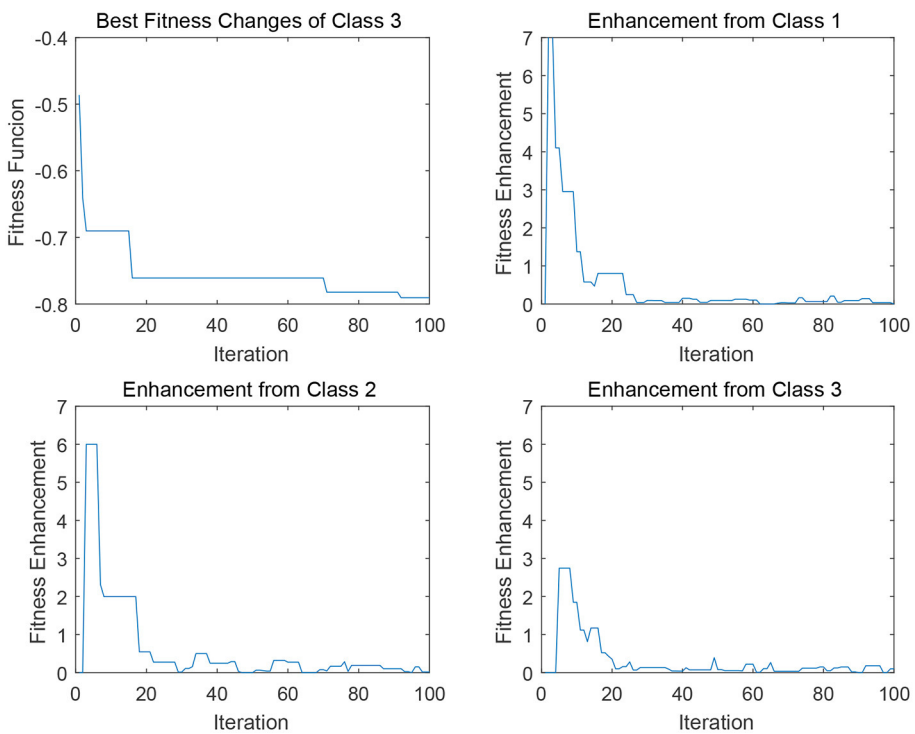

FIGURE 6 | Degrees of assistance to class 3 from various "source domains" in EMC-GEP-DE2.

the fitness enhancement from the archive A_j of each class. To clarify, the fitness enhancement here indicates the improvement of the fitness value of population POP_i after the transfer process Transfer (POP_i, A_j), which can be also concerned as the assistance from archive A_j of class j to the population POP_i. Since the transfer interval δ is configured as 10 and the maximum iteration is 1,000, the maximum visible iteration of these sub-figures is 100.

In **Figure 5**, for data 2 (i.e., DLBCL-B), the best fitness variation process of the first binary classifier of class 1 is provided, along with the fitness enhancement from each source archive A_j in each transfer iteration. In the first sub-figure, a relatively good behavior of the convergence tendency of the class-1 binary classification is indicated, since the evolution can achieve the stepwise decrease in the fitness value so as to avoid the local optima. Generally, the self transfer process (i.e., $POP_i \leftarrow A_i$), can imitate the process of the self evolution of the given population, (i.e., $POP_i \leftarrow POP_i$), and the archive from class 1 does offer relatively stable transfer performance in the first several iterations as well as the continuous enhancement in the last 30 iteration. Furthermore, as depicted in the **Figure 5**, compared to the class 1, archive of class 2 can also supply a satisfying improvement in early 20 iterations, and the archive of class 3 can offer a stable support from iteration 20 to iteration 60 to help the target population get higher-quality solution when trapped in the local optima, thereby potentially enhancing the performance of the binary classifiers in POP_1. Hence, in EMCGEP-DE1, it can be concluded that the transfer operation upon the operator "Current-to-Best" is capable of achieving performance enhancement by self transfer process, $POP_i \leftarrow A_i$, and the cross transfer process, $POP_i \leftarrow A_j$.

Similarly, as for the transfer operation upon the "Frequency-based Sampling," the strategy also can offer a satisfying convergence trend as depicted in **Figure 6**. There are several stepwise fitness improvements for the convergence curve in the first sub-figure in **Figure 6**. In the very first improvement in iteration 18, three archives can offer similar support considering the fitness enhancement from each class. Whereas, in terms of the improvement in iteration 70, the abrupt change in class 2 and class 1 played the predominant factors for helping binary classifiers of class 3 to escape from the local minima. This change elaborates our assumption that, transfer process can possibly enhance the original self evolution phase. It is notable that, in this scenario, the cross transfer of class 1 and class 2 both can offer more effective and stable fitness enhancement compared to the self transfer from class 3, which indicates the promising potential of knowledge transfer for multi-classification. However, the limit of Evolutionary Multitasking is also clear from the discussions above. Since it is uncertain which knowledge source is more beneficial for the current target population according to the unstable scale of fitness enhancement illustrated in **Figures 5**, **6**, so that it is hard to design elaborated and accurate algorithm for the given problems by Evolutionary Multitasking.

6. CONCLUSION

In this paper, knowledge transfer strategies upon canonical GEP operator and DE-based GEP operator are employed to alleviate the output conflict for multi-classification problem. In the proposed framework, a stepwise transfer is adopted to enable the segment-based transfer, DE-based transfer, as well as the feature transfer. The comparison results indicate that the DE-based transfer along with feature transfer generally can obtain significantly better performance compared to the baseline methods. Albeit the segment-based transfer for canonical GEP in this study can make no difference, some of the results and attributes of segment-based transfer still can make it special and promising, so that we concluded that this high-level transfer mechanism still require more algorithmic concern in detail. Although it is believed that knowledge transfer can enhance the existing multi-classifier, the Evolutionary Multitasking cannot tackle the intrinsic drawbacks like the randomness of the evolutionary classifiers. Furthermore, it is hard to capture the exact behavior of knowledge transfer for the evolution process, which makes it hard to design an elaborated and precise algorithm pipeline. Hence, it is deemed that both limits of Evolutionary Multitasking remain to be investigated and entails further discussion.

DATA AVAILABILITY STATEMENT

The raw data supporting the conclusions of this manuscript will be made available by the authors, without undue reservation, to any qualified researcher.

AUTHOR CONTRIBUTIONS

TW and JZ contributed to the design of the study, discussions of the results, writing, and reviewing of the manuscript. TW performed the experiments and wrote the main manuscript text.

ACKNOWLEDGMENTS

This work was supported by the National Natural Science Foundation of China (Grant No. 61602181), Program for Guangdong Introducing Innovative and Entrepreneurial Teams (Grant No. 2017ZT07X183), Guangdong Natural Science Foundation Research Team (Grant No. 2018B030312003), and the Guangdong-Hong Kong Joint Innovation Platform (Grant No. 2018B050502006).

REFERENCES

Aeberhard, S., Coomans, D., and De Vel, O. (1992). *Comparison of Classifiers in High Dimensional Settings*. Tech. Rep., Department of Mathematics and Statistics, James Cook University, North Queensland, 92.

Alizadeh, A. A., Eisen, M. B., Davis, R. E., Ma, C., Lossos, I. S., Rosenwald, A., et al. (2000). Distinct types of diffuse large b-cell lymphoma identified by gene expression profiling. *Nature* 403:503. doi: 10.1038/35000501

Aly, M. (2005). Survey on multiclass classification methods. Available online at: http://www.cs.utah.edu/~piyush/teaching/aly05multiclass.pdf

Ardeh, M. A., Mei, Y., and Zhang, M. (2019). "Genetic programming hyper-heuristic with knowledge transfer for uncertain capacitated arc routing problem," in *Proceedings of the Genetic and Evolutionary Computation Conference Companion* (Prague: ACM), 334–335.

Armstrong, S. A., Staunton, J. E., Silverman, L. B., Pieters, R., den Boer, M. L., Minden, M. D., et al. (2001). Mll translocations specify a distinct gene expression profile that distinguishes a unique leukemia. *Nat. Genet.* 30:41. doi: 10.1038/ng765

Bali, K. K., Ong, Y.-S., Gupta, A., and Tan, P. S. (2019). Multifactorial evolutionary algorithm with online transfer parameter estimation: Mfea-ii. *IEEE Trans. Evol. Comput*. doi: 10.1109/TEVC.2019.2906927

Breiman, L. (2001). Random forests. *Mach. Learn.* 45, 5–32. doi: 10.1023/A:1010933404324

Chang, C.-C., and Lin, C.-J. (2011). Libsvm: a library for support vector machines. *ACM Trans. Intell. Syst. Technol.* 2:27. doi: 10.1145/1961189.1961199

Chen, Q., Zhang, M., and Xue, B. (2017). Feature selection to improve generalization of genetic programming for high-dimensional symbolic regression. *IEEE Trans. Evol. Comput*. 21, 792–806. doi: 10.1109/TEVC.2017.2683489

Chen, Y., Zhong, J., Feng, L., and Zhang, J. (2019). "An adaptive archive-based evolutionary framework for many-task optimization," in *IEEE Transactions on Emerging Topics in Computational Intelligence*. doi: 10.1109/TETCI.2019.2916051

Chen, Y., Zhong, J., and Tan, M. (2018). "A fast memetic multi-objective differential evolution for multi-tasking optimization," in *2018 IEEE Congress on Evolutionary Computation (CEC)* (Rio de Janeiro: IEEE), 1–8.

Cheng, T., and Zhong, J. (2018). "An efficient cooperative co-evolutionary gene expression programming," in *2018 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)* (Guangzhou: IEEE), 1422–1427. doi: 10.1109/SmartWorld.2018.00246

Da, B., Gupta, A., and Ong, Y.-S. (2018). Curbing negative influences online for seamless transfer evolutionary optimization. *IEEE Trans. Cybernet*. 49, 4365–4378. doi: 10.1109/TCYB.2018.2864345

Dietterich, T. G., and Bakiri, G. (1994). Solving multiclass learning problems via error-correcting output codes. *J. Artif. Intell. Res.* 2, 263–286. doi: 10.1613/jair.105

Feng, L., Zhou, L., Zhong, J., Gupta, A., Ong, Y.-S., Tan, K.-C., et al. (2018). Evolutionary multitasking via explicit autoencoding. *IEEE Trans. Cybernet*. 49, 3457–3470. doi: 10.1109/TCYB.2018.2845361

Feng, L., Zhou, W., Zhou, L., Jiang, S., Zhong, J., Da, B., et al. (2017). "An empirical study of multifactorial pso and multifactorial de," in *2017 IEEE Congress on Evolutionary Computation (CEC)* (San Sebastian: IEEE), 921–928. doi: 10.1109/CEC.2017.7969407

Ferreira, C. (2002). "Gene expression programming in problem solving," in *Soft Computing and Industry* (London: Springer), 635–653.

Gong, M., Tang, Z., Li, H., and Zhang, J. (2019). Evolutionary multitasking with dynamic resource allocating strategy. *IEEE Trans. Evol. Comput*. 23, 858–869. doi: 10.1109/TEVC.2019.2893614

Gray, H., Maxwell, R., Martinez-Perez, I., Arus, C., and Cerdan, S. (1996). "Genetic programming for classification of brain tumours from nuclear magnetic resonance biopsy spectra," in *Proceedings of the 1st Annual Conference on Genetic Program*, 424–430.

Gupta, A., Ong, Y., Da, B., Feng, L., and Handoko, S. (2016a). "Measuring complementarity between function landscapes in evolutionary multitasking," in *2016 IEEE Congress on Evolutionary Computation* (Vancouver, CA).

Gupta, A., Ong, Y.-S., Da, B., Feng, L., and Handoko, S. D. (2016b). "Landscape synergy in evolutionary multitasking," in *2016 IEEE Congress on Evolutionary Computation (CEC)* (Vancouver, CA: IEEE), 3076–3083. doi: 10.1109/CEC.2016.7744178

Gupta, A., Ong, Y.-S., and Feng, L. (2015). Multifactorial evolution: toward evolutionary multitasking. *IEEE Trans. Evol. Comput.* 20, 343–357. doi: 10.1109/TEVC.2015.2458037

Gupta, A., Ong, Y.-S., and Feng, L. (2017). Insights on transfer optimization: because experience is the best teacher. *IEEE Trans. Emerg. Top. Comput. Intell.* 2, 51–64. doi: 10.1109/TETCI.2017.2769104

Gupta, A., Ong, Y.-S., Feng, L., and Tan, K. C. (2016c). Multiobjective multifactorial optimization in evolutionary multitasking. *IEEE Trans. Cybernet.* 47, 1652–1665. doi: 10.1109/TCYB.2016.2554622

Hancer, E., Xue, B., Zhang, M., Karaboga, D., and Akay, B. (2018). Pareto front feature selection based on artificial bee colony optimization. *Informat. Sci.* 422, 462–479. doi: 10.1016/j.ins.2017.09.028

Hong, Z.-Q., and Yang, J.-Y. (1991). Optimal discriminant plane for a small number of samples and design method of classifier on the plane. *Patt. Recogn.* 24, 317–324. doi: 10.1016/0031-3203(91)90074-F

Hoshida, Y., Brunet, J.-P., Tamayo, P., Golub, T. R., and Mesirov, J. P. (2007). Subclass mapping: identifying common subtypes in independent disease data sets. *PLoS ONE* 2:e1195. doi: 10.1371/journal.pone.0001195

Huang, Z., Zhong, J., Liu, W., and Wu, Z. (2018). "Multi-population genetic programming with adaptively weighted building blocks for symbolic regression," in *Proceedings of the Genetic and Evolutionary Computation Conference Companion* (Kyoto: ACM), 266–267. doi: 10.1145/3205651.3205673.

Johnson, B., and Xie, Z. (2013). Classifying a high resolution image of an urban area using super-object information. *ISPRS J. Photogrammet. Remote Sens.* 83, 40–49. doi: 10.1016/j.isprsjprs.2013.05.008

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems* (Lake Tahoe, NV), 1097–1105.

Kwon, E.-Y., Shin, S.-K., Cho, Y.-Y., Jung, U. J., Kim, E., Park, T., et al. (2012). Time-course microarrays reveal early activation of the immune transcriptome and adipokine dysregulation leads to fibrosis in visceral adipose depots during diet-induced obesity. *BMC Genom.* 13:450. doi: 10.1186/1471-2164-13-450

Lapointe, J., Li, C., Higgins, J. P., Van De Rijn, M., Bair, E., Montgomery, K., et al. (2004). Gene expression profiling identifies clinically relevant subtypes of prostate cancer. *Proc. Natl. Acad. Sci. U.S.A.* 101, 811–816. doi: 10.1073/pnas.0304146101

Liaw, R.-T., and Ting, C.-K. (2019). "Evolutionary manytasking optimization based on symbiosis in biocoenosis," in *Thirty-Third AAAI Conference on Artificial Intelligence* (Honolulu, HI). doi: 10.1609/aaai.v33i01.33014295

Liu, D., Huang, S., and Zhong, J. (2018). "Surrogate-assisted multi-tasking memetic algorithm," in *2018 IEEE Congress on Evolutionary Computation (CEC)* (Rio de Janeiro: IEEE), 1–8. doi: 10.1109/CEC.2018.8477830.

Min, A. T. W., Ong, Y.-S., Gupta, A., and Goh, C.-K. (2017). Multiproblem surrogates: transfer evolutionary multiobjective optimization of computationally expensive problems. *IEEE Trans. Evol. Comput.* 23, 15–28. doi: 10.1109/TEVC.2017.2783441

Muller, B., Al-Sahaf, H., Xue, B., and Zhang, M. (2019). "Transfer learning: a building block selection mechanism in genetic programming for symbolic regression," in *Proceedings of the Genetic and Evolutionary Computation Conference Companion* (Prague: ACM), 350–351. doi: 10.1145/3319619.3322072.

Muni, D. P., Pal, N. R., and Das, J. (2004). A novel approach to design classifiers using genetic programming. *IEEE Trans. Evol. Comput.* 8, 183–196. doi: 10.1109/TEVC.2004.825567

Nag, K., and Pal, N. R. (2015). A multiobjective genetic programming-based ensemble for simultaneous feature selection and classification. *IEEE Trans. Cybernet.* 46, 499–510. doi: 10.1109/TCYB.2015.2404806

O'Neill, D., Al-Sahaf, H., Xue, B., and Zhang, M. (2017). "Common subtrees in related problems: a novel transfer learning approach for genetic programming," in *2017 IEEE Congress on Evolutionary Computation (CEC)* (San Sebastian: IEEE), 1287–1294. doi: 10.1109/CEC.2017.7969453

Ong, Y.-S., and Gupta, A. (2016). Evolutionary multitasking: a computer science view of cognitive multitasking. *Cogn. Comput.* 8, 125–142. doi: 10.1007/s12559-016-9395-7

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., et al. (2011). Scikit-learn: machine learning in python. *J. Mach. Learn. Res.* 12, 2825–2830.

Poli, R., Langdon, W. B., McPhee, N. F., and Koza, J. R. (2008). *A Field Guide to Genetic Programming*. Available online at: https://www.lulu.com/

Rauss, P. J., Daida, J. M., and Chaudhary, S. (2000). "Classification of spectral imagery using genetic programming," in *Proceedings of the 2nd Annual*

Conference on Genetic and Evolutionary Computation (Las Vegas, NV: Morgan Kaufmann Publishers Inc.), 726–733.

Sakprasat, S., and Sinclair, M. C. (2007). "Classification rule mining for automatic credit approval using genetic programming," in *2007 IEEE Congress on Evolutionary Computation* (Singapore: IEEE), 548–555. doi: 10.1109/CEC.2007.4424518

Stanhope, S. A., and Daida, J. M. (1998). "Genetic programming for automatic target classification and recognition in synthetic aperture radar imagery," in *International Conference on Evolutionary Programming* (Berlin; Heidelberg: Springer), 735–744.

Storn, R., and Price, K. (1997). Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *J. Glob. Optimizat.* 11, 341–359. doi: 10.1023/A:1008202821328

Sun, L., Hui, A.-M., Su, Q., Vortmeyer, A., Kotliarov, Y., Pastorino, S., et al. (2006). Neuronal and glioma-derived stem cell factor induces angiogenesis within the brain. *Cancer Cell* 9, 287–300. doi: 10.1016/j.ccr.2006.03.003

Tran, B., Xue, B., and Zhang, M. (2018). Variable-length particle swarm optimization for feature selection on high-dimensional classification. *IEEE Trans. Evol. Comput.* 23, 473–487. doi: 10.1109/TEVC.2018.2869405

Wilcoxon, F. (1992). "Individual comparisons by ranking methods," in Kotz S. and Johnson N.L., editors. *Breakthroughs in Statistics* (New York, NY: Springer), 196–202.

Xue, B., Zhang, M., and Browne, W. N. (2012). Particle swarm optimization for feature selection in classification: a multi-objective approach. *IEEE Trans. Cybernet.* 43, 1656–1671. doi: 10.1109/TSMCB.2012.2227469

Zhong, J., Feng, L., Cai, W., and Ong, Y.-S. (2018a). Multifactorial genetic programming for symbolic regression problems. *IEEE Trans. Syst. Man Cybernet. Syst.* doi: 10.1109/TSMC.2018.2853719

Zhong, J., Feng, L., and Ong, Y.-S. (2017). Gene expression programming: a survey. *IEEE Comput. Intell. Mag.* 12, 54–72. doi: 10.1109/MCI.2017.2708618

Zhong, J., Li, L., Liu, W.-L., Feng, L., and Hu, X.-M. (2019). "A co-evolutionary cartesian genetic programming with adaptive knowledge transfer," in *2019 IEEE Congress on Evolutionary Computation (CEC)* (Wellington: IEEE), 2665–2672. doi: 10.1109/CEC.2019.8790352

Zhong, J., Lin, Y., Lu, C., and Huang, Z. (2018b). "A deep learning assisted gene expression programming framework for symbolic regression problems," in *International Conference on Neural Information Processing* (Siem Reap: Springer), 530–541. doi: 10.1007/978-3-030-04239-4_48

Zhong, J., Luo, L., Cai, W., and Lees, M. (2014). "Automatic rule identification for agent-based crowd models through gene expression programming," in *Proceedings of the 2014 International Conference on Autonomous Agents and Multi-agent Systems* (Paris: International Foundation for Autonomous Agents and Multiagent Systems), 1125–1132.

Zhong, J., Ong, Y.-S., and Cai, W. (2015). Self-learning gene expression programming. *IEEE Trans. Evol. Comput.* 20, 65–80. doi: 10.1109/TEVC.2015.2424410

Zhou, C., Xiao, W., Tirpak, T. M., and Nelson, P. C. (2003). Evolving accurate and compact classification rules with gene expression programming. *IEEE Trans. Evol. Comput.* 7, 519–531. doi: 10.1109/TEVC.2003.819261

Zhou, L., Feng, L., Zhong, J., Ong, Y.-S., Zhu, Z., and Sha, E. (2016). "Evolutionary multitasking in combinatorial search spaces: a case study in capacitated vehicle routing problem," in *2016 IEEE Symposium Series on Computational Intelligence (SSCI)* (Athens: IEEE), 1–8. doi: 10.1109/SSCI.2016.7850039

Zhou, L., Feng, L., Zhong, J., Zhu, Z., Da, B., and Wu, Z. (2018). "A study of similarity measure between tasks for multifactorial evolutionary algorithm," in *Proceedings of the Genetic and Evolutionary Computation Conference Companion* (Kyoto: ACM), 229–230. doi: 10.1145/3205651.3205736

Zuo, J., Tang, C.-J., Li, C., Yuan, C.-A., and Chen, A.-L. (2004). "Time series prediction based on gene expression programming," in *International Conference on Web-Age Information Management* (Berlin; Heidelberg: Springer), 55–64. doi: 10.1007/978-3-540-27772-9_7

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.

frontiers in Neurorobotics

ORIGINAL RESEARCH
published: 17 January 2020
doi: 10.3389/fnbot.2019.00109

A Fireworks Algorithm Based on Transfer Spark for Evolutionary Multitasking

Zhiwei Xu[1], *Kai Zhang*[1,2]*, *Xin Xu*[1,2] *and Juanjuan He*[1,2]

[1] School of Computer Science and Technology, Wuhan University of Science and Technology, Wuhan, China, [2] Hubei Province Key Laboratory of Intelligent Information Processing and Real-Time Industrial System, Wuhan, China

In recent years, lots of multifactorial optimization evolutionary algorithms have been developed to optimize multiple tasks simultaneously, which improves the overall efficiency using implicit genetic complementarity between different tasks. In this paper, a novel multitask fireworks algorithm is proposed with novel transfer sparks to solve multitask optimization problems. For each task, some transfer sparks would be generated with adaptive length and promising direction vector, which are very helpful to transfer useful genetic information between different tasks. Finally, the proposed algorithm is compared against some chosen state-of-the-art evolutionary multitasking algorithms. The experimental results show that the proposed algorithm provides better performance on several single objectives and multiobjective MTO test suites.

Keywords: evolutionary multitasking, multitask optimization, fireworks algorithm, transfer spark, evolutionary algorithm

OPEN ACCESS

Edited by:
Liang Feng,
Chongqing University, China

Reviewed by:
Zexuan Zhu,
Shenzhen University, China
Jinghui Zhong,
South China University of Technology, China

***Correspondence:**
Kai Zhang
zhangkai@wust.edu.cn

Received: *30 September 2019*
Accepted: *09 December 2019*
Published: *17 January 2020*

Citation:
Xu Z, Zhang K, Xu X and He J (2020) A Fireworks Algorithm Based on Transfer Spark for Evolutionary Multitasking. Front. Neurorobot. 13:109. doi: 10.3389/fnbot.2019.00109

INTRODUCTION

Traditional evolutionary algorithms aim to find the optimal solution for a single optimization problem by applying the reproduction and selection operators to generate better individuals iteratively (Coello et al., 2006). With the complexity of the problem increasing, simultaneously solving multiple optimization problems efficiently and quickly becomes an urgent problem (Ong and Gupta, 2016). In this context, inspired by multitasking learning in the machine learning field (Chandra et al., 2017), evolutionary multitasking (EMT) is proposed to solve the multitask optimization (MTO) problem by encoding the solutions from different tasks into a unified search space and utilizing the information of potential complementarity and similarity of different tasks to improve the convergence speed and the quality of the solutions (Gupta et al., 2016b).

The best known and the first instructive work in the EMT area is the multifactorial evolutionary algorithm (MFEA) (Gupta et al., 2016b, 2017). The MFEA algorithm is inspired by the multifactorial inheritance (Rice et al., 1978; Cloninger et al., 1979). Each task corresponds to a cultural bias block, and each cultural bias block will have an impact on the development of the offspring. When individuals with different cultural biases hybridize, they exchange information about each other's cultures and promote optimization by exploiting the potential genetic complementarity between multiple tasks (Gupta and Ong, 2016). Intuitively, an inferior solution of a task may be an exceptional solution for the other task. Similarly, the same solution in a unified space can also be excellent in multiple tasks concurrently. In both cases, the MFEA allows multiple tasks to bundle together to optimize and share genetic information to improve the overall efficiency of the search process (Gupta et al., 2018). To this end, MFEA also proposed

the mechanisms of assortative mating and vertical cultural transmission to ensure the efficiency and intensity of information exchange between tasks. These ideas have a profound impact on subsequent algorithms.

Currently, the research on EMT can approximately be summarized into three categories, the practical application of EMT (Sagarna and Ong, 2016; Yuan et al., 2016; Zhou et al., 2016; Cheng et al., 2017; Binh et al., 2018; Thanh et al., 2018; Lian et al., 2019; Wang et al., 2019) and the improved algorithm based on the MFEA framework (Bali et al., 2017; Feng et al., 2017; Wen and Ting, 2017; Joy et al., 2018; Li et al., 2018; Tuan et al., 2018; Zhong et al., 2018; Binh et al., 2019; Liang et al., 2019; Yin et al., 2019; Yu et al., 2019; Zheng et al., 2019; Zhou et al., 2019) and the perfection of EMT theory (Gupta et al., 2016a; Hashimoto et al., 2018; Liu et al., 2018; Zhou et al., 2018; Bali et al., 2019; Chen et al., 2019; Feng et al., 2019; Huang et al., 2019; Shang et al., 2019; Song et al., 2019; Tang et al., 2019). From the above studies, a consensus can be summarized that efficiently utilizing the inter-task related information is the key to improve overall search efficiency in EMT. Therefore, many studies focus on analyzing and optimizing knowledge transfer between tasks. Zhong et al. (2018) proposed a multitask genetic programming algorithm, which adopted a novel scalable chromosome representation to allow cross-domain coding of multiple solutions in a unified representation. The improved evolutionary mechanism takes both the implicit transfer of useful features between tasks and the ability of exploration into account. Liang et al. (2019) introduced genetic transform strategy and hyper-rectangle search strategy to the MFEA to improve the efficiency of knowledge transfer between tasks in the late iteration of the traditional MFEA. Huang et al. (2019) proposed an efficient surrogate-assisted multitask evolutionary framework with adaptive knowledge transfer, which is very superior for solving expensive optimization tasks. The surrogate model is constructed according to the historical search information of each task and reduces the evaluation times. A universal similarity measurement mechanism and an adaptive knowledge transfer mechanism are proposed to help knowledge transfer efficiently. Chen et al. (2019) presented the adaptive selection mechanism to evaluate the correlation between tasks and cumulative return on knowledge transferring to select the appropriate assisted task for a given task to prevent the influence of negative tasks. Feng et al. (2019) proposed an explicit genetic transferring EMT algorithm by autoencoding. This explicit genetic transfer method effectively utilizes multiple preferences embedded in different evolutionary operators to improve search performance. Bali et al. (2019) adopted the online learning mechanism into EMT and initiated a data-driven parameter tuning multitasking approach to mitigate harmful interactions between unrelated tasks to enhance overall optimization efficiency.

It is noted that most of the existing EMT algorithms are affected by the well-known MFEA algorithm. Individuals exchange genetic information through the chromosomal crossover. The hybridization of individuals with the same cultural background contributes to exploit, while individuals from different cultural backgrounds share information about their respective tasks. However, there are two drawbacks. First, the crossover sites and offset directions are randomly generated; therefore, the information transferred from the other task might not necessarily contribute to the optimization of the target task. Second, the intensity of information exchange is artificially set, and the optimization performance lacks effective feedback on it, which makes the search effect of EMT algorithm sensitive to the relationship between the tasks optimized simultaneously.

Swarm intelligence algorithms have the potential to transfer potential genetic information between tasks due to their inherent parallelism (Feng et al., 2019; Song et al., 2019). Inspired by coevolution (Cheng et al., 2017), by mapping multiple tasks into different subpopulations, the same type of subpopulations compete with each other, and subpopulations with different types cooperate, and potentially helpful knowledge blocks can be efficiently transferred between populations and utilized. The fireworks algorithm (FWA) (Tan and Zhu, 2010) is a recently proposed evolutionary algorithm based on swarm intelligence. First, a fixed number of positions in the search space are chosen as fireworks. Then, a set of sparks is generated through the explosion operation from the fireworks. Afterward, the superior solutions from the whole fireworks and sparks are selected as the fireworks for the next generation to continually improve the quality of the solution iteratively. Benefiting from the powerful global search and information utilization capabilities of FWA, it has attracted much research interest (Zheng et al., 2013; Liu et al., 2015; Li et al., 2017; Li and Tan, 2018) and has demonstrated excellent performance in many real-world problems (Yang and Tan, 2014; Bacanin and Tuba, 2015; Bouarara et al., 2015; Ding et al., 2015; Rahmani et al., 2015). In this paper, an innovative transfer vector (TV) is introduced to represent the bias of knowledge transfer between tasks. The TV is constructed by the current fitness information of other tasks and has promising direction and adaptive length. A potential superiority solution with the probability to navigate other tasks called transfer spark (TS) is generated by adding the TV as the bias to the current firework. A novel multitask optimization fireworks algorithm (MTO-FWA) utilizing the TS to exchange implicit information between tasks is proposed.

The rest of this paper is organized as follows. Section Preliminary introduces the basics of MTO and the benchmark EMT algorithm MFEA. Section Method describes the basic FWA algorithm, the proposed MTO-FWA, and the promotion of MTO-FWA on multiobjective optimization problems. Section Experiments demonstrates the experiment results on both single-objective and multiobjective MTO problems to assess the effectiveness of MTO-FWA. Finally, Section Conclusion concludes this paper and elaborates on future work.

PRELIMINARY

The section presents the key concept of MTO and the benchmark EMT algorithm MFEA.

Multitask Optimization

In general, conventional optimization problems can be divided into two categories: single-objective optimization (SOO) problems and multiobjective optimization (MOO) problems

(Liang et al., 2019). They are both committed to seeking the optimal solution of an optimization task. The difference is that SOO has only one objective function, while MOO needs to optimize multiple conflicting objective functions. The purpose of the SOO is to search out the solution with the best function value, while the goal of the MOO problem is to obtain a solution set with splendid convergence and diversity. Inspired by the cognitive ability of humans to multitasking, the knowledge acquired from solving the problem can enlighten the optimization of related problems (Gupta et al., 2016b). MTO is devoted to implementing an evolutionary search on multiple optimization tasks simultaneously to improve the convergence by seamlessly transferring knowledge between multiple optimization problems.

Unlike SOO and MOO, MTO is a new paradigm that aims to seek out the optimal solutions for multiple tasks at once. As shown in **Figure 1**, the input to the MTO consists of multiple optimization tasks, each of which can be a SOO or MOO problem. All the tasks are handled by the MTO paradigms simultaneously, so the output of the MTO contains the optimal solution for each task separately.

$$\{X_1, X_2, \cdots, X_K\} = \{\text{argmin}T_1(X_1), \text{argmin}T_2(X_2), \cdots, \text{argmin}T_K(X_K)\} \quad (1)$$

The formal representation of MTO is shown in formula (1), where X_j denotes the optimal solution of the jth task T_j ($j = 1,2,\ldots K$).

Multifactorial Evolutionary Algorithm

Inspired by the multifactorial inheritance (Rice et al., 1978; Cloninger et al., 1979), a novel EMT paradigm multifactorial optimization is proposed. Each task T_j is considered as a factor affecting individual evolution in the K-factorial environment [4, 5]. MFEA is a popular implementation that integrates genetic operators in genetic algorithm into multifactorial optimization (Gupta et al., 2016b, 2017; Bali et al., 2017; Feng et al., 2017; Wen and Ting, 2017; Binh et al., 2018, 2019; Li et al., 2018; Thanh et al., 2018; Zhong et al., 2018; Zhou et al., 2018, 2019; Liang

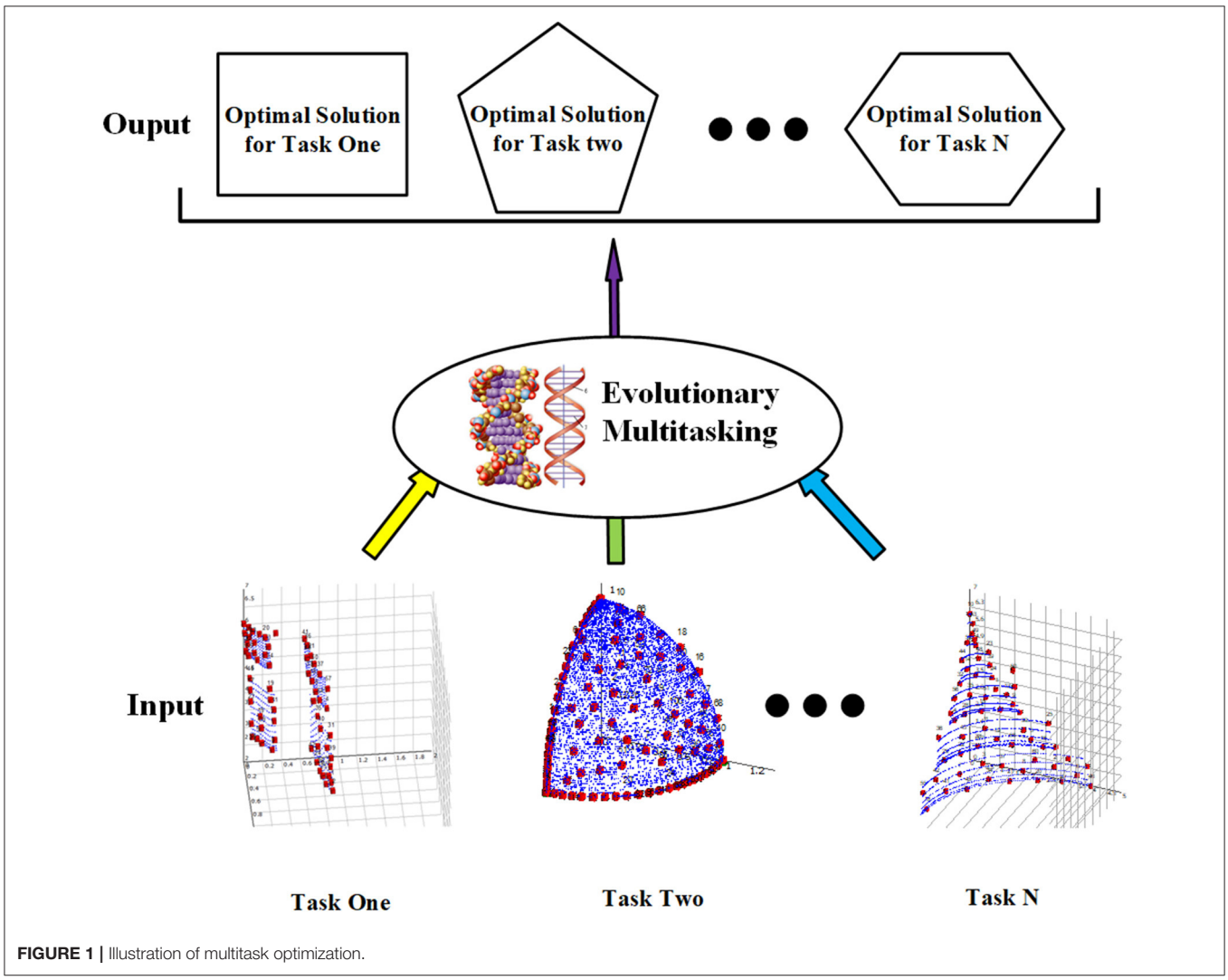

FIGURE 1 | Illustration of multitask optimization.

et al., 2019; Shang et al., 2019; Yin et al., 2019; Yu et al., 2019; Zheng et al., 2019). All the individuals are encoded into a unified search space Y, and each individual can be decoded to optimize different component problems to effectively realize cross-domain knowledge transfer. In general, Y is normalized to $[0, 1]^D$, where D is the number of dimensions of the unified search space. $D = \max\{D_{j \in \{1,2,\ldots K\}}\}$, where D_j indicates the number of dimensions of the jth task. By coding, a single chromosome $y \in Y$ can signify a combination of chromosomes corresponding to K different tasks. By decoding, the chromosomes in the unified search space can be differentiated into K chromosomes specific to the task. To evaluate the performance of a solution in the uniform search space on different tasks, MFEA proposes some definitions.

Factorial Cost: The factorial cost of individual p_i is defined as ψ_j^i which is applied to measure the performance of individual p_i on a specific task T_j. When the p_i is the feasible solution of task T_j and satisfies the constraint conditions, ψ_j^i is the fitness value of T_j. Otherwise, ψ_j^i is a very large value and indicates that the individual p_i is not a candidate solution of task T_j.

Factorial Rank: The factorial rank r_j^i indicates the rank of fitness values ψ_j^i for an individual p_i on a given task T_j by sorting the ψ_j^i in ascending order.

Scalar Fitness: To illustrate the best performance that an individual can achieve in all tasks. The scalar fitness φ_i is defined based on the best factorial rank of individual p_i among all the tasks that can be expressed as $\varphi_i = \frac{1}{min_{j \in \{1,2,\ldots k\}} r_j^i}$.

Skill Factor: The skill factor τ_i of individual p_i represents the task that p_i shows the best performance, which is defined as $\tau_i = argmin_j\{ r_j^i\}$.

Besides the traditional genetic operators, MFEA also applies the assortative mating to control the strength of genetic information transfer between tasks and vertical cultural transmission to enhance the efficiency of implicit knowledge transfer.

Assortative Mating: For two randomly selected individuals, if their skill factor is the same or satisfied the threshold called random mating probability (RMP), they can perform crossover to exchange their respective genetic information or they can only mutate. Intuitively, individuals with the uniform skill factor have a high probability of performing the crossover operator but individuals from different tasks can only exchange their genetic information in a small probability.

Vertical Cultural Transmission: Inspired by the multifactorial inheritance, MFEA believes that offspring will share the same cultural environment with their parents; that is, offspring should inherit their skill factors from their parents. If the offspring is obtained by the crossover operator, it will inherit the skill factor of either parent with equal probability. Otherwise, if the offspring is generated by the mutation operator, its skill factor will be completely inherited from the only parent. Based on the previous definitions, the pseudocode of the basic MFEA algorithm is shown in Algorithm 1.

Algorithm 1: The Pseudocode of MFEA

```
N, the size of population;
K, the number of the optimization tasks;
Randomly generate N individuals as the initial population P.
Assign initial skill factor to each individual in P randomly.
Evaluate the factorial cost of each individual
while the maximum number of evaluations is not reached:
    Generate the offspring population Q according to assortative
    mating mechanism.
    Offspring inherit the skill factor based on vertical cultural
    transmission strategy.
    Evaluate individuals in Q.
    Merge P and Q to generate new population R = P ⋃ Q.
    Update the scalar fitness φ and skill factor τ of every
    individual in R.
    Select the fittest N individuals from R as the new P.
end while
```

METHODS

This section introduces the basic FWA, the MTO-FWA based on the TS, and the extended multiobjective MTO-FWA.

The Basic FWA

Illuminated by the phenomenon that fireworks exploding to generate some explosion sparks and illuminate a surrounding area, a novel swarm intelligence algorithm FWA is proposed (Tan and Zhu, 2010). It believes that the fireworks explosion phenomenon is analogical to the process of searching the optimal solution. If there is a promising area around the current search space, fireworks will migrate to that area and generate explosion sparks to perform the local search.

The prime procedure of FWA is as follows: first, randomly initialize a set of fireworks and evaluate each firework according to the objective function Then, each firework performs a local search through an explosion operation. To save computational resources and improve search efficiency, the resource allocation strategy is used to allocate the scope and frequency of each fireworks local search. In general, individuals with better fitness function values are considered more likely to lead to global optimum, and therefore are allocated more search resources. Based on the above ideas, fireworks with better fitness values will generate a mass of sparks and possess smaller explosion amplitudes, and fireworks with worse fitness values can only generate a smaller amount of sparks and have wider explosion amplitudes relatively. After the explosion, the Gaussian mutation operation is applied to produce Gaussian mutation sparks to increase the diversity of the population. Finally, the next generation of fireworks is selected from the candidate set including fireworks, and the sparks produced by explosion and Gaussian mutation based on their performance. The processes repeat iteratively until the maximum number of evaluations is reached.

Explosion Operation

In the basic FWA algorithm (Tan and Zhu, 2010), the number of sparks and explosion amplitude of each firework x_i are shown in formula (2) and (3), respectively:

$$S_i = \hat{S} \cdot \frac{f_{max} - f(x_i) + \epsilon}{\sum_{i=1}^{N} (f_{max} - f(x_i)) + \epsilon} \tag{2}$$

$$A_i = \hat{A} \cdot \frac{f(x_i) - f_{min} + \epsilon}{\sum_{i=1}^{N} (f(x_i) - f_{min}) + \epsilon} \tag{3}$$

where $\hat{S}$ and $\hat{A}$ are two artificial parameters to control the total number of fireworks and the total amount of explosion amplitude, respectively, N represents the population size, $f_{\max}$ and $f_{\min}$ denote the maximum and minimum objective values among the total fireworks, and ϵ indicate a tiny real value to prevent zero as the denominator. To avoid this, good fireworks have too many explosion sparks, but bad fireworks have very few explosion sparks. Two other constants parameters a, $b \in [0,1]$ are introduced to bound the S_i to a proper range.

$$S_i = \begin{cases} round\left(a \cdot \hat{S}\right), & x < a \cdot \hat{S} \\ round\left(b \cdot \hat{S}\right), & x > b \cdot \hat{S} \\ round\left(\hat{S}\right), & otherwise \end{cases} \tag{4}$$

Conventional FWA does not conduct the explosion operation on each dimension of fireworks, but randomly selects $D_{explosion}$ dimensions for explosion operation. Each dimension d of explosion spark e_{is}, which can be indicated as e_{is}^d with $s \in [1, S_i]$, $d \in [1, D_{explosion}]$, conducts explosion operation according to formula (5).

$$e_{is}^d = x_i^d + A_i \cdot random(-1, 1) \tag{5}$$

The spark generated by the explosion may exceed the boundary of the search space. FWA proposed the mapping rule to map it back to the search space as expressed in formula (6).

$$e_{is}^d = x_{min}^d + e_{is}^d mod(x_{max}^d - x_{min}^d) \tag{6}$$

The outline of the explosion process is provided in Algorithm 2.

Algorithm 2: The Pseudocode of explosion

```
for s= 1 → S_i do
    Initialize the explosion spark: e_is = x_i
            D_explosion = round(D · random(0, 1))
    Stochastically choose D_explosion dimensions of e_is.
    for each dimension d of D_explosion dimensions do
        e_is^d = e_is^d + A_i · random(−1, 1)
      if e_is^d is out of the threshold value then
            e_is^d = x_min^d + e_is^d mod(x_max^d − x_min^d)
      end if
    end for
end for
```

Gaussian Mutation Operator

Some specific sparks are generated by the Gaussian explosion, which adds an offset that satisfies a Gaussian distribution to the spark to increase the diversity of population. The process of the Gaussian explosion is shown in formula (7).

$$\breve{e}_i^d = x_i^d \cdot Gaussian(1, 1) \tag{7}$$

Similar to the explosion process, the Gaussian mutation also randomly selects $D_{gaussian}$ dimensions to mutate. $\breve{e}_i^d$ indicates the d dimension of the Gaussian mutation spark with $d \in [1, D_{gaussian}]$.

Selection Mechanism

At each iteration of the algorithm, N individuals should be retained for the next generation. The individual with the best fitness is preferentially kept among all the current sparks and fireworks. Then, the remaining $N - 1$ individuals are chosen with the probability that is proportional to their distance from other individuals to maintain the diversity of sparks. Manhattan distance (Chiu et al., 2016) is usually used to measure the distance between a solution with other solutions. The choosing probability of the individual x_i represents as $Pb(x_i)$ defined in formula (8), where M denotes the solution set containing all the current individuals of both fireworks and sparks.

$$Pb(x_i) = \frac{Manhattan\ distance(x_i)}{\sum_{i \in M} Manhattan\ distance(x_i)} \tag{8}$$

The Structure of the FWA

Algorithm 3 summarizes the FWA framework. After the fireworks explode, the explosion sparks and Gaussian mutation sparks are generated based on Algorithm 2 and formula (7), respectively. The explosion sparks are generated according to the explosion operator, and the number and amplitude of the spark depend on the fitness of the firework. The Gaussian mutation sparks are generated by the Gaussian explosion process, whose number is denoted by *Gas*. Finally, N individuals remain for the next generation according to the selection mechanism.

Multitask Optimization Firework Algorithm

For MTO problems, the objective function landscape is heterogeneous, and the worst case is that they are not similar or intersecting. The key of EMT is to effectively utilize the implicit genetic information complementation from different tasks to improve the overall efficiency. Therefore, the interaction and transfer of information between different tasks are very important.

Swarm intelligence algorithms frequently possess multiple populations, which can grow the cognition of search space and further the diversity of solutions. This is very promising for exploring the heterogeneous search space of MTO problems. Different tasks can be assigned to different populations, and the cooperation between different populations provides an interpretable theoretical basis for information interaction between tasks. Different from the crossover process of randomly

Algorithm 3: The Pseudocode of FWA

N, the size of population;
Gas, the number of Gaussian mutation spark;
Randomly generate *N* initial fireworks.
while the maximum number of evaluations is not reached:
    for each firework x_i do
        Calculate the number of sparks S_i and the explosion amplitude A_i according to formula (2) and (3).
        Generate explosion sparks of the firework x_i based on Algorithm 2.
    end for
    for $gas = 1$:*Gas* do
        Obtain a Gaussian mutation spark for a randomly selected firework x_j using formula (7).
    end for
    Evaluate all the fireworks and sparks.
    Select *N* suitable solutions to constitute the fireworks of next iteration according to the selection mechanism.
end while

selected individuals in MFEA, information interaction between populations utilizes information from the whole population, which can effectively avoid random noise and negative knowledge transfer.

Unlike other swarm intelligence algorithms, FWA naturally possesses multiple populations on account of that every spark is generated near its parent firework and therefore they have similar properties. Just based on such an evolutionary strategy, each firework and its generated sparks are constituted as a task module, and each one is allocated a specific task. Disparate task modules exchange information to facilitate the exchange of implicit genetic information and individuals within a module compete with each other to promote convergence.

Compared with the conventional FWA, the main motivation of MTO-FWA can be summarized as two points.

1) Combine fireworks and their sparks into a task module to solve a specific task. Competition comes from within modules, and communication between tasks is based not on individuals but the module population. The comparison between the task module structure and the conventional FWA structure is shown in **Figure 2**.
2) A TS is proposed to solve information transfer and knowledge reuse between different tasks.

Explosion Operation

The traditional method controlling the number of sparks is sensitive to the maximum fitness value in the population, and the resource allocation gap between individuals is uncontrollable. The individuals with the highest adaptive value may get all the resources, while those with the lowest adaptive value may not get any resources. The traditional FWA solves this problem by setting thresholds, but this is crude and inelegant. Therefore, we use the power-law distribution (Li et al., 2017) to allocate spark number, through fitness rank rather than the fitness value to determine the number of spark explosion fireworks, which is shown in formula (9).

$$S_r = \hat{S} \cdot \frac{r^{-\alpha}}{\sum_{r=1}^{N} r^{-\alpha}} \tag{9}$$

N represents the total number of fireworks, r denotes the fitness rank of fireworks, and α indicates the artificial parameter controlling the distribution of resource allocation. The larger the α, more explosion sparks a good firework produces.

For the amplitude, the dynamic control algorithm (Li et al., 2017) is used, and the explosion amplitude of all fireworks is controlled dynamically, as shown in formula (10).

$$f(x) = \begin{cases} A_i^1, & g = 1 \\ C_r A_i^{g-1}, & f(x_i^g) \geqslant f(x_i^{g-1}) \\ C_a A_i^{g-1}, & f\left(x_i^g\right) < f(x_i^{g-1}) \end{cases} \tag{10}$$

where A_i^g denotes the explosion amplitude of the ith firework in generation g. In the initialization generation, the explosion amplitude is set to a large real value, usually the diameter of the search space. If the function value of the offspring firework is larger than that of the parent fireworks, the explosion amplitude will be multiplied by a shrink coefficient $C_r < 1$ to reduce the explosion amplitude so as to exploit a better solution in the local scope. Instead, the amplitude of the explosion is multiplied by an amplification coefficient $C_a > 1$ to attempt to make the largest progress. In other words, the explosion amplitude is very large at the beginning of the iteration and shrinks to a smaller value in the later stages of the iteration by the dynamic tuning strategy.

It should be emphasized that the proposed MTO-FWA has the same mapping rules as FWA. The difference is that the explosion operator works in each dimension of fireworks instead of the $D_{explosion}$ dimensions randomly selected, which has been proven to be more effective than the method of randomly selected dimensions (Li and Tan, 2018).

Guiding Spark

Different from the conventional FWA, the proposed MTO-FWA uses the guiding spark (GS) (Li et al., 2017) instead of the Gaussian mutation operator. The GS can guide the fireworks in a good direction by adding a guiding vector that indicates the dominant direction and step size to the fireworks location. The guiding vector is obtained by calculating the average of the differences between the pre-σS_i sparks and the post-σS_i sparks after all the sparks are sorted by their fitness values $f(e_{is})$ in the ascending order. By using the deviation between the top population and the bottom population, the random noise can be effectively reduced, the fireworks can be guided in the right direction, and the step length can be adjusted adaptively with the distance from the minimum value of the objective function. The generation of GS for the ith firework is shown in formula (11).

$$\begin{aligned} \Delta_i &= \frac{1}{\sigma S_i}\left(\sum_{s=1}^{\sigma S_i} e_{is} - \sum_{s=S_i-\sigma S_i+1}^{S_i} e_{is}\right) \\ GS_i &= x_i + \Delta_i \end{aligned} \tag{11}$$

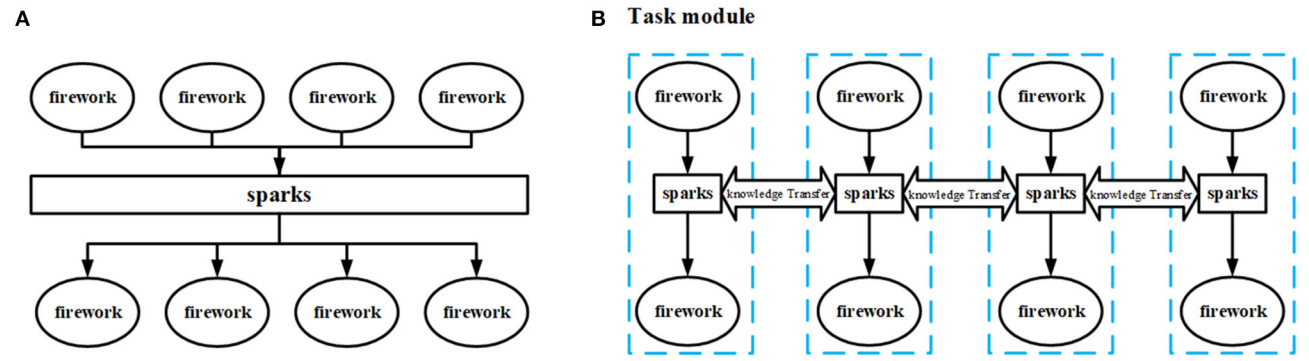

FIGURE 2 | (A) The framework of conventional FWA vs. **(B)** the framework of MTO-FWA.

where σ is the ratio parameter, e_{is} represents the sth explosion spark generated by the ith fireworks, Δ_i indicates the guiding vector of the i th fireworks, and GS_i denotes the GS of the i th fireworks. It is worth noting that only one GS is generated for each firework.

Transfer Spark

The TS is proposed to exchange information between different tasks in MTO-FWA. Each firework, explosion spark, and GS will be assigned a skill factor, and the spark inherits the skill factor from their parents. The firework and its sparks constitute a task module with the same skill factor. To avoid excessive evaluations, individuals will only evaluate the fitness values of the tasks they are assigned. In the MTO problem, according to the concept of implicit genetic information complementation, the location information of a task module can greatly help optimize another task. Based on this, assume the ith firework for the optimization task j denoted as FW_j^i, it generates a unique spark for optimizing the task k according to the information from the task k. This information from task k is denoted as TV_{jk}^i. This spark is different from other sparks generated by FW_j^i as its skill factor is k. Since it can transfer the information from other tasks, this type of spark is named TS. The TS generated by FW_j^i under the guiding of TV_{jk}^i is represented as TS_{jk}^i. TV_{jk}^i and TS_{jk}^i can be obtained by equations (12) and (13), respectively.

$$TV_{jk}^i = \frac{2}{\sigma M_j + \sigma M_k}\left(\sum_{i=1}^{\sigma M_k} x_k^i - \sum_{i=1}^{\sigma M_j} x_j^i\right)\frac{r^{-\alpha}}{\sum_{r=1}^{N_j} r^{-\alpha}} \quad (12)$$

$$TS_{jk}^i = FW_j^i + TV_{jk}^i \quad (13)$$

where M_k and M_j denote the total number of the individuals that the skill factor is k and j, respectively. In general, M_k is equal to M_j. σM_k represents the best σM_k th individuals in ascending order of fitness value of task k, and σM_j indicates the best σM_j th individuals of task j. The average value of the difference of each of the best σMth individuals is taken as a deviation. Then, each firework will be assigned deviation using the power-law distribution according to the fitness rank. The fireworks that perform better on task j are considered to have more genetic advantages and will be given more information from task k. In contrast, individuals who perform poorly on the original task can only be assigned a small amount of exchanged genetic information.

Conventional EMT algorithms randomly select individuals with different skill factors to crossover for genetic information transfer. In FWA, the locations and fitness values of the sparks generated by the explosion contain a lot of information about the objective function. Even the inferior solution that will be eliminated in the selection process still contains the genetic information that can play a great positive role in understanding the fitness landscape of the objective function and transferring the genetic information between tasks. In general, this information is ignored and not effectively utilized. Given this, we use dominant subpopulations rather than a single optimal individual for transferring genetic information in MTO. Second, by using subpopulations for information transfer, the uncorrelated values will be canceled out. Most of the dimensions of the best spark are good, but the rest are not, which means that to learn from the single best individual is to learn its good and bad at the same time. However, learning from a good population is another matter. Only the common characteristics of the population will be transferred, and other information will be regarded as random noise canceling each other, so the transferred knowledge will be more accurate.

Most EMT algorithms use crossover operators to transfer knowledge between tasks, such as SBX crossover operators. The idea is to do a local search around the parents from different tasks, and most of the offspring will fall closer to their parents, and a few will fall in between. TV, which is essentially a similar effect, can be thought of as the average of the σM vectors pointing from task j to task k, and the generated solution TS_{jk} will fluctuate between the superior subpopulations of x_j and x_k.

Selection Mechanism

All the individuals in the same task module have the same skill factor, and an individual with the best fitness in a task module is kept as candidate firework, instead of selecting from the entire individual pool. Then, all candidate fireworks and TSs are then combined and grouped according to skill factor. Afterward, the selection probability is assigned according to the fitness value of the individual, and each group will select N solutions according to this probability as the next generation of fireworks. For task j, the selection strategy is shown in Algorithm 4.

Algorithm 4: Selection mechanism

N_j, the population size of task j;
Keep the individual with the best fitness in each task module with skill factor τ_j as the candidate solution.
Merge the candidate solutions and the all the TS with skill factor τ_j as set U_j.
Assign the selection probability of the solution in U_j according to factorial rank r_j^i.
Select N_j solutions in U_j according to the selection probability as the fireworks in the next generation.

Algorithm 5: The overall framework of MTO-FWA

```
Randomly initialize fireworks.
Evaluate the objective values of different tasks for each firework.
Assign skill factor τ to each firework according to the fitness value
while not reach stop criteria
    for each firework x_i do
        Calculate the number of sparks s_i according to formula (9), the explosion
        amplitude A_i based on formula (10).
        Obtain locations of explosion sparks of the firework x_i and assign skill factor τ.
        Generate a GS according to formula (11) and assign skill factor τ.
        for each remaining K-1 tasks do
            Produce a TS according to formula (12) and (13), then assign skill factor τ̌.
        end for
    end for
    for each task do
        Select the solutions for the next generation according to Algorithm 4.
    end for
end while
```

The Structure of the MTO-FWA

Algorithm 5 summarizes the MTO-FWA framework. Assume that K tasks are optimized simultaneously; first, all the fireworks are initialized randomly and each one is evaluated by all the tasks. Then, each firework is assigned a skill factor τ according to their performance. After a firework exploding, S_i sparks with different explosion amplitude A_i are generated according to formulas (9) and (10). After that, a GS is generated by using the knowledge of exploding fireworks according to formula (11), and the skill factors of the explosion sparks and the GS are all set to τ. Afterward, $K-1$ TS are generated for other $K-1$ tasks, respectively, to share knowledge according to formulas (12) and (13). Finally, each task applies the selection strategy to pick the appropriate solutions for the next generation according to Algorithm 4.

Multitask Optimization Firework Algorithm for MOO

Multiobjective problems have two or more conflicting objectives for simultaneous optimization. Due to the lack of prior knowledge of the objective functions, we always study plentiful obtained solutions and retain the non-dominated solutions, the Pareto solution set, as the approximation of the true Pareto optimal set. Based on the fact that FWA is adept in using a single indicator to conclude the number of explosion sparks and the explosion amplitude, considering that MOO requires both convergence and diversity, the S-metric indicator (Liu et al., 2015) is introduced into FWA instead of the fitness value to select and evaluate the solutions. It should be noted that in the proposed multitask firework algorithm for MOO (MOMTO-FWA), except for the indicator modified to S-metric, the explosion operator, the GS, the TS, and the MTO-FWA are consistent. The following sections highlight the S-metric and the external archive mechanism for preserving non-dominated solutions.

S-Metric

The S-metric indicator can be regarded as the size of the space dominated by the solution or solution set (While et al., 2006). The S-metric for a solution set $M = \{m_1, m_2, \cdots m_i \cdots m_n\}$ is indicated as formula (14) (Emmerich et al., 2005).

$$S(M) := \wedge(\bigcup_{m \in M} \{x | m \prec x \prec x_{ref}\}) \tag{14}$$

where $\wedge$ denotes the Lebesgue measure, $\prec$ denotes the dominance relationship, and x_{ref} indicates the reference point dominated by all the solutions. Homoplastically, the S-metric for a single solution is represented as formula (15).

$$S(m_i) = \Delta S(M, m_i) := S(M) - S(M \backslash \{m_i\}) \tag{15}$$

The S-metric of a solution m_i can be considered as the region that is only dominated by m_i but not by other solutions in the population.

External Archive Mechanism

To ensure the quality of the solution, MOMTO-FWA uses an external archive mechanism to save the advantageous solutions for the entire iteration of each task. The number of individuals in the external archive remains at a fixed value E. For a single task k, the E_k solutions are selected from a pool of candidates M_k consisting of all fireworks, explosion sparks, GS, and TS with the same skill factor of τ_k. By selecting the optimal solution

with the largest S-metric and updating the S-metric of remaining solutions, the selected E_k solutions gain the maximum S-metric in all the E_k sets. It has been proven that the solution set that has the theoretic maximum of S-metric comes necessarily from the True Pareto Front (Fleischer, 2003). The concrete mechanism of update the external archive of the specific task is shown in Algorithm 6.

Algorithm 6: Updating strategy for the external archive

M_k, all the individuals with the same skill factor τ_t including the fireworks, explosion sparks, *TS*, and *GS*; EA_k, the external archive; $\widetilde{EA}_k$, the external archive of next generation; $\breve{ea}_k$, the selected individual save into EA_k;
Candidates pool $U_k = M_k \bigcup EA_k$.
Calculate the S-metric for every candidate in U_k.
    While $|\widetilde{EA}_k| < E_k$
    $\breve{ea}_k \leftarrow arg\ max_{u \in U_k}(S - metric(u))$.
    Save $\breve{ea}_k$ into $\widetilde{EA}_k$.
    remove $\breve{ea}_k$ from U_k.
    Update the S-metric for each candidate in U_k.
end while
Save $\widetilde{EA}_k$ as the external archive of next generation.

EXPERIMENTS

In this section, the proposed MTO-FWA is compared with other state-of-the-art EMT algorithms. The performance of MTO-FWA is comprehensively evaluated by the single-objective MTO test suite and the performance of MOMTO-FWA is assessed by the multiobjective MTO test suite.

Experiments on MTO for Single-Objective Problems

The performances of EMT algorithms are evaluated by the classical single-objective MTO test suite presented in the evolutionary MTO technical report (Da et al., 2017). The similarity of the fitness landscape and the degree of intersection of the global optima are the two key factors affecting genetic complementarity between different tasks. In other words, if the values of the corresponding dimensions of the global optima of different tasks are closer, the genetic information of the task is more likely to generate complementarity. Homoplastically, the more similar the fitness landscape of the optimization functions of the different tasks, the more helpful the knowledge an individual learns from one task to optimize other tasks indirectly. Therefore, based on the degree of intersection of the global optima, the designed benchmark problems can be divided into complete intersection (CI), partial intersection (PI), and no intersection (NI) categories. According to the similarity in the fitness landscape, the designed benchmark problems can be categorized as High Similarity (HS), Medium Similarity (MS), and Low Similarity (LS) classes. Based on the combination of the above two classification strategies, nine continuous MTO benchmark problems for SOO are proposed, each problem consisting of two classical SOO functions including the Sphere, Rosenbrock, Ackley, Rastrgin, Griewank, Weierstrass, and Schwefel functions.

As a typical swarm intelligence algorithm, the proposed MTO-FWA is compared not only with the classical basic MFEA algorithm but also with MFDE and MFPSO (Feng et al., 2017), the two swarm intelligence EMT algorithms. For a fair comparison, the population number for a single task is set to 100, and the maximum number of fitness evaluation for a single task is set to 100,000, using the average results of 30 independent runs for comparison. The MFEA uses simulated binary crossover operator (SBX) and polynomial mutation methods produce offspring to reproduce offspring, the RMP is set to 0.3, p_c and η_c in SBX are set to 1 and 2, respectively, and the parameters in polynomial mutation p_m and η_m are set to 1 and 5, respectively. In MFPSO, the w decreases linearly from 0.9 to 0.4; $c1$, $c2$, and $c3$ are all set to 0.2; and the RMP is also set to 0.3. In MFDE, the RMP is set to 0.3, and F and CR are set to 0.5 and 0.9. To ensure fairness, in MTO-FWA, the RMP is also set as 0.3; C_r, C_a, σ, and α are set to 0.9, 1.2, 0.2, and 0.

Table 1 shows the average and standard deviation of the objective function values of all algorithms that run 30 times independently on the classical single-objective MTO test suite. The superior average objective value results are highlighted in bold. The Wilcoxon rank sum test is performed at the significance level of 5%, and the proposed MTO-FWA is compared with other EMT algorithms. Significantly better and worse results than the basic MFEA are presented as "+" and "−".

As can be seen from **Table 1**, MTO-FWA shows obvious advantages in the average objective value of all the tasks in the classic MTO test problems compared with the basic MFEA. Compared with MFPSO and MFDE, MTO-FWA also shows better performance on both 15 out of 18 tasks, respectively, in the classical single-objective MTO test suite. The above statistical results verify the competitiveness and potential of the MTO-FWA algorithm in solving single-objective MTO. It is worth emphasizing that MTO-FWA reveals better performance than other EMT algorithms in most low and medium similarity test problems such as CIMS, CILS, PIMS, PILS, NIMS, and NILS. It is mainly due to the fact that the proposed TS can provide useful direction and step size and reduce the probability of negative information transfer by using information about the entire population rather than individual individuals. MFEA cannot mitigate the impact of negative knowledge transfer, which leads to the crossing process randomly happening with a lot of noise. Compared to MFPSO, MTO-FWA achieved better results on NIMS and NILS problems, because TV integrates information about the many sparks around the fireworks; therefore, it can provide better directions than the vector in PSO. Compared with MFDE, the MTO-FWA achieved better results on CIMS, CILS, PILS, and NIMS problems. It can be considered that the information used is the difference between two or more randomly selected individuals in DE, which is unpredictable. The information used in MTO-FWA comes

TABLE 1 | Averaged objective value and standard deviation obtained by MTO-FWA, MFPSO, MFDE, and MFEA on the single-objective multitask problem.

		MTO-FWA	MFPSO	MFDE	MFEA
CIHS	T1	**4.638E−7+** (3.264E−7)	2.147E−1+ (4.836E−2)	9.696E−4+ (3.625E−3)	3.684E−1 (6.462E−2)
	T2	**8.672E−5+** (6.218E−5)	7.865E0+ (3.692E1)	2.256E0+ (7.854E0)	1.875E2 (3.854E1)
CIMS	T1	**8.239E−5+** (1.173E−4)	5.871E−2+ (3.106E−2)	9.872E−4+ (2.765E−3)	4.426E0 (5.832E−1)
	T2	**9.634E−6+** (2.928E−5)	5.938E0+ (2.812E1)	3.672E−3+ (1.361E−2)	2.234E2 (5.364E1)
CILS	T1	**2.316E0+** (4.176E−2)	5.326E0+ (9.162E0)	2.203E1− (3.851E−2)	2.017E1 (6.797E2)
	T2	1.173E4− (1.161E3)	**2.172E3+** (4.163E3)	1.183E4− (1.506E3)	3.694E3 (5.361E2)
PIHS	T1	**7.124E1+** (1.763E1)	2.012E2+ (1.368E2)	7.629E1+ (1.128E1)	5.768E2 (9.744E1)
	T2	**5.647E−6+** (4.293E−6)	3.625E3− 1.367E2	2.196E−5+ (2.861E−5)	9.736E0 (1.852E0)
PIMS	T1	**7.072E−4+** (8.106E−4)	2.953E0+ (3.157E−1)	9.529E−4**+** (8.694E−4)	3.573E0 (5.821E−1)
	T2	8.168E1+ (1.632E1)	1.176E2+ (1.583E2)	**6.654E1+** (2.216E1)	6.914E2 (3.128E2)
PILS	T1	1.263E−1+ (2.684E−1)	**9.521E−3+** (5.130E−2)	3.613E−1+ (5.148E−1)	2.001E1 (9.424E−2)
	T2	**3.564E−2+** (6.845E−2)	4.672E−2+ (1.396E−1)	2.175E−1+ (4.673E−1)	1.962E1 (2.765E0)
NIHS	T1	8.521E1+ (3.262E1)	**4.216E1+** (2.723E1)	8.812E1+ (4.171E1)	9.894E2 (4.328E2)
	T2	2.716E1+ (9.864E0)	3.672E1+ (1.128E2)	**1.976E1+** (1.493E1)	2.627E2 (7.632E1)
NIMS	T1	**1.184E−3+** (2.651E−3)	4.691E−1− 2.966E−1	1.987E−3+ (4.282E−3)	4.248E−1 (6.384E−2)
	T2	**2.658E0+** (1.113E0)	1.332E1+ (1.942E0)	2.968E0+ (1.062E0)	2.772E1 (2.961E0)
NILS	T1	1.012E2+ (2.106E1)	3.167E2+ (1.176E2)	**9.478E1+** (1.971E1)	6.271E2 (1.034E2)
	T2	**2.125E3+** (2.946E2)	9.116E3− (7.126E3SS)	3.916E3− (7.136E2)	3.643E3 (3.767E2)

"+" and "−" denote the algorithm statistically significant better and worse than MFEA, respectively.

TABLE 2 | Averaged value and standard deviation of the IGD obtained by MOMTO-FWA, MOMFEA, and NSGA-II on the multiobjective multitask problem.

		MOMTO-FWA	MOMFEA	NSGA-II
CIHS	T1	**2.437E−4+** (5.507E−5)	3.422E−4 (9.643E−5)	1.733E−3− (2.345E−4)
	T2	**2.757E−4+ (8.144E−5)**	2.339E−3 (5.491E−4)	4.418E−3− (6.989E−4)
CIMS	T1	1.066E−1− (1.303E−2)	**5.932E−2** (7.136E−2)	1.306E−1− (5.421E−2)
	T2	1.263E−2− (9.682E−3)	**1.259E−2** (9.080E−3)	2.714E−2− (1.589E−2)
CILS	T1	**1.466E−4+** (1.013E−5)	2.701E−4 (2.943E−5)	2.524E−1− (6.195E−2)
	T2	**1.448E−4+** (6.575E−6)	1.867E−4 (8.093E−6)	2.022E−4− (8.687E−6)
PIHS	T1	**3.186E−4+** (9.145E−5)	8.317E−4 (1.179E−3)	1.0581E−3− (3.854E−4)
	T2	**3.424E−4+** (1.470E−4)	4.091E−2 (1.885E−2)	5.480E−2− (2.087E−2)
PIMS	T1	**7.767E−4+** (3.510E−4)	2.862E−3 (1.257E−3)	5.033E−3− (1.367E−3)
	T2	**1.094E1+** (3.423E0)	1.388E1 (4.159E0)	1.559E1− (3.700E0)
PILS	T1	4.307E−4− (6.266E−4)	3.495E−4 (3.003E−4)	**2.209E−4+** (1.357E−4)
	T2	**3.834E−4+** (1.044E−4)	1.109E−2 (2.350E−3)	6.343E−1− (5.097E−4)
NIHS	T1	**1.465E0+** (1.072E−2)	1.552E0 (1.469E−2)	9.376E1− (7.172E0)
	T2	**2.709E−4+** (6.558E−5)	4.961E−4 (1.058E−4)	8.450E−4− (1.731E−4)
NIMS	T1	**1.571E−1+** (6.445E−2)	2.133E−1 (2.352E−1)	5.846E−1− (5.182E−1)
	T2	**2.623E−3+** (1.667E−3)	3.541E−2 (6.654E−2)	6.518E−2− (5.992E−2)
NILS	T1	1.574E−3− (1.121E−3)	8.351E−4 (5.645E−5)	**8.277E−4+** (5.807E−5)
	T2	**3.827E−3+** (5.133E−4)	6.432E−1 (4.165E−4)	6.422E−1+ (3.896E−4)

"+" and "−" denote the algorithm statistically significant better and worse than MOMFEA, respectively.

from the difference between the two populations, so it is more specific.

Experiments on MTO for Multiobjective Problems

Similar to the above study for single-objective MTO, this experimental study considers the nine multiobjective multitask problems built in the recent technical report (Yuan et al., 2017). Analogously, the test problems can be classified as high similarity (HS), medium similarity (MS), and low similarity (LS), three categories according to the similarity in the fitness landscape, and each category can be divided into three sub-categories, complete intersection (CI), partial intersection (PI), and no intersection (NI) by the degree of intersection of the value of optima in each dimension. Each MTO problem consists of two MOO problems, each consisting of two or three objective functions commonly studied in the literature. Meanwhile, the proposed MOMTO-FWA is also compared with the well-known NSGA-II (Deb et al., 2002), since it is frequently applied as the underlying basic solver by many multiobjective EMT algorithms. For a

fair comparison, the population number for a single task is set to 100, and the maximum number of fitness evaluation for a single task is set to 100,000, using the average results of 30 independent runs for comparison. Both MOMFEA and NSGA-II use SBX, and polynomial variations use the same parameter values. In SBX, p_c and η_c are set to 0.9 and 20, respectively. As for polynomial mutation, p_m and η_m are set to $1/D^6$ and 20, respectively.

Table 2 shows the average and standard deviation of the IGD of all algorithms that run 30 times independently on the classical multiobjective MTO test suite. The superior average IGD values are highlighted in bold. The Wilcoxon rank sum test is performed at the significance level of 5%, and the proposed MOMTO-FWA is compared with other multiobjective EMT algorithms. Significantly better and worse results than the basic MOMFEA are presented as "+" and "−."

As can be seen from **Table 2**, MOMTO-FWA shows obvious advantages in the average IGD value on 14 out of 18 tasks in the classic multiobjective MTO test problems compared with the basic MOMFEA. Compared with NSGA-II, MOMTO-FWA also shows better performance on 16 out of 18 tasks in the multiobjective MTO test suite. It is worth emphasizing that MOMTO-FWA reveals better performance than other multiobjective EMT algorithms in most low and medium similarity test problems such as CILS, PIMS, PILS-T2, NIMS, and NILS-T2 problems. Compared to MOMFEA, MOMTO-FWA achieved better results on CIHS, CILS, PIHS, PIMS, PILS-T2, NIHS, NIMS, and NILS-T2 problems, Even if it cannot surpass the performance of MOMFEA on CIMS, PILS-T1, and NILS-T1 problems, the performance of MOMTO-FWA is not much different. This may be because MOMFEA uses non-dominant ranking, while MOMTO-FWA uses S-metric as the evaluation index. In the later stage of the algorithm, the archiving-based mechanism reduces the diversity of solutions. Encouragingly, MOMTO-FWA achieves much better results than MOMFEA and NSGA-II on PIHS-T2, PIMS-T1, PILS-T2, NIMS-T2, and NILS-T2. It can be considered that the knowledge learning from simple tasks provides inspiration for solving difficult tasks and thus improves accuracy.

Figure 3 shows the average IGD values of MOMFEA, NSGA-II, and the proposed MOMTO-FWA after 30 independent runs on the classic multiobjective multitask test set. It should be noted that to indicate the changes in IGD more clearly, the starting

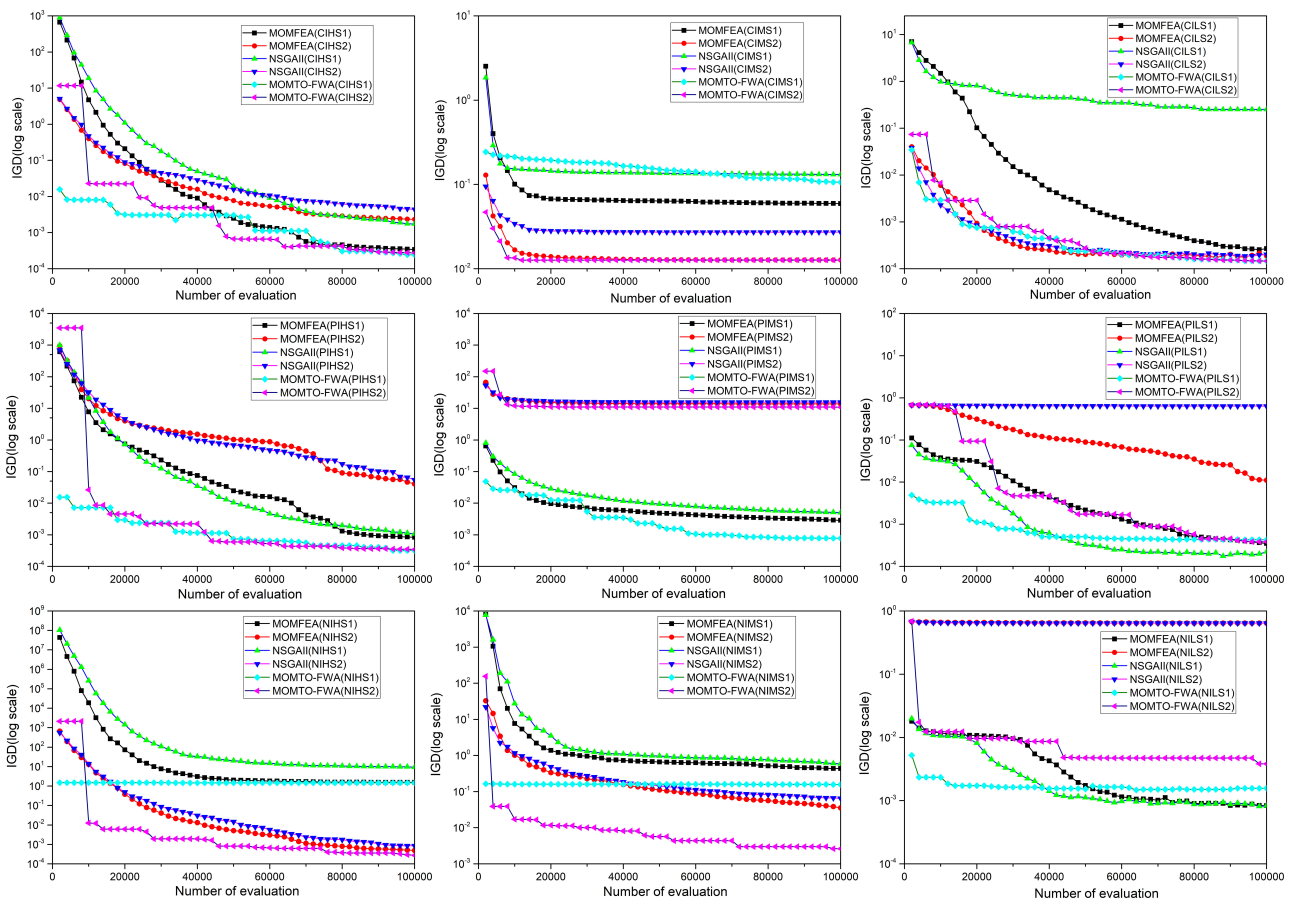

FIGURE 3 | The average IGD with the number of evaluation for MOMFEA, NSGA-II, and MOMTO-FWA on the multiobjective multitask benchmark problem.

point of the evaluation in **Figure 3** starts from the 2000th evaluation, not from the 0th evaluation. Therefore, the algorithm has a preeminent starting point on some test problems, which does not mean that the random initialization of the population has undergone artificial intervention, but the population has converged to a state with a better IGD value within 2,000 evaluations. It is obvious from **Figure 3** that the proposed MOMTO-FWA has terrific exploration ability and can quickly find out a better solution when the value of the fitness function of the initial population is terrible. In all the test problems, MOMTO-FWA is always on top in terms of IGD value within 20,000 evaluations. Besides, the proposed MOMTO-FWA converges faster than MOMFEA and NSGA-II in most problems.

CONCLUSION

In this paper, we propose the strategy named TS to enable the FWA to solve MTO problems. The core idea is to bind a firework and its generated explosion sparks and GS into a task module to solve a specific problem. Through the performance of other task modules, a TS is generated around the firework to transfer the implicit genetic information between tasks. For the single-objective MTO problem, the objective function value corresponding to the task is used as the indicator to measure the performance of the task module to control the number of explosion sparks and the explosion amplitude. For multiobjective multitask problems, S-metric is applied to evaluate individual performance. The evaluation method based on the indicator is simple and effective, which is unified for utilizing the FWA to solve the SOO and MOO in MTO. Experimental results have shown that the proposed MTO-FWA can get promising results compared with the state-of-the-art multitask evolutionary algorithms on both SOO and MOO. There are several future research directions. One direction is to improve the efficiency of information sharing and transfer between fireworks. In addition, our current research focuses on the numerical optimization of two tasks. The many task problems and the simultaneous optimization of discrete and numerical tasks are the focus of the next phase of our research.

DATA AVAILABILITY STATEMENT

The datasets generated for this study are available on request to the corresponding author.

AUTHOR CONTRIBUTIONS

ZX: code implementation and writing the experiment. KZ, JH, and XX: guidance, revision of paper, and discussion.

FUNDING

This work was supported by the National Natural Science Foundation of China (Grant Nos. U1803262, 61702383, 61602350, and 61472293) and by the Hubei Province Key Laboratory of Intelligent Information Processing and Real-time Industrial System under grant 2016znss11B.

ACKNOWLEDGMENTS

The authors would also like to thank the anonymous reviewers for their valuable remarks and comments.

REFERENCES

Bacanin, N., and Tuba, M. (2015). "Fireworks algorithm applied to constrained portfolio optimization problem," in *2015 IEEE Congress on Evolutionary Computation (CEC)* (Sendai), 1242–1249. doi: 10.1109/CEC.2015.7257031

Bali, K. K., Gupta, A., Feng, L., Ong, Y. S., and Siew, T. P. (2017). "Linearized domain adaptation in evolutionary multitasking," in *2017 IEEE Congress on Evolutionary Computation (CEC)* (San Sebastian), 1295–1302. doi: 10.1109/CEC.2017.7969454

Bali, K. K., Ong, Y., Gupta, A., and Tan, P. S. (2019). "Multifactorial evolutionary algorithm with online transfer parameter estimation: MFEA-II," in *IEEE Transactions on Evolutionary Computation*, 1. doi: 10.1109/TEVC.2019.2906927

Binh, H. T., Thanh, P. D., Trung, T. B., and Thao, L. P. (2018). "Effective multifactorial evolutionary algorithm for solving the cluster shortest path tree problem," in *2018 IEEE Congress on Evolutionary Computation (CEC)* (Rio de Janeiro), 1–8.

Binh, H. T. T., Tuan, N. Q., and Long, D. C. T. (2019). "A multi-objective multi-factorial evolutionary algorithm with reference-point-based approach," in *2019 IEEE Congress on Evolutionary Computation (CEC)* (Wellington), 2824–2831. doi: 10.1109/CEC.2019.8790034

Bouarara, H. A., Hamou, R. M., Amine, A., and Rahmani, A. (2015). A fireworks algorithm for modern web information retrieval with visual results mining. *Int. J. Swarm Intell. Res.* 6, 1–23. doi: 10.4018/IJSIR.2015070101

Chandra, R., Ong, Y.-S., and Goh, C.-K. (2017). Co-evolutionary multi-task learning with predictive recurrence for multi-step chaotic time series prediction. *Neurocomputing* 243, 21–34. doi: 10.1016/j.neucom.2017.02.065

Chen, Y., Zhong, J., Feng, L., and Zhang, J. (2019). An adaptive archive-based evolutionary framework for many-task optimization. *IEEE Trans. Emerg. Top. Comput. Intell.* 1–16. doi: 10.1109/TETCI.2019.2916051

Cheng, M.-Y., Gupta, A., Ong, Y.-S., and Ni, Z.-W. (2017). Coevolutionary multitasking for concurrent global optimization: With case studies in complex engineering design. *Eng. Appl. Artif. Intell.* 64, 13–24. doi: 10.1016/j.engappai.2017.05.008

Chiu, W., Yen, G. G., and Juan, T. (2016). Minimum manhattan distance approach to multiple criteria decision making in multiobjective optimization problems. *IEEE Trans. Evol. Comput.* 20, 972–985. doi: 10.1109/TEVC.2016.2564158

Cloninger, C. R., Rice, J., and Reich, T. (1979). Multifactorial inheritance with cultural transmission and assortative mating. II. a general model of combined polygenic and cultural inheritance. *Am. J. Hum. Genet.* 31, 176–198.

Coello, C. A. C., Lamont, G. B., and Veldhuizen, D. A. V. (2006). *Evolutionary Algorithms for Solving Multi-Objective Problems (Genetic and Evolutionary Computation)*. Springer-Verlag. Available online at: http://dl.acm.org/citation.cfm?id=1215640 (accessed September 21, 2019).

Da, B., Ong, Y.-S., Feng, L., Qin, A. K., Gupta, A., Zhu, Z., et al. (2017). Evolutionary multitasking for single-objective continuous optimization: benchmark problems, performance metric, and baseline results. *arXiv:1706.03470*.

Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* 6, 182–197. doi: 10.1109/4235.996017

Ding, K., Chen, Y., Wang, Y., and Tan, Y. (2015). "Regional seismic waveform inversion using swarm intelligence algorithms," in *2015 IEEE Congress on Evolutionary Computation (CEC)* (Sendai), 1235–1241. doi: 10.1109/CEC.2015.7257030

Emmerich, M., Beume, N., and Naujoks, B. (2005). "An EMO algorithm using the hypervolume measure as selection criterion," in *Evolutionary Multi-Criterion Optimization Lecture Notes in Computer Science,* eds C. A. C. Coello, A. Hernández Aguirre, and E. Zitzler (Berlin; Heidelberg: Springer), 62–76. doi: 10.1007/978-3-540-31880-4_5

Feng, L., Zhou, L., Zhong, J., Gupta, A., Ong, Y., Tan, K., et al. (2019). Evolutionary multitasking via explicit autoencoding. *IEEE Trans. Cybernet.* 49, 3457–3470. doi: 10.1109/TCYB.2018.2845361

Feng, L., Zhou, W., Zhou, L., Jiang, S. W., Zhong, J. H., Da, B. S., et al. (2017). "An empirical study of multifactorial PSO and multifactorial DE," in *2017 IEEE Congress on Evolutionary Computation (CEC)* (San Sebastian), 921–928. doi: 10.1109/CEC.2017.7969407

Fleischer, M. (2003). "The measure of pareto optima applications to multi-objective metaheuristics," in *Evolutionary Multi-Criterion Optimization Lecture Notes in Computer Science,* eds C. M. Fonseca, P. J. Fleming, E. Zitzler, L. Thiele, and K. Deb (Berlin; Heidelberg: Springer), 519–533. doi: 10.1007/3-540-36970-8_37

Gupta, A., and Ong, Y. (2016). "Genetic transfer or population diversification? Deciphering the secret ingredients of evolutionary multitask optimization," in *2016 IEEE Symposium Series on Computational Intelligence (SSCI)* (Athens), 1–7. doi: 10.1109/SSCI.2016.7850038

Gupta, A., Ong, Y., and Feng, L. (2016b). Multifactorial evolution: multitasking. *IEEE Trans. Evol. Comput.* 20, 343–357. doi: 10.1109/TEVC.2015.2458037

Gupta, A., Ong, Y., and Feng, L. (2018). Insights on transfer optimization: because experience is the best teacher. *IEEE Trans. Emerg. Top. Comput. Intell.* 2, 51–64. doi: 10.1109/TETCI.2017.2769104

Gupta, A., Ong, Y., Feng, L., and Tan, K. C. (2017). Multiobjective multifactorial optimization in evolutionary multitasking. *IEEE Trans. Cybernet.* 47, 1652–1665. doi: 10.1109/TCYB.2016.2554622

Gupta, A., Ong, Y. S., Da, B., Feng, L., and Handoko, S. D. (2016a). "Landscape synergy in evolutionary multitasking," in *2016 IEEE Congress on Evolutionary Computation (CEC)* (Vancouver, BC), 3076–3083. doi: 10.1109/CEC.2016.7744178

Hashimoto, R., Ishibuchi, H., Masuyama, N., and Nojima, Y. (2018). "Analysis of evolutionary multi-tasking as an island model," in *Proceedings of the Genetic and Evolutionary Computation Conference Companion GECCO'18* (New York, NY: ACM), 1894–1897. doi: 10.1145/3205651.3208228

Huang, S., Zhong, J., and Yu, W. (2019). Surrogate-assisted evolutionary framework with adaptive knowledge transfer for multi-task optimization. *IEEE Trans. Emerg. Topics Comput.* 1. doi: 10.1109/TETC.2019.2945775

Joy, C. P., Tang, J., Chen, Y., Deng, Z., and Xiang, Y. (2018). "A group-based approach to improve multifactorial evolutionary algorithm," in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI-18)*, 3870–3876. Available online at: https://www.ijcai.org/proceedings/2018/538 (accessed September 22, 2019).

Li, G., Zhang, Q., and Gao, W. (2018). "Multipopulation evolution framework for multifactorial optimization," in *Proceedings of the Genetic and Evolutionary Computation Conference Companion GECCO'18* (New York, NY: ACM), 215–216. doi: 10.1145/3205651.3205761

Li, J., and Tan, Y. (2018). Loser-out tournament-based fireworks algorithm for multimodal function optimization. *IEEE Trans. Evol. Comput.* 22, 679–691. doi: 10.1109/TEVC.2017.2787042

Li, J., Zheng, S., and Tan, Y. (2017). The effect of information utilization: introducing a novel guiding spark in the fireworks algorithm. *IEEE Trans. Evol. Comput.* 21, 153–166. doi: 10.1109/TEVC.2016.2589821

Lian, Y., Huang, Z., Zhou, Y., and Chen, Z. (2019). "Improve theoretical upper bound of jumpk function by evolutionary multitasking," in *Proceedings of the 2019 3rd High Performance Computing and Cluster Technologies Conference HPCCT 2019* (Guangzhou: ACM), 44–50. doi: 10.1145/3341069.3342982

Liang, Z., Zhang, J., Feng, L., and Zhu, Z. (2019). A hybrid of genetic transform and hyper-rectangle search strategies for evolutionary multi-tasking. *Expert Syst. Appl.* 138, 112798. doi: 10.1016/j.eswa.2019.07.015

Liu, D., Huang, S., and Zhong, J. (2018). "Surrogate-assisted multi-tasking memetic algorithm," in *2018 IEEE Congress on Evolutionary Computation (CEC)* (Janeiro), 1–8. doi: 10.1109/CEC.2018.8477830

Liu, L., Zheng, S., and Tan, Y. (2015). "S-metric based multi-objective fireworks algorithm," in *2015 IEEE Congress on Evolutionary Computation (CEC)* (Sendai), 1257–1264. doi: 10.1109/CEC.2015.7257033

Ong, Y.-S., and Gupta, A. (2016). Evolutionary multitasking: a computer science view of cognitive multitasking. *Cogn. Comput.* 8, 125–142. doi: 10.1007/s12559-016-9395-7

Rahmani, A., Amine, A., Hamou, R. M., Rahmasni, M. E., and Bouarara, H. A. (2015). Privacy preserving through fireworks algorithm based model for image perturbation in big data. *Int. J. Swarm Intell. Res.* 6, 41–58. doi: 10.4018/IJSIR.2015070103

Rice, J., Cloninger, C. R., and Reich, T. (1978). Multifactorial inheritance with cultural transmission and assortative mating. I. Description and basic properties of the unitary models. *Am. J. Hum. Genet.* 30, 618–643.

Sagarna, R., and Ong, Y. (2016). "Concurrently searching branches in software tests generation through multitask evolution," in *2016 IEEE Symposium Series on Computational Intelligence (SSCI)* (Athens), 1–8. doi: 10.1109/SSCI.2016.7850040

Shang, Q., Zhang, L., Feng, L., Hou, Y., Zhong, J., Gupta, A., et al. (2019). "A preliminary study of adaptive task selection in explicit evolutionary many-tasking," in *2019 IEEE Congress on Evolutionary Computation (CEC)* (Wellington), 2153–2159. doi: 10.1109/CEC.2019.8789909

Song, H., Qin, A. K., Tsai, P., and Liang, J. J. (2019). "Multitasking multi-swarm optimization," in *2019 IEEE Congress on Evolutionary Computation (CEC)* (Wellington), 1937–1944. doi: 10.1109/CEC.2019.8790009

Tan, Y., and Zhu, Y. (2010). "Fireworks algorithm for optimization," in *Advances in Swarm Intelligence Lecture Notes in Computer Science,* eds. Y. Tan, Y. Shi, and K. C. Tan (Berlin; Heidelberg: Springer), 355–364. doi: 10.1007/978-3-642-13495-1_44

Tang, Z., Gong, M., Jiang, F., Li, H., and Wu, Y. (2019). "Multipopulation optimization for multitask optimization," in *2019 IEEE Congress on Evolutionary Computation (CEC)* (Wellington), 1906–1913. doi: 10.1109/CEC.2019.8790234

Thanh, P. D., Dung, D. A., Tien, T. N., and Binh, H. T. T. (2018). "An effective representation scheme in multifactorial evolutionary algorithm for solving cluster shortest-path tree problem," in *2018 IEEE Congress on Evolutionary Computation (CEC)* (Janeiro), 1–8. doi: 10.1109/CEC.2018.8477684

Tuan, N. Q., Hoang, T. D., and Binh, H. T. T. (2018). "A guided differential evolutionary multi-tasking with powell search method for solving multi-objective continuous optimization," in *2018 IEEE Congress on Evolutionary Computation (CEC)* (Rio de Janeiro), 1–8. doi: 10.1109/CEC.2018.8477860

Wang, C., Ma, H., Chen, G., and Hartmann, S. (2019). "Evolutionary multitasking for semantic web service composition," in *2019 IEEE Congress on Evolutionary Computation (CEC)* (Wellington), 2490–2497. doi: 10.1109/CEC.2019.8790085

Wen, Y., and Ting, C. (2017). "Parting ways and reallocating resources in evolutionary multitasking," in *2017 IEEE Congress on Evolutionary Computation (CEC)* (San Sebastian), 2404–2411. doi: 10.1109/CEC.2017.7969596

While, L., Hingston, P., Barone, L., and Huband, S. (2006). A faster algorithm for calculating hypervolume. *IEEE Trans. Evol. Comput.* 10, 29–38. doi: 10.1109/TEVC.2005.851275

Yang, X., and Tan, Y. (2014). "Sample index based encoding for clustering using evolutionary computation," in *Advances in Swarm Intelligence Lecture Notes in Computer Science*, eds Y. Tan, Y. Shi, and C. A. C. Coello (Springer International Publishing), 489–498. doi: 10.1007/978-3-319-11857-4_55

Yin, J., Zhu, A., Zhu, Z., Yu, Y., and Ma, X. (2019). "Multifactorial evolutionary algorithm enhanced with cross-task search direction," in *2019 IEEE Congress on Evolutionary Computation (CEC)* (Wellington), 2244–2251. doi: 10.1109/CEC.2019.8789959

Yu, Y., Zhu, A., Zhu, Z., Lin, Q., Yin, J., and Ma, X. (2019). "Multifactorial differential evolution with opposition-based learning for multi-tasking

optimization," in *2019 IEEE Congress on Evolutionary Computation (CEC)* (Wellington), 1898–1905. doi: 10.1109/CEC.2019.8790024

Yuan, Y., Ong, Y., Gupta, A., Tan, P. S., and Xu, H. (2016). "Evolutionary multitasking in permutation-based combinatorial optimization problems: realization with TSP, QAP, LOP, and JSP," in *2016 IEEE Region 10 Conference (TENCON)* (Singapore), 3157–3164. doi: 10.1109/TENCON.2016.7848632

Yuan, Y., Ong, Y.-S., Feng, L., Qin, A. K., Gupta, A., Da, B., et al. (2017). Evolutionary multitasking for multiobjective continuous optimization: benchmark problems, performance metrics and baseline results. *arXiv:1706.02766*.

Zheng, X., Lei, Y., Qin, A. K., Zhou, D., Shi, J., and Gong, M. (2019). "Differential evolutionary multi-task optimization," in *2019 IEEE Congress on Evolutionary Computation (CEC)* (Wellington), 1914–1921. doi: 10.1109/CEC.2019.8789933

Zheng, Y.-J., Song, Q., and Chen, S.-Y. (2013). Multiobjective fireworks optimization for variable-rate fertilization in oil crop production. *Appl. Soft Comput.* 13, 4253–4263. doi: 10.1016/j.asoc.2013.07.004

Zhong, J., Feng, L., Cai, W., and Ong, Y.-S. (2018). Multifactorial genetic programming for symbolic regression problems. *IEEE Trans. Syst. Man Cybern. Syst.* 1–14. doi: 10.1109/TSMC.2018.2853719

Zhou, L., Feng, L., Liu, K., Chen, C., Deng, S., Xiang, T., et al. (2019). "Towards effective mutation for knowledge transfer in multifactorial differential evolution," in *2019 IEEE Congress on Evolutionary Computation (CEC)* (Wellington), 1541–1547. doi: 10.1109/CEC.2019.8790143

Zhou, L., Feng, L., Zhong, J., Ong, Y., Zhu, Z., and Sha, E. (2016). "Evolutionary multitasking in combinatorial search spaces: a case study in capacitated vehicle routing problem," *in 2016 IEEE Symposium Series on Computational Intelligence (SSCI)*(Athens: IEEE), 1–8. doi: 10.1109/SSCI.2016.7850039

Zhou, L., Feng, L., Zhong, J., Zhu, Z., Da, B., and Wu, Z. (2018). "A study of similarity measure between tasks for multifactorial evolutionary algorithm," in *Proceedings of the Genetic and Evolutionary Computation Conference Companion GECCO'18* (New York, NY: ACM), 229–230. doi: 10.1145/3205651.3205736

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.

frontiers in Neurorobotics

ORIGINAL RESEARCH
published: 21 January 2020
doi: 10.3389/fnbot.2019.00113

Droplet-Transmitted Infection Risk Ranking Based on Close Proximity Interaction

Shihui Guo[1], Jubo Yu[1], Xinyu Shi[1], Hongran Wang[1], Feibin Xie[2], Xing Gao[1] and Min Jiang[1]**

[1] School of Informatics, Xiamen University, Xiamen, China, [2] Department of Orthopaedic Trauma, Zhongshan Hospital, Xiamen University, Xiamen, China

We propose an automatic method to identify people who are potentially-infected by droplet-transmitted diseases. This high-risk group of infection was previously identified by conducting large-scale visits/interviews, or manually screening among tons of recorded surveillance videos. Both are time-intensive and most likely to delay the control of communicable diseases like influenza. In this paper, we address this challenge by solving a multi-tasking problem from the captured surveillance videos. This multi-tasking framework aims to model the principle of Close Proximity Interaction and thus infer the infection risk of individuals. The complete workflow includes three essential sub-tasks: (1) person re-identification (REID), to identify the diagnosed patient and infected individuals across different cameras, (2) depth estimation, to provide a spatial knowledge of the captured environment, (3) pose estimation, to evaluate the distance between the diagnosed and potentially-infected subjects. Our method significantly reduces the time and labor costs. We demonstrate the advantages of high accuracy and efficiency of our method. Our method is expected to be effective in accelerating the process of identifying the potentially infected group and ultimately contribute to the well-being of public health.

Keywords: influenza-like infection, person re-identification, multi-person pose estimation, infection risk ranking, multi-tasking

OPEN ACCESS

Edited by:
Huajin Tang,
Zhejiang University, China

Reviewed by:
Zexuan Zhu,
Shenzhen University, China
Jinghui Zhong,
South China University of Technology, China

****Correspondence:***
Xing Gao
gaoxing@xmu.edu.cn
Min Jiang
minjiang@xmu.edu.cn

Received: *31 August 2019*
Accepted: *13 December 2019*
Published: *21 January 2020*

Citation:
Guo S, Yu J, Shi X, Wang H, Xie F, Gao X and Jiang M (2020) Droplet-Transmitted Infection Risk Ranking Based on Close Proximity Interaction. Front. Neurorobot. 13:113. doi: 10.3389/fnbot.2019.00113

1. INTRODUCTION

The most frequent infectious diseases in humans—and those with the highest potential for rapid pandemic spread—are usually transmitted via droplets during close proximity interactions (Salathé et al., 2010). Such infectious diseases include influenza, common colds, whooping cough, SARS-CoV, and many others. Influenza alone leads to a projected annual cost of 2.0-5.8 billion USD for the American health-care system (Yan et al., 2017). It is critical to identify the group of individuals who are in close contact with the diagnosed patient, in order to understand and mitigate the spread of the aforementioned pandemic diseases.

Previous attempts model the contact networks relevant for disease transmission by using online questionnaire (Ibuka et al., 2016), surveys (Leung et al., 2017), and wearable devices (Smieszek et al., 2016; Ozella et al., 2018). Manual approaches (surveys and interviews) require a significant amount of human efforts, while wearable devices introduce additional cost and are limited to small-scale study. Open challenges remain in the development of methods to fast capture the contact networks. Given the high density of surveillance cameras in metropolitans, the impact of using captured videos to identify the contact networks is under-explored. However, two significant challenges exist for this vision-based method: (1) re-identify the diagnosed patient in non-overlapping monitor cameras and (2) assess the potential risk of infection in the exposed population. The most popular

solution to identify a specific person from videos is currently face recognition. However, poor illumination and camera viewpoint make it difficult for existing face recognition method to achieve satisfactory performance. Overlapping and occlusion of multiple faces also create significant difficulties. Meanwhile, it is non-trivial to assess the infectious risk from the captured video quantitatively. How to obtain a robust estimation of the interaction between the detected subjects in the video is still an open question.

We propose a novel framework to automatically evaluate the infection risk based on the principle of *Close Proximity Interaction*. Our success leverages the advantages of Artificial Intelligence (AI) systems over human beings in solving multiple tasks simultaneously. The accurate identification of this potentially-infected group can only be achieved with an integrative understanding of personal identity, spatial and temporal contexts from the video sequences. Such a wide range of information is processed by individual sub-tasks, including person detection, re-identification, depth and pose estimation. The user study shows that our method is effective in reducing the time and labor costs, and produces consistent results as human screening.

To this end, we made the following contributions:

- We propose a novel framework to evaluate the infection risk of identified individuals. This framework is constructed upon multi-tasking capabilities of modern techniques of computer vision. Our method effectively addresses the problem of infectious disease prevention, greatly reducing labor and time costs.
- We quantitatively model the principle of *Close Proximity Interaction* for assessing and ranking the infection risk. This is achieved by robustly reconstructing the 3D joint trajectories, based on 3D depth and pose estimation. The proposed metric takes distance as well as mutual contact between subjects into account.
- We evaluate our method in real-world environments including indoor office, and other scenarios with massive human traffic (e.g., shopping mall, hospital, public transport). The results show that our automatic method is not only time-efficient but also produces consistent prediction results as human observers.

The rest of this paper is structured as follows. Section 2 summarizes the related works, and section 3 describes the proposed framework to model the principle of *Close Proximity Interaction*, including the cornerstones to build this framework. Section 4 presents the results from our experiments, and section 5 discusses the failure cases and limitations of our method. Section 7 concludes this work and points out the directions for future efforts.

2. RELATED WORK

2.1. Infectious Disease Monitor

Monitoring the spread of infectious disease is critical for taking prompt actions to control the expansion. The contact in close distance between an infectious individual and the population leads to the spread of respiratory infections (Leung et al., 2017). This paper investigates the diseases transmitted via droplets.

The conventional methods started with social surveys, by asking participants to report their contact patterns, including the number/duration of contacts and other demographical information (including age, gender, household size) (Eames et al., 2012; Read et al., 2014; Dodd et al., 2015). Understanding the contact pattern allows us to build parameterized models and capture the transmission patterns. Leung et al. (2017) proposed a diary-based design, using both paper and online questionnaires, and found out that the approach of using paper questionnaires leads to an increasing report of contacts and longer contact duration than using online questionnaires. However, conducting such social surveys and questionnaires requires a significant amount of time and effort.

Researchers use wearable devices to analyze the contact patterns among a group of individuals. A recent work measured face-to-face proximity between family members within 16 households with infants younger than six months for 2-5 consecutive days of data collection (Ozella et al., 2018). Researchers compared the two methods of reporting with paper diaries and recording with wearable sensors, to monitor the contact pattern at a conference (Smieszek et al., 2016). They found out that reporting was notably incomplete for contacts <5 min, and participants appear to have overestimated the duration of their contacts. The typical device is RFID-based and proves to be useful in a variety of scenarios, including a pediatric hospital (Isella et al., 2011), a tertiary care hospital (Voirin et al., 2015), and a primary school (Stehlé et al., 2011). The merit of using wearable devices is a high-resolution measurement of contact matrices between individuals with the device. However, it is not feasible to apply to a wide, dynamic, and unconstrained scenario.

Different from existing methods, our work utilizes the surveillance cameras as the capture device and process the video input with the state-of-the-art techniques in computer vision. Our method quantitatively modeled the principle of close proximity interaction and introduced a graph structure to represent the contact pattern.

2.2. Person Re-identification

Person re-identification is a long-standing and significant problem that has profound application value for a wide range of fields such as security, health care, business. It aims at re-identifying the person of interest from a collection of images or videos taken by multiple non-overlapping cameras in a large distributed space over a prolonged period. Re-ID is fundamentally challenging due to three difficulties: (1) diverse visual appearance changes caused by variations in view angle, lighting, background clutter, and occlusion. (2) difficulties in producing discriminating feature representation invariant to background clutter. (3) over-fitting problem due to the limited scale of a tagged dataset.

Two types of solutions are proposed to address these problems. One is to learn a more distinctive feature representation to make a trade-off between recognition accuracy and generalization ability. The other is to leverage the Siamese neural network and triplet loss to minimize the

loss of images with the same identity and maximize that with different identities. We briefly survey the person re-identification literature from these two aspects in this paper.

2.2.1. Improvements in Feature Representation

Improvements in feature representation mainly achieved by leveraging local parts of the person. Representative methods applied part-informed features such as segmentation mask, pose, gait, etc. Pose sensitive model proposed by Saquib Sarfraz et al. (2018) incorporates both fine- and coarse-grained pose information into CNN to learn the feature representation without explicitly modeling body parts. The combined representation includes both the view captured by the camera and joint locations, which ensures the discriminating embedding. Song et al. (2018) proposed a mask-guided contrastive attention model to learn features separately from the background and human bodies. Their work takes the binary body mask as input to remove the background in pixel-level and use gait information as features. However, failure cases will happen when discriminative body parts are missing. Horizontal Pyramid Matching (HPM) approach is proposed by Fu et al. (2018), solving this problem by using partial feature representations at different horizontal pyramid scales and adopting average and max pooling for inter-person variations. For similarity measurement, metric learning approaches are exploited such as cross-view quadratic discriminant analysis (Liao et al., 2015), relative distance comparison optimization (PRDC algorithm) (Zheng et al., 2011), locally-adaptive decision functions (LADF) (Li et al., 2013) and etc.

2.2.2. Siamese Neural Network Architecture

Siamese neural network architecture is also adopted to tackle the problem of person re-identification by taking image pairs or triplets (Ding et al., 2015) as input. Siamese CNN (S-CNN) for person re-identification was presented in Yi et al. (2014) and Li et al. (2014). Improvements such as Gated Siamese CNN (Varior et al., 2016) aimed at acquiring finer local patterns for discriminative capacity enhancement. Cheng et al. (2016) proposed a Multi-Channel Parts-Based CNN with improved triplet loss consisting of multiple channels to jointly learn the global full-body and local body-parts features. Triplet loss is also widely used to learn fine-grained similarity image metrics (Wang et al., 2014). Quadruplet loss Chen et al. (2017c) strengthens the generalization capability and leads the model to output with a larger inter-class variation and a smaller intra-class variation superior to triplet loss.

2.3. Multi-Person Pose Estimation

Multi-person pose estimation aims at recognizing and locating key points on multiple persons in the image, which is the basis for resolving the technical challenges such as human action recognition (HAR) and motion analysis. Single person pose estimation is based on the assumption that the person dominates the image content. Deep learning methods perform well when the assumption is satisfied. However, for our specific problem in this paper, the case of a single person in one captured image seldom happens. Thus, we focus on the survey of multiple people pose estimation problem here. Cases such as occluded or invisible key points and background clutter lead to significant difficulties for multi-person pose estimation. State-of-the-art approaches built on CNN can be mainly divided into two categories: bottom-up approaches and top-down approaches.

2.3.1. Bottom-Up Approaches

Bottom-up approaches (Insafutdinov et al., 2016; Pishchulin et al., 2016; Cao et al., 2017) mainly adopt the strategy of detecting all key points in the image first and then matching poses to individuals. Deepcut (Pishchulin et al., 2016) casts the problem in the form of an Integer Linear Program (ILP), and the proposed partitioning and labeling formulation jointly solve the task of detection and pose estimation. A follow-up work, Deepercut (Insafutdinov et al., 2016), achieves better success by adopting image-conditioned pairwise terms with deeper ResNet (He et al., 2016). An open-source effort, Openpose (Cao et al., 2017), uses a non-parametric representation referred to as Part Affinity Fields (PAFs) for associating body parts with individuals, achieving real-time performance with high accuracy.

2.3.2. Top-Down Approaches

Top-down approaches (Fang et al., 2017; Huang et al., 2017; Papandreou et al., 2017; Chen et al., 2018) are opposed to the former, locating and partitioning all persons in the image followed by utilizing single person pose estimation caches individually for each person. Cascaded Pyramid Network (CPN) (Chen et al., 2018) takes two steps to cope with overlapping or obscured keypoints: GlobalNet for easy recognized keypoints and RefineNet for hard one. Papandreou et al. (2017) leverages the Faster RCNN (Ren et al., 2015) as the person detector and the fully convolutional ResNet to predict heatmaps and offsets. The recent work based on Mask-RCNN (He et al., 2017) extends Faster RCNN to predict human keypoints by combining the human bounding box and the corresponding feature map.

2.4. Multi-Tasking Intelligence

Multi-tasking refers to the capability of solving many tasks simultaneously. The current advances of artificial intelligence outperform human beings in effortlessly handling multiple tasks without switching costs. There are a couple of mainstream techniques for solving multi-tasking problems.

One of the popular techniques is to use the evolutionary algorithm to tackle the problem of multi-tasking. This is referred to as evolutionary multi-tasking optimization. In classic EAs, different optimization problems are typically solved independently. Researchers proposed a variety of techniques, such as multi-factorial memetic algorithm (Chen et al., 2017a), opposition-based learning (Yu et al., 2019), cross-task search direction (Yin et al., 2019), explicit autoencoding (Feng et al., 2018), and cooperative co-evolutionary memetic algorithm (Chen et al., 2017b), for the purpose of solving the multi-tasking problem. Evolutionary multi-tasking algorithms share knowledge among individual tasks and accelerate the convergence of multiple optimization tasks (Liang et al., 2019).

Relevant domains to multi-task are transfer learning and multi-objective optimization. A linearized domain adaptation

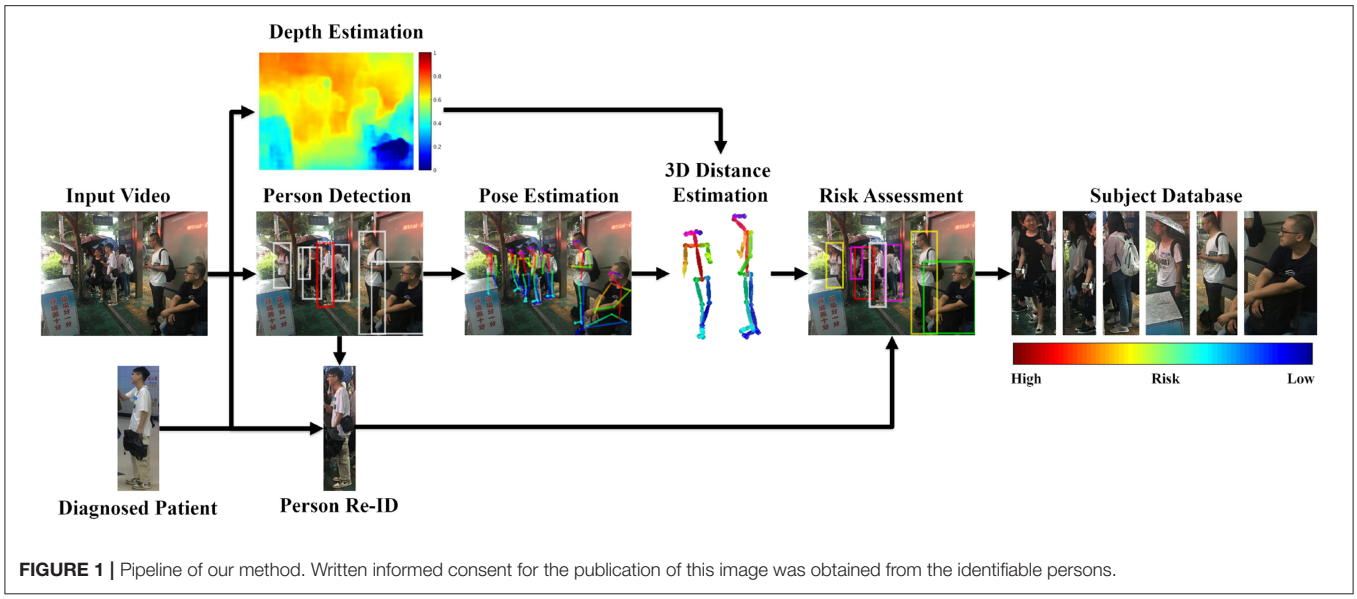

FIGURE 1 | Pipeline of our method. Written informed consent for the publication of this image was obtained from the identifiable persons.

(LDA) strategy transforms the search space of a simple task to the search space similar to its constitutive complex task (Bali et al., 2017). Researchers explored the use of transfer learning to tackle the problem of dynamic multi-objective optimization (Jiang et al., 2018). This method can significantly speed up the evolutionary process by reusing past experience and generating an effective initial population pool. The formulation of multi-objective optimization allows us to share the underlying similarity between different optimization exercises and automates the information transfer, which improves the convergence (Gupta et al., 2016).

Inspired by the methods mentioned above, our method solves a multi-tasking problem by effectively taking advantage of the information from a few building blocks. Our method directly applies to real-world scenarios to identify potentially-infected subjects. So far, we found that this problem is under-explored.

3. METHODOLOGY

The key contribution of our method is to quantitatively model the principle of Close Proximity Interaction (Salathé et al., 2010), based on the state-of-the-art techniques in computer vision. The input to our workflow is video sequences $\mathbf{VS}_i, i = 1, 2, 3, \cdots, N_c$, captured by multiple ($N_c$) cameras. These cameras are potentially non-overlapping and installed at different locations. The search starts with a diagnosed patient $\mathbf{P}^*$, who is confirmed in the clinic with the pandemic disease. The goal of this work is to identify the contact graph (**CG**) and quantitatively evaluate their potential infection risk (**PR**) with the principle of close proximity interaction. The workflow of our method is presented in **Figure 1**.

Our method successfully evaluates the infection risk and requires to solve multiple problems simultaneously. The tasks range from the fundamental problem to extract human from an image, to identify the same subject across different cameras and eventually to evaluate the infection risk for potential subjects. The knowledge learned from one task is harnessed for use in other tasks. The final goal of infection assessment can only be achieved by integrating the knowledge from prior sub-tasks. We describe our method as two main stages: (1) identifying the potentially-infected group, (2) modeling close proximity interaction.

3.1. Identifying the Contact Graph

The first step of our method is to identify the potentially-infected group of subjects. This includes a couple of sub-tasks: (1) segmenting the persons from the image, (2) re-identifying the diagnosed patient $\mathbf{P}^*$ across different cameras, (3) constructing the contact graph (**CG**) by adding the individuals who appear in the same image with the patient $\mathbf{P}^*$.

Faster R-CNN (Ren et al., 2015) is used for person segmentation as the first step of our method. Faster R-CNN extends Fast R-CNN by unifying the Region Proposal Networks (RPNs) with the original network architecture to break the bottleneck of computing time cost. RPNs are a kind of fully-convolutional network (FCN) for generating detection proposals, sharing convolutional layers with Fast R-CNN. RPNs and Fast R-CNN are trained independently. The unified architecture provides convolutional features for both object detection and region proposal tasks.

We leverage an open-source project, SVDNet (Sun et al., 2017), for person re-identification. We choose this method because of its mesrits in computational performance and comparable accuracy as the state-of-the-art. This work optimizes the deep representation learning process with Singular Vector Decomposition (SVD). It is motivated by the observation that after training a convolutional neural network (CNN) for classification, the weight vectors within a fully-connected layer (FC) are usually highly correlated.

We use a graph representation to model the contact network. Each edge **E** is a sequence involving two subjects S_A, S_B as the graph nodes. Two nodes can be connected with

multiple edges since two subjects can encounter each other at multiple locations.

3.2. Modeling Close Proximity Interaction

We model the principle of close proximity interaction by extracting contextual knowledge from the surveillance videos. The knowledge includes personal identity (acquired from the previous stage), spatial and temporal information. The latter two components describe the movement trajectories of individual persons in the 3D space. These are used o evaluate the extent of interaction proximity among subjects in the contact graph (**CG**). This is based on the assumption that the infection transmitted via droplet is critically related to the physical distance between individuals.

For each edge **E** in the contact graph **CG**, we segment the sequences from the video containing both subjects S_A, S_B on the edge **E**. For each sequence, we perform three tasks: (1) depth estimation, (2) posture estimation, and (3) risk evaluation.

For the task of depth estimation, we use the existing method (Zhou et al., 2017). This method estimates the depth information from unstructured video sequences captured by a monocular camera. The acquired depth information is used to estimate the joint trajectories in the 3D world robustly.

For the task of posture estimation, we use OpenPose (Cao et al., 2017), an open-source real-time multi-person pose estimation system. We use the provided body and hands detector to obtain the 24 key points of each individual in the image. Two-dimensional position information can be acquired by the pre-trained model.

Third, we calculate the Euclidean distances between all visible keypoints of two people separately and seek the joint on the identified patient with the smallest distance to a potential subject. The distance of joints in the 3D world can be computed with the pose positions on a 2D image and the extracted depth information. We compute the infection risk as:

$$R = \frac{1}{N_j} \sum_{i=1}^{N_j} D(J_i, J_m^*) \tag{1}$$

$$J_m^* = \arg\min_j D(J_i, J_j^*), i, j = 1, \cdots, N_j \tag{2}$$

where N_j is the number of joints. J_m^* indicates the joint on the identified patient with the minimum distance to the potentially-infected subject. $D(J_i, J_m^*)$ computes the distance between the J_i joint on the potentially-infected subject and J_m^* on the patient. The risk R indicates the average distance of all joints on the potentially-infected subject to J_m^* on the patient. We iterate this process for all identifiable subjects in the image.

4. RESULTS

4.1. Hardware and Software

Our algorithm runs on a standard PC (CPU: Intel i7 9700, GPU: RTX1080Ti, RAM:16G). The algorithm is implemented in the Python environment. The deep learning models are implemented with the open-source framework, TensorFlow.

4.2. Person Detection

The model is trained on COCO dataset for 160k iterations, starting from a learning rate of 0.02 and reducing it by 10 at 60k and 80k iterations. In RPN network, we use 5 scales with box area of the square of 32, 64, 128, 256 and 512 pixels for anchors and 3 ratios of 0.5, 1, 2. There are 256 anchors per image to use for training in total. The Faster R-CNN outputs the individual detection results. The average time cost for this task is 0.011 s.

4.3. Person Re-identification

We use the database of CASIA (Yu et al., 2009; Chen et al., 2017c) to train our network model for the task of person re-identification. The task of person re-identification achieved 88.24% top-1 accuracy, mAP = 70.68% only with softmax loss. The training strategy of combining Part-based Convolutional Baseline (PCB) and ResNet50 achieves state-of-the-art performance. We use Adam Optimizer with the learning rate of 0.1, the batch size of 32, and the stride of 2. Dropout strategy is adopted to avoid the over-fitting problem, and the drop rate is set to be 0.5. The process of the training is presented in **Figure 2**.

The number of people in the image critically affects the computation load of our method. The initial process for person segmentation leveraging the Faster R-CNN is insensitive to the number of people. The average time cost of one single image is 0.8 s. However, the amount of time spent on subsequent steps is affected by the number of people involved. The person re-identification method takes segmented individuals as input and seeks the target person among these people. The increase in the number of people leads to greater time consumption, increasing from 0.8 s of 5 persons to 1.4 s of 70 persons (shown in **Figure 3**). The time cost of multi-person pose estimation based on OpenPose is 0.8 s for $4,032 \times 3,024$ pixels' image. Thus, the total time cost of our method is no more than 3 s, far below the average time required by labor.

4.4. Experiment on Public Dataset

We here evaluate our method on a public dataset, HDA (Nambiar et al., 2014). We choose this dataset because they offer the video sequences in an uncropped way so that the depth information can be obtained. The HDA dataset is originally constructed for person re-identification, with 18 cameras recorded simultaneously during 30 min in a typical indoor office scenario at a busy hour (lunchtime) involving more than 80 persons. The cameras are located on three floors, and 13 cameras have been fully labeled. The floor plans are offered on the dataset website[1]. To accurately evaluate our method, we choose four labeled cameras (camera ID: 50/54/58/60) on Floor 7 and analyze the contact patterns between the detected persons. Camera 50 and 60 are placed toward the corridor, Camera 54 captures an indoor office room, and Camera 58 monitors a lobby at the lift. These are typical scenarios in an office environment.

Figure 4 plots the distance between subjects (marked as ID: 15, 22, 24, 32) in Camera 50. We here assume that the subject of ID:24 is the diagnosed patient and compute the relative distance

[1] Available online at: http://vislab.isr.ist.utl.pt/hda-dataset/.

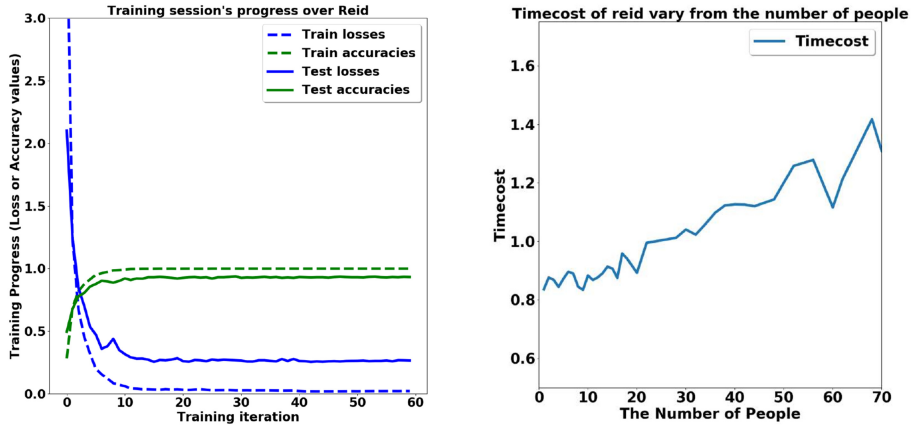

FIGURE 2 | Time performance in the task of person re-identification. **(Left)** The training stage. **(Right)** The time cost given different number of persons.

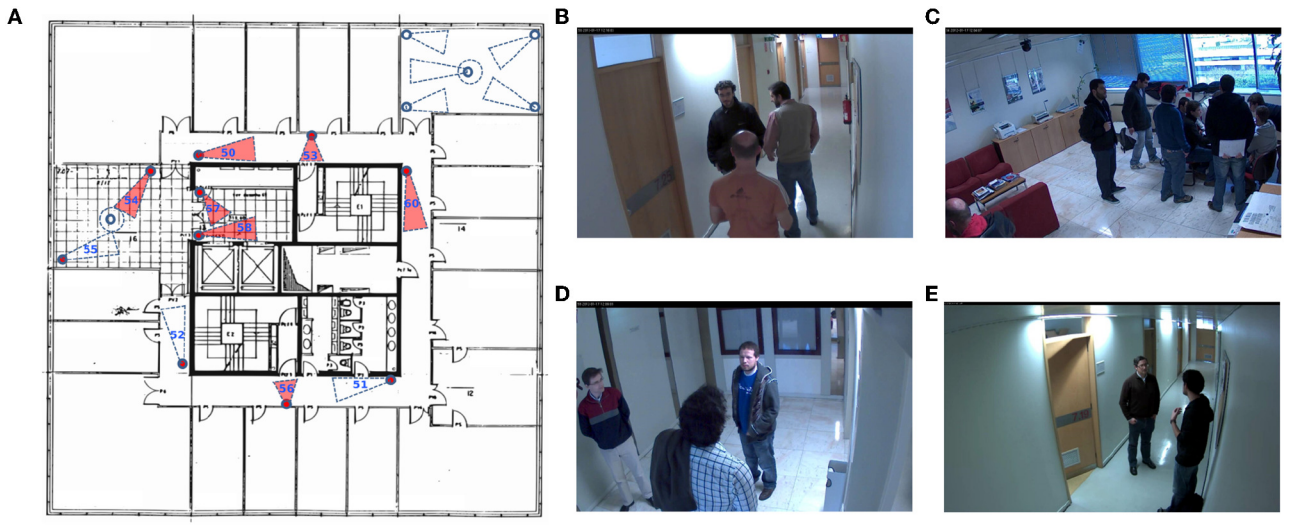

FIGURE 3 | Illustration of public HDA dataset (Nambiar et al., 2014). **(A)** Camera layout; **(B)** Camera 50; **(C)** Camera 54; **(D)** Camera 58; **(E)** Camera 60.

with other subjects who appear in this camera. Because there is no direct body contact in this scenario, we use the distance between the body centers (the hip joints) of two subjects for the visual demonstration. The results show that the predicted distance between the two subjects is consistent with the perception in the real world. It shows that our method can reliably capture the interaction within close proximity.

4.5. Multiple Scenarios

To further verify our method, we consider public places with a massive flow of people where the infectious disease spreads quickly. Three typical scenarios are considered here: a bus station, a bus compartment, and a hospital.

4.5.1. Bus Station

The scene is rainy and the background is chaotic (**Figure 5**). Many people are partially shielded by umbrellas. In the middle of the image, the crowd is so dense that only the heads can be seen. Another point worth noting is that the distance between the person and the camera varies greatly. Thus, the relative size of the skeleton varies greatly, which is prone to influence the results of risk ranking. However, through the robust method combining depth and posture estimation, risk ranking results are satisfactory.

4.5.2. Bus Compartment

Insufficient light in the bus compartment makes it harder to achieve the person retrieval (**Figure 6**). Besides, the target person is photographed from a side view rather than the same angle as his identity photo. Different perspectives are also an important factor causing difficulties in person retrieval. Results show that our method is robust to the view variations.

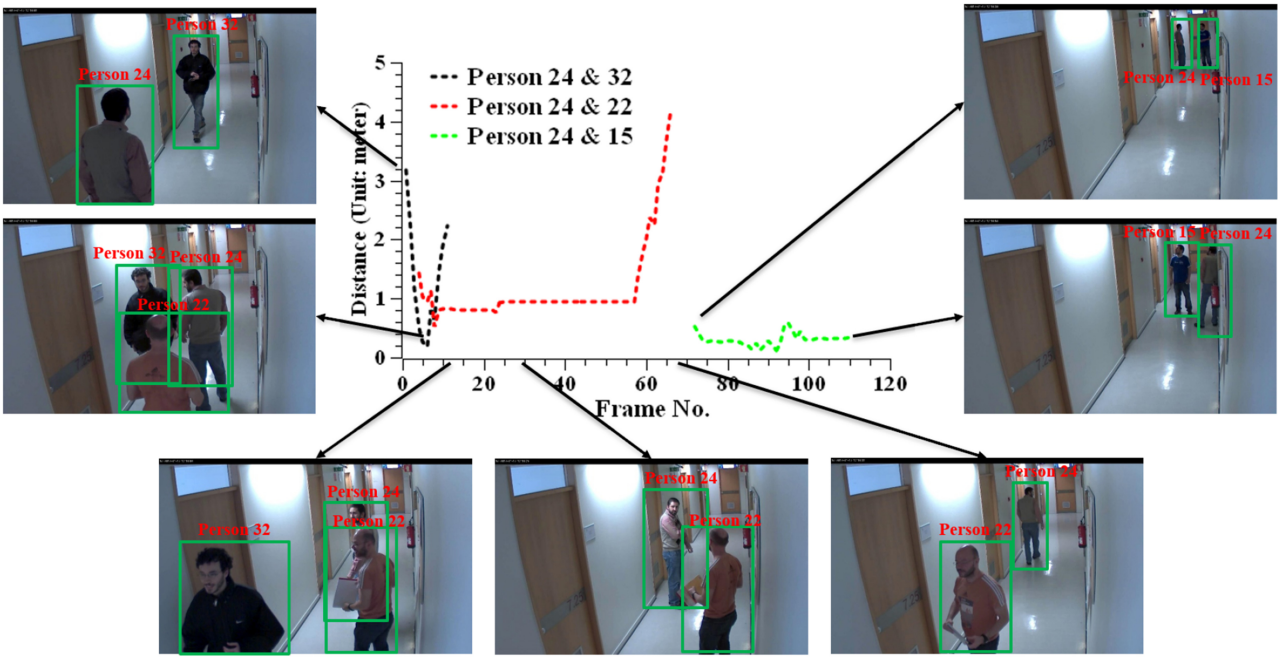

FIGURE 4 | Measuring the distance between subjects in Camera 50 from the public HDA dataset (Nambiar et al., 2014).

FIGURE 5 | Experimental results at a bus station. From left to right: the detected persons, the pose estimation and the ranking order of infection risk.Written informed consent for the publication of this image was obtained from the identifiable persons.

FIGURE 6 | Experimental results in a bus compartment. From left to right: the detected persons, the pose estimation and the ranking order of infection risk. Written informed consent for the publication of this image was obtained from the identifiable persons.

FIGURE 7 | Experimental results at a hospital.From left to right: the detected persons, the pose estimation and the ranking order of infection risk. Written informed consent for the publication of this image was obtained from the identifiable persons.

4.5.3. Hospital

The mutual occlusion between individuals is significant in this case (**Figure 7**). Considering the pose information we use is two dimensional, it is difficult to determine the exact distance between people. With the depth information, infection risks we obtained are consistent with our visual, intuitive judgment.

4.6. Comparison With User Study

To evaluate the reasonableness of our method in assessing the risk of infection, we used the risk assessment obtained by human subjects as a comparison baseline.

4.6.1. Participants

Ten volunteers (5 males and 5 females) with an average age of 21 and SD of 3.5 were recruited in this study. They are all undergraduate and graduate students in the department of information science. Written agreement to participate in this study was obtained from individual participant after explanations of this study. They all agreed to join this study for free.

4.6.2. Procedures

Participants were invited to the lab and conducted this experiment. After explaining the task details, they signed the agreement of participation. They were instructed to rank the infection risk of all detected persons in each video, given the diagnosed subject. They were not aware of the purpose of this study, as the comparison baseline of our proposed algorithm.

We used all three scenarios (bus station, bus compartment, and hospital) in the previous section. Participants were presented with a short sequence of videos, They were instructed to sort the infection risk of all detected individuals in the image based on common sense or intuition. Starting from the candidate with the highest perceived risk, they associated with the candidate with the rank number from 1 to N (N is the number of candidates in each image). No judging criteria were given. We started the timer when the participant sees the image and started marking it without explicitly informing the user of timekeeping. Interviews were conducted after participants finished the previous procedure by asking open questions and collecting their subjective feedback on how they perceived and ranked the risk. Each participant spent around 20 min to complete the study.

4.6.3. Quantitative Findings

We compared the ranking result from our method and the user experiments (**Figure 8**). The bar plots show the distribution of the ranking order, while the number on top of each box is the proposed order by our method. The results show that the ranking order of infection risk is consistent between our method and human subjects. Participants achieved a higher degree of consensus with the highest and lowest ranking candidates. For the examples of both the bus station and the hospital, all participants identified the person (ID = 3 and ID = 4 in these two respective examples) closest to the diagnosed patient as the top candidate of infection risk. For the example of the bus compartment, the choice of the top candidate is distributed to two options (ID = 3 and ID = 4). However, for other options between the highest and lowest ranking candidate, participants showed a higher degree of variation.

In terms of time cost, our method requires far less time (3 s) to process one image, than the time cost required by our participants (2 min). During the decision flow, when the participants ranked the risk order, we observed that it requires significantly less time to identify the person of the highest risk than the rest choices. This is consistent with the high degree of consensus in the candidate selection. We believe this shows the advantages of our method in accuracy and efficiency. The reasoning behind the decision process of human participants is to be explained in the following paragraph.

4.6.4. Qualitative Findings

We interviewed the participants and collected their feedback and comments. We asked about how they decided the ranking order, and all participants mentioned the factor of the distance between the candidate and the diagnosed patient. This confirms the principle of close proximity interaction. Six participants explicitly pointed out that the fact that the top candidate is conversing with the diagnosed patient in the examples of the bus station and hospital critically shapes their decision. This is also consistent with the transmission route of the droplet. When people are having a face-to-face conversation, the droplets are more likely

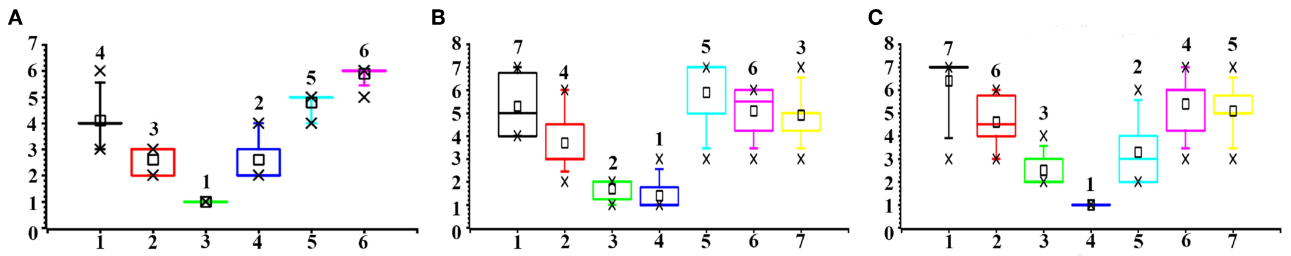

FIGURE 8 | Comparison of manual ranking and our method. Horizontal axis: the candidate ID in the image (from left to right). Vertical axis: the ranking order. A smaller value of the ranking order indicates a higher risk of infection. **(A)** Bus station; **(B)** Bus compartment; **(C)** Hospital.

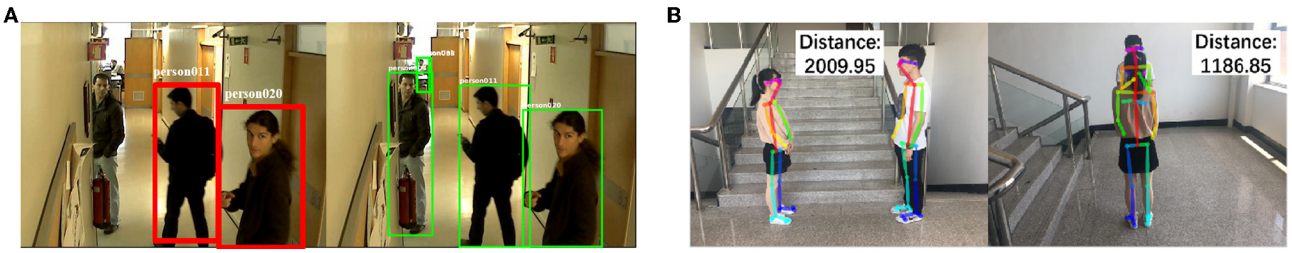

FIGURE 9 | Failure cases of our method. Images in **(A)** are from the public dataset HDA (Nambiar et al., 2014). Written informed consent for the publication of image **(B)** was obtained from the identifiable persons.

to spread out to the person in the conversational group. The carried virus in the droplet causes the infection. For the ranking decision with lower possibility, participants agreed that it is more difficult to decide since more than one candidate is located at a similar distance with the diagnosed patient. However, they also mentioned that since the rest of the candidates are not exposed to the high infection risk, their significance to infection control deserves less attention.

5. DISCUSSION

In this section, we discuss the insights we learned from our experience, in particular typical failure cases and limitations in our experiments.

5.1. Failure Case Analysis

The building blocks critically determine the success of inferring the close proximity interaction in the upstream workflow. Here we identify the failure cases caused by two components: person re-identification and distance estimation.

The state-of-the-art methods in person re-identification still face significant challenges in a complicated environment. The current accuracy of re-identification in our method is 88%. In selected scenarios, the method in our work fails to identify the same person in two different camera views. This is caused by the relative perspective between the person in the view and the camera perspective. Improving the method of re-identification is the solution to this problem. **Figure 9A** presents one typical failure case. The person on the left of the image is about to exit from the corridor and partially occluded. This creates a detection failure.

Reliable reconstruction of 3D information from the 2D image is still an open question in the domain of computer vision. Although we propose an efficient method to infer the depth information and integrate with the 2D posture, failure cases still arise due to occlusion and viewpoint perspective. For the former case, if the two persons are standing in line with the camera (shown in the right image of **Figure 9B**), the detected key points will be almost mixed together. At this time, it is significantly challenging to predict the distance between the subjects.

6. LIMITATIONS

First, only direct infection is considered, while the indirect infection is neglected. Some bacteria or viruses will remain on objects such as escalator rail, doorknob, shopping cart, etc. handled by infected patients. Though their infection may be weakened to varying degrees, it still poses potential threats to indirect infection. We did not take these contaminated objects into account yet. Object detection and tracking techniques will help to locate these objects. It is challenging to accurately determine whether a person is in direct contact with an object rather than just being close to it due to the factors of occlusion and overlapping. When the contaminated object is sheltered from persons or multiple objects overlap each other, the visible part of the object is insufficient to provide sufficient information for making a judgment.

Second, formulating the infection risk assessment criteria based on vision-level rather than chemical analysis also presents

a unique set of challenges. Obtaining the exact distance between people in practical circumstances is necessary for verifying the estimated distance by our method. Besides, the potential risk of infection varies according to different environmental set-ups and transmission routes of infectious diseases. A confined space like a room may lead to a higher risk than an open space. The cumulative effect of continuously contact over a while rather than a particular moment is difficult to measure. Besides, it is worth pointing out that we do not take the intra-person variations of immunity into account since it cannot be measured at the vision-level.

7. CONCLUSION

This paper proposes a novel method to represent the potentially-infected group of people as a graph structure. We also model the principle of close proximity interaction by robustly analyzing the physical distance between subjects in the 3D world. This vision-based approach can re-identify diagnosed patients with infectious diseases and evaluate the infection risk of people who have contacted them. We evaluated our method in various scenarios, including indoor office, bus station, bus compartment, hospital. The comparison with the process of manual analysis shows that our method achieves consistent results but significantly reduces the time cost.

There are a few directions for our future work. Our current work focuses on the direct contact between the subjects and neglects the indirect contact between subjects via objects. It is highly likely that the objects in close contact with the diagnosed subject contain the virus and thus lead to disease spread. Investigating the indirect infection caused by contaminated objects is in line with our future work. Besides, deploying our method in an in-the-wild study could validate the effectiveness of our method in the real world. One potential scenario is to predict the absentee statistics of the childcare center, given the surveillance camera videos. This could offer advice to parents and administrators concerning the status of the disease infection on both individual and group levels.

DATA AVAILABILITY STATEMENT

Publicly available datasets were analyzed in this study. This data can be found here: http://vislab.isr.ist.utl.pt/hda-dataset/.

ETHICS STATEMENT

The studies involving human participants were reviewed and approved by School of Informatics, Xiamen University. The patients/participants provided their written informed consent to participate in this study.

AUTHOR CONTRIBUTIONS

SG conceived, designed the analysis, and wrote the majority of this manuscript. MJ and XG conceived, designed the analysis, and revised the manuscript. XS conceived the analysis and participated in the writing of the manuscript. JY and HW collected and performed the analysis. FX read and revised the manuscript.

FUNDING

This work was supported by National Natural Science Foundation of China (61702433, 61673328, 61661146002) and the Fundamental Research Funds for the Central Universities.

REFERENCES

Bali, K. K., Gupta, A., Feng, L., Ong, Y. S., and Siew, T. P. (2017). "Linearized domain adaptation in evolutionary multitasking," in *2017 IEEE Congress on Evolutionary Computation (CEC)* (San Sebastian: IEEE), 1295–1302.

Cao, Z., Simon, T., Wei, S.-E., and Sheikh, Y. (2017). "Realtime multi-person 2d pose estimation using part affinity fields," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Honolulu, HI: IEEE), 7291–7299.

Chen, Q., Ma, X., Sun, Y., and Zhu, Z. (2017a). "Adaptive memetic algorithm based evolutionary multi-tasking single-objective optimization," in *Asia-Pacific Conference on Simulated Evolution and Learning* (Shenzhen: Springer), 462–472.

Chen, Q., Ma, X., Zhu, Z., and Sun, Y. (2017b). "Evolutionary multi-tasking single-objective optimization based on cooperative co-evolutionary memetic algorithm," in *2017 13th International Conference on Computational Intelligence and Security (CIS)* (Hong Kong: IEEE), 197–201.

Chen, W., Chen, X., Zhang, J., and Huang, K. (2017c). "Beyond triplet loss: a deep quadruplet network for person re-identification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Honolulu, HI: IEEE), 403–412.

Chen, Y., Wang, Z., Peng, Y., Zhang, Z., Yu, G., and Sun, J. (2018). "Cascaded pyramid network for multi-person pose estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Salt Lake City, UT), 7103–7112.

Cheng, D., Gong, Y., Zhou, S., Wang, J., and Zheng, N. (2016). "Person re-identification by multi-channel parts-based cnn with improved triplet loss function," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Las Vegas, NV: IEEE), 1335–1344.

Ding, S., Lin, L., Wang, G., and Chao, H. (2015). Deep feature learning with relative distance comparison for person re-identification. *Patt Recog.* 48, 2993–3003. doi: 10.1016/j.patcog.2015.04.005

Dodd, P. J., Looker, C., Plumb, I. D., Bond, V., Schaap, A., Shanaube, K., et al. (2015). Age-and sex-specific social contact patterns and incidence of mycobacterium tuberculosis infection. *Am. J. Epidemiol.* 183, 156–166. doi: 10.1093/aje/kwv160

Eames, K. T., Tilston, N. L., Brooks-Pollock, E., and Edmunds, W. J. (2012). Measured dynamic social contact patterns explain the spread of h1n1v influenza. *PLoS Comput. Biol.* 8:e1002425. doi: 10.1371/journal.pcbi.1002425

Fang, H.-S., Xie, S., Tai, Y.-W., and Lu, C. (2017). "Rmpe: regional multi-person pose estimation," in *Proceedings of the IEEE International Conference on Computer Vision* (Venice: IEEE), 2334–2343.

Feng, L., Zhou, L., Zhong, J., Gupta, A., Ong, Y.-S., Tan, K.-C., et al. (2018). Evolutionary multitasking via explicit autoencoding. *IEEE Trans. Cybernet.* 49, 3457–3470. doi: 10.1109/TCYB.2018.2845361

Fu, Y., Wei, Y., Zhou, Y., Shi, H., Huang, G., Wang, X., et al. (2018). Horizontal pyramid matching for person re-identification. *arXiv [preprint]. arXiv:1804.05275.*

Gupta, A., Ong, Y.-S., Feng, L., and Tan, K. C. (2016). Multiobjective multifactorial optimization in evolutionary multitasking. *IEEE Trans. Cybernet.* 47, 1652–1665. doi: 10.1109/TCYB.2016.2554622

He, K., Gkioxari, G., Dollár, P., and Girshick, R. (2017). "Mask r-cnn," in *Proceedings of the IEEE International Conference on Computer vision* (Las Vegas, NV: IEEE), 2961–2969.

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), 770–778.

Huang, S., Gong, M., and Tao, D. (2017). "A coarse-fine network for keypoint localization," in *Proceedings of the IEEE International Conference on Computer Vision* (Venice: IEEE), 3028–3037. doi: 10.1109/ICCV.2017.329

Ibuka, Y., Ohkusa, Y., Sugawara, T., Chapman, G. B., Yamin, D., Atkins, K. E., et al. (2016). Social contacts, vaccination decisions and influenza in japan. *J. Epidemiol. Commun. Health* 70, 162–167. doi: 10.1136/jech-2015-205777

Insafutdinov, E., Pishchulin, L., Andres, B., Andriluka, M., and Schiele, B. (2016). "Deepercut: a deeper, stronger, and faster multi-person pose estimation model," in *European Conference on Computer Vision* (Amsterdam: Springer), 34–50.

Isella, L., Romano, M., Barrat, A., Cattuto, C., Colizza, V., Van den Broeck, W., et al. (2011). Close encounters in a pediatric ward: measuring face-to-face proximity and mixing patterns with wearable sensors. *PLoS ONE* 6:e17144. doi: 10.1371/journal.pone.0017144

Jiang, M., Huang, Z., Qiu, L., Huang, W., and Yen, G. G. (2018). Transfer learning-based dynamic multiobjective optimization algorithms. *IEEE Trans. Evol. Comput.* 22, 501–514. doi: 10.1109/TEVC.2017.2771451

Leung, K., Jit, M., Lau, E. H., and Wu, J. T. (2017). Social contact patterns relevant to the spread of respiratory infectious diseases in hong kong. *Sci. Rep.* 7:7974. doi: 10.1038/s41598-017-08241-1

Li, W., Zhao, R., Xiao, T., and Wang, X. (2014). "Deepreid: deep filter pairing neural network for person re-identification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Columbus, OH: IEEE), 152–159.

Li, Z., Chang, S., Liang, F., Huang, T. S., Cao, L., and Smith, J. R. (2013). "Learning locally-adaptive decision functions for person verification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Portland, OR: IEEE), 3610–3617.

Liang, Z., Zhang, J., Feng, L., and Zhu, Z. (2019). A hybrid of genetic transform and hyper-rectangle search strategies for evolutionary multi-tasking. *Expert Syst. Appl.* 138:112798. doi: 10.1016/j.eswa.2019.07.015

Liao, S., Hu, Y., Zhu, X., and Li, S. Z. (2015). "Person re-identification by local maximal occurrence representation and metric learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Boston, MA), 2197–2206. doi: 10.1109/CVPR.2015.7298832

Nambiar, A., Taiana, M., Figueira, D., Nascimento, J., and Bernardino, A. (2014). A multi-camera video dataset for research on high-definition surveillance. *Int. J. Mach. Intell. Sens. Signal Process.* 1, 267–286. doi: 10.1504/IJMISSP.2014.066428

Ozella, L., Gesualdo, F., Tizzoni, M., Rizzo, C., Pandolfi, E., Campagna, I., et al. (2018). Close encounters between infants and household members measured through wearable proximity sensors. *PLoS ONE* 13:e0198733. doi: 10.1371/journal.pone.0198733

Papandreou, G., Zhu, T., Kanazawa, N., Toshev, A., Tompson, J., Bregler, C., et al. (2017). "Towards accurate multi-person pose estimation in the wild," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Seattle, WA), 4903–4911.

Pishchulin, L., Insafutdinov, E., Tang, S., Andres, B., Andriluka, M., Gehler, P. V., et al. (2016). "Deepcut: joint subset partition and labeling for multi person pose estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Seattle, WA), 4929–4937.

Read, J. M., Lessler, J., Riley, S., Wang, S., Tan, L. J., Kwok, K. O., et al. (2014). Social mixing patterns in rural and urban areas of southern china. *Proc. Roy. Soc. B Biol. Sci.* 281:20140268. doi: 10.1098/rspb.2014.0268

Ren, S., He, K., Girshick, R., and Sun, J. (2015). "Faster r-cnn: towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems* (Montreal, QC), 91–99.

Salathé, M., Kazandjieva, M., Lee, J. W., Levis, P., Feldman, M. W., and Jones, J. H. (2010). A high-resolution human contact network for infectious disease transmission. *Proc. Natl. Acad. Sci. U.S.A.* 107, 22020–22025. doi: 10.1073/pnas.1009094108

Saquib Sarfraz, M., Schumann, A., Eberle, A., and Stiefelhagen, R. (2018). "A pose-sensitive embedding for person re-identification with expanded cross neighborhood re-ranking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*(Salt Lake City, UT: IEEE), 420–429.

Smieszek, T., Castell, S., Barrat, A., Cattuto, C., White, P. J., and Krause, G. (2016). Contact diaries versus wearable proximity sensors in measuring contact patterns at a conference: method comparison and participants' attitudes. *BMC Infect. Dis.* 16:341. doi: 10.1186/s12879-016-1676-y

Song, C., Huang, Y., Ouyang, W., and Wang, L. (2018). "Mask-guided contrastive attention model for person re-identification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Salt Lake City, UT: IEEE), 1179–1188.

Stehlé, J., Voirin, N., Barrat, A., Cattuto, C., Isella, L., Pinton, J.-F., et al. (2011). High-resolution measurements of face-to-face contact patterns in a primary school. *PLoS ONE* 6:e23176. doi: 10.1371/journal.pone.0023176

Sun, Y., Zheng, L., Deng, W., and Wang, S. (2017). "Svdnet for pedestrian retrieval," in *Proceedings of the IEEE International Conference on Computer Vision*, 3800–3808.

Varior, R. R., Haloi, M., and Wang, G. (2016). "Gated siamese convolutional neural network architecture for human re-identification," in *European Conference on Computer Vision* (Amsterdam: Springer), 791–808.

Voirin, N., Payet, C., Barrat, A., Cattuto, C., Khanafer, N., Régis, C., et al. (2015). Combining high-resolution contact data with virological data to investigate influenza transmission in a tertiary care hospital. *Infect. Control Hosp. Epidemiol.* 36:254–260. doi: 10.1017/ice.2014.53

Wang, J., Song, Y., Leung, T., Rosenberg, C., Wang, J., Philbin, J., et al. (2014). "Learning fine-grained image similarity with deep ranking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Columbus, OH: IEEE), 1386–1393.

Yan, S., Weycker, D., and Sokolowski, S. (2017). Us healthcare costs attributable to type a and type b influenza. *Hum. Vacc. Immunotherapeut.* 13, 2041–2047. doi: 10.1080/21645515.2017.1345400

Yi, D., Lei, Z., and Li, S. (2014). Deep metric learning for practical person re-identification. *ArXiv e-prints*. doi: 10.1109/ICPR.2014.16

Yin, J., Zhu, A., Zhu, Z., Yu, Y., and Ma, X. (2019). "Multifactorial evolutionary algorithm enhanced with cross-task search direction," in *2019 IEEE Congress on Evolutionary Computation (CEC)* (Wellington: IEEE), 2244–2251.

Yu, S., Tan, T., Huang, K., Jia, K., and Wu, X. (2009). A study on gait-based gender classification. *IEEE Trans. Image Process.* 18, 1905–1910. doi: 10.1109/TIP.2009.2020535

Yu, Y., Zhu, A., Zhu, Z., Lin, Q., Yin, J., and Ma, X. (2019). "Multifactorial differential evolution with opposition-based learning for multi-tasking optimization," in *2019 IEEE Congress on Evolutionary Computation (CEC)* (Wellington: IEEE), 1898–1905.

Zheng, W.-S., Gong, S., and Xiang, T. (2011). "Person re-identification by probabilistic relative distance comparison," in *CVPR 2011* (Providence, RI: IEEE), 649–656. doi: 10.1109/CVPR.2011.5995598

Zhou, T., Brown, M., Snavely, N., and Lowe, D. G. (2017). "Unsupervised learning of depth and ego-motion from video," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Honolulu, HI: IEEE), 1851–1858. doi: 10.1109/CVPR.2017.700

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.

frontiers in Neuroscience

ORIGINAL RESEARCH
published: 23 January 2020
doi: 10.3389/fnins.2020.00001

Multi-Task Network Representation Learning

Yu Xie[1], *Peixuan Jin*[1], *Maoguo Gong*[2], *Chen Zhang*[1] *and Bin Yu*[1]*

[1] School of Computer Science and Technology, Xidian University, Xi'an, China, [2] Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education, School of Electronic Engineering, Xidian University, Xi'an, China

Networks, such as social networks, biochemical networks, and protein-protein interaction networks are ubiquitous in the real world. Network representation learning aims to embed nodes in a network as low-dimensional, dense, real-valued vectors, and facilitate downstream network analysis. The existing embedding methods commonly endeavor to capture structure information in a network, but lack of consideration of subsequent tasks and synergies between these tasks, which are of equal importance for learning desirable network representations. To address this issue, we propose a novel multi-task network representation learning (MTNRL) framework, which is end-to-end and more effective for underlying tasks. The original network and the incomplete network share a unified embedding layer followed by node classification and link prediction tasks that simultaneously perform on the embedding vectors. By optimizing the multi-task loss function, our framework jointly learns task-oriented embedding representations for each node. Besides, our framework is suitable for all network embedding methods, and the experiment results on several benchmark datasets demonstrate the effectiveness of the proposed framework compared with state-of-the-art methods.

Keywords: multi-task learning, representation learning, node classification, link prediction, graph neural network

OPEN ACCESS

Edited by:
Liang Feng,
Chongqing University, China

Reviewed by:
Yaqing Hou,
Dalian University of Technology (DUT), China
Xiaofen Lu,
Southern University of Science and Technology, China

***Correspondence:**
Bin Yu
yubin@mail.xidian.edu.cn

Specialty section:
This article was submitted to Decision Neuroscience, a section of the journal Frontiers in Neuroscience

Received: *31 August 2019*
Accepted: *03 January 2020*
Published: *23 January 2020*

Citation:
Xie Y, Jin P, Gong M, Zhang C and Yu B (2020) Multi-Task Network Representation Learning. Front. Neurosci. 14:1. doi: 10.3389/fnins.2020.00001

1. INTRODUCTION

Networks are ubiquitous in the real world, and can be organized in the form of graphs where nodes represent various objects and edges represent relationships between objects. For examples, in a protein-protein interaction network (Wang et al., 2019), the physical interactions among proteins constitute the networks of protein complexes where each individual protein is an independent node and the interaction represents an edge. In medical practice (Litjens et al., 2017), analyzing protein-protein networks can gain new insights into biochemical cascades and guide the discovery of putative protein targets of therapeutic interest. For efficiently mining these complex networks, it is necessary to learn an informative and discriminative representation for each node in the complex network. Therefore, network representation learning (Cui et al., 2019), also known as graph embedding (Yan et al., 2005), has attracted a great deal of attention in recent years.

Existing network representation learning methods can be generally divided into two categories, including unsupervised and semi-supervised methods. Unsupervised network representation learning methods (Khosla et al., 2019), such as DeepWalk (Perozzi et al., 2014), node2vec (Grover and Leskovec, 2016), and GraphGAN (Wang et al., 2018), explore specific proximities and topological information in a complex network and optimize the carefully designed unsupervised loss for learning node representations, which can be used for subsequent node classification (Kazienko and Kajdanowicz, 2011) and link prediction (Liben-Nowell and Kleinberg, 2007; Lü and Zhou, 2011). Semi-supervised network representation learning methods (Li et al., 2017), such

as GraphSAGE (Hamilton et al., 2017), GAT (Veličković et al., 2018), and so on, develop end-to-end graph neural network architectures for semi-supervised node classification based on the partial labeled nodes and other unlabeled nodes in hand. However, all of these methods are lack of adequate consideration for subsequent network analysis tasks. More specifically, unsupervised network representation learning methods inherently ignore the category attributes of nodes. Both unsupervised and semi-supervised network representation learning methods are not supervised by the link prediction task in the process of learning desirable node representations. The only existing work is that, Tran et al. presented a densely connected autoencoder architecture (Zhu et al., 2016), namely local neighborhood graph autoencoder (LoNGAE, αLoNGAE) (Tran, 2018), to learn a joint representation of both local graph structure and available external node features for the multi-task learning (Yu and Qiang, 2017) of node classification and link prediction. Nevertheless, it has poor scalability on general network embedding methods due to the use of autoencoder.

As a bridge between the graph structured network data and the underlying network analysis task, network representation learning algorithms should not only preserve the proximities and complex topological structure, but also learn high-quality node representations for enhancing the performance of relevant tasks. Fortunately, multi-task learning (MTL) is a standard paradigm that takes full advantage of the synergy among tasks to make multiple learning tasks promote each other (Yu and Qiang, 2017). In deep learning (LeCun et al., 2015), multi-task learning (Caruana, 1993) is usually implemented by sharing the soft or hard parameters of the hidden layer. Each task has its own parameters and models when sharing soft parameters. The distance between model parameters is regularized to encourage parameter similarity. Sharing the hard parameter is the most common method of multi-task learning on neural networks, which significantly reduces the risk of overfitting.

Inspired by this, we attempt to propose a universal multi-task network representation learning (MTNRL) framework, which can be implemented on general network embedding methods for link prediction and node classification. To enable the traditional network embedding methods to effectively learn multiple tasks synchronously, two different network analysis tasks share parameters of the feature extraction module and retain its own task-specific module in our framework. The shared feature extraction module is utilized for learning the latent low-dimensional representations of nodes in a complex network. The task-specific module takes the obtained node representations as input and incorporates the losses of node classification and link prediction tasks. Through jointly optimizing the overall losses, we can learn the desirable network representations and improve the classification or prediction results of different tasks. Besides, our proposed MTNRL framework has good universality and can be applied to almost all of the existing network representation learning approaches.

The main contributions of this paper are summarized as follows:

- We propose a novel multi-task network representation learning (MTNRL) framework, which simultaneously performs multiple tasks including node classification and link prediction by sharing the intermediate embedding representations of nodes.
- The proposed framework is implemented on state-of-the-art graph attention neural networks in detail for illustration.
- We conduct empirical evaluation on three datasets and the experimental results demonstrate that the proposed framework achieves similar or even better results than existing original network representation learning methods.

The rest of this paper is arranged as follows. We first summarize related works in section 2. Section 3 presents our proposed multi-task network representation learning framework for node classification and link prediction. Section 4 describes the experimental settings and results, while conclusions are discussed in section 5.

2. RELATED WORK AND MOTIVATION

2.1. Network Representation Learning

Recently, network representation learning has attracted an increasing research attention in various fields. Existing network representation learning techniques can roughly be divided as unsupervised and semi-supervised. Given a complex network with all nodes being unlabeled, unsupervised methods learn node representations through optimizing the carefully designed objective to capture proximities and topology in the network graph, which can facilitate identifying the class labels for the nodes. Deepwalk (Perozzi et al., 2014) regards the sequence of nodes generated by random walk (Tong et al., 2006) as a sentence, the nodes in the sequence as words in the text, and obtains node representations through optimizing the Skip-Gram model (Lazaridou et al., 2015). LINE (Tang et al., 2015) characterizes the first-order proximity observed from the connections among nodes, and preserves the second-order proximity through calculating the number of common neighbors for two nodes without direct connection. Node2vec (Grover and Leskovec, 2016) extends the Deepwalk algorithm by introducing a pair of hyper-parameters for adding flexibility in exploring neighborhoods, and generates random walk sequences by breadth-first search (Beamer et al., 2013) and depth-first search (Barták, 2004).

Unsupervised learning begins with clustering and then characterization, while supervised learning is carried out simultaneously with classification and characterization. Semi-supervised learning is a classic paradigm of machine learning between supervised learning and unsupervised learning. In this paradigm, a small amount of labeled data and a large number of unlabeled data are used to train the learning model. In practice, it is arduous to obtain a great deal of labeled data and semi-supervised learning is capable of improving the performance of purely supervised learning algorithms through modeling the distribution of unlabeled data. Therefore, semi-supervised learning has received considerable attention in recent years. Semi-supervised learning methods utilize partial nodes being

labeled and others remaining unlabeled to learn high-quality node representations supervised by partial nodes. For examples, graph convolution networks (GCN) (Kipf and Welling, 2017) generalizes the original convolutional neural networks on grid-like images to non-grid graphs through considering the localized first-order approximation of spectral graph convolutions for encoding graph structure and optimizing the cross-entropy loss over labeled node examples for semi-supervised node classification. Given a graph composed of instance nodes, Planetoid (Yang et al., 2016) presents a semi-supervised learning framework based on graph embeddings which can train an embedding for each instance to jointly predict the class label and the neighborhood context in the graph. This method has both transduction variables and induction variables. While in the inductive variant, the embeddings are defined as a parametric function of the feature vectors, so predictions can be made on instances not seen during training. GraphSAGE (Hamilton et al., 2017) is an inductive network representation learning framework that learns an embedding function for generating node representations through sampling a fixed-size set of neighbors of each node, and then performing a specific aggregator over neighboring nodes (such as the mean over all the sampled neighbors' feature vectors, or the result of feeding them through a recurrent neural network). Graph attention networks (GAT) (Veličković et al., 2018) operate on graph-structured data, leveraging masked self-attentional layers (Zhang et al., 2018) to address the shortcomings of prior methods based on graph convolutions. These methods are all implemented as a single task, but multi-task learning can be used to improve the performance of multiple tasks simultaneously.

2.2. Multi-Task Learning

Multi-task learning is a promising area of machine learning that leverages the useful information contained in multiple learning tasks to help learn each task more accurately. Multi-task learning is capable of learning more than one learning task simultaneously, because each task can take advantage of the knowledge of other related tasks. Traditional multi-task learning methods (Doersch and Zisserman, 2017) can be classified into many kinds, including multi-task supervised learning, multi-task unsupervised learning (Kim et al., 2017), and multi-task semi-supervised learning (Zhuang et al., 2015). Multi-task supervised learning implies that each task in multi-task learning is a supervised learning task, which models the function mapping from examples to labels. Different from the multi-task supervised learning with labeled examples, the training set of multi-task unsupervised learning only consists of unlabeled examples to mine the information contained in the dataset.

2.3. Motivation

In many practical applications, there is usually only a small amount of labeled graph data, because manual annotation wastes labor and time considerably (Navon and Goldschmidt, 2003). For example, in biology, the structure and function analysis of a protein network may take a long time, while large amounts of unlabeled data are easily available. Hence, semi-supervised learning methods are widely used to improve learning performance of graph analysis. Unfortunately, all of the aforementioned semi-supervised learning methods applied on graphs, such as GCN, GraphSAGE, and GAT only learn the latent node representations in a single-task oriented manner and lack consideration of the synergy among subsequent graph analytic tasks. In reality, tasks of node classification and link prediction usually share some common characteristics and can be conducted simultaneously for facilitating each other.

As far as we know, the only existing work is the local neighborhood graph autoencoder (LoNGAE, αLoNGAE), which implements the multi-task network representation learning based on a densely connected symmetrical autoencoder and is model dependent. The model utilizes the parameter sharing between encoders and decoders to learn expressive non-linear latent node representations from local graph neighborhoods. Motivated by this, we innovatively propose a general multi-task network representation learning (MTNRL) framework, which is model-agnostic and can be applied on arbitrary network representation models. It optimizes the losses of two tasks jointly to learn the desirable node representations followed by node classification and link prediction tasks that performed on the embedding vectors.

3. METHODOLOGY

In this section, we formally define the problems of network representation learning and multi-task learning. Then the proposed MTNRL framework and its implementation on graph attention networks are elaborated in detail.

3.1. Problem Formulation and Notations

A network is usually denoted as $G = (V, E)$, where $V = \{v_1, \cdots, v_n\}$ represents a set of nodes and n is the number of nodes. $E = \left\{e_{i,j}\right\}_{i,j=1}^{n}$ denotes the set of edges between any two nodes. Each edge $e_{i,j}$ can be associated with a weight $a_{i,j} \geq 0$, which is an element of the adjacency matrix A for the network G. In an unweighted graph, for nodes v_i and v_j not linked by an edge, $a_{i,j} = 0$, otherwise, $a_{i,j} = 1$. Formally, we define the following two problems closely related to our work.

Definition 1 (**Network representation learning**). Given a network $G = (V, E)$, network representation learning aims to learn a function $f : V \rightarrow R^{n \times d}$, that maps each node into a d-dimensional embedding space. Meanwhile, d is the dimension of latent representations and $d \ll n$.

Definition 2 (**Multi-task learning**). Given multiple related learning tasks, the goal of multi-task learning is to improve the performance of each task by jointly learning these related tasks and mining the useful information contained in these tasks.

The main symbols used throughout this paper are listed in **Table 1**.

3.2. Framework

Aiming to obtain the compact and expressive representation of a complex network, network representation learning is widely

TABLE 1 | Notations and their descriptions.

Notations	Descriptions
G	The given network
V	Set of nodes in the given network
E	Set of edges in the given network
v	A node $v \in V$
$e_{i,j}$	An edge between nodes v_i and v_j
n	Number of nodes in the given network
c	The number of class labels for nodes in V
A	The adjacency matrix of G
d	The dimension of learned node representations
Z	The initial feature matrix of nodes
H	The embedding representation matrix of nodes

used in a variety of applications, including node classification, link predication, and so on.

As one of the most important application for network representation learning, node classification attempts to assign the predicted class label to each node in the network based on the patterns learnt from the partially labeled nodes. Intuitively, similar nodes in a complex network should have the same labels. The results of node classification are often used in recommendation systems and data mining systems. Because in these practical applications, nodes in a complex network are only partially labeled due to high labeling costs, and a large portion of vertices in networks do not have ground truth. According to the number of labels of each node in a network, node classification can be categorized into multi-class node classification and multi-label node classification. In multi-label node classification, each node may correspond multiple labels, while each node only has one label in multi-class node classification. Essentially, node classification based on existing network representation learning techniques typically consist of two stages: representation learning and node classification.

With the carefully designed network embedding algorithm, a network graph G can be taken as input to the embedding model f for learning the low-dimensional dense representation H in an unsupervised or semi-supervised manner, which is expressed as:

$$H = f(A, Z) \tag{1}$$

A denotes the adjacency matrix of G and Z is the initial feature representation of nodes, which can be represented by nodes' feature property or other properties. For unsupervised network representation learning, the obtained node representations are then utilized to train a supervised classifier for node classification. Semi-supervised network representation learning directly trains a classifier well for classification while training the embedding model. With the well-trained classifier, we can infer the labels of the remaining nodes. The performance of node classification is reflected by the predicted accuracy for node labels. The loss function of node classification can be defined as follow:

$$\mathcal{L}_{NC} = -\sum_{v \in V_L} \sum_{k=1}^{c} y_{v,k} * \log(P_{v,k}) \tag{2}$$

where V_L is the set of labeled nodes and c denotes the number of class labels. $y_{v,k}$ represents an indicator variable of node v, which is equal to 1 if node v belongs to class k, otherwise 0. P_v is the predicted probability vector of node v and can be calculated by $P_v = \text{softmax}(W^T h_v + b)$, in which h_v is the embedding representation of node v, W is the weight matrix, and b is the bias in the final fully connected layer.

Another fundamental application for network representation learning is link predication. Link prediction endeavors to predict the existing possibility of edges between two nodes in a network that are unobserved or missing by utilizing available network nodes and topological structure. In general, we randomly hide a portion of the existing links for simulation and use the left edges to train an unsupervised network embedding model. To seamlessly integrate the tasks of link prediction and node classification, we design a loss function for link prediction as:

$$\mathcal{L}_{LP} = -\sum_{i=1}^{n} \sum_{j=1}^{n} \left[A_{i,j} * \log\left(S_{i,j}\right) + \left(1 - A_{i,j}\right) * \log\left(1 - S_{i,j}\right) \right] \tag{3}$$

where $A_{i,j}$ is an element of the adjacency matrix of a network G and n indicates the number of nodes. $S_{i,j} = \text{s}(h_i, h_j)$ is a score of the predicted link between nodes v_i and v_j, which can be calculated with the inner product or other similarity measure between embedding representations h_i and h_j. A larger score usually implies that the two nodes may have a higher likelihood to be linked. With the loss in Equation (3), we can learn the structural representations for each node in the network graph and then utilize the obtained representations to predict the unobserved link.

To benefit subsequent tasks of both node classification and link prediction, we learn informative and discriminative graph representations collaboratively supervised by these two tasks. More specifically, the overall loss function for multi-task network representation learning (MTNRL) can be formulated as:

$$\mathcal{L} = \mathcal{L}_{NC} + \alpha \mathcal{L}_{LP} \tag{4}$$

where α is a tradeoff factor for balancing losses of node classification and link prediction. For illustration, our MTNRL framework is shown in **Figure 1**. A network graph is taken as the input to a network representation learning model. By virtue of the network representation learning model for graph-structured data, the proximity and topological structure will be preserved in the embedding representations. Furthermore, we simultaneously perform node classification and link prediction tasks through optimizing the carefully designed multi-task loss function on the node representations obtained from the representation learning module. As a result, we jointly learn task-oriented embedding representations for each node, which are capable of improving the performance of a variety of graph analytics applications.

3.3. Implementation on Graph Attention Networks

Graph attention networks (GAT) (Veličković et al., 2018) introduce an attention-based architecture to learn the

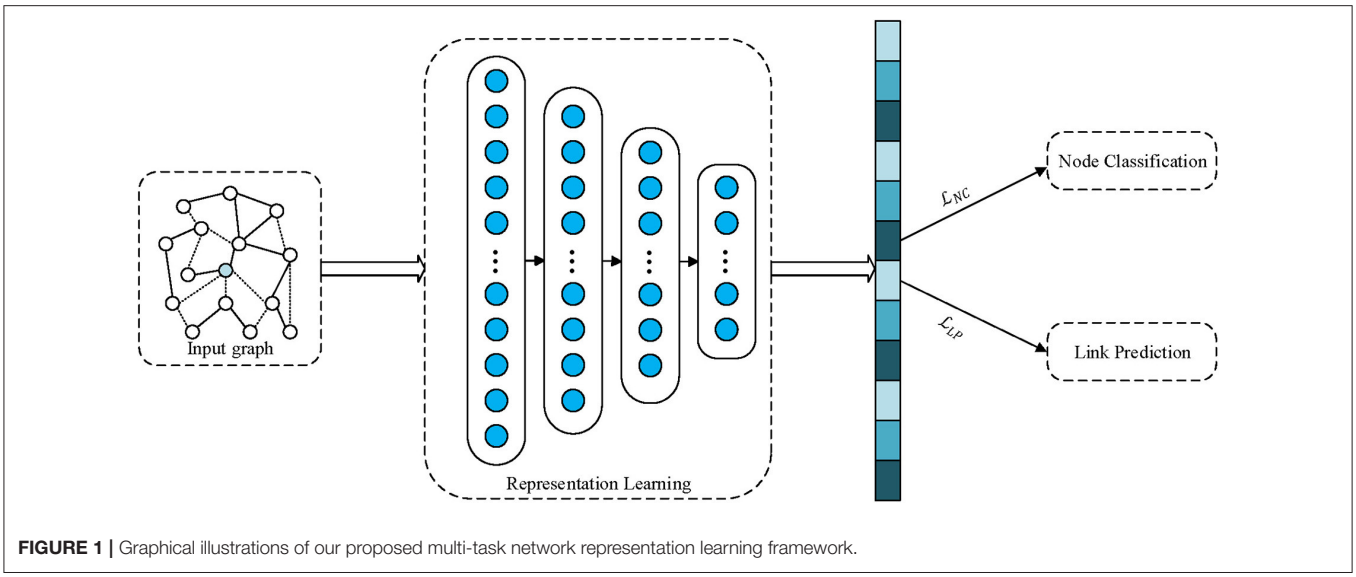

FIGURE 1 | Graphical illustrations of our proposed multi-task network representation learning framework.

node-focused representations for node classification on graph-structured data. GAT is based on the classical neighbor aggregation schema for generating low-dimensional node representations and extends the pioneering graph convolutional networks through exploring the importance of different neighboring nodes. Based on the attention mechanism widely used in sequence-based tasks, GAT calculates an attention coefficient $e_{ij} = a\left(\mathbf{W}\vec{h}_i, \mathbf{W}\vec{h}_j\right)$ for pairwise nodes. Suppose $\mathbf{h} = \left\{\vec{h}_1, \vec{h}_2, \ldots, \vec{h}_N\right\}, \vec{h}_i \in \mathbb{R}^F$ is a set of node features used as the input to the attention layer, where N is the number of nodes, and F is the number of features for each node. A shared linear transformation, parameterized by a weight matrix, $\mathbf{W} \in \mathbb{R}^{F' \times F}$, is applied to every node. Then the shared attentional mechanism $a : \mathbb{R}^{F'} \times \mathbb{R}^{F'} \rightarrow \mathbb{R}$ is utilized to calculate e_{ij}. With the normalized attention coefficients $\alpha_{ij} = \text{softmax}_j\left(e_{ij}\right) = \frac{\exp(e_{ij})}{\sum_{k \in \mathcal{N}_i} \exp(e_{ik})}$, we can pay different attention to the neighboring nodes when attending over its neighbors for generating the latent representation of each node. Therefore, the normalized attention coefficients are used to compute a linear combination of the features corresponding to them, to serve as the final output features for every node (after potentially applying a non-linear function σ): $\vec{h}'_i = \sigma\left(\sum_{j \in \mathcal{N}_i} \alpha_{ij} \mathbf{W}\vec{h}_j\right)$, where $\mathbf{h}' = \left\{\vec{h}'_1, \vec{h}'_2, \ldots, \vec{h}'_N\right\}, \vec{h}'_i \in \mathbb{R}^{F'}$ is a new set of node features produced by the attention layer. By optimizing the loss of semi-supervised node classification, GAT learns the representation of nodes. By stacking to multiple layers, a deep graph attention network can be constructed for capturing the high-order topological relationship among nodes in a graph.

The proposed MTNRL framework can be implemented on arbitrary network representation learning methods. In this subsection, we introduce an implementation of the MTNRL framework on graph attention networks (MT-GAT) as an example. The original graph attention networks adopt a two-layer GAT model for inductive learning, which can predict the labels of nodes in a semi-supervised manner based on the masked self-attention operated on graph-structured data. In our implementation of MT-GAT, node classification and link prediction tasks are predicted simultaneously. As shown in **Figure 2**, a network graph is taken as input to graph attention networks that can output compact embedding representations of nodes. Then we use the learned low-dimensional node representations for multi-task learning. In the MT-GAT, all parameters in the network except the softmax layer for node classification are shared. In this implementation, the loss function of node classification employs a negative log likelihood loss and the loss function of link prediction adopts a two-class cross entropy loss, which is in consistent with Equations (2) and (3).

3.4. Discussion

To further demonstrate that our MTNRL is a universal framework, we explain how it can be used in Graph Convolutional Networks (GCN) (Kipf and Welling, 2017). GCN is a classical convolutional neural network architecture applied to graph-structured data, which can explicitly characterize the first-order neighboring structure and be stacked to multiple layers for encoding high-order proximities in a network. The original GCN only optimizes the semi-supervised node classification loss for learning latent node representations. Under the proposed MTNRL framework, we can optimize the loss functions of both node classification and link prediction tasks at the same time. Through further assigning the proper weights to the losses of two tasks, we can complete the implementation of our MTNRL framework on GCN.

4. EXPERIMENT

We conduct the experimental evaluation of the proposed multi-task network representation learning framework on graph attention networks (MT-GAT), compared with state-of-the-art methods. This section first introduces the specifics of experimental datasets and several baselines. Then, we present the details of the implementation, followed by experimental results

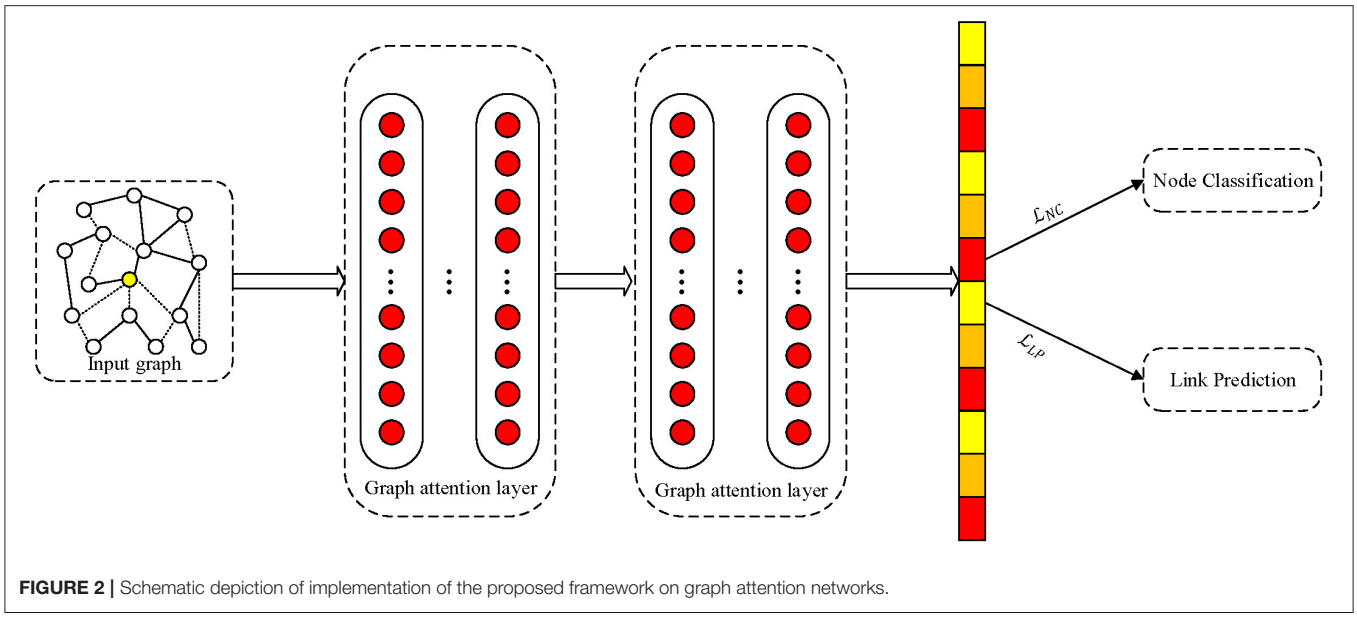

FIGURE 2 | Schematic depiction of implementation of the proposed framework on graph attention networks.

TABLE 2 | Statistics of benchmark datasets used in our experiments.

Datasets	Cora	Citeseer	Pubmed
Nodes	2,708	3,327	19,717
Edges	5,429	4,732	44,338
Text feature dimension	1,433	3,703	500
Classes	7	6	3

and analysis of different algorithms. Finally, we analyze the sensitivity of the hyperparameters.

4.1. Datasets

We adopt three benchmark citation network datasets for evaluation, including Cora, Citeseer, and Pubmed (Sen et al., 2008), whose detailed statistics are summarized in **Table 2**. For these citation networks, each paper is denoted as a node and the words of each paper are encoded as the features of nodes which is a vocabulary containing multiple words. Each node only corresponds a class label. The features of the paper consist of a string of binary codes, which indicate whether the paper contains this word.

- The Cora dataset consists of 2,708 papers from machine learning area and these papers are divided into the seven categories: Case Based, Genetic Algorithms, Neural Networks, Probabilistic Methods, Reinforcement Learning, Rule Learning, Theory. The citation network consists of 5,429 edges that represent citation relationships. The text information of each publication is encoded by a tf-idf vector of 1,433 dimensions indicating the importance of the corresponding words.
- The Citeseer dataset consists of 3,312 scientific publications from the CiteSeer web database, and are categorized into six classes: Agents, Artificial Intelligence, Data Base, Information Retrieval, Machine Language, and HCI. The citation network consists of 4,732 links. Each publication in the dataset is described by a 0/1-valued word vector indicating the absence/presence of the corresponding word from the dictionary. The dictionary consists of 3,703 unique words.
- The Pubmed dataset consists of 19,717 scientific publications from PubMed database pertaining to diabetes classified into three classes: Diabetes Mellitus Experimental, Diabetes Mellitus Type 1, Diabetes Mellitus Type 2. The citation network consists of 44,338 links. Each publication in the dataset form a dictionary which is made up of 500 unique words.

4.2. Baselines

We compare our MT-GAT against the following baselines: graph convolution networks (GCN), graph autoencoder (GAE, VGAE), graph attention networks (GAT), local neighborhood graph autoencoder (LoNGAE, αLoNGAE).

- GCN (Kipf and Welling, 2017) performs a convolution operation on each node's neighbors for feature aggregation in each graph convolutional layer, which can be stacked to deeper networks for semi-supervised node classification tasks.
- GAE and VGAE (Kipf and Welling, 2016) utilize a graph convolutional network (GCN) encoder and a simple inner product decoder. The advantage of this method is that it can naturally incorporate node features compared to most existing unsupervised models for link prediction.
- GAT (Veličković et al., 2018) is a novel neural network architecture that operates on graph-structured data, leveraging masked self-attentional layers to address the shortcomings of prior methods based on graph convolution or their approximation.
- LoNGAE and αLoNGAE (Tran, 2018) introduce a densely connected autoencoder architecture to learn a joint representation of both local graph structure and available external node features for the multi-task learning of link

TABLE 3 | Accuracy of semi-supervised node classification on Cora.

Method	90%	80%	70%	60%	50%	40%	30%	20%	10%
GCN	0.842	0.842	0.828	0.828	0.821	0.821	0.807	0.807	0.800
αLoNGAE	0.803	0.793	0.790	0.783	0.780	0.777	0.770	0.767	0.763
GAT	0.824	0.822	0.816	0.808	0.806	0.804	0.798	0.796	0.794
MT-GAT (ours)	**0.874**	**0.864**	**0.861**	**0.856**	**0.855**	**0.850**	**0.848**	**0.832**	**0.827**

The best results are shown in bold, and our MT-GAT with significant improvements over the baselines is shown with underlines.

TABLE 4 | Accuracy of semi-supervised node classification on Citeseer.

Method	90%	80%	70%	60%	50%	40%	30%	20%	10%
GCN	0.846	0.824	0.824	0.824	0.813	0.802	0.802	0.780	**0.780**
αLoNGAE	0.733	0.727	0.723	0.716	0.710	0.706	0.697	0.690	0.683
GAT	0.718	0.716	0.710	0.708	0.706	0.704	0.700	0.698	0.696
MT-GAT (ours)	**0.852**	**0.845**	**0.841**	**0.835**	**0.830**	**0.820**	**0.816**	**0.800**	**0.780**

The best results are shown in bold, and our MT-GAT with significant improvements over the baselines is shown with underlines.

prediction and node classification. LoNGAE and αLoNGAE adopt the densely connected symmetrical autoencoder, where αLoNGAE uses node features and LoNGAE does not. In our node classification experiments, we only adopt αLoNGAE for comparison due to its superiority.

4.3. Experimental Settings

We implement our MT-GAT with the Pytorch-GPU backend, along with several additional details. Gradient descent optimization is employed with a fixed learning rate of 0.005. Two layers of dropout are used in the model with dropout rate of 0.1 to prevent the problem of overfitting. The number of attention heads in the graph attention layer is set to 8, consistent with the setting for transductive learning in GAT. We train for 300 epochs for MT-GAT. The loss of node classification is negative log likelihood loss while the loss of link prediction is binary cross entropy. The tradeoff factor between node classification and link prediction tasks α is 1. For fair comparison, we use mean classification accuracy to measure the performance of the node classification task, and use AUC and AP to evaluate the results of link prediction. The evaluation metric AUC is the area under the ROC curve. In the context of unbalanced categories, even if the number of certain categories increases significantly, the growth of the curve is not obvious, and therefore we choose it to eliminate the impact of a lot of imbalanced classes. AP is just the average accuracy score.

4.4. Results and Analysis

We use different methods to obtain embedding vectors of nodes, and adopt softmax as classifier. For comparison, the training ratio of the classifier is ranged from 10 to 90% with a step of 10% in each dataset for all methods. We run each method 10 times, respectively at a given training ratio and report the average performance.

Tables 3–5 demonstrate the comparison of mean classification accuracy on semi-supervised node classification for GCN, αLoNGAE, GAT, and our MT-GAT. For clarity, the best results

TABLE 5 | Accuracy of semi-supervised node classification on Pubmed.

Method	90%	80%	70%	60%	50%	40%	30%	20%	10%
GCN	**0.871**	0.838	0.838	0.806	0.806	0.774	0.774	0.741	0.741
αLoNGAE	0.807	0.803	0.800	0.797	0.796	0.793	0.790	0.787	0.786
GAT	0.794	0.792	0.790	0.788	0.786	0.784	0.782	0.780	0.788
MT-GAT (ours)	0.854	**0.847**	**0.843**	**0.836**	**0.831**	**0.824**	**0.822**	**0.816**	**0.806**

The best results are shown in bold, and our MT-GAT with significant improvements over the baselines is shown with underlines.

TABLE 6 | AUC and AP performance of different methods on link prediction.

Method	Cora		Citeseer		Pubmed	
	AUC	AP	AUC	AP	AUC	AP
GAE	0.910	0.920	0.895	0.899	0.964	0.965
VGAE	0.914	0.926	0.908	0.920	0.944	0.947
LoNGAE	0.896	0.915	0.860	0.892	0.926	0.930
αLoNGAE	**0.943**	0.952	**0.956**	**0.964**	0.960	0.963
GCN	0.809	0.811	0.811	0.822	0.828	0.834
MT-GAT (ours)	0.930	**0.963**	0.931	0.963	**0.968**	**0.970**

The best results are shown in bold.

are shown in bold. For node classification, GCN and our MT-GAT exhibit better performance compared with LoNGAE and GAT. Although GCN occasionally outperforms our MT-GAT on the Pubmed dataset when the training ratio is 90%, it is inferior to our MT-GAT in all other cases. It is shown that on this task, the performance of our MT-GAT is relatively stable and splendid compared with baselines, which fully demonstrates the superiority of our multi-task network representation learning framework. Furthermore, we conduct the t-test in **Tables 3–5** and our MT-GAT with significant improvements over the baselines is shown with underline as measured by a t-test with a p-value $\leqslant 0.05$.

Table 6 shows the comparison of AUC and AP performance on link prediction for GAE, VGAE, LoNGAE, αLoNGAE, GCN, and MT-GAT. For link prediction, the LoNGAE that only captures graph structure without node features is less than satisfactory, but the αLoNGAE with node features performs slightly better. Although αLoNGAE occasionally outperforms our MT-GAT on the Cora and Citeseer datasets, αLoNGAE is restrictive and obviously provides no flexibility in extending to general network representation learning methods. In the meantime, the performance of GAE and VGAE is mediocre because it is potentially a poor choice in combination with an inner product decoder, and the generative model is not flexible enough. Note that in this task, our MT-GAT performs comparable or more excellent than other methods, due to the capability of our framework for collaboratively learning task-oriented embedding representations.

Overall, our MT-GAT achieves more outstanding and stable performance on both tasks of node classification and link prediction. However, these baselines mostly learn network representations based on a model-dependent framework without careful consideration of the follow-up tasks to optimize the embedding model. Our MT-GAT is simultaneously supervised

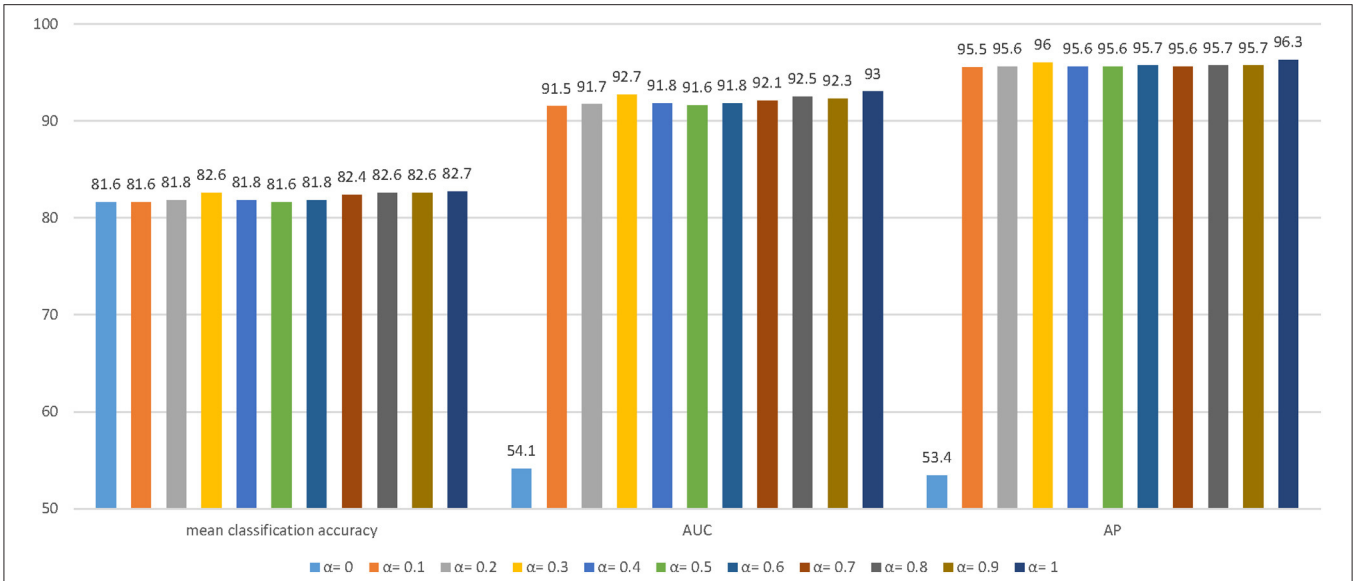

FIGURE 3 | The effect of different hyperparameters α on the Cora dataset. We choose the AUC and AP scores of link prediction and the classification accuracy of node classification to demonstrate the effect of different hyperparameters for the experiments.

by node classification and link prediction tasks, and is capable of learning comprehensive and desirable node representations. Through the joint learning of two different loss functions, our model is able to achieve more effective, complete, and stable predictions.

4.5. Parameter Sensitivity

The parameter sensitivity of MT-GAT is investigated in this section. More specifically, we evaluate how different values of hyperparameter α can affect the performance of node classification and link prediction. The hyperparameter α is varied from 0 to 1 with an increment of 0.1. We report the three evaluation metrics: mean classification accuracy for node classification, AUC score for link prediction, and AP scores for link prediction. The histogram in **Figure 3** displays the results of evaluation metrics with different parameter settings for the Cora dataset. We notice that the performance of node classification and link prediction on the Cora dataset fluctuates from $\alpha = 0$ to 1. It slightly boosts at first and reaches the local optimum at $\alpha = 0.3$. After the value of α is over 0.3, it gradually declines and slightly increases to the peak at $\alpha = 1$. The AUC and AP scores of link prediction are more sensitive to parameters than the classification accuracy of node classification. Especially, when parameter α is 0, the optimization of the link prediction loss is completely separated from that of the network embedding model, thus causing AUC and AP scores of link prediction to always float around the starting value of 0.5. It empirically suggests that the consideration of the weight parameter α between node classification and link prediction tasks can facilitate learning network representations more effectively.

5. CONCLUSION

In this paper, we propose a multi-task network representation learning framework, namely MTNRL, which exploits the synergy among the node classification and link prediction tasks for facilitating their individual performance. The experimental results demonstrate the MTNRL framework on GAT is well-performed on a range of graph-structured network datasets for both node classification and link prediction. Besides, the proposed method can soundly outperform the state-of-the-art network representation learning methods. The main advantage of our MT-GAT is the performance improvement brought by the extensive parameter sharing between link prediction and node classification tasks. The proposed framework solves the single-task limitations of traditional network representation learning methods. In particular, our framework is universal and can be implemented on any arbitrary network embedding methods to improve performance. In future work, we will investigate the implementation of our framework on heterogeneous network representation methods and explore the scalability of our framework on other network analysis tasks.

DATA AVAILABILITY STATEMENT

The datasets analyzed in this manuscript are not publicly available. Requests to access the datasets should be directed to peixuanjin@gmail.com.

AUTHOR CONTRIBUTIONS

YX and PJ conceptualized the problem and the technical framework. MG and CZ developed the algorithms, supervised the experiments, and exported the data. YX, PJ, and BY implemented the multi-task representation learning architecture simulation. BY managed the project. All authors wrote the manuscript, discussed the experimental results, and commented on the manuscript.

FUNDING

This work was supported by the Key Research and Development Program of Shaanxi Province (Grant no. 2019ZDLGY17-01, 2019GY-042).

ACKNOWLEDGMENTS

We would like to thank Yu Zhang and Yiming Fan for thoughtful comments on the manuscript and language revision. We were grateful to all study participants for their time and effort.

REFERENCES

Barták, R. (2004). "Incomplete depth-first search techniques: a short survey," in *Proceedings of the 6th Workshop on Constraint Programming for Decision and Control*, ed J. Figwer (Washington, DC), 7–14.

Beamer, S., Asanović, K., and Patterson, D. (2013). Direction-optimizing breadth-first search. *Sci. Program.* 21, 137–148. doi: 10.5555/2388996.2389013

Caruana, R. A. (1993). "Multitask learning: a knowledge-based source of inductive bias," in *Proceedings of the 10th International Conference on Machine Learning* (Amherst, MA), 41–48.

Cui, P., Wang, X., Pei, J., and Zhu, W. (2019). A survey on network embedding. *IEEE Trans. Knowl. Data Eng.* 31, 833–852. doi: 10.1109/TKDE.2018.2849727

Doersch, C., and Zisserman, A. (2017). "Multi-task self-supervised visual learning," in *Proceedings of the IEEE International Conference on Computer Vision* (Venice), 2051–2060.

Grover, A., and Leskovec, J. (2016). "Node2vec: scalable feature learning for networks," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (San Francisco, CA: ACM), 855–864.

Hamilton, W., Ying, Z., and Leskovec, J. (2017). "Inductive representation learning on large graphs," in *Advances in Neural Information Processing Systems* (Long Beach, CA), 1024–1034.

Kazienko, P., and Kajdanowicz, T. (2011). Label-dependent node classification in the network. *Neurocomputing* 75, 199–209. doi: 10.1016/j.neucom.2011.04.047

Khosla, M., Anand, A., and Setty, V. (2019). A comprehensive comparison of unsupervised network representation learning methods. *arXiv* 1903.07902.

Kim, J., Bukhari, W., and Lee, M. (2017). Feature analysis of unsupervised learning for multi-task classification using convolutional neural network. *Neural Process. Lett.* 47, 1–15. doi: 10.1007/s11063-017-9724-1

Kipf, T., and Welling, M. (2017). "Semi-supervised classification with graph convolutional networks," in *Proceedings of the 5th International Conference on Learning Representations* (Toulon).

Kipf, T. N., and Welling, M. (2016). Variational graph auto-encoders. *CoRR* abs/1611.07308.

Lazaridou, A., Pham, N. T., and Baroni, M. (2015). Combining language and vision with a multimodal skip-gram model. *arXiv* 1501.02598.

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

Li, C., Li, Z., Wang, S., Yang, Y., Zhang, X., and Zhou, J. (2017). "Semi-supervised network embedding," in *International Conference on Database Systems for Advanced Applications* (Suzhou: Springer), 131–147.

Liben-Nowell, D., and Kleinberg, J. (2007). The link-prediction problem for social networks. *J. Am. Soc. Inf. Sci Technol.* 58, 1019–1031. doi: 10.1002/asi.20591

Litjens, G., Kooi, T., Bejnordi, B. E., Aaa, S., Ciompi, F., Ghafoorian, M., et al. (2017). A survey on deep learning in medical image analysis. *Med. Image Anal.* 42, 60–88. doi: 10.1016/j.media.2017.07.005

Lü, L., and Zhou, T. (2011). Link prediction in complex networks: a survey. *Phys. A* 390, 1150–1170. doi: 10.1016/j.physa.2010.11.027

Navon, R., and Goldschmidt, E. (2003). Monitoring labor inputs: automated-data-collection model and enabling technologies. *Autom. Constr.* 12, 185–199. doi: 10.1016/S0926-5805(02)00043-2

Perozzi, B., Al-Rfou, R., and Skiena, S. (2014). "Deepwalk: online learning of social representations," in *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (New York, NY: ACM), 701–710.

Sen, P., Namata, G., Bilgic, M., Getoor, L., Galligher, B., and Eliassi-Rad, T. (2008). Collective classification in network data. *AI Mag.* 29:93. doi: 10.1609/aimag.v29i3.2157

Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J., and Mei, Q. (2015). Line: Large-scale information network embedding. In *Proceedings of the 24th International Conference on World Wide Web*, pages 1067–1077.

Tong, H., Faloutsos, C., and Pan, J. Y. (2006). "Fast random walk with restart and its applications," in *Proceedings of 6th International Conference on Data Mining* (Hong Kong: IEEE), 613–622.

Tran, P. V. (2018). "Learning to make predictions on graphs with autoencoders," in *IEEE 5th International Conference on Data Science and Advanced Analytics* (Turin: IEEE), 237–245.

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., and Bengio, Y. (2018). "Graph attention networks," in *Proceedings of the 6th International Conference on Learning Representations* (Vancouver).

Wang, H., Jia, W., Wang, J., Miao, Z., Zhang, W., Zhang, F., et al. (2018). "Graphgan: graph representation learning with generative adversarial nets," in *Proceedings of the 32nd AAAI Conference on Artificial Intelligence* (New Orleans, LA).

Wang, Y., You, Z. H., Yang, S., Li, X., and Zhou, X. (2019). A high efficient biological language model for predicting protein-protein interactions. *Cells* 8:122. doi: 10.3390/cells8020122

Yan, S., Xu, D., Zhang, B., and Zhang, H.-J. (2005). "Graph embedding: a general framework for dimensionality reduction," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Vol. 2 (San Diego, CA: IEEE), 830–837.

Yang, Z., Cohen, W. W., and Salakhutdinov, R. (2016). "Revisiting semi-supervised learning with graph embeddings," in *Proceedings of the 33nd International Conference on Machine Learning* (NewYork, NY), 40–48.

Yu, Z., and Qiang, Y. (2017). A survey on multi-task learning. *arXiv* 1707.08114.

Zhang, H., Goodfellow, I., Metaxas, D., and Odena, A. (2018). "Self-attention generative adversarial networks," in *Proceedings of the 36th International Conference on Machine Learning* (Long Beach, CA).

Zhu, Z., Wang, X., Bai, S., Yao, C., and Bai, X. (2016). Deep learning representation using autoencoder for 3d shape retrieval. *Neurocomputing.* 204, 41–50. doi: 10.1016/j.neucom.2015.08.127

Zhuang, F., Luo, D., Jin, X., Xiong, H., Luo, P., and He, Q. (2015). "Representation learning via semi-supervised autoencoder for multi-task learning," in *International Conference on Data Mining* (Atlantic City, NJ: IEEE), 1141–1146.

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.

frontiers in Neuroscience

ORIGINAL RESEARCH
published: 14 February 2020
doi: 10.3389/fnins.2020.00040

Electroencephalographic Workload Indicators During Teleoperation of an Unmanned Aerial Vehicle Shepherding a Swarm of Unmanned Ground Vehicles in Contested Environments

Raul Fernandez Rojas[1]*, *Essam Debie*[1], *Justin Fidock*[2], *Michael Barlow*[1], *Kathryn Kasmarik*[1], *Sreenatha Anavatti*[1], *Matthew Garratt*[1] and *Hussein Abbass*[1]

[1] School of Engineering & IT, University of New South Wales, Canberra, NSW, Australia, [2] Defence Science and Technology Organisation, Adelaide, SA, Australia

OPEN ACCESS

Edited by:
Liang Feng,
Chongqing University, China

Reviewed by:
Zexuan Zhu,
Shenzhen University, China
Abhishek Gupta,
*Agency for Science, Technology and Research (A*STAR), Singapore*

***Correspondence:**
Raul Fernandez Rojas
r.fernandezrojas@adfa.edu.au

Specialty section:
This article was submitted to Decision Neuroscience, a section of the journal Frontiers in Neuroscience

Received: *01 November 2019*
Accepted: *13 January 2020*
Published: *14 February 2020*

Citation:
Fernandez Rojas R, Debie E, Fidock J, Barlow M, Kasmarik K, Anavatti S, Garratt M and Abbass H (2020) Electroencephalographic Workload Indicators During Teleoperation of an Unmanned Aerial Vehicle Shepherding a Swarm of Unmanned Ground Vehicles in Contested Environments. Front. Neurosci. 14:40. doi: 10.3389/fnins.2020.00040

Background: Although many electroencephalographic (EEG) indicators have been proposed in the literature, it is unclear which of the power bands and various indices are best as indicators of mental workload. Spectral powers (Theta, Alpha, and Beta) and ratios (Beta/(Alpha + Theta), Theta/Alpha, Theta/Beta) were identified in the literature as prominent indicators of cognitive workload.

Objective: The aim of the present study is to identify a set of EEG indicators that can be used for the objective assessment of cognitive workload in a multitasking setting and as a foundational step toward a human-autonomy augmented cognition system.

Methods: The participants' perceived workload was modulated during a teleoperation task involving an unmanned aerial vehicle (UAV) shepherding a swarm of unmanned ground vehicles (UGVs). Three sources of data were recorded from sixteen participants (n = 16): heart rate (HR), EEG, and subjective indicators of the perceived workload using the Air Traffic Workload Input Technique (ATWIT).

Results: The HR data predicted the scores from ATWIT. Nineteen common EEG features offered a discriminatory power of the four workload setups with high classification accuracy (82.23%), exhibiting a higher sensitivity than ATWIT and HR.

Conclusion: The identified set of features represents EEG indicators for the objective assessment of cognitive workload across subjects. These common indicators could be used for augmented intelligence in human-autonomy teaming scenarios, and form the basis for our work on designing a closed-loop augmented cognition system for human-swarm teaming.

Keywords: augmented intelligence, cognitive load, human-autonomy teaming, human-swarm teaming, shepherding, mental load, cognitive indicators, EEG

1. INTRODUCTION

Mental workload refers to the depletion of mental resources due to mental demands imposed by a task on an individual. When task difficulty increases, mental workload increases due to the reduction in available cognitive resources. Research has shown that when an individual is under high cognitive workload and the cognitive workload approaches the individual's cognitive capacity, suboptimal decisions and human errors are expected. In the absence of any increase of task demand, prolonged mental activities also leads to depletion of cognitive resources (Kamzanova et al., 2014). Low workload can also lead to errors, due to boredom and the possibility for human distraction from the main task due to environmental influencing factors.

Humans have a limited amount of resources (both physically and mentally); therefore, optimizing these resources toward specific sets of tasks is likely to produce better results. However, it is challenging to understand these human limitations within a work environment due to many factors, such as demographic factors (gender, age, ethnicity), intrinsic motivation, mood states (happy, sad, anxious, etc.), previous experience, and different problem-solving strategies due to mental abilities, education, and skills. For example, the level of difficulty to accomplish a task might be seen differently by two operators; operator A could see the task difficult at first, but then find a good strategy to solve the task, while operator B could find the task extremely difficult, get discouraged, and fail to complete the task. As human resources are limited, there is a problem when a task demands more resources (Maior et al., 2014).

In many domains, the ability to process information, to react to different environments, and to make accurate decisions is vital. For instance, air traffic controllers (ATCs) generally perform in a highly cognitively-demanding environment, working for long periods of time, and under stress (Dasari et al., 2017). This scenario can lead to depletion of cognitive resources and thus degradation of performance. Another clear example is doctors and nurses in critical care units, they face large volumes of work, need to act quickly, and stay alert after many hours of intense work. In this case, errors and compromised standards signify that quality and safety of patient care might be endangered (MacPhee et al., 2017). It is, therefore, evident that there is a need to measure mental workload to identify the changes of cognitive demands on an individual while completing a task, which can potentially help reduce errors, task failure, accidents, and thus improve and maintain performance longer.

A number of metrics have been proposed for measuring mental workload. In the literature, these metrics can be divided into two main groups: subjective and objective measures. Subjective metrics are based on an operator's opinions, answers to questionnaires, and interviews. A popular technique for the subjective assessment of an operator's mental workload is the NASA Task Load indeX (NASA-TLX) (Hart and Staveland, 1988). This method uses six dimensions: mental demand, physical demand, temporal demand, performance, frustration level, and effort, each with 10- or 20-point scale. An overall rating is then calculated as the weighted mean of all six ratings. One of the limitations of NASA-TLX is the lack of continuous measurement while the task is performed, since participants typically answer the survey questions after a task is completed and they may be unable to recall the workload experienced during a trial. The Air Traffic Workload Input Technique (ATWIT) (Stein, 1985) is less pruned to this problem. Although, it is a workload rating scale designed for use in air traffic control studies, it has been successfully applied in other domains (Loft et al., 2015). This technique uses a scale from 1 (low workload) to 7 (high workload), which is administered by freezing the simulation. At each freeze, participants are asked to report their level of workload. An advantage of using this technique is that it enables a more accurate evaluation, since the participant can report the workload as it changes, instead of waiting until the end of the task/scenario to report workload.

Objective measures are generally based on experimental methods used to collect physiological and/or behavioral information by a single sensor or a combination of different types of sensors, simultaneously (Debie et al., 2019). In contrast with subjective measures, objective techniques offer a continuous measure of workload in real time, and also their implementations do not interfere with the performance of the task at hand (Wang et al., 2015). In general, objective measures can be classified either as neurophysiological, physiological, or behavioral. Neurophysiological measures include electroencephalography (EEG) and functional near-infrared spectroscopy (fNIRS) (Hirshfield et al., 2009). Physiological measures include electrocardiography (ECG) (Veltman and Gaillard, 1996), heart rate and heart rate variability (HRV) (Elkin-Frankston et al., 2017), pupil dilation (Pomplun and Sunkara, 2003), blink frequency and blink duration (Tsai et al., 2007), and saccades (Ahlstrom and Friedman-Berg, 2006). Behavioral measures include keystroke dynamics, mouse tracking, and body positioning (Mota and Picard, 2003). Most objective measures (physiological and neurophysiological) rely on the assumption that changes in cognitive demands are reflected in the autonomic nervous system (ANS) (Mulder, 1989; Veltman and Gaillard, 1996). Although, physiological measures can be used as indicators of mental workload, neurophysiological techniques are considered the most direct indicators of different cognitive states (Debie et al., 2019).

There are two main techniques with appropriate temporal resolution to measure cognitive workload using brain signals: fNIRS and EEG. fNIRS measures cognitive workload by examining the levels of oxygenated (HbO) and deoxygenated (HbR) hemoglobin concentration in the cerebral cortex (Rojas et al., 2017b), and alertness, and indicative of loss of cortical arousal (Kamzanova et al., 2014). In this regard, fNIRS is commonly used to measure the amount of effort exerted in a given brain region in response to a given task. Different studies have reported that increased levels of HbO in the pre-frontal cortex correlates with increased task engagement which is used to indicate increased cognitive workload (Ayaz et al., 2012; Herff et al., 2014). On the other hand, EEG measures the brain's electrical activity and pattern analysis of this activity is used to indicate different levels of cognitive workload. Spectral analysis is

used to decompose EEG signals into their constituent frequency components. Typically, EEG data are partitioned into five bands (from slowest to fastest: delta, theta, alpha, beta, and gamma). The power spectral density (PSD) in each band is computed and used to compare the conditions being studied (i.e., low vs. high workload). EEG is considered the most popular approach in the literature to objectively assess cognitive states (Gevins et al., 1997; Abbass et al., 2014; Dong et al., 2016; Rojas et al., 2019a).

Although many indicators have been proposed in the literature, it is unclear which of the power bands and various indices is the most optimal for mental workload. In the following section, we present the most prominent indicators in the literature and their relationship with cognitive workload. The intent is not to provide an exhaustive literature review, but identify EEG metrics that could be potentially used as indicators of mental workload in our experiment.

1.1. EEG Indicators of Mental Workload

In the literature, EEG correlates of spectral powers at different cortical locations have been proposed for the assessment of cognitive workload. For example, theta band (4–8 Hz) has been linked to mental fatigue and mental workload (Gevins et al., 1995). Theta spectral power is thought to increase with increase demands on cognitive resources (Vidulich and Tsang, 2012; Xie et al., 2016), with higher task difficulty (Antonenko et al., 2010), and with increase of working memory (Borghini et al., 2012); particularly, theta power increases in tasks requiring a sustained concentration (Gevins and Smith, 2003). In addition, increase in theta power is related to lower mental vigilance and alertness, and indicative of loss of cortical arousal (Kamzanova et al., 2014). An increase in theta power monitored over the frontal cortex has been linked to an increase in task difficulty and use of higher memory resources (Parasuraman and Caggiano, 2002), frontal theta also increases during vigilance (Paus et al., 1997).

Alpha band (8–12 Hz) power has shown sensitivity to experiments in mental workload (Sterman and Mann, 1995; Xie et al., 2016; Puma et al., 2018), cognitive fatigue (Borghini et al., 2012), and also with reduction in attention or alertness (Kamzanova et al., 2014). In general, alpha band increases in relaxed states with eyes closed and decreases when the eyes are open (Antonenko et al., 2010). An increase in alpha power is related to lower mental vigilance and alertness (MacLean et al., 2012; Kamzanova et al., 2014) and therefore a decrease in the attention resources allocated to the task (Vidulich and Tsang, 2012). On the other hand, a progressive suppression of alpha waves has been linked to increasing levels of task difficulty (Mazher et al., 2017). Cortical areas that have been associated with alpha band changes are parietal and occipital areas (Dasari et al., 2017; Puma et al., 2018).

Beta band (12–30 Hz) has been linked to visual attention (Wróbel, 2000), short-term memory (Tallon-Baudry et al., 1999; Palva et al., 2011), and hypothesized to react to an increase in working memory (Spitzer and Haegens, 2017). An increase in beta power is associated with elevated mental workload levels during mental tasks (Coelli et al., 2015) and concentration (Kakkos et al., 2019). In addition, beta band activity reflects an arousal of the visual system during increased visual attention (Wróbel, 2000). An increase in beta activity has been observed in the parieto-occipital channels during visual working memory tasks (Mapelli and Özkurt, 2019).

In addition, the use of multiple EEG frequency bands (ratios or indices) has been proposed as an indicator of mental workload. This is based on the assumption that by combining information from multiple bands, the assessment of workload can be enhanced. For example, beta/(alpha + theta) (or Engagement Index, EI) has been used to study alertness and task engagement (Pope et al., 1995; Freeman et al., 1999; Mikulka et al., 2002), mental attention investment (MacLean et al., 2012), and mental effort (Smit et al., 2005). When alpha reduction was observed to correlate with increases in activity in frontal-parietal cortical areas, beta power increased while theta decreased, indicating a state of high vigilance (MacLean et al., 2012). When alpha reduction was seen to correlate with increases in activity in occipital and parietal areas, beta decreased and theta increased, indicating a state of drowsiness, or low vigilance (MacLean et al., 2012).

Another index used to explore the assessment of workload is the theta/alpha ratio (or Task Load Index, TLI). This index is based on the assumption that an increase of mental load is associated with a decrease in alpha power and an increase in theta power (Stipacek et al., 2003; Käthner et al., 2014). While an increased level of fatigue is related to increase of alpha and theta powers (Käthner et al., 2014; Xie et al., 2016). Research has shown that workload manipulations increased theta power at anterior frontal and frontal midline regions and decreased alpha power at parietal regions (Gevins and Smith, 2003). In general, an increase of cognitive workload has been associated with an increase of theta power together with a decrease of alpha power (Fairclough and Venables, 2004).

Theta/beta ratio has been used to study attention-deficit/hyperactitivty disorder (ADHD) and working memory problems in children (Lansbergen et al., 2011). This ratio shows increased theta power and decreased beta power during resting state in individuals with ADHD (Barry et al., 2003). Theta/beta ratio has been negatively correlated with mean reaction time in adults, indicating an increased theta/beta ratio linked to shorter, faster reaction time (Loo and Makeig, 2012). Theta/beta ratio has been used for monitoring sleepiness and wakefulness in car drivers (Sun et al., 2015). This ratio has been used to discriminate distraction from attentive driving as measured in the parietal lobe (Zhao et al., 2013). This index is based on the assumption that an increase in alertness and task engagement results in an increase in beta power and a decrease in theta power (Gale and Edwards, 1983). **Table 1** presents a summary of EEG indicators for the assessment of cognitive workload identified in the literature.

The present study was conducted to directly address the challenge to identify a set of indicators that can be used for the objective assessment of cognitive workload in a multitasking setting. Consequently, we have designed a simulation environment which affords manipulation of task complexity by varying the quality of information in the simulation. It has been shown that information quality affects

TABLE 1 | Summary of EEG correlates of spectral powers for the assessment of cognitive workload in the literature.

Indicator	Type of cognitive behavior	Description
Theta	Workload, vigilance, and concentration.	Theta spectral power is thought to increase with increase cognitive resources demand. Theta increases in tasks requiring a sustained focus of concentration and vigilance.
Alpha	Workload, cognitive fatigue, and attention.	Alpha band increases in relaxed states with eyes closed and decreases when the eyes are open. An increase in alpha power is related to lower mental vigilance and alertness.
Beta	Workload, visual attention, and concentration.	An increase in beta power is associated with elevated mental workload levels during mental tasks and concentration. Beta band activity reflects an arousal of the visual system during increased visual attention.
$\frac{Beta}{Alpha + Theta}$	Mental Effort, vigilance, and attention.	It has been used to study alertness and task engagement, mental attentional investment, and mental effort.
$\frac{Theta}{Alpha}$	Workload, mental effort.	This index is based in the assumption that an increase of mental load is associated with a decrease in alpha power and an increase in theta power.
$\frac{Theta}{Beta}$	Working memory, attention, and sleepiness.	This index is based in the assumption that an increases in alertness and task engagement result in an increase in beta power and a decrease in theta power.

cognitive workload (Young et al., 2016). Finally, we aim to identify EEG indices that may be used to trigger technological support to maintain performance.

2. METHODS

2.1. Participants

Sixteen participants (four females) were recruited. Their age ranged from 22 to 50 years old (mean age 33 ± 8.1 std). The experiment was approved by the University of New South Wales (UNSW) Research Ethics Committee (protocol ID: HC180554). All participants provided written informed consent prior to participating in the study. A demographics questionnaire was given to the participants before the start of the experiments. Participants did not receive monetary compensation for their participation in this study.

2.2. Description of the Experiment

Participants were seated on a fixed chair in front of a computer screen placed on a desk. An introduction to the experimental procedure and a practice session were provided to the participants before the start of the study. After that, the EEG head cap was mounted on the participants' head. To minimize any muscle movement artifacts, the participants were instructed to remain as still as possible while holding the mouse at all times during the experiment. Next, a 1 min baseline recording was obtained, in the first 30 s, the participants were told to close their eyes; then in the remaining 30 s, the participants were told to keep their eyes open and fixed on a point in the center of the screen. Finally, the participants were instructed to start the experiment after a 2-min break; the complete session lasted ∼50 min.

The experimental task was to teleoperate an unmanned aerial vehicle (UAV) to guide a swarm formation of autonomous unmanned ground vehicles (UGVs). Only the UAV remote-operator knows the destination defined by the mission profile. The UGVs consist of a group of four vehicles with capabilities to self-organize to autonomously maintain a formation during the mission. The operator's graphical user interface (presented in **Figure 1**) displays sufficient information to successfully guide the UAV with information display on the UAV (e.g., speed, altitude), mission state information, navigation map, and localization of the UGVs. This experiment is designed to run the simulation that combines four scenarios of different levels of information quality. Each scenario lasts 4 min and is repeated two times per participant. Simultaneously, EEG data and heart beat were recorded continuously during the experimental task. During each experimental condition, participants rated their mental effort using a computer version of the ATWIT questionnaire.

2.3. Simulation Environment

The experimental task is undertaken using the Virtual Battlespace Simulation 3 (VBS3) (Bohemia Interactive Simulations, Orlando Florida, USA) environment. The VBS3 software was used under the Australian Defence Force (ADF) Enterprise Licence Agreement with BISSimulation Australia. Information latency and loss were modeled to impact the operator's control station (as illustrated in **Figure 1**). This interface was programmed in C# (Microsoft Corporation, Washington, USA) since VBS3 does not have the capability for simulating information latency and loss as required. The interface has two main graphic displays located side by side on the top. On the left side, there is a lateral view of the UAV and UGVs' positions on a map. The UAV is presented by a green rectangle and the UGVs are visualized as blue rectangles. A blue star marks the UGVs' destination on the map. On the right side, real-time video streamed from the UAV camera is provided to the operator. At the bottom of the interface, detailed textual information on the UAV and UGVs' status including their positions, headings and speeds are provided. In the middle of the interface, a panel lists all possible UGV formation options; however, for this study we limit the formation to a boxing formation alone.

2.4. Experimental Design

A within-subject design with four different experimental conditions determined by the levels of quality of information was used in this study. The four experimental conditions (scenarios) are: (1) low latency/delay and low dropout; (2) low delay and high dropout; (3) high delay and low dropout; and (4) high

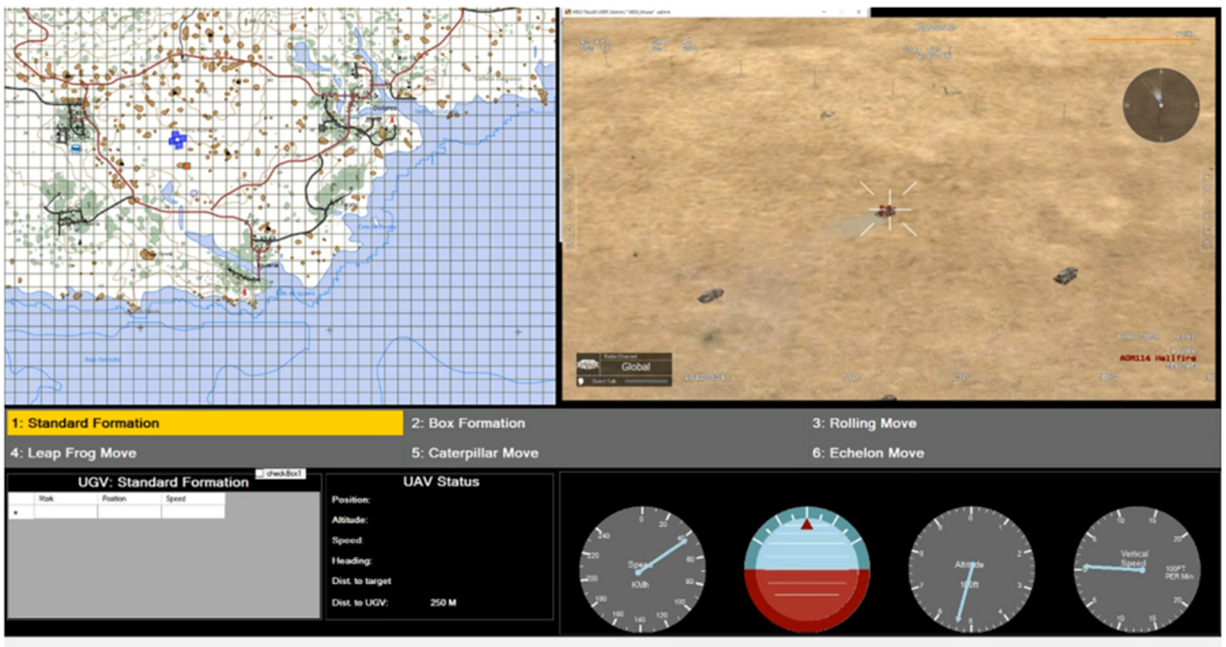

FIGURE 1 | UAV pilot interface.

delay and high dropout. The experiment is counterbalanced by using the composite 3 × 3 Latin Square design to avoid confounding due to order effects. In our experiment, information latency is the amount of time a video frame from the UAV camera and the status of all vehicles to traverse in the camera' s field of view are delayed to the interface; while, information loss is the rate in which video frames and data about the status of vehicles are not transmitted during data transmission. Ideally, information latency should be unnoticeable to the UAV operator and the delivery of information should be operationally assured.

However, to modulate the participants' perceived workload, information latency and information loss are injected into the simulation. Thus, it has been hypothesized that the latency and loss of information affect the subjects' perceived cognitive workload. Information latency and information loss are modeled using two parameters, d for the delay time (Low $d = 1$ s, High $d = 9$ s) of information transmission, and lf for the number of video frames lost per second (Low $lf = 1$ s, High $lf = 9$ s) in transmission. **Table 2** lists the parameter values corresponding to the corresponding levels of information latency and loss, respectively.

TABLE 2 | Variables used in information latency and loss.

Variable	Level	Parameter value
Information latency	Low	$d = 1$ s
	High	$d = 9$ s
Information loss	Low	$l_f = 1$ s
	High	$l_f = 9$ s

2.5. Heart Rate Measurement

A mouse (Mionix Naos QG) equipped with heart rate (HR) and galvanic skin response (GSR) sensors is used as the main input method during the simulation. The biometric sensors are designed to measure the physiological data on the palm of the user; thus, the user must maintain the mouse in their hand at all times while the simulation is running. The mouse uses a sample rate of eight samples per second. In addition, the mouse can also record different mouse metrics, such as: number of scrolls, clicks, and movements. In this study, the heart rate information is only used to corroborate the design of the experimental conditions due to its high sensitivity to mental load measure (Cinaz et al., 2010).

2.6. Electroencephalographic (EEG) Measurement

A wireless EEG acquisition system (Emotiv EPOC) was used to record neural activity. This device has a resolution of 14 channels (plus 2 reference channels) with a sampling frequency of 128 samples per second. Some advantages of using the Emotiv EPOC is its low cost, good signal-to-noise ratio, and ease of use (Duvinage et al., 2013). In addition, the EPOC has shown satisfactory results in diverse research studies in emotion recognition (Ramirez and Vamvakousis, 2012), brain computer interface (Holewa and Nawrocka, 2014), and cognitive workload (Lim et al., 2015). **Figure 2** presents the headset and the channel positions based on the international 10–20 EEG system of electrode placement. Channel locations

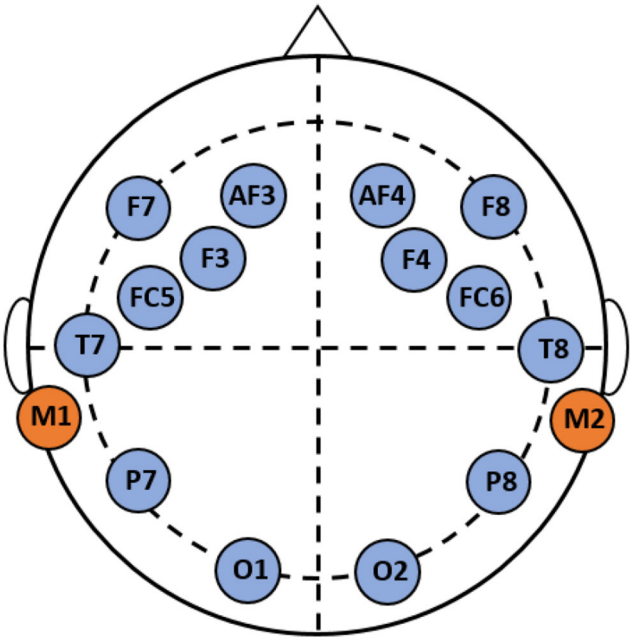

FIGURE 2 | Cortical areas covered by the electrodes of the EEG EMOTIV Epoc.

correspond to: AF3, F7, F3, FC5, T7, P7, O1, O2, P8, T8, FC6, F4, F8, AF4, M1, and M2. M1 is used as the ground reference channel for measuring the voltage of the other channels, while M2 is used as a feed-forward reference point to reduce external electrical interference (Badcock et al., 2015). A saline solution was employed to reduce the electrode impedance and facilitate sensitivity between each electrode and the scalp (Duvinage et al., 2013).

2.6.1. EEG Pre-processing

EEG pre-processing was performed in Matlab (version 2018b, The MathWorks Inc.) by using custom software and the EEGLab toolbox (Delorme and Makeig, 2004). Baseline correction was performed by subtracting the corresponding mean from a pretrial (200ms) period from each channel. Then, EEG signals were band-pass filtered between 2 and 43 Hz using a FIR filter, which helps remove high-frequency artifacts and low-frequency drifts. Electrode movement artifacts were manually removed from the data; these artifacts produce large spikes that are several orders of magnitude bigger than the neural response produced by EEG. Artifacts from eye blinks and movements were corrected using the multiple artifact rejection algorithm (MARA) which evaluates ICA-derived components (Winkler et al., 2014).

2.6.2. Feature Extraction

Feature extraction was carried out using spectral analysis. First, the power distribution from each channel was studied by transforming the EEG into power spectral density (PSD) using a fast-Fourier transform (FFT) and using 10-s windows with 50% overlapping windows multiplied by the Hamming function to reduce spectral leakage (Chaouachi et al., 2011). Second, from each window, the EEG channels were decomposed into sub-bands: delta (2–4 Hz), theta (4–8 Hz), alpha (8–12 Hz), beta (12–30 Hz), and gamma (30–40 Hz). Third, the PSD results of each frequency band were normalized (1/f) to obtain the relative PSD of each band to the baseline time period. This normalization helps to make quantitative comparisons of power across frequency bands (Cohen, 2014). Finally, the resulting PSD values in each band were averaged to obtain the power spectral features used for classification.

2.6.3. Feature Selection

Feature selection was carried out to reduce the number of features and build a more accurate learning model. The selection criteria was based on the joint mutual information algorithm (JMI), this method ranks the features with the largest mutual information (MI) that produces most of the MI between the feature vector and the class label (Yang and Moody, 1999). The reason to choose JMI is that it presents better tradeoff in terms of accuracy, stability, and flexibility than other ranking methods (Brown et al., 2012; Rojas et al., 2019b). A disadvantage of this method is the fact that there is no stopping criteria to reach the best subset of features, and the user needs to select the number of features from the ranking list to form the optimal subset.

2.6.4. Classification

The classification task is to determine the level of mental effort based on the recorded EEG signals from each participant. To identify the four levels of mental effort, we used the linear discriminant analysis (LDA) algorithm for offline analysis. The reason to choose LDA is because it is the most popular classifier in brain computer interface (BCI) research due to its good performance and low computational cost, attributes needed for the development of an on-line assessment of mental effort in our future work. To measure the classifier's performance, the data was divided into two parts with 70% for training and the remaining 30% used for testing and to report generalization performance. *k*-fold cross validation ($k = 10$) was performed on the training set; the training set was randomly divided into *k* partitions. Then, *k*-1 partitions are used to fit the learning model and the remaining partition used to validate the model, this process is repeated *k* times, and each time using a different partition to validate the model. The final generalization results are presented as the average and standard deviation on the 30% untouched test set.

2.7. Validation of Experimental Design

In order to validate the experimental conditions, the response to the ATWIT questionnaire and the heart rate information were analyzed. The research hypothesis of this study is that different levels in the quality of information (delay and dropout) significantly affect the perceived mental effort of the participants during the experimental task. In order to corroborate the research hypothesis and the design of this experiment, a repeated measures model was used to appraise statistical difference for the four different experimental conditions. Therefore, it was expected that the level of mental effort in each condition is significantly different and this difference can be observed by the subjective

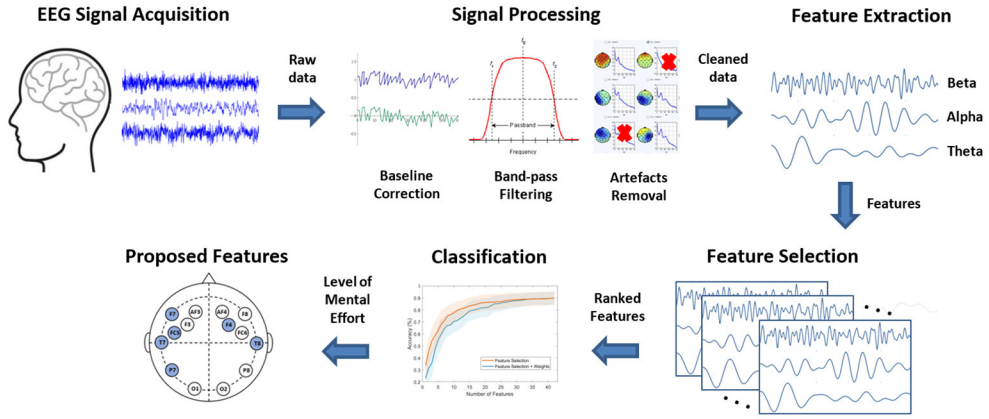

FIGURE 3 | Analysis workflow used in the present study.

and objective metrics. A p value that is >0.05 was not considered statistically significant.

ATWIT scores and heart rate data were tested for normality using the Shapiro-Wilk test. Both tests showed that the data significantly deviated from a normal distribution, $p = 0.01$ for ATWIT scores and $p = 0.033$ for heart rate. Then, a logarithmic transformation was applied to reinforce the linearity of both data, which resulted in meeting the normality assumption ($p > 0.05$) after a subsequent normality test for both data. However, after checking normality visually using Q–Q plots (quantile–quantile plots), the distribution of both data was non-normal. Therefore, the non-parametric Friedman test was applied to both data for testing the difference between experimental conditions and Wilcoxon signed ranks test as *post-hoc* test.

Figure 3 illustrates a summary of the analysis workflow used to obtain the results presented in this study. First, acquired EEG signals are cleaned through a series of signal processing techniques, then decomposition (feature extraction) of EEG signals into sub-bands (beta, alpha, theta) is carried out. The obtained features from each participant are then ranked using a feature selection technique. Each rank is then evaluated using an LDA classifier. Finally, the list of most prominent features contributing to the accuracy of the classifier are identified.

3. RESULTS

3.1. Validation of Experimental Conditions

Two methods were used to evaluate the experimental design. First, the subjective workload assessment using ATWIT scores was evaluated for each experimental condition. Second, heart rate (HR) is used to corroborate the cognitive modulation with respect to each condition. The experimental assumption of this study is that in conditions with low quality communication (e.g., high delay and high dropout condition), the participants' perceived workload will be significantly different than in conditions with high quality communication (e.g., low delay and low dropout).

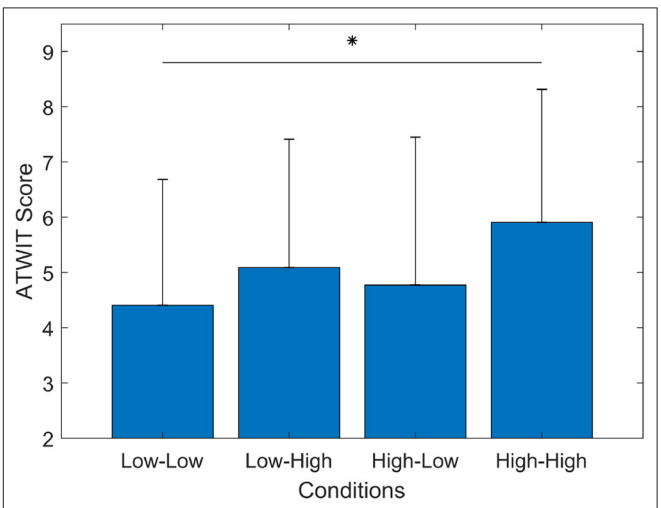

FIGURE 4 | The bar graph represents the mean and standard deviation of ATWIT scores. The Wilcoxon test showed a significant increase of workload ($p = 0.002$). $^*p < 0.0083$.

3.1.1. ATWIT Scores

Figure 4 shows the results of the subjective workload evaluation using ATWIT test. The recorded ATWIT response for each condition was averaged among the participants. The overall trend of subjects' perceived workload showed the lowest workload in the Low-Low condition ($mean = 4.4, std = 2.2$), medium workload in the Low-High ($mean = 5.0, std = 2.3$) and High-Low ($mean = 4.7, std = 2.7$) conditions, and the highest workload in the High-High ($mean = 5.9, std = 2, 4$) condition. Overall, ATWIT scores showed an increase of perceived workload in experimental conditions with low quality communication compared to experimental conditions with high quality of communication.

A Friedman test of differences among repeated measures was carried out to examine changes in ATWIT scores under the four conditions. This test was used with the following research

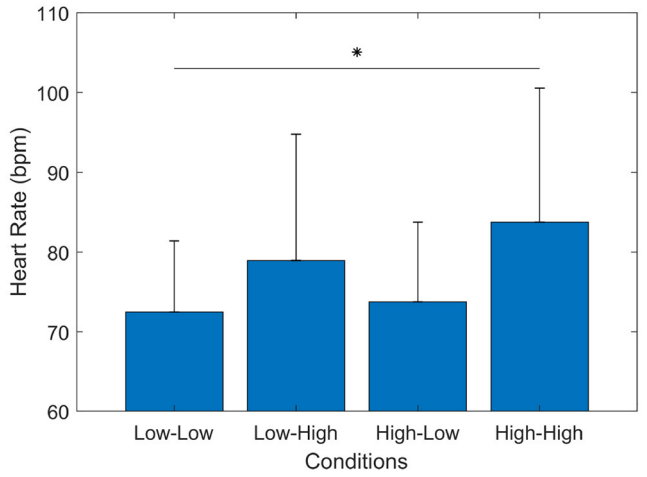

FIGURE 5 | Subjects' heart rate (HR) in beats per minute (bpm). The bar graph represents the mean and the standard deviation of participants' HR during the four experimental conditions. The Wilcoxon test showed a significant increase between High-High and Low-Low ($p = 0.003$) conditions. *$p < 0.0083$.

hypothesis *Ho: There are no significant differences between the mean ATWIT scores among the experimental conditions.* In other words, the distribution of the answer to the ATWIT questionnaire is independent of the experimental condition (no difference in perceived workload). A statistically significant difference in perceived workload depending on the experimental conditions $[\chi^2(n = 16) = 10.471, p = 0.015]$ was obtained. *Post-hoc* tests using multiple two-sided Wilcoxon signed-rank tests were performed with Bonferroni correction applied, resulting in a significance level set at $p < 0.0083$. There were no significant differences between the Low-Low and Low-High ($p = 0.178$), the Low-Low and High-Low ($p = 0.502$), the High-Low and Low-High ($p = 0.303$), the High-High and Low-High ($p = 0.025$), or the High-High and High-Low ($p = 0.011$) conditions. However, this statistical test showed a significant increase ($p = 0.002$) in perceived workload as declared in the ATWIT scores by the participants in the Low-Low and High-High scenarios.

3.1.2. Heart Rate Information

Another metric used to validate the experimental design was the participants' heart rate (HR). **Figure 5** presents the results of the heart rate value between the four different conditions in the experiment. Heart rate has been shown to be a physiological indicator directly related to mental workload (Luque-Casado et al., 2016). In this case, the experimental assumption (refer to section methods) was that delay and dropout of information affect the cognitive workload of the participants and this can be observed by measuring the participants' heart rate. Overall, the results showed that during the Low-Low condition the participants exhibited the lowest HR ($mean = 72.47, std = 8.9$), medium HR during the Low-High ($mean = 78.94, std = 15.83$) and High-Low ($mean = 73.76, std = 9.9$), and the highest HR ($mean = 83.73, std = 16.8$) during the High-High condition.

TABLE 3 | Reference values for classification accuracy and standard deviation (std) using LDA.

	Power bands			Ratios		
	Theta	**Alpha**	**Beta**	**Theta/ Beta**	**Beta/ (Alpha + Theta)**	**Theta/ Alpha**
Accuracy	60.28	53.13	69.89	55.50	56.44	49.10
Std (±)	8.16	10.12	6.48	8.51	8.36	5.92

Results are presented in percentage.

A Friedman test was carried out to examine changes in heart rate under the four conditions. The Friedman test on the heart rate information revealed a significant difference among the scenarios $[\chi^2(n = 16) = 15.60, p = 0.001]$. *Post-hoc* tests using multiple two-sided Wilcoxon signed-rank test with Bonferroni correction applied showed that the participants' heart rate in High-High conditions increased statistically significant ($p < 0.0083$) compared to the heart rate in Low-Low ($p = 0.003$), while in the other conditions there were no significant differences between the Low-Low and Low-High ($p = 0.039$), the Low-Low and High-Low ($p = 0.408$), the High-Low and Low-High ($p = 0.079$), the High-High and Low-High ($p = 0.023$), or the High-High and High-Low ($p = 0.01$) conditions. These results showed that in experimental conditions with low quality communication (e.g., High-High) the participants' heart rate increased significantly, which also suggest an increase in cognitive workload as a result.

These two results (ATWIT and HR) validate the assumption that the different cognitive demands are affected due to the experimental conditions. In addition, high delay and high dropout exhibited the most significant increase in ATWIT score and heart rate, which suggests that this experimental condition induced the highest cognitive demand in the experiment. On the other hand, experimental conditions with low delay and low dropout exhibited the lowest ATWIT scores and lowest heart rate, which suggest that it produced the least cognitive demand in the experiment.

3.2. Evaluation of EEG Indicators

Based on the indicators identified in the literature, this study explores the classification performance using only three frequency bands: Theta (4–7.5 Hz), Alpha (8–12 Hz), and Beta (13–35 Hz).

3.2.1. Reference Values

First, the indicators are investigated separately to obtain a reference performance value. **Table 3** presents the grand average results from the classification task using each indicator. The results represent the classification accuracy (in percentage %) and standard deviation using each indicator separately. Each indicator was obtained from each channel. The highest overall accuracy ($75.99 \pm 6.48\%$) was achieved using the Beta band, while the lowest accuracy (49.10 ± 5.92) was obtained with Alpha band only.

3.2.2. Feature Selection Evaluation

Second, using a feature selection method based on mutual information, to identify any kind of statistical dependency between variables, a subset of indicators was obtained. In this step, only the power bands (Theta, Alpha, Beta) were used in the feature selection process to avoid introducing correlated variables into the subset of indicators. **Figure 6** presents the correlation analysis of all the indicators. In this figure, it is possible to observe that all the ratios (e.g., theta/beta) are highly correlated with the power bands (e.g., beta, alpha). Therefore, removing these variables from the analysis will make the feature selection more efficient.

The objective of feature selection is to find a good representation of the data, improve estimators' performance by reducing the dimensionality of the data and eliminating redundant and irrelevant data from each participant's data (Rojas et al., 2019b). After applying joint mutual information (JMI), the features were ranked according to the their relevance to the class label. This process returned 16 ranks (one rank per participant), each rank contains the ranking of 42 features (e.g., 3 indicators * 14 channels = 42 features). These ranks represent the importance of each indicator and channel with respect to the class label from each participant. However, a limitation of this process is that each rank is different from one another, which complicates direct comparisons between participants.

In order to obtain a subset of common features that potentially describe most of the data for all the participants, the frequency of appearances and ranking position were considered for each feature and for each participant. To achieve that, a new list containing the top 10 features from each rank were chosen (160 features in total). Then, the number of appearances of each feature in the list was counted and a weight [from 1 (most important) to 0.1 (least important)] for its position in each rank was given. For example, a feature (Theta in T8) appeared two times in the list (i.e., this feature was in the top 10 features only in two participants), in the first rank it appeared in the first position ($weight = 1.0$) and in the second rank in the seventh position ($weight = 0.4$); thus, its total value is 1.4 (please refer to **Figure 7**).

The complete results obtained during the weight process mentioned above is presented in **Figure 7**. Beta band in channel T7 (BetaT7) exhibited the highest value from this list, it suggests that this feature is the most important feature in our sample population. On the other hand, features from channels O1, FC6, and AF4 in the Theta and Alpha bands showed the lowest value from this list, which suggest that these features are less important among the 16 ranks. Based on this frequency of appearance, a common set of features can be obtained across the participants, which represents the most relevant features in the data set.

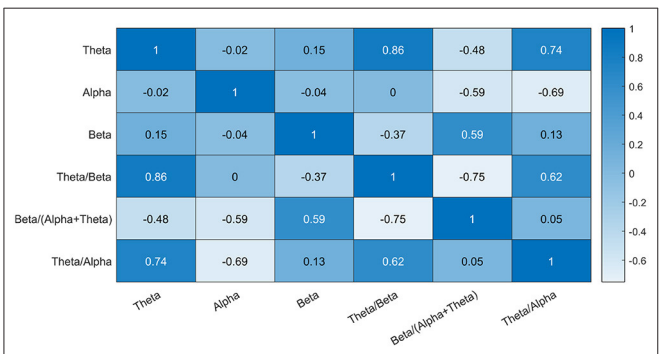

FIGURE 6 | Strength and direction of correlations among the EEG features.

3.2.3. Classification Results

Using both ranks, after feature selection (FS) and after application of weights (FS + weights), classification was carried out to obtain performance results and compare these with the obtained reference values (refer to section 3.2.1). **Figure 8** presents the classification results of both methods. FS presented the highest accuracy (89.84 ± 5.60%) using the top 40 features (in total, 42 features), while the FS + weights method presented slightly lower accuracy (89.43 ± 5.47%) using the same number of features. It is to note, that using the top 10 features from both methods produced a higher accuracy (77.57 ± 8.39% for FS, and 70.60 ± 8.98% for FS + weights) than the highest reference accuracy value (69.89 ± 6.48) using any of the frequency bands and ratios separately, which was achieved using the 14 channels in the Beta band. Overall, both methods showed comparable results, which suggests that the group of common features across our sample population can be used as indicators of cognitive load in our experiment.

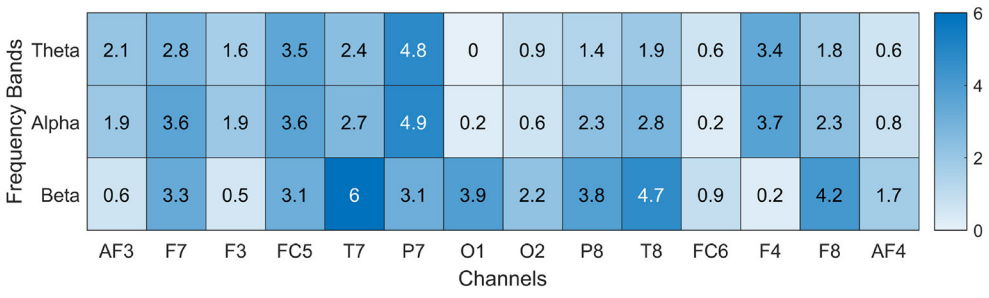

FIGURE 7 | Frequency of appearance of each indicator. Each indicator was weighted according to its ranking (e.g., rank = 1 weight = 1, rank = 2 weight = 0.9) and number of occurrences in the list.

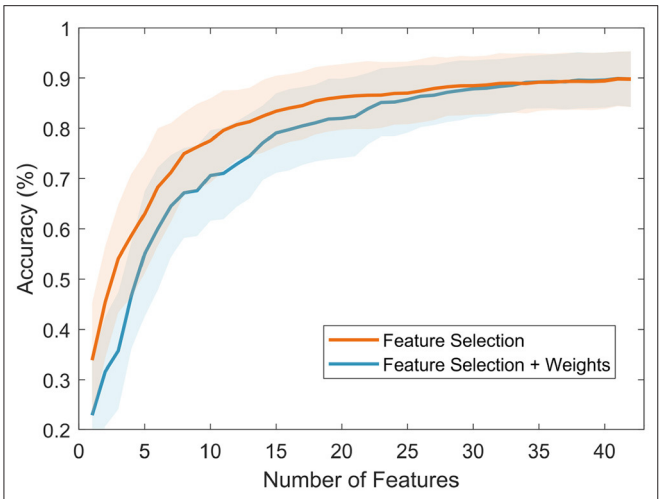

FIGURE 8 | Classification results using the features ranked after the feature selection (using JMI) process (in orange) and after the feature selection + weights process (in blue).

However, in order to identify the appropriate number of features to be used as indicators of workload in our experiment, a stopping criterion was introduced. This criterion is based on comparing consecutive classification results using two-sample *t*-tests. Using this method, the searching process is stopped when three consecutive non-statistically significant results were obtained. The final number of features is the one that produced the first non-statistically significant result. After this step, the appropriate number of features is 18 for FS (85.44 ± 6.70%) and 19 for FS + weights (82.83 ± 8.01%). These top 19 features are presented in **Table 4**.

3.2.4. Activated Cortical Areas

The majority of features from the identified subset (top 19 features) are from the Beta band and the frontal area. **Figure 9** presents the cortical location of each feature with respect to their frequency band (Theta, Alpha, Beta). Three channels (F4, F7, FC5) from the frontal area, one channel from the temporal area (T7) and one channel (P7) from the parietal area were obtained in the Theta band. In the Alpha band seven channels were identified, the same three channels in the frontal area (F4, F7, FC5), bilateral activation in the temporal area (T7, T8), and one channel (P7) from the parietal area. The Beta band exhibited the largest number of channel within the top 19 features in four cortical locations, in the frontal (F7, FC5, F8), bilateral activation in both the temporal (P7, P8) and parietal (P7, P8) cortex, and in the occipital area (O1). Another interesting finding is that most of the features in the top 19 corresponded to the left hemisphere.

3.2.5. Evaluation of the Weight Process

Two more filter feature selection methods were used to evaluate the weight process to capture the most common features across the sample population. These two techniques are Information Gain (InfoGain) and student's *t*-test, their criteria to rank each feature are entropy and statistical based (Novaković, 2016), respectively. These two feature selection techniques were implemented and a group of 16 different ranks (i.e., one rank per subject) was obtained from each technique. Then, the weight process was applied to each technique separately using the top 10 features from each participant. Please refer to section 3.2.2 for a more detailed description.

TABLE 4 | Top 19 features after feature selection and weight procedure (FS + weights).

Ranking	Channel	Band	Ranking	Channel	Band
1	T7	Beta	11	FC5	Theta
2	P7	Alpha	12	F4	Theta
3	P7	Theta	13	F7	Beta
4	T8	Beta	14	FC5	Beta
5	F8	Beta	15	P7	Beta
6	O1	Beta	16	F7	Theta
7	P8	Beta	17	T8	Alpha
8	F4	Alpha	18	T7	Alpha
9	FC5	Alpha	19	T7	Theta
10	F7	Alpha			

Figure 10 presents the classification results of both techniques using LDA. It was expected that each technique will produce different rankings and different classification results. This is mainly due to the different ranking strategies followed by different techniques. In addition, similar to the JMI technique, InfoGain and *t*-test lack a stopping criterion to obtain the best feature subset; therefore, three consecutive non-statistically significant results were used to stop the searching process for each method. For the InfoGain method, the stopping criterion led to 16 features (85.06 ± 6.04%) and 19 for InfoGain + weights (83.36 ± 6.63%). For the *t*-test method the stopping criterion led to 17 features (84.40 ± 6.74%) and 27 for *t*-test + weights (85.77 ± 5.38%).

These results highlight that the weight process captures the most common features among the sample population. Introducing the weight process after feature selection allows the classifier to maintain a comparable performance than the reliance on individual rankings for each participant. Therefore, the use of common features not only facilitates making comparisons across subjects but also reduces the complexity of the analysis by focusing on a smaller set of features.

3.3. Sensitivity of EEG Indicators

In order to examine the sensitivity of the proposed set of EEG indicators to differentiate between the four experimental conditions (i.e., four levels of workload), a test for differences was conducted using the Friedman Test. This test was used with the following research hypothesis *Ho: There are no significant differences between the mean EEG values among the experimental conditions.* In other words, the distribution of EEG values is independent of the experimental conditions (the EEG indicators do not capture a difference in workload). **Figure 11** presents the results of this test.

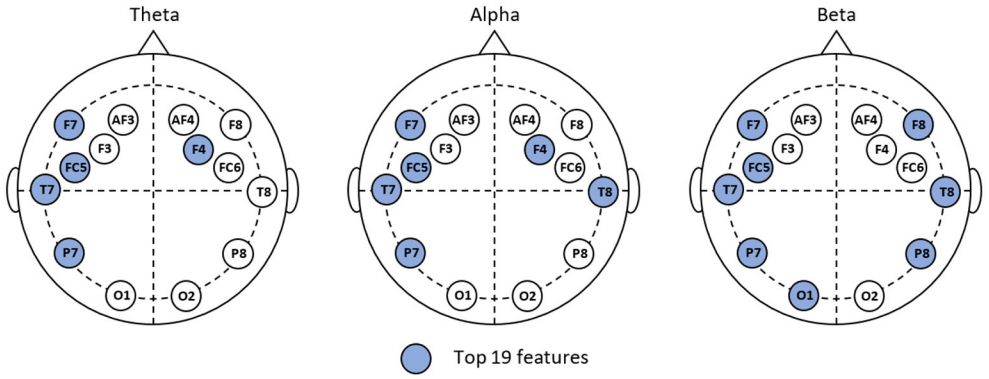

FIGURE 9 | Cortical locations of the top 19 features in the three bands explored in this study.

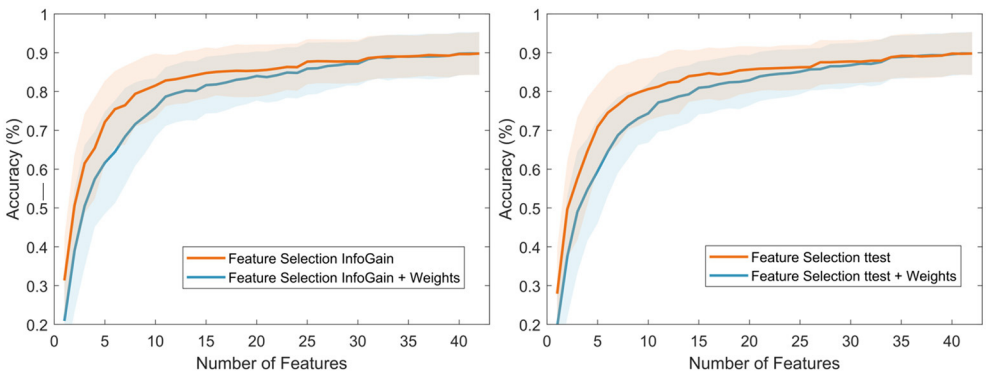

FIGURE 10 | Classification results using Information Gain **(left panel)** and *t*-test **(right panel)**.

The results exhibited a statistically significant difference in EEG values depending on the experimental conditions [$\chi^2(n = 16) = 31.27, p = 0.000$]. *Post-hoc* tests using multiple two-sided Wilcoxon signed-rank tests were performed with Bonferroni correction applied ($p < 0.0083$). There were no significant differences between the Low-Low and Low-High ($p = 0.030$), the Low-Low and High-Low ($p = 0.01$), or the High-Low and Low-High ($p = 0.026$) conditions. However, this statistical test showed a significant difference in the Low-Low and High-High ($p = 0.000$), the High-High and Low-High ($p = 0.005$), and the High-High and High-Low ($p = 0.000$) conditions. This result suggests that using the proposed set of EEG features presents higher sensitivity to measure cognitive load during our experiment, than the ATWIT questionnaire and the heart rate.

4. DISCUSSIONS

The primary goal of the current investigation was to examine different EEG indicators for the objective assessment of cognitive workload. An experiment was designed to modulate the participants' perceived workload. EEG indicators of spectral powers at different cortical locations (based on theta, alpha, and beta bands) were compared and investigated. Using a feature selection technique, the most important features were obtained for each subject, then a weight procedure was applied to identify a set of common features across our sample population. The identified set of features represents a group of possible EEG indicators for the objective assessment of cognitive workload.

The experimental conditions and overall assumption of the experiment were validated. The research hypothesis about the use of delay and dropout to modulate the participant's perceived workload was confirmed by using statistical analysis performed on both the averaged ATWIT response and heart rate (HR) data. The overall trend exhibited that the participants faced a significantly higher ($p < 0.0083$) cognitive workload during the high delay and high dropout (High-High), a similar trend was also observed using the EEG indicators. This finding suggests that the increase of participants' cognitive workload in scenarios with high delay and high dropout reflects the difficulty in understanding and identifying new information after the loss of an already-familiar scenario. This is in line with previous studies on the relationship between information quality and workload. For instance, in an experiment to study the effect of audio communication latency on cognitive workload (Krausman, 2013), it was found that increased audio communication latency led to increased cognitive workload and lower task accuracy. Similarly, increased workload has been reported in participants

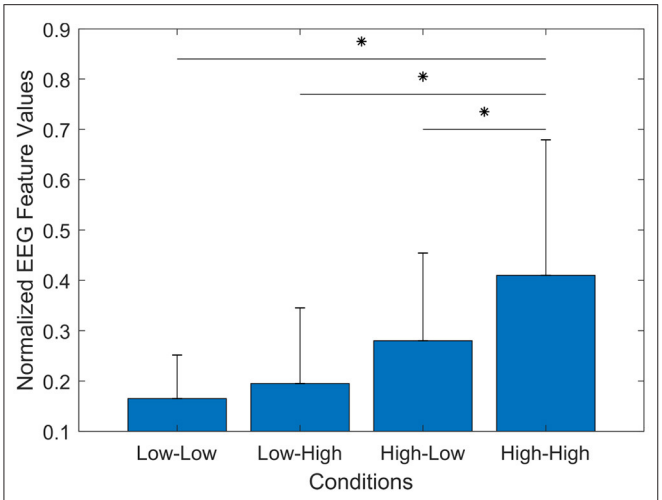

FIGURE 11 | The bar graph represents the mean and standard deviation of the proposed set of EEG features. The Wilcoxon test showed a significant increase of workload the Low-Low and High-High ($p = 0.000$), the High-High and Low-High ($p = 0.005$), and the High-High and High-Low ($p = 0.000$) conditions. $^*p < 0.0083$.

after the use of automation in teleoperated systems, where participants face new information after the use of automation to complete a task (Chen et al., 2017).

The weighting process after feature selection (FS + weight) helped obtain a common set of features across our sample population. In the machine learning and the data mining literature, feature selection is an important preprocessing step in regression and classification problems (Vergara and Estévez, 2014). An advantage of using feature selection in comparison with other dimensionality reduction methods (e.g., PCA) is that feature selection does not alter or transform the data; thus, attempting to understand the underlying process that produced a given classification result can be achieved (Bennasar et al., 2015). In our experiment, although feature selection was used to determine the most important features for each subject and also to identify the irrelevant features to be discarded, it produced sixteen different rankings that made it difficult to deduce a common set of features. Thus, the weighting process helped determine a common set of features by using the individual rankings of each feature from each participant. The resulting set (**Table 4**) represents the most frequent features in the complete feature set. It is worth mentioning that by using the ranked features according to their relevance to the class label, the weight process retains useful intrinsic groups of interdependent features, which helped avoid redundant and irrelevant features in the FS process. This common set of features (top 19) represents less than half ($\sim$ 45%) of the total number of features.

The common set of features showed objective confirmation of the different levels of perceived workload during the classification task. The classification task exhibited a much better performance (82.23%) using the top-19 features than any of the reference values (**Table 3**) using each indicator separately. In addition, the obtained feature set represents a combination of well-known EEG power bands that have been linked to cognitive workload as identified in our literature review. These frequency bands (theta, alpha, beta) are generally associated with a different dimension of workload (e.g., attention, vigilance, or mental fatigue). For instance, theta band has been successfully used to study mental fatigue and alertness (Gevins et al., 1995; Kamzanova et al., 2014), alpha band has been employed to assess mental vigilance, attention and alertness (Antonenko et al., 2010; Borghini et al., 2012; MacLean et al., 2012), while beta band has been used to study visual attention or short-term memory (Tallon-Baudry et al., 1999; Wróbel, 2000; Palva et al., 2011). Therefore, using a combination multiple frequency bands will make the assessment of workload more robust to other intrinsic cognitive processes that are carried out simultaneously. This is particular important, since our experiment reflects a multitasking environment where different dimensions are present at the same time, e.g., navigation while maintaining orientation, or planning while maintaining communication distance with the alpha vehicle.

The identified feature set also helped identify the most relevant cortical areas associated with the assessment of cognitive workload in our experimental task. The majority of identified channels are from the frontal, temporal, and parietal regions, cortical areas that have been associated to cognitive workload in previous studies. For instance, increase in theta band power over the frontal cortex has been associated with an increase in task difficulty and use of more working memory resources (Parasuraman and Caggiano, 2002). Suppression of Alpha power has been observed in the parietal and occipital areas during increase of mental workload (Mazher et al., 2017; Puma et al., 2018). Increase in Beta power over the parietal and occipital cortical regions has been observed during visual working memory tasks (Mapelli and Özkurt, 2019). In addition, bilateral activation was identified in the beta band. These activations were found in the frontal (F7 and F8), temporal (T7 and T8), and parietal (P7 and P8) areas. While in the alpha band, a bilateral activation was only found in temporal areas (T7 and T8). The observed bilateral activation in different cortical areas suggests that there is is no single brain region or hemisphere that solely responds to mental workload. In addition, as many other cognitive tasks, the brain functions as a system rather than separated brain areas working independently (Rojas et al., 2016).

We acknowledge that this study presents some limitations that should be addressed in our future research. The use of a small number of electrodes to monitor the cortical activity restricts our ability to make generalizations to other cerebral regions from the proposed set of EEG features. Advantages of using the Emotive EPOC is that it is less uncomfortable to be worn for longer periods of time and less unpleasant for participants since it uses dry electrodes. However, in future research a larger number of electrodes to record activity in more areas of the cerebral cortex should be considered. Another limitation of this study is that each sensor modality was analyzed separately to study workload. Debie et al. (2019) highlighted the disadvantages of using a single sensor modality to capture changes in cognitive workload. For instance, a given measure may respond to a particular task (e.g., attention or engagement) but may fail to capture workload change in other tasks (e.g., working memory or

mental fatigue). Thus, combining multiple sensors can measure different aspects of workload and can potentially complement one another to provide a better assessment of cognitive workload in multitasking situations.

Finally, the results of this study expand earlier findings from previous research of cognitive workload assessment using EEG. However, direct comparisons with other studies are difficult because of the use of different experimental conditions, EEG acquisition system, sampled population and with different demographics, validation methods, and classification models (Rojas et al., 2017a). Therefore, the contributions of this study can be summarized as follows: (1) it offers an exploratory study that aims to compare different EEG indicators identified in the literature for the objective assessment of cognitive workload; (2) it introduces a framework to extend the feature selection process to identify the most important features among the sample population; and (3) it presents a group of features (EEG power bands and cortical regions) as possible indicators for the objective assessment of cognitive workload in multitasking environments.

5. CONCLUSIONS

This study investigated different EEG power bands to identify a set of indicators that can be used for the objective assessment of cognitive workload. Results showed that our experimental study was valid at increasing mental workload in the participants as measured by three metrics (ATWIT, HR, and EEG). The use of a weighting process after feature selection (FS + weights) helped identify common features across all participants. In addition, a set of indicators (including EEG power bands and cortical regions) was identified as objective metric of workload in our multitasking environment. The proposed set of indicators exhibited higher sensitivity to various levels of cognitive workload than the subjective metric (ATWIT) and the physiological measure (heart rate). Finally, future research will adopt the proposed EEG indicators to trigger adaptive automation to maintain performance in human-swarm teaming.

DATA AVAILABILITY STATEMENT

The datasets for this article are not publicly available due to confidentiality requirements with our funder. Requests to access the datasets should be directed to Hussein Abbass: h.abbass@unsw.edu.au.

ETHICS STATEMENT

The studies involving human participants were reviewed and approved by University of New South Wales (UNSW) Research Ethics Committee. The patients/participants provided their written informed consent to participate in this study.

AUTHOR CONTRIBUTIONS

RF and ED carried out the experiment. ED implemented the simulation environment. RF analyzed the data and wrote the manuscript in consultation with HA. HA conceived the study and was in charge of overall direction and planning. JF, MB, KK, SA, and MG offered domain knowledge in the design of the system, discussed the results, and commented on the manuscript.

FUNDING

The Commonwealth of Australia supported this research through the Australian Army and a Defence Science Partnerships agreement with the Defence Science and Technology Group, as part of the Human Performance Research Network.

REFERENCES

Abbass, H. A., Tang, J., Amin, R., Ellejmi, M., and Kirby, S. (2014). "Augmented cognition using real-time eeg-based adaptive strategies for air traffic control," in *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, Vol. 58 (Los Angeles, CA: SAGE Publications Sage CA), 230–234.

Ahlstrom, U., and Friedman-Berg, F. J. (2006). Using eye movement activity as a correlate of cognitive workload. *Int. J. Ind. Ergonom.* 36, 623–636. doi: 10.1016/j.ergon.2006.04.002

Antonenko, P., Paas, F., Grabner, R., and Van Gog, T. (2010). Using electroencephalography to measure cognitive load. *Educ. Psychol. Rev.* 22, 425–438. doi: 10.1007/s10648-010-9130-y

Ayaz, H., Shewokis, P. A., Bunce, S., Izzetoglu, K., Willems, B., and Onaral, B. (2012). Optical brain monitoring for operator training and mental workload assessment. *Neuroimage* 59, 36–47. doi: 10.1016/j.neuroimage.2011.06.023

Badcock, N. A., Preece, K. A., de Wit, B., Glenn, K., Fieder, N., Thie, J., et al. (2015). Validation of the emotiv EPOC EEG system for research quality auditory event-related potentials in children. *PeerJ* 3:e907. doi: 10.7717/peerj.907

Barry, R. J., Clarke, A. R., and Johnstone, S. J. (2003). A review of electrophysiology in attention-deficit/hyperactivity disorder: I. Qualitative and quantitative electroencephalography. *Clin. Neurophysiol.* 114, 171–183. doi: 10.1016/S1388-2457(02)00362-0

Bennasar, M., Hicks, Y., and Setchi, R. (2015). Feature selection using joint mutual information maximisation. *Expert Syst. Appl.* 42, 8520–8532. doi: 10.1016/j.eswa.2015.07.007

Borghini, G., Vecchiato, G., Toppi, J., Astolfi, L., Maglione, A., Isabella, R., et al. (2012). "Assessment of mental fatigue during car driving by using high resolution EEG activity and neurophysiologic indices," in *2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society* (San Diego, CA: IEEE), 6442–6445.

Brown, G., Pocock, A., Zhao, M.-J., and Luján, M. (2012). Conditional likelihood maximisation: a unifying framework for information theoretic feature selection. *J. Mach. Learn. Res.* 13, 27–66.

Chaouachi, M., Jraidi, I., and Frasson, C. (2011). "Modeling mental workload using eeg features for intelligent systems," in *International Conference on User Modeling, Adaptation, and Personalization* (Girona: Springer), 50–61.

Chen, S. I., Visser, T. A., Huf, S., and Loft, S. (2017). Optimizing the balance between task automation and human manual control in simulated submarine track management. *J. Exp. Psychol. Appl.* 23:240. doi: 10.1037/xap0000126

Cinaz, B., La Marca, R., Arnrich, B., and Tröster, G. (2010). "Monitoring of mental workload levels," in *International Conference on e-Health. sn: IADIS*, Vol. 189 (Freiburg), 193.

Coelli, S., Sclocco, R., Barbieri, R., Reni, G., Zucca, C., and Bianchi, A. M. (2015). "EEG-based index for engagement level monitoring during sustained attention," in *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)* (Milan: IEEE), 1512–1515.

Cohen, M. X. (2014). *Analyzing Neural Time Series Data: Theory and Practice*. Cambridge, MA: MIT press.

Dasari, D., Shou, G., and Ding, L. (2017). ICA-derived EEG correlates to mental fatigue, effort, and workload in a realistically simulated air traffic control task. *Front. Neurosci.* 11:297. doi: 10.3389/fnins.2017.00297

Debie, E., Rojas, R., Fidock, J., Barlow, M., Kasmarik, K., Anavatti, S., et al. (2019). Multimodal fusion for objective assessment of cognitive workload: a review. *IEEE Trans. Cybernet.* doi: 10.1109/TCYB.2019.2939399

Delorme, A., and Makeig, S. (2004). EEGlab: an open source toolbox for analysis of single-trial EEG dynamics including independent component analysis. *J. Neurosci. Methods* 134, 9–21. doi: 10.1016/j.jneumeth.2003.10.009

Dong, S.-Y., Kim, B.-K., and Lee, S.-Y. (2016). Eeg-based classification of implicit intention during self-relevant sentence reading. *IEEE Trans. Cybernet.* 46, 2535–2542. doi: 10.1109/TCYB.2015.2479240

Duvinage, M., Castermans, T., Petieau, M., Hoellinger, T., Cheron, G., and Dutoit, T. (2013). Performance of the emotiv epoc headset for p300-based applications. *Biomed. Eng. Online* 12:56. doi: 10.1186/1475-925X-12-56

Elkin-Frankston, S., Bracken, B. K., Irvin, S., and Jenkins, M. (2017). "Are behavioral measures useful for detecting cognitive workload during human-computer interaction?" in *Advances in The Human Side of Service Engineering* (Florida, FL: Springer), 127–137.

Fairclough, S. H., and Venables, L. (2004). Psychophysiological candidates for biocybernetic control of adaptive automation. *Hum. Factors Des.* 177–189. doi: 10.1037/e577062012-018

Freeman, F. G., Mikulka, P. J., Prinzel, L. J., and Scerbo, M. W. (1999). Evaluation of an adaptive automation system using three EEG indices with a visual tracking task. *Biol. Psychol.* 50, 61–76. doi: 10.1016/S0301-0511(99)00002-2

Gale, A., and Edwards, J. (1983). The EEG and human behavior. *Physiol. Corr. Hum. Behav.* 2, 99–127.

Gevins, A., Leong, H., Du, R., Smith, M. E., Le, J., DuRousseau, D., et al. (1995). Towards measurement of brain function in operational environments. *Biol. Psychol.* 40, 169–186. doi: 10.1016/0301-0511(95)05105-8

Gevins, A., and Smith, M. E. (2003). Neurophysiological measures of cognitive workload during human-computer interaction. *Theor. Issues Ergonom. Sci.* 4, 113–131. doi: 10.1080/14639220210159717

Gevins, A., Smith, M. E., McEvoy, L., and Yu, D. (1997). High-resolution EEG mapping of cortical activation related to working memory: effects of task difficulty, type of processing, and practice. *Cereb. Cortex* 7, 374–385. doi: 10.1093/cercor/7.4.374

Hart, S. G., and Staveland, L. E. (1988). Development of NASA-TLX (task load index): results of empirical and theoretical research. *Adv. Psychol.* 52, 139–183. doi: 10.1016/S0166-4115(08)62386-9

Herff, C., Heger, D., Fortmann, O., Hennrich, J., Putze, F., and Schultz, T. (2014). Mental workload during n-back task—quantified in the prefrontal cortex using fNIRS. *Front. Hum. Neurosci.* 7:935. doi: 10.3389/fnhum.2013.00935

Hirshfield, L. M., Chauncey, K., Gulotta, R., Girouard, A., Solovey, E. T., Jacob, R. J., et al. (2009). "Combining electroencephalograph and functional near infrared spectroscopy to explore users' mental workload," in *International Conference on Foundations of Augmented Cognition* (San Diego, CA: Springer), 239–247.

Holewa, K., and Nawrocka, A. (2014). "Emotiv EPOC neuroheadset in brain-computer interface," in *Proceedings of the 2014 15th International Carpathian Control Conference (ICCC)* (Ostrava: IEEE), 149–152.

Kakkos, I., Dimitrakopoulos, G. N., Gao, L., Zhang, Y., Qi, P., Matsopoulos, G. K., et al. (2019). Mental workload drives different reorganizations of functional cortical connectivity between 2D and 3D simulated flight experiments. *IEEE Trans. Neural Syst. Rehabil. Eng.* 27, 1704–1713. doi: 10.1109/TNSRE.2019.2930082

Kamzanova, A. T., Kustubayeva, A. M., and Matthews, G. (2014). Use of EEG workload indices for diagnostic monitoring of vigilance decrement. *Hum. Factors* 56, 1136–1149. doi: 10.1177/0018720814526617

Käthner, I., Wriessnegger, S. C., Müller-Putz, G. R., Kübler, A., and Halder, S. (2014). Effects of mental workload and fatigue on the p300, alpha and theta band power during operation of an erp (p300) brain–computer interface. *Biol. Psychol.* 102, 118–129. doi: 10.1016/j.biopsycho.2014.07.014

Krausman, A. S. (2013). *Understanding the Effect of Audio Communication Delay on Distributed Team Interaction*. Technical report, Army Research Lab Aberdeen Proving Ground MD Human Research And Engineering Directorate.

Lansbergen, M. M., Arns, M., van Dongen-Boomsma, M., Spronk, D., and Buitelaar, J. K. (2011). The increase in theta/beta ratio on resting-state EEG in boys with attention-deficit/hyperactivity disorder is mediated by slow alpha peak frequency. *Prog. Neuropsychopharmacol. Biol. Psychiatry* 35, 47–52. doi: 10.1016/j.pnpbp.2010.08.004

Lim, W. L., Sourina, O., Liu, Y., and Wang, L. (2015). "EEG-based mental workload recognition related to multitasking," in *2015 10th International Conference on Information, Communications and Signal Processing (ICICS)* (Singapore: IEEE), 1–4.

Loft, S., Bowden, V., Braithwaite, J., Morrell, D. B., Huf, S., and Durso, F. T. (2015). Situation awareness measures for simulated submarine track management. *Hum. Factors* 57, 298–310. doi: 10.1177/0018720814545515

Loo, S. K., and Makeig, S. (2012). Clinical utility of EEG in attention-deficit/hyperactivity disorder: a research update. *Neurotherapeutics* 9, 569–587. doi: 10.1007/s13311-012-0131-z

Luque-Casado, A., Perales, J. C., Cárdenas, D., and Sanabria, D. (2016). Heart rate variability and cognitive processing: the autonomic response to task demands. *Biol. Psychol.* 113, 83–90. doi: 10.1016/j.biopsycho.2015.11.013

MacLean, M. H., Arnell, K. M., and Cote, K. A. (2012). Resting EEG in alpha and beta bands predicts individual differences in attentional blink magnitude. *Brain Cogn.* 78, 218–229. doi: 10.1016/j.bandc.2011.12.010

MacPhee, M., Dahinten, V. S., and Havaei, F. (2017). The impact of heavy perceived nurse workloads on patient and nurse outcomes. *Admin. Sci.* 7:7. doi: 10.3390/admsci7010007

Maior, H. A., Pike, M., Wilson, M. L., and Sharples, S. (2014). "Continuous detection of workload overload: an fNIRS approach," in *Contemporary Ergonomics and Human Factors 2014: Proceedings of the International Conference on Ergonomics & Human Factors 2014* (Southampton: CRC Press), 450.

Mapelli, I., and Özkurt, T. E. (2019). Brain oscillatory correlates of visual short-term memory errors. *Front. Hum. Neurosci.* 13:33. doi: 10.3389/fnhum.2019.00033

Mazher, M., Aziz, A. A., Malik, A. S., and Amin, H. U. (2017). An EEG-based cognitive load assessment in multimedia learning using feature extraction and partial directed coherence. *IEEE Access* 5, 14819–14829. doi: 10.1109/ACCESS.2017.2731784

Mikulka, P. J., Scerbo, M. W., and Freeman, F. G. (2002). Effects of a biocybernetic system on vigilance performance. *Hum. Factors* 44, 654–664. doi: 10.1518/0018720024496944

Mota, S., and Picard, R. W. (2003). "Automated posture analysis for detecting learner's interest level," in *Conference on Computer Vision and Pattern Recognition Workshop, 2003. CVPRW'03*, Vol. 5 (Madison: IEEE), 49–49.

Mulder, L. (1989). Cardiovascular reactivity and mental workload. *Int. J. Psychophysiol.* 7, 321–322. doi: 10.1016/0167-8760(89)90258-4

Novaković, J. (2016). Toward optimal feature selection using ranking methods and classification algorithms. *Yugoslav J. Oper. Res.* 21, 119–135. doi: 10.2298/YJOR1101119N

Palva, S., Kulashekhar, S., Hämäläinen, M., and Palva, J. M. (2011). Localization of cortical phase and amplitude dynamics during visual working memory encoding and retention. *J. Neurosci.* 31, 5013–5025. doi: 10.1523/JNEUROSCI.5592-10.2011

Parasuraman, R., and Caggiano, D. (2002). *Mental Workload*. San Diego, CA.

Paus, T., Zatorre, R. J., Hofle, N., Caramanos, Z., Gotman, J., Petrides, M., et al. (1997). Time-related changes in neural systems underlying attention and arousal during the performance of an auditory vigilance task. *J. Cogn. Neurosci.* 9, 392–408. doi: 10.1162/jocn.1997.9.3.392

Pomplun, M., and Sunkara, S. (2003). "Pupil dilation as an indicator of cognitive workload in human-computer interaction," in *Proceedings of the International Conference on HCI*, Vol. 2003. Crete.

Pope, A. T., Bogart, E. H., and Bartolome, D. S. (1995). Biocybernetic system evaluates indices of operator engagement in automated task. *Biol. Psychol.* 40, 187–195. doi: 10.1016/0301-0511(95)05116-3

Puma, S., Matton, N., Paubel, P.-V., Raufaste, É., and El-Yagoubi, R. (2018). Using theta and alpha band power to assess cognitive workload in multitasking environments. *Int. J. Psychophysiol.* 123, 111–120. doi: 10.1016/j.ijpsycho.2017.10.004

Ramirez, R., and Vamvakousis, Z. (2012). "Detecting emotion from EEG signals using the emotive EPOC device," in *International Conference on Brain Informatics* (Macau: Springer), 175–184.

Rojas, R. F., Debie, E., Fidock, J., Barlow, M., Kasmarik, K., Anavatti, S., et al. (2019a). "Encephalographic assessment of situation awareness in teleoperation of human-swarm teaming," in *International Conference on Neural Information Processing* (Sydney, NSW: Springer), 530–539.

Rojas, R. F., Huang, X., Hernandez-Juarez, J., and Ou, K.-L. (2017a). "Physiological fluctuations show frequency-specific networks in fNIRS signals during resting state," in *2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)* (Jeju: IEEE), 2550–2553.

Rojas, R. F., Huang, X., and Ou, K.-L. (2016). Region of interest detection and evaluation in functional near infrared spectroscopy. *J. Near Infrared Spectrosc.* 24, 317–326. doi: 10.1255/jnirs.1239

Rojas, R. F., Huang, X., and Ou, K.-L. (2017b). Toward a functional near-infrared spectroscopy-based monitoring of pain assessment for nonverbal patients. *J. Biomed. Opt.* 22:106013. doi: 10.1117/1.JBO.22.10.106013

Rojas, R. F., Huang, X., and Ou, K.-L. (2019b). A machine learning approach for the identification of a biomarker of human pain using fNIRS. *Sci. Rep.* 9:5645. doi: 10.1038/s41598-019-42098-w

Smit, A. S., Eling, P. A., Hopman, M. T., and Coenen, A. M. (2005). Mental and physical effort affect vigilance differently. *Int. J. Psychophysiol.* 57, 211–217. doi: 10.1016/j.ijpsycho.2005.02.001

Spitzer, B., and Haegens, S. (2017). Beyond the status quo: a role for beta oscillations in endogenous content (re) activation. *Eneuro* 4:ENEURO.0170-17.2017. doi: 10.1523/ENEURO.0170-17.2017

Stein, E. S. (1985). *Air Traffic Controller Workload: An Examination of Workload Probe*. Atlantic City.

Sterman, M., and Mann, C. (1995). Concepts and applications of eeg analysis in aviation performance evaluation. *Biol. Psychol.* 40, 115–130. doi: 10.1016/0301-0511(95)05101-5

Stipacek, A., Grabner, R., Neuper, C., Fink, A., and Neubauer, A. (2003). Sensitivity of human eeg alpha band desynchronization to different working memory components and increasing levels of memory load. *Neurosci. Lett.* 353, 193–196. doi: 10.1016/j.neulet.2003.09.044

Sun, H., Bi, L., Chen, B., and Guo, Y. (2015). EEG-based safety driving performance estimation and alertness using support vector machine. *Int. J. Security Appl.* 9, 125–134. doi: 10.14257/ijsia.2015.9.6.13

Tallon-Baudry, C., Kreiter, A., and Bertrand, O. (1999). Sustained and transient oscillatory responses in the gamma and beta bands in a visual short-term memory task in humans. *Vis. Neurosci.* 16, 449–459. doi: 10.1017/S0952523899163065

Tsai, Y.-F., Viirre, E., Strychacz, C., Chase, B., and Jung, T.-P. (2007). Task performance and eye activity: predicting behavior relating to cognitive workload. *Aviat. Space Environ. Med.* 78, B176–B185.

Veltman, J., and Gaillard, A. (1996). Physiological indices of workload in a simulated flight task. *Biol. Psychol.* 42, 323–342. doi: 10.1016/0301-0511(95)05165-1

Vergara, J. R., and Estévez, P. A. (2014). A review of feature selection methods based on mutual information. *Neural Comput. Appl.* 24, 175–186. doi: 10.1007/s00521-013-1368-0

Vidulich, M. A., and Tsang, P. S. (2012). Mental workload and situation awareness. *Handb. Hum. Factors Ergonom.* 4, 243–273. doi: 10.1002/9781118131350.ch8

Wang, S., Gwizdka, J., and Chaovalitwongse, W. A. (2015). Using wireless EEG signals to assess memory workload in the *n*-back task. *IEEE Trans. Hum. Mach. Syst.* 46, 424–435. doi: 10.1109/THMS.2015.2476818

Winkler, I., Brandl, S., Horn, F., Waldburger, E., Allefeld, C., and Tangermann, M. (2014). Robust artifactual independent component classification for BCI practitioners. *J. Neural Eng.* 11:035013. doi: 10.1088/1741-2560/11/3/035013

Wróbel, A. (2000). Beta activity: a carrier for visual attention. *Acta Neurobiol. Exp.* 60, 247–260.

Xie, J., Xu, G., Wang, J., Li, M., Han, C., and Jia, Y. (2016). Effects of mental load and fatigue on steady-state evoked potential based brain computer interface tasks: a comparison of periodic flickering and motion-reversal based visual attention. *PLoS ONE* 11:e0163426. doi: 10.1371/journal.pone.0163426

Yang, H., and Moody, J. (1999). "Feature selection based on joint mutual information," in *Proceedings of International ICSC Symposium on Advances in Intelligent Data Analysis* (Rochester, NY: Citeseer), 22–25.

Young, J. Q., Irby, D. M., Barilla-LaBarca, M.-L., ten Cate, O., and O'Sullivan, P. S. (2016). Measuring cognitive load: mixed results from a handover simulation for medical students. *Perspect. Med. Educ.* 5, 24–32. doi: 10.1007/s40037-015-0240-6

Zhao, G., Wu, C., and Ou, B. (2013). "The electrocortical correlates of daydreaming during simulated driving tasks," in *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, Vol. 57 (Los Angeles, CA: SAGE Publications Sage CA), 1904–1908.

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.

frontiers in Computational Neuroscience

HYPOTHESIS AND THEORY
published: 03 March 2020
doi: 10.3389/fncom.2020.00016

BrainOS: A Novel Artificial Brain-Alike Automatic Machine Learning Framework

Newton Howard[1]*, *Naima Chouikhi*[2], *Ahsan Adeel*[3,4], *Katelyn Dial*[5], *Adam Howard*[5] and *Amir Hussain*[6]

[1] Etats-Unis, Department of Neurosurgery, Nuffield Department of Surgical Sciences, John Radcliffe Hospital, Oxford, United Kingdom, [2] REGIM-Lab: REsearch Groups in Intelligent Machines, National Engineering School of Sfax (ENIS), University of Sfax, Sfax, Tunisia, [3] School of Engineering, Computing and Mathematics, University of Plymouth, Plymouth, United Kingdom, [4] School of Mathematics and Computer Science, University of Wolverhampton, Wolverhampton, United Kingdom, [5] Howard Brain Sciences Foundation, Providence, RI, United States, [6] School of Computing, Edinburgh Napier University, Edinburgh, United Kingdom

Human intelligence is constituted by a multitude of cognitive functions activated either directly or indirectly by external stimuli of various kinds. Computational approaches to the cognitive sciences and to neuroscience are partly premised on the idea that computational simulations of such cognitive functions and brain operations suspected to correspond to them can help to further uncover knowledge about those functions and operations, specifically, how they might work together. These approaches are also partly premised on the idea that empirical neuroscience research, whether following on from such a simulation (as indeed simulation and empirical research are complementary) or otherwise, could help us build better artificially intelligent systems. This is based on the assumption that principles by which the brain seemingly operate, to the extent that it can be understood as computational, should at least be tested as principles for the operation of artificial systems. This paper explores some of the principles of the brain that seem to be responsible for its autonomous, problem-adaptive nature. The brain operating system (BrainOS) explicated here is an introduction to ongoing work aiming to create a robust, integrated model, combining the connectionist paradigm underlying neural networks and the symbolic paradigm underlying much else of AI. BrainOS is an automatic approach that selects the most appropriate model based on the (a) input at hand, (b) prior experience (a history of results of prior problem solving attempts), and (c) world knowledge (represented in the symbolic way and used as a means to explain its approach). It is able to accept diverse and mixed input data types, process histories and objectives, extract knowledge and infer a situational context. BrainOS is designed to be efficient through its ability to not only choose the most suitable learning model but to effectively calibrate it based on the task at hand.

Keywords: human brain, artificial intelligence, architecture design, hyperparameters, automatic machine learning, BrainOS

OPEN ACCESS

Edited by:
Liang Feng,
Chongqing University, China

Reviewed by:
Jinghui Zhong,
South China University of Technology, China
Yaqing Hou,
Dalian University of Technology, China

***Correspondence:**
Newton Howard
nhmit@me.com

Received: *11 November 2019*
Accepted: *10 February 2020*
Published: *03 March 2020*

Citation:
Howard N, Chouikhi N, Adeel A, Dial K, Howard A and Hussain A (2020) BrainOS: A Novel Artificial Brain-Alike Automatic Machine Learning Framework. Front. Comput. Neurosci. 14:16. doi: 10.3389/fncom.2020.00016

1. INTRODUCTION

As humans are constantly surrounded by data, their survival depends on their capability to understand and evaluate their observations of the external environment. They formulate and extract knowledge from received information by transforming the data into specific patterns and models. To this end, a number of biological processes and aspects of the brain are involved (Hernandez et al., 2010). Once established, brain agents create and refer to these models with each observation.

Both researchers and theorists specializing in neuroscience agree that these brain agents support the task of analyzing external data, processing them and making decisions using fundamental units of thought. Howard and Hussain (2018) describe this process of the fundamental code unit as cognitive minimums of thought where n to N information exchange is expressed in an assembly-like language at the neuronal cellular level. The Fundamental Code Unit addresses the question of whether input signals feed to the brain in their analogical form or if they are transformed beforehand. Bierdman's theory of components recognition and Yin's review of theories of geometry of perception supports the FCU model where an infinite combination of patterns are created from a fixed number of components (Yin, 2008). The conclusions regarding brain processes derived from the field of neuroscience are applied in parallel to the field of artificial intelligence (AI) (Wang et al., 2016). The finest example of this is Machine Learning (ML), which is inspired by the brain's methods of processing external signals (input data) (Wang et al., 2016). ML can mimic human brain behavior (Louridas and Ebert, 2016) by providing a set of appropriate and intelligent techniques to perform data analysis (Howard and Lieberman, 2014). ML automates data manipulation by extracting sophisticated analytical models. Within this branch of AI, systems are capable of learning from data and distributions, distinguishing patterns and making autonomous decisions, which considerably decreases the need for human intervention.

The appeal of ML is considerably rising due to factors, such as the growing demands of data mining tools (Bredeche et al., 2006). Indeed, in a world replete with data, intelligent computation is gainful in terms of expense and performance (Wang and Yan, 2015). Automated data handling has yielded valuable systems able to solve increasingly complex problems and provide more accurate outcomes.

The three big challenges that ML still face are (1) that it requires a great deal of training data and is domain-dependent, (2) it can produce inconsistent results for different types of training or parameter tweaking, and (3) it produces results that may be difficult to interpret when such black-box algorithms are used. Here, we propose a novel automatic approach to address such shortcomings in a multidisciplinary approach that aims to bridge the gap between statistical Natural Language Processing (NLP) (Cambria et al., 2014) and the many other disciplines necessary for understanding human language, such as linguistics, common sense reasoning and computing. Our proposed approach, "Brain OS" is an intelligent adaptive system that combines input data types, processes history and objectives, researches knowledge and situational context to determine what is the most appropriate mathematical model, chooses the most appropriate computing infrastructure on which to perform learning, and proposes the best solution for a given problem. BrainOS has the capability to capture data on different input channels, perform data enhancement, use existing AI models, create others and fine-tune, validate and combine models to create more powerful collection of models. To guarantee efficient processing, BrainOS can automatically calibrate the most suitable mathematical model and choose the most appropriate computing learning tool based on the task to handle. Thus, it arrives at "optimal" or pre-optimal solutions. BrainOS leverages both symbolic and sub-symbolic methods as it uses models, such as semantic networks and conceptual dependency representations to encode meaning but it also uses deep neural networks and multiple kernel learning to infer syntactic patterns from data. The architecture of BrainOS uses concepts from the critic-selector model of mind and from brain pathology treatment approaches.

Herein, a thorough evaluation of the state of the art of Automatic ML is discussed, and specifically the proposed automatic BrainOS is presented in detail. The advantages of BrainOS over state of the art models are enumerated, and an empirical study is presented in order to validate the proposed framework.

2. STATE-OF-THE-ART: AUTOMATIC ML FRAMEWORKS

ML has several models, which apply one or more techniques to one or more applications. ML models include support vector machine (SVM) (Mountrakis et al., 2011), bayesian networks (BNs) (Bielza and Larranaga, 2014), deep learning (DL) (Bengio et al., 2013), decision trees (DTs) (Kotsiantis, 2013), clustering (Saxena et al., 2017), artificial neural networks (ANNs) (Dias et al., 2004), etc.

Each ML model is an intelligent computing mean that is trained to perform a well-defined task according to a set of observations. These intelligent models require a set of related data to extract knowledge about the problem at hand. The construction of these data is a crucial factor by which the performance of the model is judged. The more the data, the better the performance becomes.

All ML models undergo three principle steps: (1) receiving input data (signals), (2) processing these data, and finally (3) deriving outputs according to the handled task. To check if the system achieves a good learning level, an evaluation metric is computed. It is then tested on a number of patterns not previously observed and is then judged whether it has acquired a good generalization capability or not.

For any given application, there are a number of specific models that can perform better than the others. The choice of the best model for a well-determined task does not obey to any rule. Rather, there are only instructions on how these models proceed. Thus, there is no way to understand how to choose the best model for a problem.

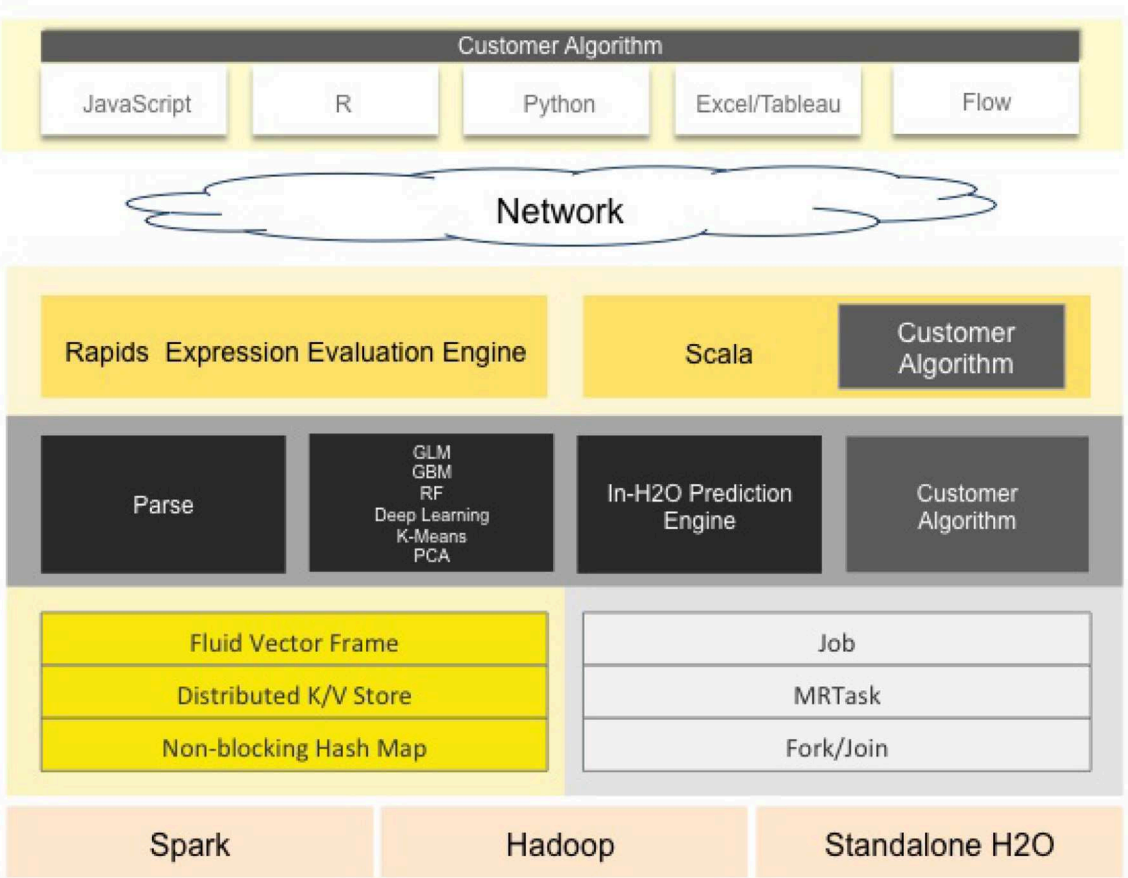

FIGURE 1 | H2O's standard architecture. H2O is designed mainly on Java language with some blocks based on Python, JavaScript, R, and Flow. The software stack is composed of the top and bottom sections, divided by the network cloud. The top part highlights some of REST API customers, while the bottom illustrates the constituents undergoing the Java virtual machine (image courtesy of H2O.ai).

While classic ML focuses on developing new models and techniques without regard to the resulting increase in complexity, automatic ML (AML), affirms that these tools can be employed in an easier manner. AML platforms computerize the majority of ML tasks in less time and implementation costs. Therefore, automatic ML has become a hot topic not only for industrial users, but also for academic purposes.

Fine-tuning or optimization is a key component to provide suitable models Hutter et al. (2019). AML framework addresses issues, such as the best ML model for different problems, model tuning or hyper-parameters optimization, etc. (Yao et al., 2019). Simple classical methods, Bayesian optimization and metaheuristics are among the most used tools of optimization in AML.

To develop such automated frameworks, researchers have developed and proposed several solutions e.g., H2O, Google Cloud AutoML, and Auto-sklearn depicted in **Figures 1–3**, respectively. These frameworks have certainly solved several problems but are still far from the strategy behind the human brain. What can be noticed throughout the enumerated techniques is that developers are using sophisticated ML models without reasoning; hence, no explainable AI.

- **H2O**

  H2O (Landry, 2018) is an open source machine learning platform for the enterprise. The platform contains a module that employs a set of well-defined algorithms to form a pipeline. It provides a specific graphical interface to set the appropriate model, the stopping criteria and the training dataset.

  It supports several linear and complex ML models, such as Deep Neural Networks (DNN), gradient boosting machines, etc. It also supports the Cartesian and random grid searches optimization techniques. It is designed based mainly on Java developing language with some blocks on Python, Javascript, R and Flow. The standard H2O architecture is visualized in **Figure 1** (Landry, 2018).

  The H2O software stack depicted in **Figure 1** is composed of numerous components that can be divided into two parts

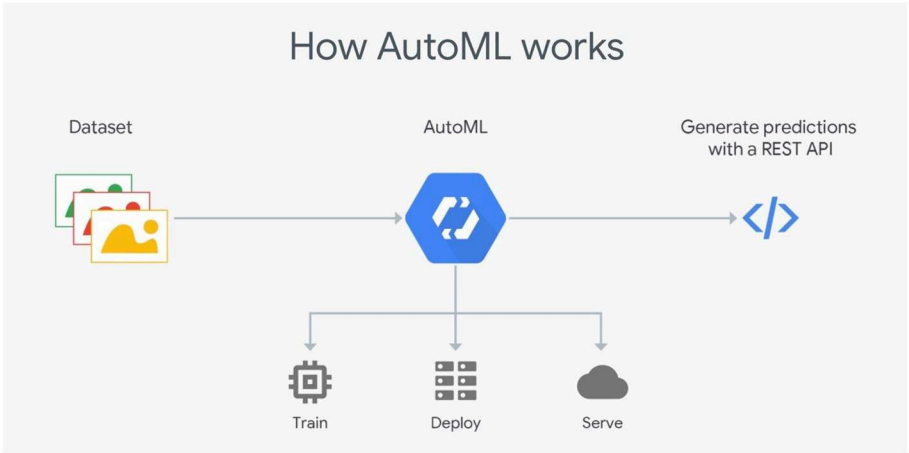

FIGURE 2 | Google Cloud AutoML's standard architecture. Cloud AML offers a simple interface for inexperienced users to exploit models according to their needs. Using DNNs and genetic algorithms, Cloud AutoML trains machine learning models, deploys models based on user data, and stores trained data in cloud storage. The framework generates predictions with a REST API (image courtesy of Google Cloud).

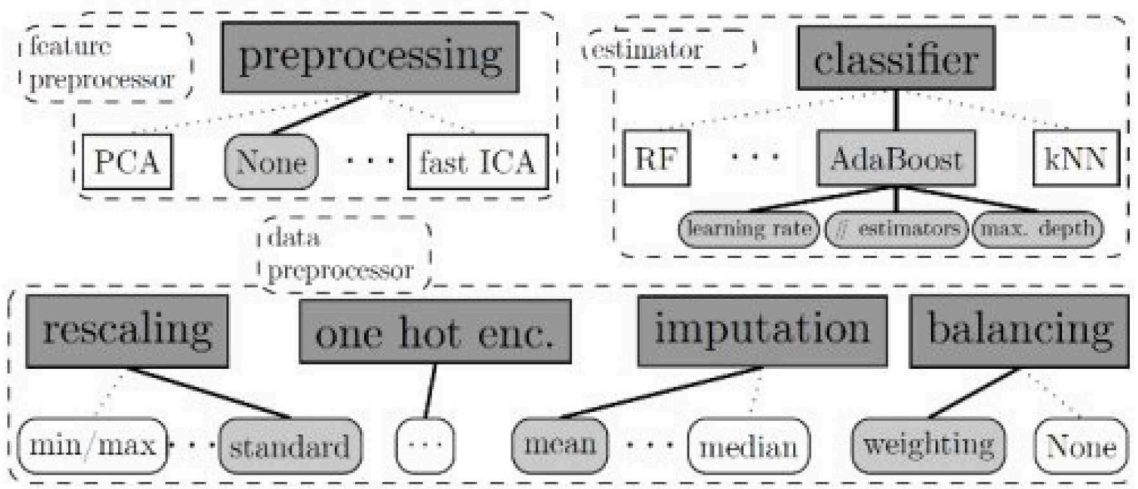

FIGURE 3 | Auto-sklearns's standard architecture. Auto-sklearn employs Bayesian fine-tuning for hyperparameter settings. The program utilizes 15 classification approaches, 14 pre-processing techniques, and four feature engineering methods.

(top and bottom). The top part highlights some of REST API customers, while the bottom part illustrates the constituents undergoing the Java virtual machine.

In spite of its ease of use especially for ML beginners and non-specialists, H2O still suffers from a lack of background in data science. Another drawback concerns the huge amount of employed resources. In fact, failures during complex executions are very likely to occur.

- **Google's Cloud AutoML**
  Cloud AutoML (Vinson, 2018) presents a series of products permitting inexperienced users to exploit well-qualified models obeying their business queries. It employs sophisticated capabilities of Google, such as transfer learning. It provides users with a simple interface so that they are able to learn, assess, improve, and unfold techniques according to their data. The products offered by this framework include AutoML Vision and video-intelligence, AutoML natural language and translation and AutoML Tables, etc. The standard Cloud AutoML's architecture is visualized in **Figure 2** (Vinson, 2018).

  This framework is mainly based on deep neural networks (DNN) and genetic algorithms. It also asks users to respect a limit of training data size. For AutoML, tables data size should not surpass 100 Go.
- **Auto-sklearn**
  Auto-sklearn, proposed by Feurer et al. (2015), employs Bayesian fine-tuning for hyperparameter settings. It is an improved version of the scikit-learn system (a preceding

automatic ML). The standard Auto-sklearn's architecture is visualized in **Figure 3**.

There are 15 classification approaches, 14 pre-processing techniques and four feature engineering methods. Although its structure is advanced, this toolkit's package does not support natural language inputs. Therefore, it can not distinguish categorical data from digital data (Feurer et al., 2015).

Although the majority of preexisting ML frameworks have efficiently solved several problems, such as object recognition and image understanding, they are still far from simulating human brain processes. ML has attempted to mimic the brain as a model for computation, for instance neural networks algorithms, however ML is still not able to perform as well as the human brain. We propose a novel automatic ML framework called "BrainOS." The proposed system architecture and operation is biologically inspired by neuron cells, designed at a very low level of abstraction.

3. BRAINOS: A NOVEL AUTOMATIC ML FRAMEWORK

Attracted by the strength of the human brain's ability to reason and analyze objects and ideas, we propose a novel automatic ML framework called "BrainOS." The system's architecture and operation is inspired by the behavior of neuronal cells.

Since existing ML models have many challenges related to over-sized task-dependent training data and uninterpretable results, BrainOS addresses these shortcomings. Indeed, it provides a multidisciplinary approach able to deal with natural language processing (NLP) so that the gap between statistical NLP and many other disciplines necessary for understanding human language is minimized. Linguistics, commonsense reasoning, and affective computing are essential to analyze the human language. BrainOS involves symbolic as well as sub-symbolic techniques by employing models like semantic networks and conceptual dependency representations to encode meaning. Furthermore, it uses DNNs to deduce syntactic aspects from data.

3.1. High-Level BrainOS Model

Thanks to its anthropomorphic and data-adaptive power, BrainOS can be of great use in various types of applications, because it has the capability to react differently according to the user's profile and preferences. Data adaptation signifies the ability to pick out the most adequate mathematical model in terms of the received input data.

The high-level BrainOS architecture is presented in **Figure 4**. The Input Data Layer is composed of data points coming from various source channels (sensors, videos, images, etc). When fed through this layer, the data undergo numerous stages of data retrieval and handling. For example, input points can be identified, typified, and pre-processed. Sampling techniques can also be employed at this level. The Data Processing Layer identifies a number of intelligent approaches according to the following stages:

- *Critic-Selector Mechanism*: combines input data types, processes history and objectives, researches knowledge and situational context to determine the most appropriate ML model for existing data and how the system should manage the processing resources.
- *Data handling using ML pipelines*: A series of vertical and horizontal pipelines to spread out the data can help prepare the data more quickly and efficiently.

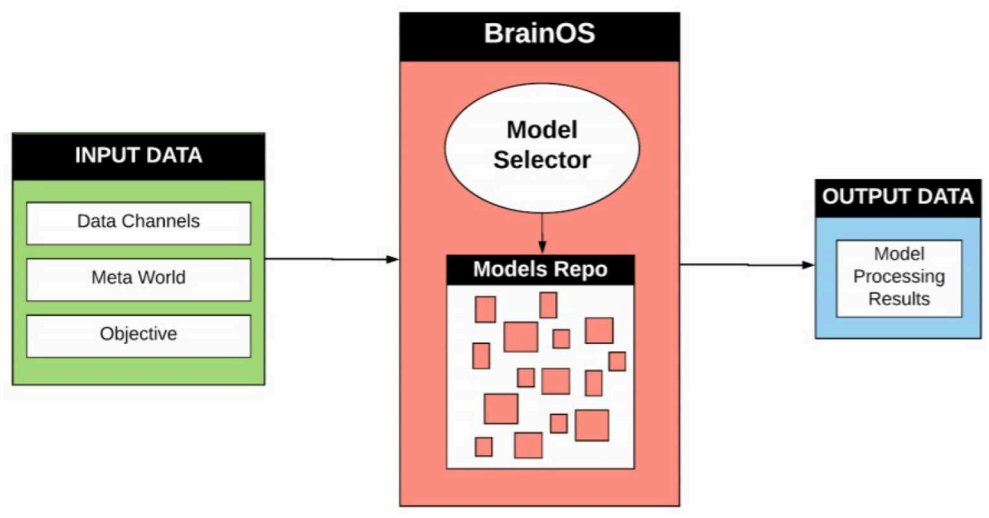

FIGURE 4 | High-level Brain OS architecture. Input data information is received from various mixed input data channels. Real world context is retrieved from the meta-world container. The objective presents the aim of the processing problem and the desired outputs. The most appropriate model is then created and stored in the model repository for future use or chosen from a preexisting model within the repository. The output data contains the results and findings achieved after undergoing data processing.

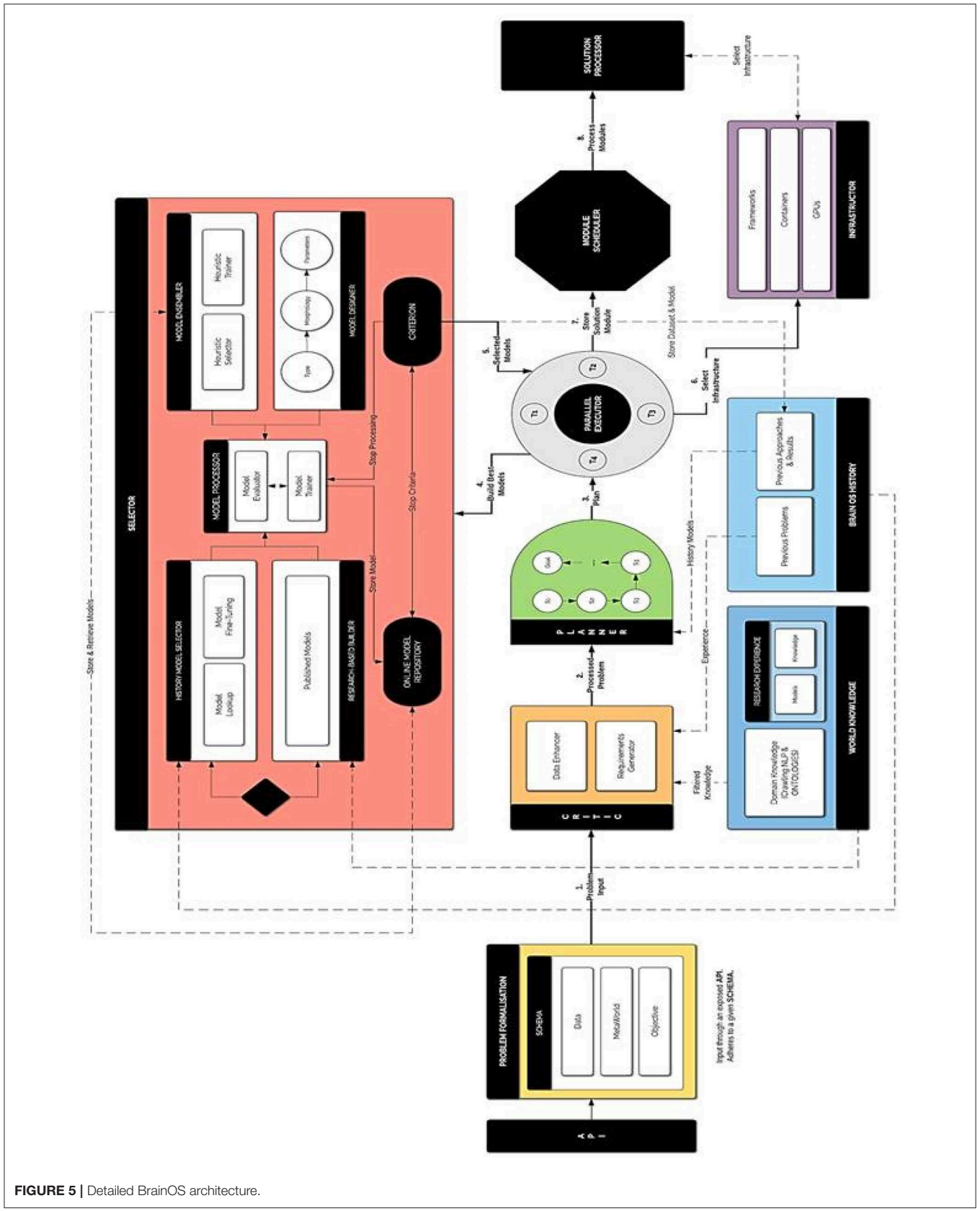

FIGURE 5 | Detailed BrainOS architecture.

- *Model training and/or transfer learning*: Not isolating algorithms and utilizing knowledge from a previous task to solve related ones increases efficiency and accuracy.

The Output Data Layer contains the results and the findings achieved after undergoing the Data Processing Layer.

BrainOS is adaptive to various data channels. It employs several data processing techniques and model selector components. Similar to the human brain, BrainOS uses an archive of data, knowledge and ML models. BrainOS is boosted by a complex qualifier-orchestrator meta-component. The critic-model selector is located within the orchestrator to give an answer to the question "What is the best tool to chose for a given problem?".

Based on the human brain, which uses different neuronal areas to process input data, depending on the receptor type, the proposed infrastructure is founded on an ensemble of resources that are managed by the critic-selector (turned on and off), much in the manner the biological mind operates.

3.2. BrainOS Fundamental Architecture

The key concept of BrainOS is its adaptability to the problem at hand. It selects the appropriate models for the nature of the input data. **Figure 5** visualizes a more thorough overview about the architecture of the whole infrastructure. As shown in **Figure 5**, BrainOS topology is characterized by a number of components. In the next section, every component is detailed.

3.3. Problem Formalization Component

Problem formalization is the principle entry point of the system. It houses three sub-components: data, meta-world information, and task objective. These three components contain all the necessary related information associated with the data and the task to be processed. The input data is held within the data container while general and real world context data is held in the meta-world container. The task objective represents the primary aim of the problem to be processed and the desired outputs.

For consistency, the input data points should comply to a specific schema. This can be done using an API to connect BrainOS to other ML packages to maintain the task's integrity and consistency. **Figure 6** presents an example of the problem formalization component.

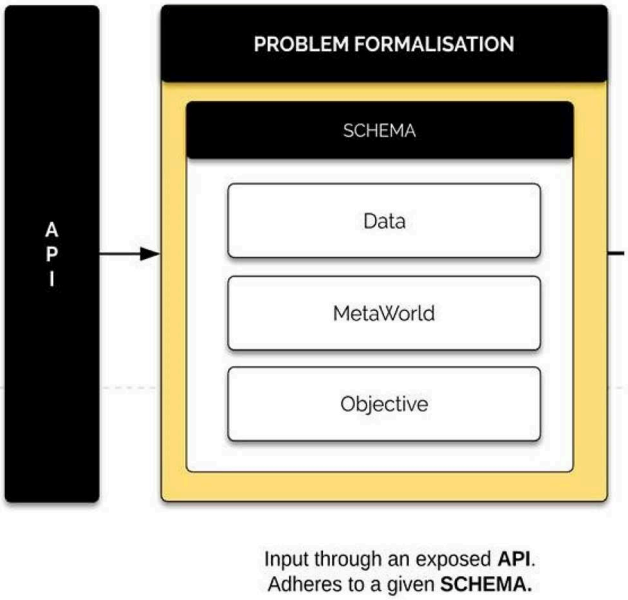

FIGURE 6 | Problem formalization component. The problem formalization component includes mixed input data, general real-world data context contained within the meta-world container, and the major objective of the processing problem, as well as the desired outcomes.

3.4. The Critic Component

The critic (qualifier) component utilizes the problem formulation and the BrainOS history (meta-world knowledge) to enhance the dataset fed to the system. It improves the data with antedate datasets, which complement the current input features in a module called the data enhancer. Furthermore, it applies qualifications, imposes constraints and builds requirements to achieve an intermediate. **Figure 7** shows the architecture of the critic component.

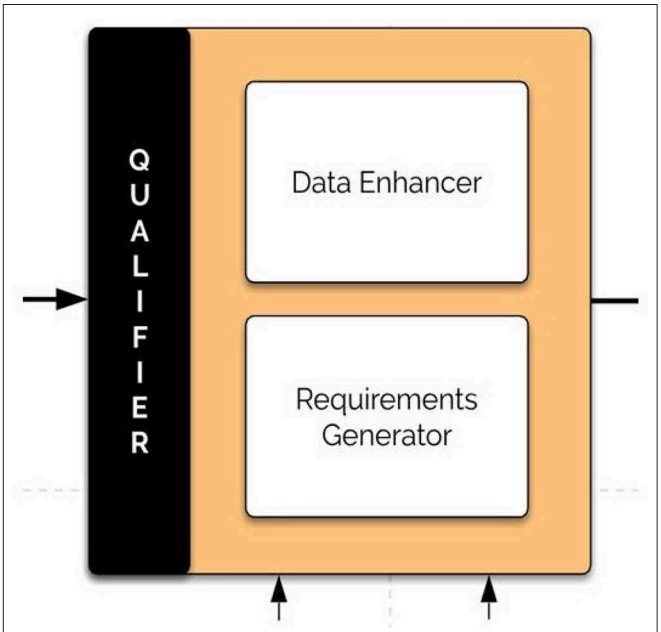

FIGURE 7 | The qualifier (critic) component. The qualifier component enhances the datasets fed to the system and applies qualifications, imposes constraints, and builds requirements to achieve an intermediate.

3.5. History Database

Proposing an adaptive learning system in a non-static space looks like the human's reasoning aspect. In fact, humans exploit their knowledge and experiences to find solutions to any kind of problem. Inspired by this extraordinary capability, BrainOS blends at least two memory sub-components: world knowledge and history. **Figure 8** shows the architecture of the history database component.

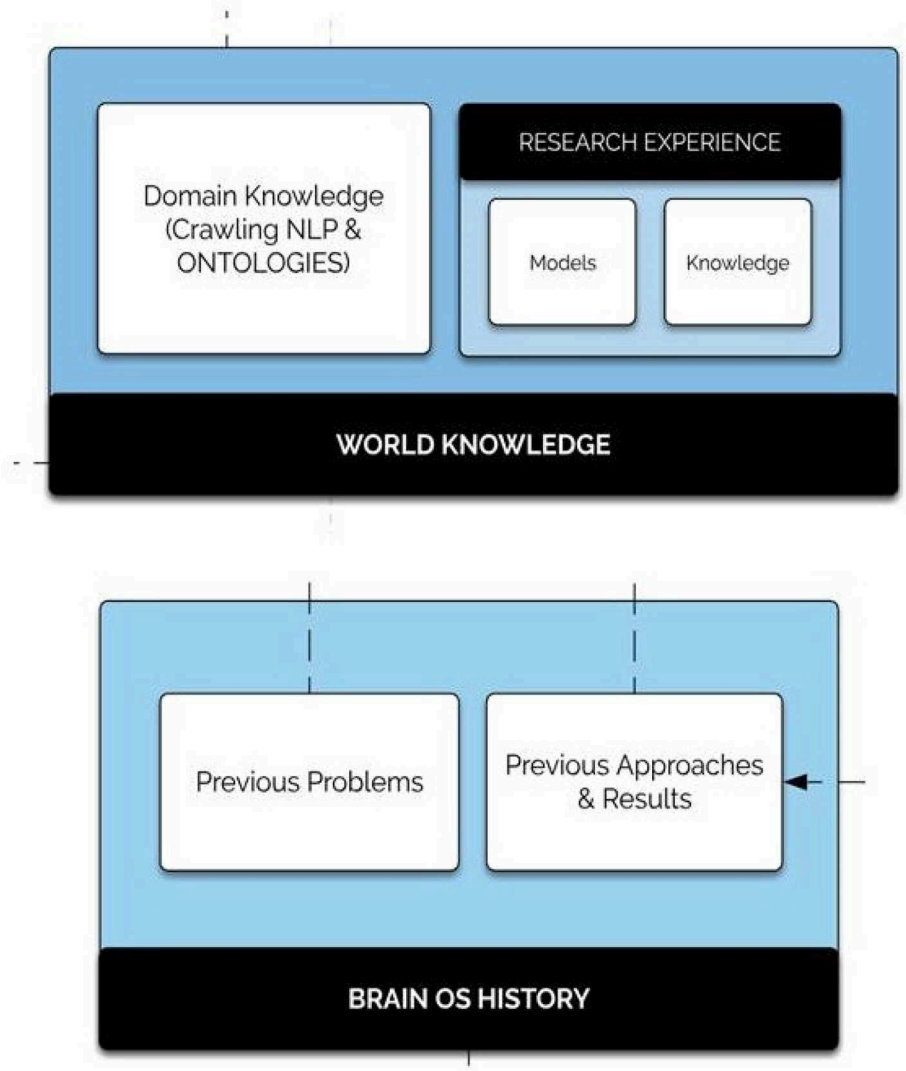

FIGURE 8 | History database component. The history database component is comprised of world knowledge as well as the Brain OS history. The world knowledge sub-component contains the domain knowledge package of crawling NLP and ontologies as well as research experience comprised of stored models and more abstract research knowledge.

1. *The BrainOS history:* includes the experience acquired over the system life cycle in terms of encountered data sets, previously employed models and achieved outcomes. Such a quick memory access resource is of great value especially in situations where the platform encounters problems already resolved. In this case, the system uses a "reflex response."
2. *The world knowledge:* holds the "common sense" world knowledge, overlaying from general to domain-specific concepts. The domain knowledge package contains numerous fields within which the infrastructure requires a knowledge expert. The integrated research experience is comprised of models and inferences drawn from real world knowledge encompassing the following two components:

- **Stored models**: include non-constrained previously discovered resources.
- **More abstract research knowledge**: a big information field. It can be carried out on specific problem formulations, distinct problem solutions, or precise datasets.

3.6. The Planner Component

The Planner is based essentially on the processed problem and the history of used models. It is able to set the most adequate processing flow for the tackled problem according to the world knowledge, objective, and the similarity of the present task with those treated in the past.

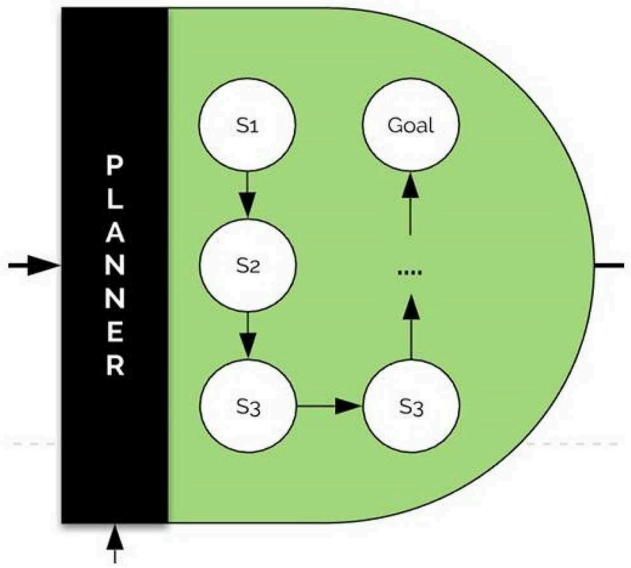

FIGURE 9 | The planner architecture. The planner sets the most adequate processing flow for the problem according to world knowledge. Within the planner, special heuristic search algorithms can be run for the detection of the proper node sequences for a given task.

As an example, for a problem of intent extraction from an image, the planner might prescribe the following steps:

- Run captioning algorithms on the image to obtain a narrativization of the image.
- Run object detection and activity recognition on the image.
- Run an algorithm to obtain an ontology for the previously extracted concepts.
- Infer intent using all the previously obtained entities and ontologies.

The planner plays the role of large bidirectional graph knowledge within which special heuristic search algorithms can be run for the detection of the proper node sequences for a given task. The architecture of the planner is visualized in **Figure 9**.

3.7. The Parallel Executor

The parallel executor plays the role of task scheduler. This component builds models, stores solution modules, and selects infrastructure. It manages when, what and how threads will be executed once they come from the selector.

The parallel executor triggers a number of threads for convenient structures. Based on the models provided by the selector, the executor creates new models or combines existing ones. It partitions the corresponding tasks in parallel threads processing simultaneously. The architecture of the parallel executor is visualized in **Figure 10**.

3.8. The Module Scheduler

The module scheduler receives threads sent by the parallel executor and plans a schedule for the solution's execution. This gives the ability of parallel execution using different resources.

3.9. The Selector Component

The Selector, the key component of BrainOS, picks out the adequate model according to the Problem Formulation. With the intention to provide suitable models, the Selector proceeds with the following steps in parallel:

1. Searching for an adequate model in BrainOS history. If a good fit is found, then the corresponding tool is optimized, trained, and evaluated.
2. Else, searching in the Research Knowledge including published papers and source codes. If a suitable candidate is found, then it is tuned, learned, and evaluated.
3. Building a tool from scratch after type and topology are defined. Thereafter, the model is tuned, trained, and assessed.
4. Performing an ensemble learning by combining several models which may give better findings than a higher accuracy model.

Therefore, before the Selector adopts the solution model for the given Problem Formulation, it analyses whether there is a combination of models that can outperform the selected model. If the Selector finds such a model combination, then the model solution is an ensemble of models. The architecture of the module selector is visualized in **Figure 11**.

The selected ensemble of models, the problem formulation and the given precision are then archived in the BrainOS history. The four approaches are executed in parallel where every module records the best model within the online model repository.

The criterion determines whether the retrieval is a fitted enough approach according to the predetermined objectives, or when one of the modules should be excluded from the search. For each part of BrainOS processing plan, appropriate models are selected. It is advisable to furnish different specialized Domain Specific Instances of the selector, each one optimized for a specific domain knowledge or problem context. For instance, for classification purposes, SVM, K-means clustering, ANNs and other tools can be employed. For time-dependant problems, recurrent architectures, such as recurrent neural networks (RNNs) (Chouikhi et al., 2017) are highly recommended. To deal with feature engineering problems, independent component analysis (ICA) (Henriquez and Kristjanpoller, 2019), independent component analysis (PCA) (Kacha et al., 2020), autoencoders (AEs) (Xu et al., 2016), matrix factorization, and various forms of clustering.

Concerning optimization tasks, there are many useful techniques, such as evolutionary computation (Chouikhi et al., 2016), global optimization, naive optimization, etc.

3.10. The Orchestrator Component

From a high level of abstraction, the BrainOS plays the role of an orchestrator-centered infrastructure as it monitors overall models. It is arranged in a graph to pick out the processing paths. The proposed framework seems to be powerful as it can employ any approach from supervised to unsupervised learning, reinforcement learning, search algorithms, or any combination of those.

The orchestrator is a meta-component which merges input data, processes history and objectives, and researches knowledge and situational context to determine the most appropriate

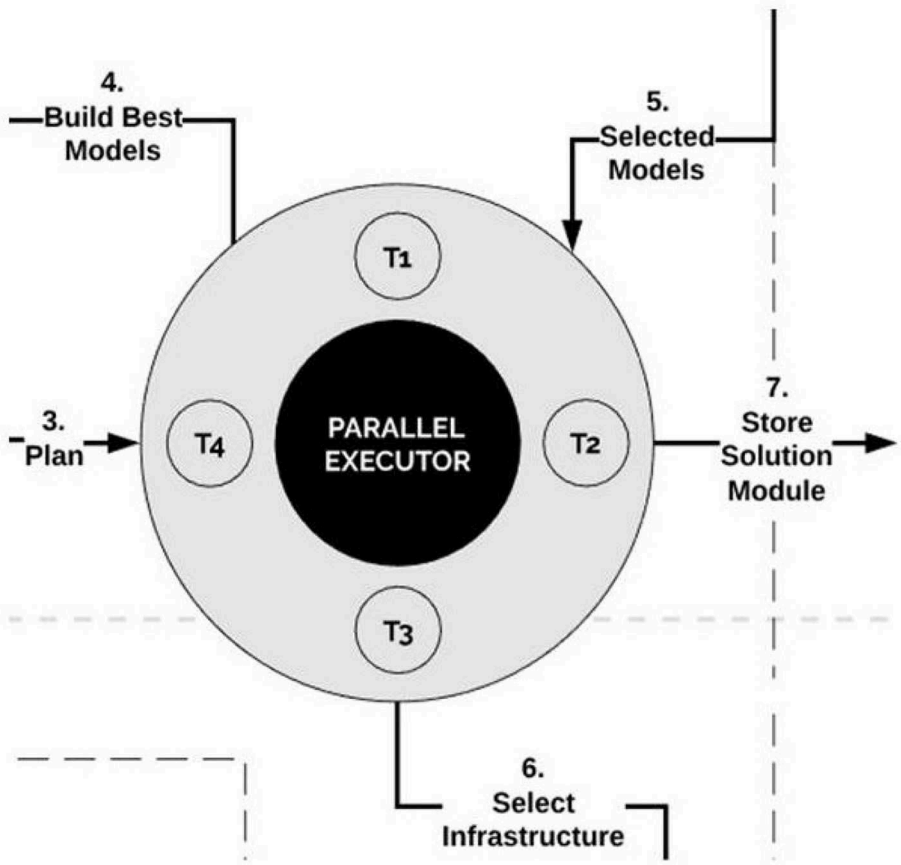

FIGURE 10 | The parallel executor. The parallel executor creates new models or combines existing ones. It partitions corresponding tasks in parallel threads processing simultaneously.

ML model for a given problem formulation. The orchestrator is comprised of four components: models selector, problem qualifier, planner and parallel executor.

4. INTERPRETATIONS

Our evaluation of BrainOS focuses on the following questions:

Question 1 *Flexibility and adaptability*: Is BrainOS capable enough to deal with a large variety of application areas?

Question 2 *Fast convergence*: When dealing with a certain task, does BrainOS proceed quickly or it takes much time to converge?

Question 3 *Accuracy*: How does BrainOS ensure the achievement of accurate results?

4.1. Flexibility and Adaptability

One of the most important characteristics of the BrainOS is its flexibility to handle several issues. BrainOS can be adapted for a large array of existing problems, and also extended for new approaches. Here, we provide just a small subset of possible application areas for the BrainOS. It can be applied to Anthropomorphism in Human/Machine Interaction problems including personality emulation and emotional intelligence. Moreover, BrainOS is relevant in dealing with brain disease diagnostics and treatment (e.g., Alzheimer, Parkinson Disease, etc.), automated manufacturing systems, energy management, etc.

In fact, the inner memory modules, incubated within the BrainOS architecture, store previous experiences and knowledge. This gives our platform the possibility to solve any kind of application, even those with a high-level of abstraction. What specifies the proposed paradigm over the state of the art, is the consistency with conceptual data, such as NLP. Indeed, it addresses the shortcomings of the existing models in solving many contextual tasks. Additionally, it provides a plenty of ML models, each of which performs in a specific field.

4.2. Fast Convergence

BrainOS can decrease the execution time. If a problem was previously tackled and another problem in the same context is about to feed to BrainOS, the model previously employed can be directly found in the BrainOS history and used to solve the new task. In this case, there is no need to proceed to the selector and the subsequent components. Furthermore, one of the common challenges of automatic ML systems is to quickly decide how to choose the model that best fits the given task. BrainOS encompasses a selector component which automatically and directly chooses better models according to the task at hand. This can be gainful in terms of run time. Furthermore,

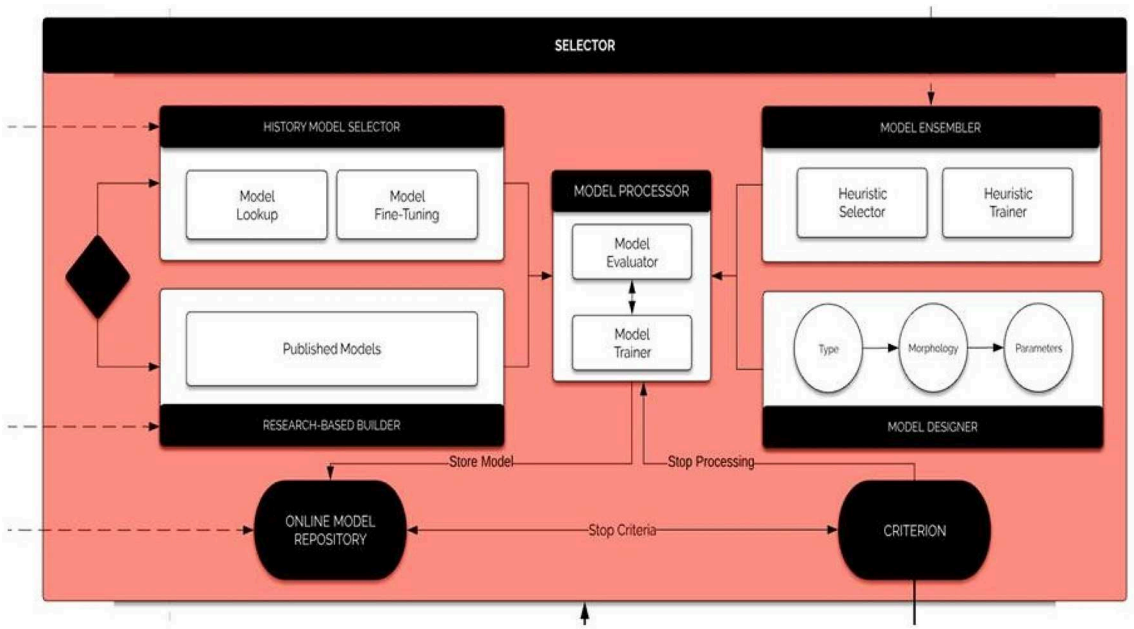

FIGURE 11 | The selector. The selector runs the history model selector, the researched-based builder, the model ensembler, and the model designer in parallel. The history model selector searches for an adequate model in BrainOS history. The Research-Based Builder searches published papers and source code to find a suitable model. The model ensembler combines several models, which may give better findings than a higher accuracy model. The model designer builds tools from scratch. The model processor evaluates and trains the selected models.

BrainOS supports parallel execution by launching several threads simultaneously through the parallel executor component. This can save much time and hasten data processing.

4.3. Accuracy

BrainOS holds many components, which constitute levels through which the data circulates. At the majority of these levels, there is a storage of historical processing and models and knowledge from world experience. Recording previous models and their findings gives a priori indications about what model to use. Furthermore, BrainOS provides several optimization techniques as well as ML models capable of affording high generalization capability. It is also possible to carry out an ensemble learning by executing many models at the same time and taking the best one.

4.4. Availability and Scalability

Data Processing Service is responsible for collecting data from different input channels, decompressing it, and storing it for later usage. There is a large number of data channels which can send data to the BrainOS. Thus, on the Cloud, there is a need for high scalability in recording this data, and there will also be a demand to store a large amount of it. There are different technologies which can support this, but the most suitable ones that can enable the constant increase of inputs and high parallelism of incoming data are those based on the Publish/Subscribe Paradigm. In this specific case of data processing, the inputs will act as data publishers while the BrainOS which processes the data, as a subscriber.

5. EMPIRICAL RESULTS

Currently the implementation of AML models, such as Google's AI solution is likely to be susceptible to high latency, computational cost and power consumption. This is due to the huge data flow presented by larger data sets. The big issue, which the industry will not overcome easily, is that it is using digital arithmetic units and Boolean gates, which themselves are a mismatch with how neurons and synapses work. This represents, therefore, a poor approach to implementing deep neural architectures. To continue solving more complex problems, using increasingly more hardware is mandatory yet unsustainable. The proposed BrainOS is under the way of implementation. We are designing and testing some BrainOS modules, and we will gather all the modules into one framework. For example, we are working with a completely new architecture for Deep Neural Networks (DNN), which we call Deep Cognitive Neural Network (DCNN) (Howard et al., 2019).

5.1. Deep Cognitive Neural Network (DCNN)

DCNN is one of the new ML models exhibiting characteristics similar to the human brain, such as perception and reasoning and is a much better fit for building Neural Networks. The value of this new architecture is that big data analysis can be run near real-time on small devices, such as mobile phones and IoT devices. The proposed DCNN architecture, shown in **Figure 12**, is comprised of one million neurons and 2.5 billion synapses. DCNN has a remarkable property of

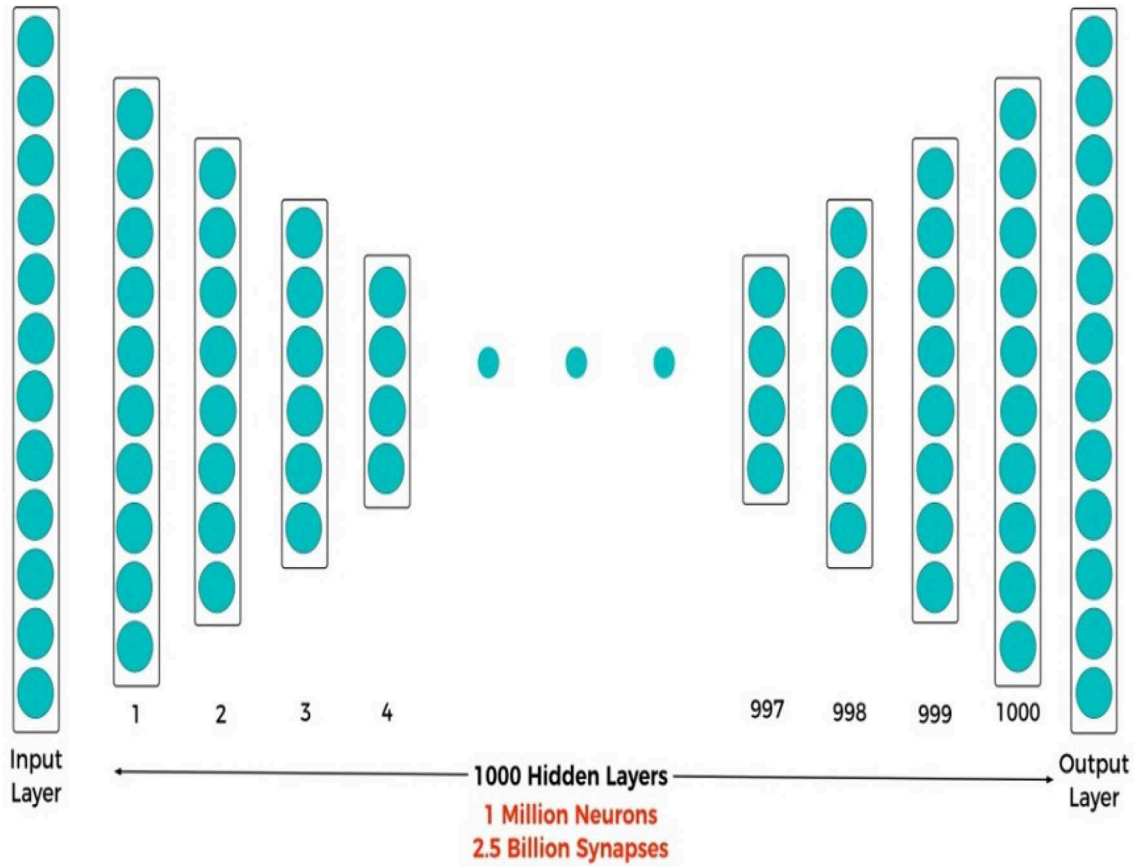

FIGURE 12 | DCNN architecture (1,000 hidden layers, 1 million neurons, and 2.5 billion synapses).

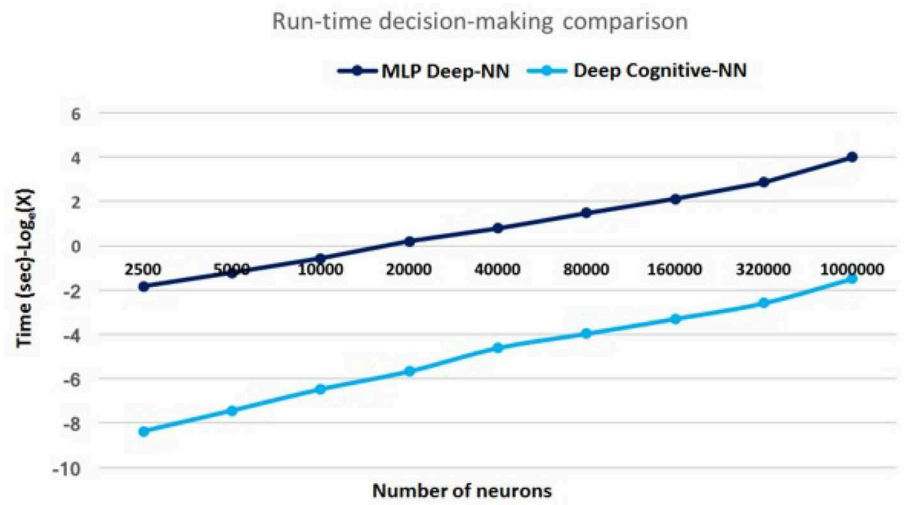

FIGURE 13 | Decision making speed: for very large scale DNN processing, simulation results of DCNN has shown 300× faster decision-making as compared to the state-of-the-art Multi-Layer Perceptron (MLP) based deep neural network comprising one million neurons and 2.5 billion synapses.

concurrently acquiring highly energy-efficient implementation, fast decision-making, and excellent generalization (long-term learning). DCNN is highly energy-efficient in computing with ultra-low energy requirements that can easily be implemented in both hardware and software, as its neurons can be represented by simple equations consisting of addition, subtraction, and

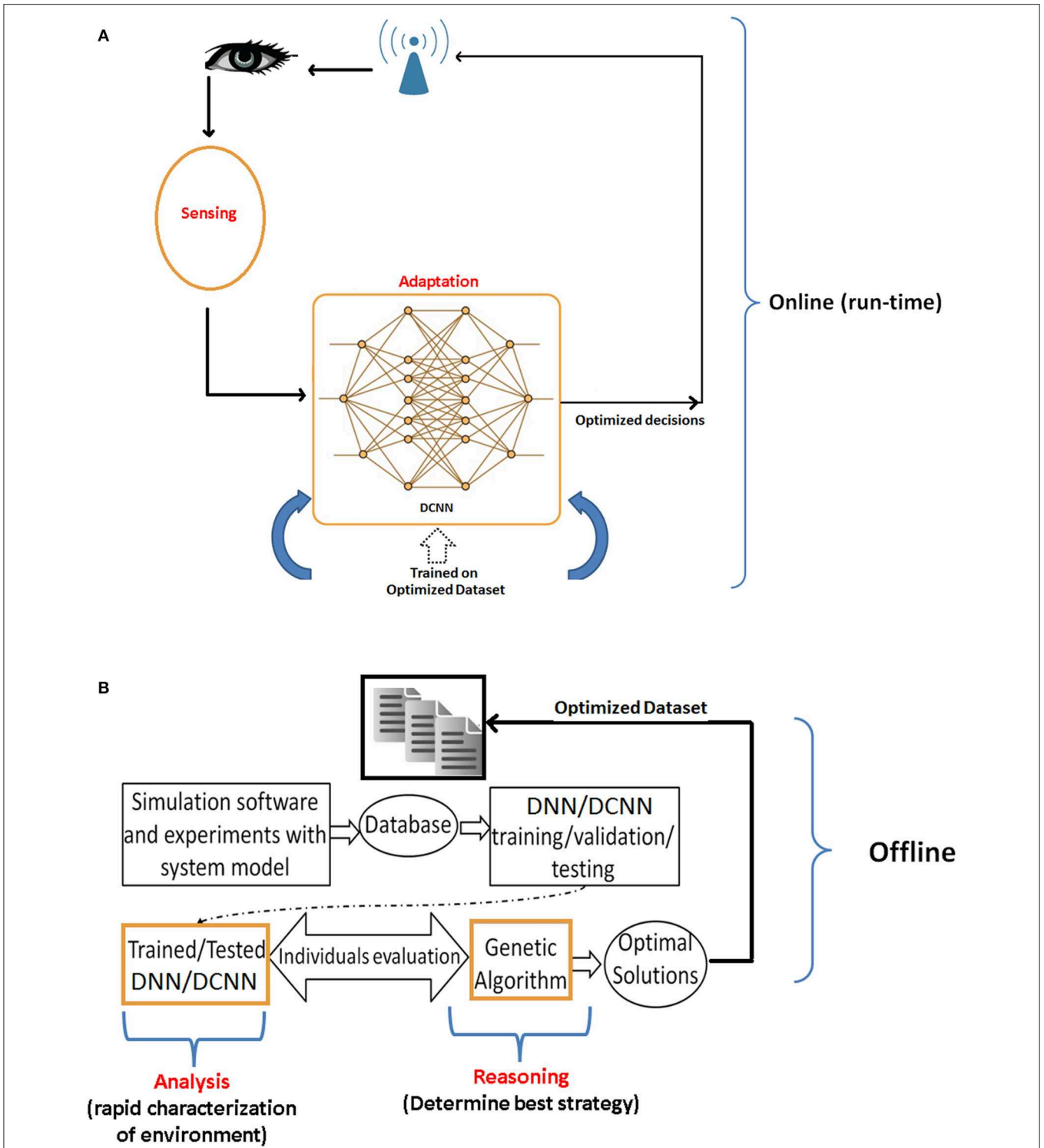

FIGURE 14 | DCNN based optimized decision-making. **(A)** DCNN based real-time optimal adaptation. **(B)** Optimized dataset extraction: the data is first collected for learning which helped GA based reasoning process to build optimized dataset.

division operations. A highly energy-efficient implementation of shallow neural networks using complementary metal-oxide semiconductor (CMOS) or Probabilistic CMOS (PCMOS) technology has revealed that they are up to 300× times more efficient in terms of energy performance product (EPP). The substantial gain per-operation is proportional, which depends on

the entire application, where large gains are expected with deep structures for large scale processing.

5.2. DCNN Fast Decision-Making

DCNN was trained and tested using the state-of-the-art MNIST dataset (LeCun et al., 1998). The decision making results are depicted in **Figure 13**. It can be seen that for very large scale processing, DCNN has shown up to 300× faster decision-making as compared to the state-of-the-art Multi-Layer Perceptron (MLP) based deep neural network.

5.3. DCNN Integration With the Reasoning Algorithm

Another unique property of the developed DCNN is its quick adaptability and convergence behavior when integrated with reasoning algorithms to acquire human-like computing (both perception and reasoning simultaneously) in real-time. Large scale simulation reported up to 80× faster decision-making. The simulated reasoning/optimization framework is demonstrated in **Figure 14**. **Figure 14A** shows the DCNN based sensing and adaptation procedure, trained on an optimized dataset produced by the optimization framework. The optimization framework is shown in **Figure 14B**, which is responsible for analysis and reasoning. In this framework, the learning module assists the reasoning process in deciding the best configurations to be used in new upcoming situation. Whereas, the reasoning module [e.g., genetic algorithm (GA)] uses learning module to maximize the utility function. The proposed framework is used for an optimized and autonomous power control in wireless uplink systems. Simulation results demonstrated significant performance improvement of DCNN + GA framework as compared to DNN+GA, in terms of real-time decision making. Specifically, in an offline optimization mode, DCNN took 0.28 s/decision as compared to DNN's 2 min/decision. Nevertheless, once the DCNN is trained on an optimized dataset, it performed 300× time faster than DNN as shown in **Figure 14**. More details on the optimization framework and dataset are comprehensively presented in Adeel et al. (2016).

We believe that our proposed DCNN is an optimal choice for future ultra-low power and energy efficient devices capable of handling massive arrays of mathematical calculations in real-time for both generalized learning and optimization applications. To acquire more flexibility for dealing with a variety of applications, we are currently implementing the DCNN regression model along with the designing and testing of other BrainOS modules. Lately, we will gather all the modules in one framework.

6. CONCLUSION

Our work was motivated both by the intellectual goal of creating a model of human intelligence that better resembles how the brain and cognition works as well as the related practical goal of building a more effective machine learning approach; an automatic-ML approach in particular. While ML and AI approaches have generally been premised on duplicating brain and cognitive functions, their varied suitability for different kinds of problems means that no one model is adequate for all problems. The way forward as many have supposed long ago, is to figure out how to select an approach (which might be one or a system of models), in an automatic, rational/explainable manner, for any particular problem at hand, to elicit optimal solutions to that problem. This means the selection and calibration (i.e., parameter selection) of a system/architecture of models. The BrainOS system described in this paper differs from existing automatic ML tools in what it automates and how it does so. It proceeds from existing taxonomies of approaches in the automatic ML literature, to develop its own architecture. Preliminary studies have convinced us that BrainOS can deal with complex high-level problems, such as natural language processing.

AUTHOR CONTRIBUTIONS

NH contributed to the design of the proposed approach. NC was responsible for the state-of-the-art review and the paper write-up. AA conceived and co-developed the original idea of DCNN and DCNN based optimized decision-making. KD contributed substantially to the writing and revising of the manuscript. AHo co-designed the proposed architectural model. AHu was responsible for the overall planning and direction of the proposed approach, including the DCNN framework.

ACKNOWLEDGMENTS

The authors would like to greatly acknowledge Mandar Gogate from Edinburgh Napier University and Hadi Larijani from Glasgow Caledonian University for their contributions in DCNN and optimization framework, which are cited here for reference.

REFERENCES

Adeel, A., Larijani, H., and Ahmadinia, A. (2016). Random neural network based novel decision making framework for optimized and autonomous power control in LTE uplink system. *Phys. Commun.* 19, 106–117. doi: 10.1016/j.phycom.2015.11.004

Bengio, Y., Courville, A., and Vincent, P. (2013). Representation learning: a review and new perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.* 35, 1798–1828. doi: 10.1109/TPAMI.2013.50

Bielza, C., and Larranaga, P. (2014). Bayesian networks in neuroscience: a survey. *Front. Comput. Neurosci.* 8:131. doi: 10.3389/fncom.2014.00131

Bredeche, N., Shi, Z., and Zucker, J. D. (2006). Perceptual learning and abstraction in machine learning: an application to autonomous robotics. *IEEE Trans. Syst. Man Cybern. C Appl. Rev.* 36, 172–181. doi: 10.1109/TSMCC.2006.871139

Cambria, E., White, B., Durrani, T., and Howard, N. (2014). Computational intelligence for natural language processing. *IEEE Comput. Intell. Mag.* 9, 19–63. doi: 10.1109/MCI.2013.2291686

Chouikhi, N., Ammar, B., Rokbani, N., and Alimi, A. M. (2017). PSO-based analysis of echo state network parameters for time series forecasting. *Appl. Soft Comput.* 55, 211–225. doi: 10.1016/j.asoc.2017.01.049

Chouikhi, N., Fdhila, R., Ammar, B., Rokbani, N., and Alimi, A. M. (2016). "Single- and multi-objective particle swarm optimization of reservoir structure in echo state network," in *International Joint Conference on Neural Networks (IJCNN)* (Vancouver, BC), 440–447.

Dias, F. M., Antunes, A., and Mota, A. M. (2004). Artificial neural networks: a review of commercial hardware. *Eng. Appl. Artif. Intell.* 17, 945–952. doi: 10.1016/j.engappai.2004.08.011

Feurer, M., Klein, A., Eggensperger, K., Springenberg, J. T., Blum, M., and Hutter, F. (2015). "Efficient and robust automated machine learning," in *Neural Information Processing Sytstem,* eds F. Hutter, L. Kothoff, and J. Vanschoren (Cham: Springer), 113–114.

Henriquez, J., and Kristjanpoller, W. (2019). A combined independent component analysis–neural network model for forecasting exchange rate variation. *Appl. Soft Comput.* 83:105654. doi: 10.1016/j.asoc.2019.105654

Hernandez, C., Sanz, R., Ramirez, J. G., Smith, L. S., Hussain, A., Chella, A., et al. (2010). "From brains to systems," in *Brain-Inspired Cognitive Systems*, eds C. Hernandez, R. Sans, J. Gomez Ramirez, L. S. Smith, A. Hussain, A. Chella, and I. Aleksander (New York, NY: Springer-Verlag), 1–250.

Howard, N., Adeel, A., Gogate, M., and Hussain, A. (2019). *Deep Cognitive Neural Network (DCNN)*. US Patent App. 16/194,721. Washington, DC: U.S. Patent and Trademark Office.

Howard, N., and Hussain, A. (2018). The fundamental code unit of the brain: towards a new model for cognitive geometry. *Cogn. Comput.* 10, 426–436. doi: 10.1007/s12559-017-9538-5

Howard, N., and Lieberman, H. (2014). Brainspace: relating neuroscience to knowledge about everyday life. *Cogn. Comput.* 6, 35–44. doi: 10.1007/s12559-012-9171-2

Hutter, F., Kotthoff, L., and Vanschoren, J. (2019). *Automated Machine Learning*. Cham: Springer.

Kacha, A., Grenez, F., Rafael, J., Arroyave, O., and Schoentgen, J. (2020). Principal component analysis of the spectrogram of the speech signal: interpretation and application to dysarthric speech. *Comput. Speech Lang.* 59, 114–122. doi: 10.1016/j.csl.2019.07.001

Kotsiantis, S. B. (2013). Decision trees: a recent overview. *Artif. Intell. Rev.* 39, 261–283. doi: 10.1007/s10462-011-9272-4

Landry, M. (2018). *Machine Learning With R and H2O*. Mountainview, CA: H2O.ai, Inc.

LeCun, Y., Cortes, C., and Burges, C. (1998). *Mnist Dataset*. Available online at: http://yann.lecun.com/exdb/mnist

Louridas, P., and Ebert, C. (2016). Machine learning. *IEEE Softw.* 33, 110–115. doi: 10.1109/MS.2016.114

Mountrakis, G., Im, J., and Ogole, C. (2011). Support vector machines in remote sensing: a review. *J. Photogramm. Rem. Sens.* 66, 247–259. doi: 10.1016/j.isprsjprs.2010.11.001

Saxena, A., Prasad, M., Gupta, A., Bharill, N., Patel, O. P., Tiwari, A., et al. (2017). A review of clustering techniques and developments. *Neurocomputing* 267, 664–681. doi: 10.1016/j.neucom.2017.06.053

Vinson, B. (2018). *Machine Learning With Google Cloud Platform*. Technical report, Google Cloud.

Wang, H., and Yan, X. (2015). Optimizing the echo state network with a binary particle swarm optimization algorithm. *Knowl. Based Syst.* 96, 182–193. doi: 10.1016/j.knosys.2015.06.003

Wang, Y., Widrow, B., Zadeh, L. A., Howard, N., Wood, S., Bhavsar, V. C., et al. (2016). Cognitive intelligence: deep learning, thinking, and reasoning by brain-inspired systems. *Int. J. Cogn. Inform. Natural Intell.* 10, 1–20. doi: 10.4018/IJCINI.2016100101

Xu, J., Xiang, L., Liu, Q., Gilmore, H., Wu, J., Tang, J., et al. (2016). Stacked sparse autoencoder (SSAE) for nuclei detection on breast cancer histopathology images. *IEEE Trans. Med. Imaging* 35, 119–130. doi: 10.1109/TMI.2015.2458702

Yao, Q., Wang, M., Chen, Y., Dai, W., Hu, Y. Q., Li, Y. F., et al. (2019). *Taking the Human Out of Learning Applications: A Survey on Automated Machine Learning*. Technical Report, arXiv:1810.13306 [cs.AI], ArXiV.

Yin, P.-Y. (2008). *Theory of Cognitive Pattern Recognition*, Chapter Pattern Recognition Techniques. Vienna: I-tech.

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.

frontiers in Psychology

ORIGINAL RESEARCH
published: 13 March 2020
doi: 10.3389/fpsyg.2020.00415

High Cognitive Flexibility Learners Perform Better in Probabilistic Rule Learning

*Xia Feng, Garon Jesse Perceval, Wenfeng Feng and Chengzhi Feng**

Department of Psychology, School of Education, Soochow University, Suzhou, China

Cognitive flexibility reflects the ability to switch quickly between tasks or stimulus sets, which is an important feature of human intelligence. Researchers have confirmed that this ability is related to the learners' academic achievement, cognitive ability, and creativity development. The number-letter switching task is an effective tool for measuring cognitive flexibility. Previous studies have found that high flexibility individuals perform better in rule-based tasks such as the Iowa Gambling Task. It is not clear whether highly flexible learners have learning advantages when the rule tasks involve probabilistic cues. Using an inter-individual differences approach, we examined whether cognitive flexibility, as assessed by the number-letter task, is associated with the learning process of a probabilistic rule task. The results showed that the high flexibility group reached a higher level of rule acquisition, and the accuracy during the post-learning stage was significantly higher than the low flexibility group. These findings demonstrate that cognitive flexibility is associated with the performance after the rule acquisition during the probabilistic rule task. Future research should explore the internal process of learning differences between high and low flexibility learners by using other technologies across multiple modes.

Keywords: cognitive flexibility, rule learning, probability, switch cost, reward

OPEN ACCESS

Edited by:
Liang Feng,
Chongqing University, China

Reviewed by:
Masataka Watanabe,
Tokyo Metropolitan Institute of Medical Science, Japan
Anthony John Porcelli,
Medical College of Wisconsin, United States

****Correspondence:***
Chengzhi Feng
fengchengzhi@suda.edu.cn

Specialty section:
This article was submitted to Decision Neuroscience, a section of the journal Frontiers in Psychology

Received: *13 August 2019*
Accepted: *24 February 2020*
Published: *13 March 2020*

Citation:
Feng X, Perceval GJ, Feng W and Feng C (2020) High Cognitive Flexibility Learners Perform Better in Probabilistic Rule Learning. Front. Psychol. 11:415. doi: 10.3389/fpsyg.2020.00415

INTRODUCTION

As a core component of executive functioning (EF), cognitive flexibility has attracted much attention in psychological research. Research from various fields has investigated the internal mechanism underlying cognitive flexibility. Animal-based research has explored the underlying mechanisms of this function from an anatomical neurology perspective (Darby et al., 2018). Developmental psychologists focus on the training and growth of cognitive flexibility in children and adolescents (Dajani and Uddin, 2015). Studies of patients with neurological impairment also provide a window for exploring internal mechanisms (Lange et al., 2017). Despite extensive attention and research, there is still no clear common definition of cognitive flexibility, which can influence how this construct is operationalized in research (Müller et al., 2014). Based on the understanding that cognitive flexibility refers to "the ability of switching between tasks and stimulus sets in a quick and flexible manner" (Diamond, 2013; Müller et al., 2014), previous studies measured cognitive flexibility using scales or cognitive tasks (e.g., Wisconsin Card Sorting Test, WCST; task- switching paradigms) and researches have proved subjects with different levels of cognitive flexibility have different behavioral and neural characteristics (Müller et al., 2014). Although there

is yet no clear conclusion about the mechanisms underlying cognitive flexibility, many researchers hold that cognitive flexibility is a prerequisite for many psychological functions, and it is one of the most important factors affecting intelligence and creativity (Diamond, 2013). For children, cognitive flexibility is a significant predictor of academic performance (Stad et al., 2018). High cognitive flexibility learners, including children (Lehto and Elorinne, 2003) and adults (Dong et al., 2016), usually show better performance on learning task, such as the Iowa Gambling task, which involves decision-making under uncertainly and has partly common neural mechanisms with rule learning (Hartstra et al., 2010).

Rule learning is based on stimulus patterns and feedback of behavioral outcomes to discover the relationship between operations and outcomes. Upon mastering the relationship, learners develop guidelines for subsequent behavioral choices, allowing them to further predict the corresponding results. This process enables an individual to recognize new information that expands upon existing knowledge. From the perspective of cognitive psychology, this process can be summarized as follows: the brain encodes stimuli, stimuli and feedback is used to construct rules, these rules are used to predict subsequent stimuli, and these rules are also applied to other similar stimuli (Dehaene et al., 2015). Hypothesis testing is at the core of rule learning (Klayman and Ha, 1989; Liu et al., 2015). During rule learning tasks, rule learning can enter the application stage smoothly if the hypothesis is successfully tested. If the hypothesis cannot explain the stimulus sequence, it must undergo further revision by the participant. This process will be repeated until the correct hypothesis is found or the experiment has ended. Successful hypothesis testing requires flexible switching among multiple possible hypotheses. High cognitive flexibility learners show better abstract induction, working memory, and feedback learning abilities during the Iowa Gambling task (Dong et al., 2016), which has some common neural basis of rule learning (Hartstra et al., 2010). We speculate that high-flexibility individuals may display more accurate and faster rule acquisition during rule learning as a result of their cognitive advantages.

In contrast to deterministic rule learning, there is no one-to-one matching relationship between cues and results in probabilistic rule learning. To use weather forecasting as an example, a "dark cloud" cue may result in "rain" in 70% of cases. Yet, in 30% of cases, the result is "cloudy." Therefore, the cue "dark cloud" cannot be fixed to a certain attribute (i.e., "rain"), and the same reaction to "dark cloud" may be reinforced as "rain" or "cloudy." It is impossible for learners to achieve complete error-free performance, and they eventually accept certain inevitable mistakes (Craig et al., 2011). This study aims to explore, for the first time, whether healthy adults with high cognitive flexibility show an advantage during a probabilistic rule learning task, just as in other rule-based learning tasks (i.e., Iowa Gambling Tasks; Dong et al., 2016). The WCST is perhaps the most widely used tool to measure cognitive flexibility in neuropsychology at present. However, compared to WCST, task-switching paradigms can provide a more pure measurement of cognitive flexibility by reducing the demand for working memory, classified learning and rule reasoning (Buchsbaum et al., 2005; Lange et al., 2018). This study uses the classical "number-letter task" to measure learners' cognitive flexibility. Performance on the "number-letter task" (switch cost) will be used to divide participants into high and low cognitive flexibility groups, and their dynamic learning characteristics in different stage of probabilistic rule learning will be explored.

MATERIALS AND METHODS

Participants

Three hundred and ten undergraduates from Soochow University completed the number-letter task. Data from 13 subjects was excluded for responding too quickly (RT below 100 ms), giving repeated responses, or misunderstanding the instructions. Data from 297 subjects (60 males) aged from 17 to 26 (M = 18.7, SD = 1.5) were used for further grouping. All subjects were right-handed, had normal or correct-to-normal vision, and no reported cognitive impairment. None of the subjects had participated in similar experiments. Participants were reimbursed according to their performance in the coin-searching task. All subjects had given written informed consent. The study protocol was approved by "the Ethical Committee of Soochow University."

Materials and Procedure

Number-Letter Task

In the classic number-letter task (Rogers and Monsell, 1995), a letter plus a number (e.g., 2U or M5) appears in a quadrant at the center of the screen. Letters are either vowels (A/E/I/U) or consonants (G/K/M/R) and numbers are either odd (3/5/7/9) or even (2/4/6/8). A letter and a number are randomly combined to form number-letter stimulus pairs. In the current study, the task consisted of practice and formal trials.

Practice trials

Letter, number and number-letter joint judgments were included in the practice trials. The sequence of letter and number judgments was balanced among subjects. For letter judgments, 32 trials (16 trials of consonants, half of them paired with an odd number) were included. Subjects were instructed to press 'E' or 'I' as quickly and accurately as possible to determine whether the letters were consonants or vowels. During the letter judgment trials, the stimulus pairs always appeared in the upper two quadrants. After an incorrect response, " × " would appear and the participants were instructed to re-press the correct key. The number judgment trials differed from the letter judgment trials in that the stimulus pairs always appeared in the two bottom quadrants and the subjects were required to determine whether the number was even or odd. In the combined number-letter trials, the stimulus pairs were presented clockwise one by one in each quadrant, and the number or letter was not the same as the previous one. For the stimulus pairs appearing in the upper two quadrants, letter judgments were needed, otherwise number judgments were required. Only when the accuracy rate was higher than 80% could participants enter the formal trials.

Formal experiment

The formal experiment consisted of 128 trials of combined number-letter judgments. " × " without chance of correction would appear after incorrect responses. When the stimulus pair jumped from the first quadrant to the fourth quadrant, the subjects needed to switch from letter judgment to number judgment accordingly. Similarly, when the stimulus pair jumped from the third to the second quadrant, the judgment should change from number to letter. We classified these as switching trials. When the stimulus pairs jumped from the fourth to the third quadrant or the second to the first quadrant, there was no need for task type switching. We classified these as non-switching trials.

The latency difference between switching and non-switching trials was regarded as the switch cost (the switch cost of latency = the average latency of the correct response in the switching trials – the average latency of the correct response in the non-switching trials). Switch cost was used to distinguish learners with high and low cognitive flexibility. High flexibility participants had a smaller switch cost, and the switch cost of low flexibility participants was greater. Participants that with switch cost scores in the upper 27th percentile were included in the low flexibility group, subjects with scores in the lower 27th percentile were included in the high flexibility group. Only these high and low flexibility groups completed the coin searching task.

Coin-Searching Task

The coin-searching task is similar to that of Bellebaum and Daum (2008). E-prime 2.0 was used to program and run the task. The stimuli were presented on a 17′ computer monitor with a resolution of 1024 × 768 pixels. Each participant sat approximately 57 cm from the screen. Responses were recorded via 'F' and 'J' keys on a computer keyboard. There were 12 regular color blocks [RGB_{red} (255, 0, 0), RGB_{white} (255, 255, 255)] on the left and right sides of a black [RGB (0, 0, 0)] background. The visual angles of stimulation were shown in **Figure 1**.

The total number of red blocks was equal on both sides, with either 4 or 8 cases. The number of red blocks in the right column on both sides was either 0 or 2 or 4 or 6. If there was no red block in the right column of the selected side, the reward probability is 0. Similarly, the reward probability was 2/6 (1/3) for two red blocks, 4/6 (2/3) for four red blocks and 6/6 (1) for six red blocks of the selected side. The combinations of reward probabilities in single trial and the number of trials are shown in **Table 1**.

Prior to beginning the task, the participants were told that: (1) Red and white blocks would appear on both sides of the fixation cross [The subjects were not informed that the total number of red blocks (4 or 8) was equal on both sides]; (2) A coin was hidden in one of the 12 colored blocks; (3) The task was to guess whether a coin was more likely to be hidden under a red block on the left ("F" key) or right ("J" key) side, and there was no need to judge a specific location for the coin; (4) There was a "rule" determining the reward and that correct identification and application of this rule would result in a greater reimbursement at study completion. Participants were not told beforehand the exact reimbursement amounts (correctly identifying rule: 50RMB, failure to identify rule: 40RMB).

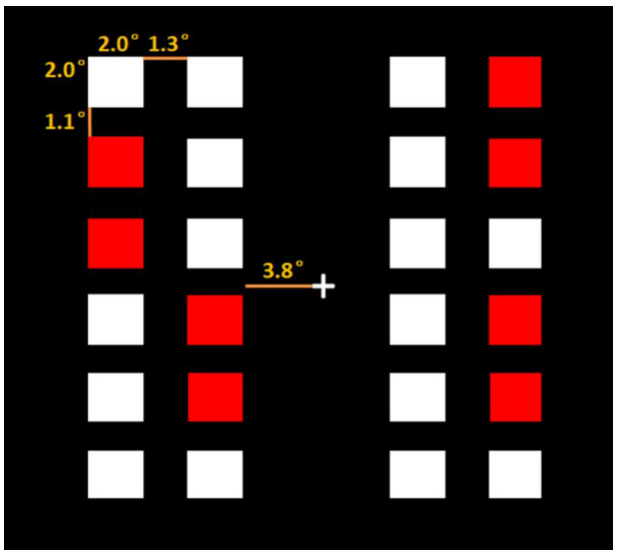

FIGURE 1 | The visual angles of stimulation.

TABLE 1 | The combinations of reward probabilities in a single trial and number of trials.

Type of stimulus		Number of trials (Including left-right balance)
Reward probability of the right column for one side	Reward probability of the right column for the other side	
0	1/3	120/540
0	2/3	90/540
1/3	2/3	240/540
1/3	1	90/540

The fixation point was presented with a random duration between 900 and 1100 ms. Then the color blocks were presented on two sides of the fixation cross. After the fixation point flashes, participants made a choice by pressing the "F" or "J" key with the left or right index finger respectively. The minimum reaction time was 1000 ms and the maximum was 2700 ms. The selected side would present for another 500 ms. After a 400—600 ms interval (black screen), feedback was presented for 500 ms. A triangle or a hexagon was represented to indicate reward or no reward respectively, which was balanced between subjects. There were three blocks, with 540 trials in total. Block 1 and Block 3 were identical. In Block 2 (trial 181–360), the participants were given additional feedback indicating the exact location of the coin: in the reward trials, the coin would appear under one of the red blocks in the right column of the selected side; and in the non-reward trials, the coin would appear under a white block in the right column of the selected side. The exact location of coins was determined at random. The procedure is shown in **Figure 2**. After finishing the experiment, subjects completed a questionnaire assessing rule identification success. In the questionnaire, subjects firstly described the rule he/she had found in as much detail as

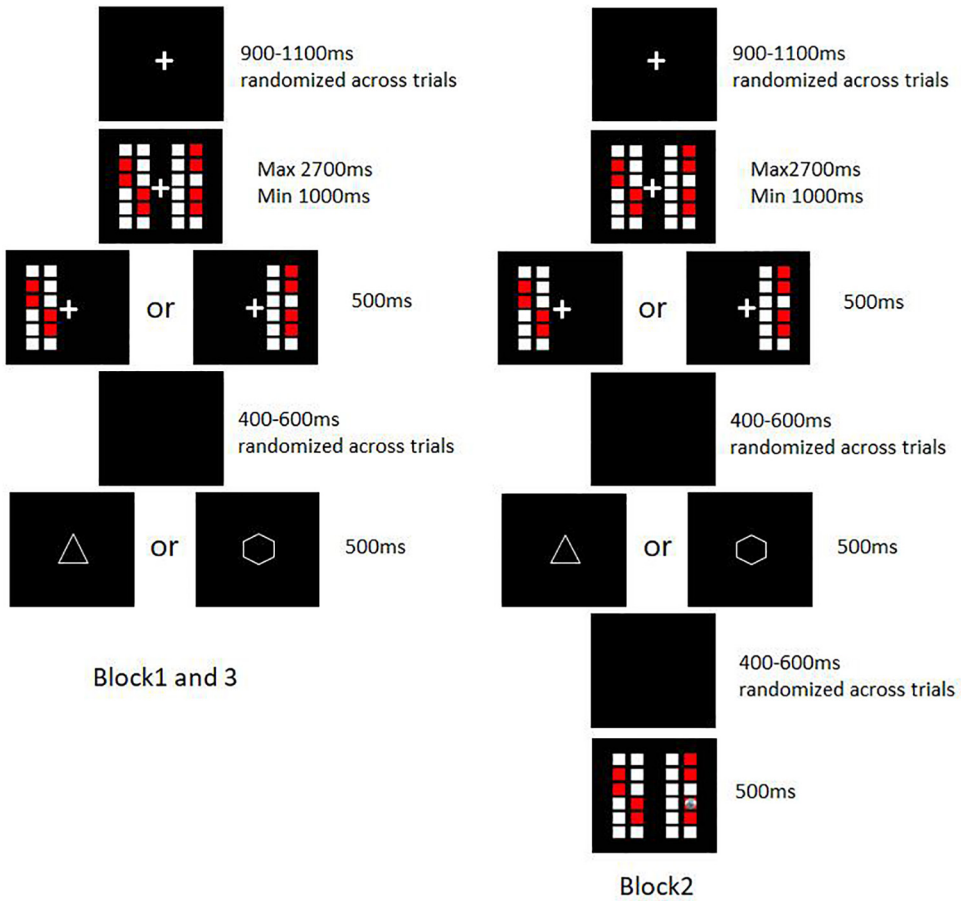

FIGURE 2 | Schematic illustration of an exemplar trial in the coin-searching task.

possible, and then evaluated his/her own confidence in the rules described before with one of six points (6-quite sure, 5-pretty sure, 4-a little sure, 3-a little not sure, 2-pretty not sure, 1-quite not sure).

The rule of the task is that the reward probability is determined by the ratio of red blocks in the right column of the chosen side. Choosing the side with a higher ratio of red blocks in the right column (i.e., a larger number of red blocks) will result in a higher likelihood of receiving a reward. In **Figure 2**, the reward probability is 1/3 for the left side and 2/3 for the right. Although there is a 1/3 probability of receiving no-reward upon choosing the right side, the right side is the correct choice since it has a higher chance of reward than the left side.

To control for potential left/right dominance effects, half of the participants were instructed to make their correct choice according to the comparison of reward probability in the two left columns of each side.

Data Analysis

For the number-letter task, participants were ranked according to their switch cost. The first 27% (smaller switch cost) of the participants were assigned to the high flexibility group, and the last 27% were assigned to the low group.

In order to analyze the dynamic learning characteristics of probabilistic rule tasks, a window analysis with 20 window lengths and 1 step length was used. A stable performance criterion of $\geq$ 80% correct choices ($\geq$ 16 correct responses within 20 successive trials) was considered successful task rule learning (learning baseline) (Bellebaum and Daum, 2008). For the subjects who learned the rule, the crossover point of the dynamic learning curve and the learning baseline (as shown in **Figure 3B**) was used as the key point to distinguish pre- and post-learning stages. If the subjects did not find any rule during the experiment, all responses were regarded as pre-learning in the subsequent analysis; similarly, if a participant learned the rule at the beginning of the experiment participant responses were regarded as post-learning only. A mixed analysis of variance (ANOVA) [2(high/low flexibility) $\times$ 4(probability pair) $\times$ 2(learning stage)] was adopted for the accuracy and latency scores of the coin-searching task. High/low cognitive flexibility was a between-subjects factor, probability pair (0-1/3, 0-2/3, 1/3-2/3, 1-1/3) and learning stage (pre-and post-learning stage) were within-subjects factors. Statistical analysis was performed using SPSS22.0.

TABLE 2 | Switching cost (ms) for high and low cognitive flexibility groups.

	Minimum	Maximum	M ± SD	Lower27th percentile	Upper 27th percentile
All participants (297)	137.5	2085.3	813.4 ± 389.2	525.3	1041.1
High cognitive flexibility group (39)	137.5	517.1	357.3 ± 109.9		
Low cognitive flexibility group (37)	855.9	2069.6	1161.1 ± 285.5		

RESULTS

Overview of the Data

High and low cognitive flexibility groups were created according to switch cost scores on the number-letter task. Participants with a score less than 525.3 ms were assigned to the high flexibility group, and participants with a score greater than 1041.1 ms were assigned to the low flexibility group (**Table 2**). 39 (15 males, age_{39} = 20.2 ± 2.0) high flexibility and 37 (5 males, age_{37} = 19.3 ± 1.6) low flexibility participants were willing to participate further. Given that this study comprises part of the first author's doctoral thesis, there is a difference between the total number of participants and participants assigned to the high and low cognitive flexibility groups.

27 out of 39 (69.2%) participants in the high flexibility group and 12 out of 37 (44.4%) participants in the low flexibility group identified the correct rule (**Figure 4**). Pearson Chi-square test showed that the number of rule discoverers in the high group was significantly higher than that in low flexibility group [χ^2 = 10.3, df = 1, p = 0.001].

In order to describe the dynamic learning process more closely, we plotted the learning curves of each participant. A typical rule learner and a non-learner are shown in **Figure 3**, and the average learning curves of the four groups (High CF – learner group: 27, and 3 out of 27 participants only had the post-learning stage since they had found the right rule with a few trials; high CF – non-learner group: 12; low CF – learner group: 12; low CF – non-learner: 25) are shown in **Figure 5**. The average learning point of all rule learners was 251 trials, which is the 71st trial in block 2 (this block contains the specific feedback about coin position). Both the high and low flexibility groups reached their learning point in the second block (high CF group – 244, low CF group – 257).

Further, the confidence scores for the described rules of four groups (high CF – learner, high CF – non-learner, low CF – learner, low CF – non-leaner) and two groups (learner, non-learn)were compared. One-way analysis of variance for four groups showed that there was significant difference among four groups [$F(3,72)$ = 3.108, p < 0.05], A Least-Squares Difference (LSD) test revealed high CF – learners' confidence score [M ± SD = 4.6 ± 1.2] was significantly higher than that of low CF – non-learners [M ± SD = 3.6 ± 1.4]. Independent sample t-test showed the confidence score of learner group (M = 4.4, SD = 1.1) was significantly higher than non-learner group (M = 3.7, SD = 1.3) [$t(76)$ = −2.761, p < 0.05, Cohen's d = −0.626].

Accuracy Analysis

As mentioned above, the participants received feedback indicating reward or non-reward during the experiment. A correct response was recorded when participants chose the side with the greater number of red blocks in the right column (or left column, n = 38). Timed-out and unresponsive trials (1.16% of trials) were not included in the analysis. Accuracy scores were analyzed (accuracy = number of correct responses/total number of responses). A mixed measures ANOVA [2(high/low

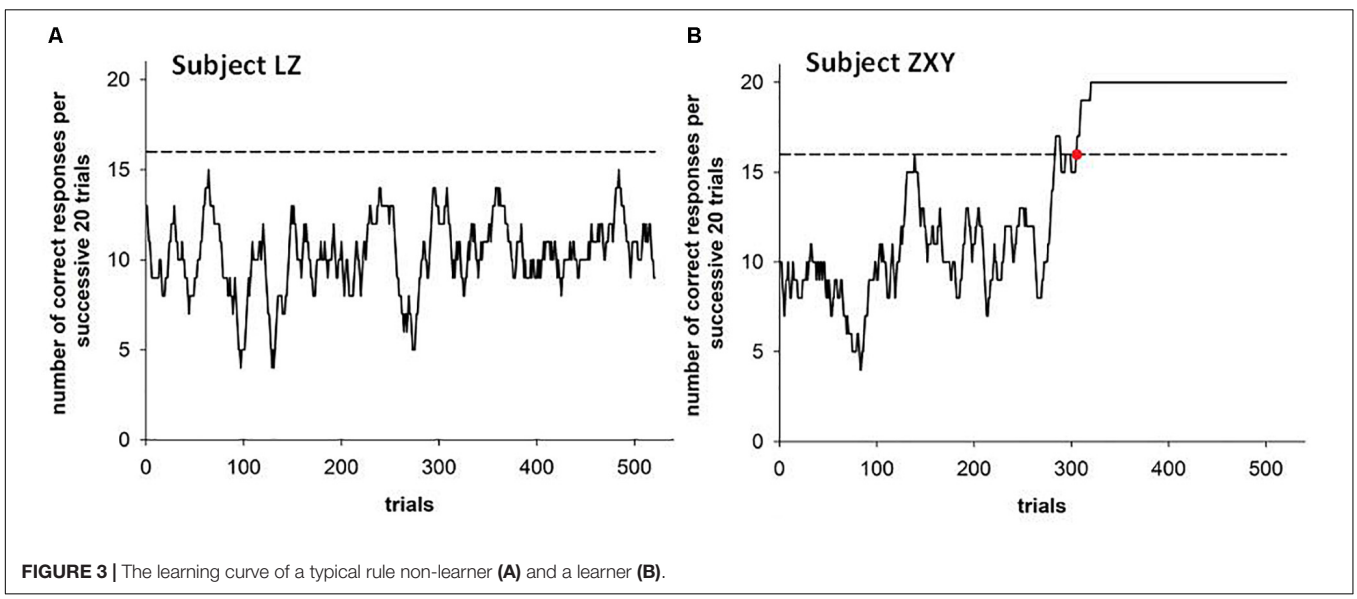

FIGURE 3 | The learning curve of a typical rule non-learner **(A)** and a learner **(B)**.

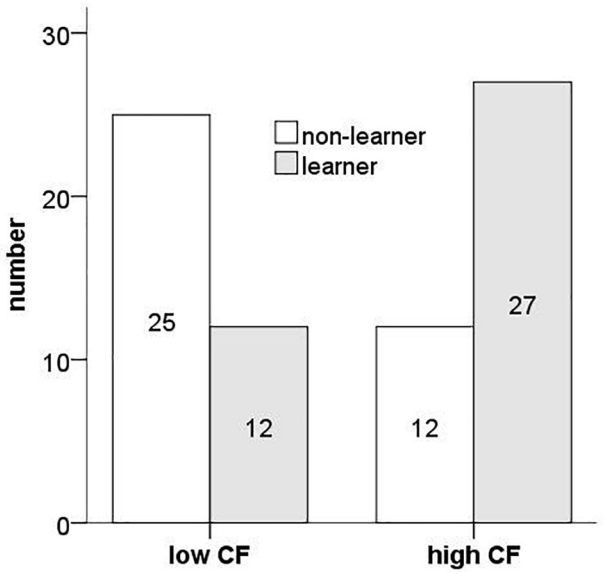

FIGURE 4 | The number of rule discovers for high and low cognitive flexibility groups.

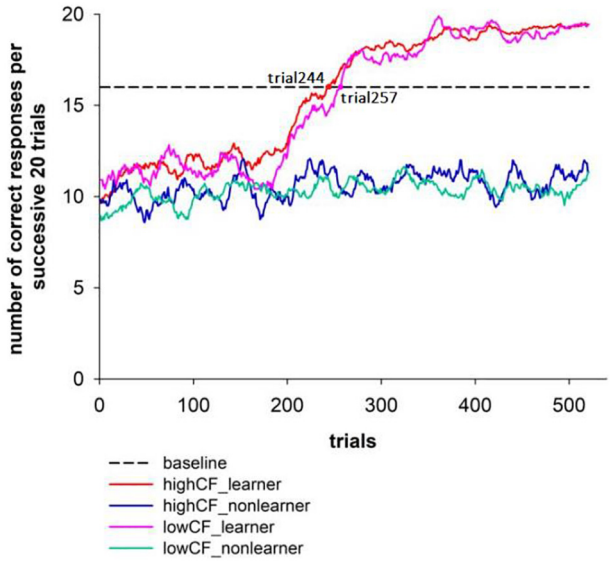

FIGURE 5 | The average learning curves of the four groups.

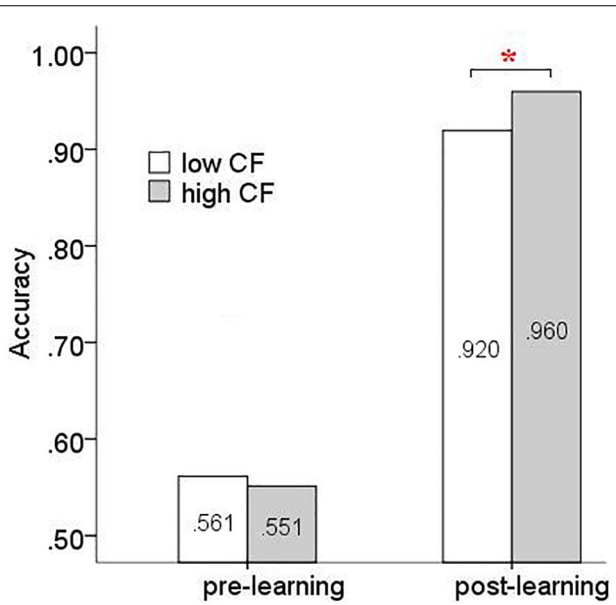

FIGURE 6 | The accuracy of high and low CF groups during different learning stages (*$p < 0.05$).

flexibility) × 4(probability pair) × 2(learning stage)] for participant accuracy scores showed a significant main effect of learning stage [$F(1,34) = 709.728$, $p < 0.001$, $\eta_p^2 = 0.954$] and probability pair [$F(3,102) = 10.942$, $p < 0.001$, $\eta_p^2 = 0.243$]. The interaction between learning stage and group was marginally significant [$F(1,34) = 3.051$, $p = 0.090$, $\eta_p^2 = 0.082$]. All other effects were not significant. An analysis of simple effects of high/low flexibility group and learning stage on accuracy showed a significant difference between the two groups after rule acquisition only [$F(1,34) = 12.651$, $p < 0.05$, $\eta_p^2 = 0.271$] (**Figure 6**). In order to further investigate the differences between groups after rule acquisition, a mixed measures ANOVA [4(probability pair) × 2(high/low flexibility)] was performed on the post-learning data. We observed a significant main effect of probability pair [$F(3,111) = 19.889$, $p < 0.001$, $\eta_p^2 = 0.350$] and group [$F(1,37) = 11.662$, $p < 0.05$, $\eta_p^2 = 0.240$]. However, the probability pair × high/low flexibility interaction was not significant (**Figure 7A**). Next we merged trials from the four probability pair conditions (0, 1/3; 0, 2/3; 1/3, 2/3; 1/3, 1) by averaging the accuracy of conditions with equal probability difference values. Two probability difference conditions were created: 1/3 probability difference (0, 1/3; 1/3, 2/3) and 2/3 probability difference (0, 2/3; 1/3, 1). A mixed measures ANOVA [2(probability difference) × 2(high/low flexibility)] showed a significant main effect of probability difference [$F(1,37) = 48.914$, $p < 0.001$, $\eta_p^2 = 0.569$] and high/low flexibility [$F(1,37) = 11.662$, $p < 0.05$, $\eta_p^2 = 0.240$]. A significant probability difference × high/low flexibility interaction was also observed [$F(1,37) = 4.875$, $p < 0.05$, $\eta_p^2 = 0.116$] (**Figure 7B**). Simple effects analysis showed there was a significant difference between high/low flexibility groups for the 1/3 probability difference [$F(1,37) = 11.314$, $p < 0.05$, $\eta_p^2 = 0.234$], and a marginally significant difference [$F(1,37) = 3.482$, $p = 0.07$, $\eta_p^2 = 0.086$] for the 2/3 probability difference.

Latency Analysis

After deleting timed-out and unresponsive trials (1.16% of trials), a mixed measures ANOVA [2(learning stage) × 4(probability pair) × 2(high/low flexibility)] showed a significant main effect of learning stage [$F(1,34) = 84.561$, $p < 0.001$,

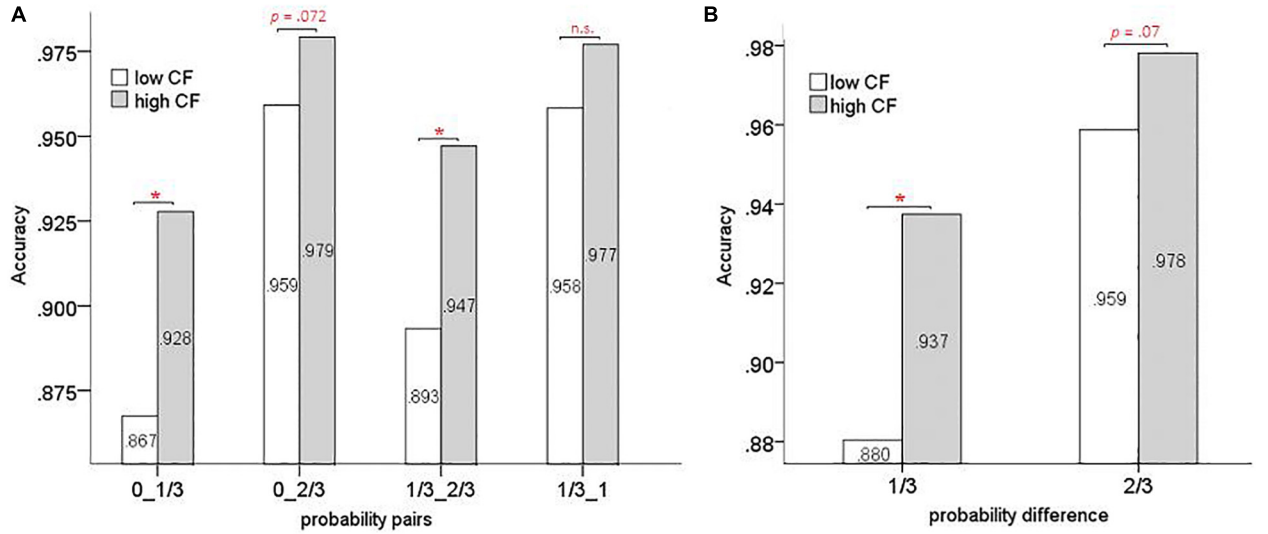

FIGURE 7 | The accuracy of high and low CF groups during post-learning stage. **(A)** Accuracy in different probability pairs. **(B)** Accuracy in different probability differences (*$p < 0.05$).

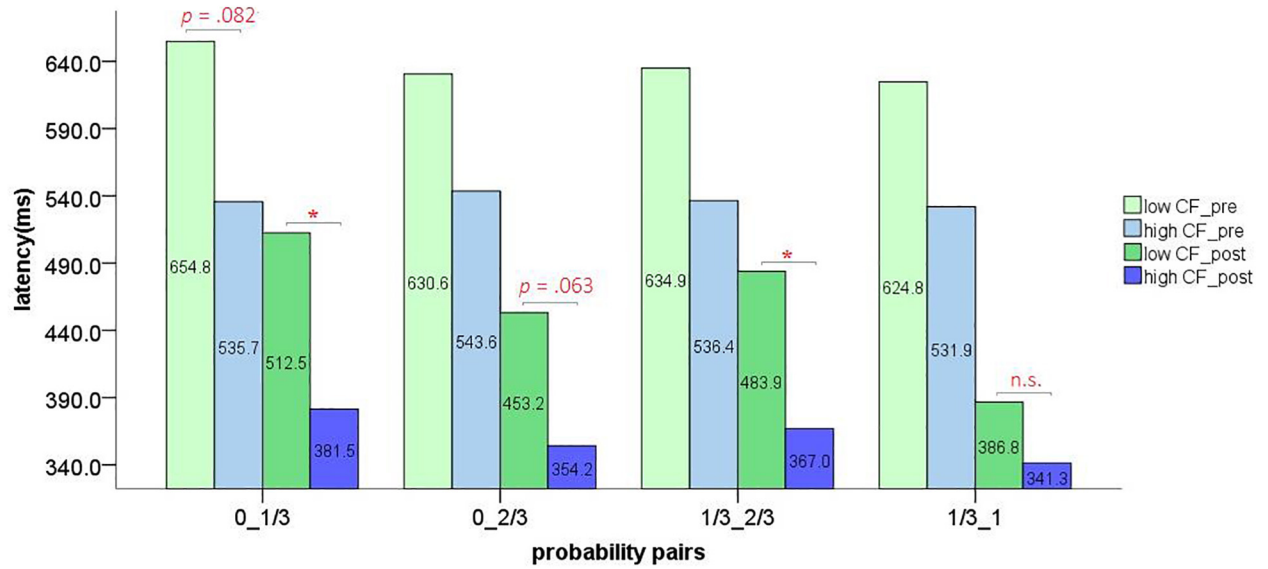

FIGURE 8 | The latency of high and low cognitive flexibility groups for pre- and post-learning stages in different probability pairs (*$p < 0.05$).

$\eta_p^2 = 0.713$] and probability pair [$F(3,102) = 10.030$, $p < 0.001$, $\eta_p^2 = 0.228$]. A marginally significant main effect of group was also observed [$F(1,34) = 3.395$, $p = 0.074$, $\eta_p^2 = 0.091$]. Significant interactions between learning stage and probability pair [$F(3,102) = 7.728$, $p < 0.001$, $\eta_p^2 = 0.185$], and probability pair and high/low flexibility were observed [$F(3,102) = 3.203$, $p < 0.05$, $\eta_p^2 = 0.086$]. The three-way interaction was marginally significant [$F(3,102) = 2.181$, $p = 0.095$, $\eta_p^2 = 0.060$] and the interaction between learning stage and group was not significant [$F(1,34) = 0.001$, $p = 0.973$, $\eta_p^2 = 0.000$]. Simple effects analysis of the four probability pairs in the different learning stages showed a significant difference among four probability pairs after rule learning [$F(3,32) = 6.752$, $p < 0.05$, $\eta_p^2 = 0.388$]. A Least-Squares Difference (LSD) test revealed significant differences between 0-1/3 and 1/3-1 probability pairs ($p < 0.05$) during post-learning. Simple effects analysis of the four probability pairs for the two flexibility groups showed that only the low flexibility group had significantly different latencies in the different probability pairs [$F(3,32) = 5.571$, $p < 0.05$, $\eta_p^2 = 0.343$]. The latency of high and low cognitive flexibility groups for pre- and post-learning stages in different probability pairs was shown in **Figure 8**. In order to further describe the reaction time differences within the low-flexibility group, a 2 × 4 repeated measures ANOVA

[2(learning stage) × 4(probability pair)] was conducted on the data of the low group only. All main and interaction effects were significant [learning stage: $F(1,11) = 24.340$, $p < 0.001$, $\eta_p^2 = 0.689$; probability pair: $F(3,33) = 6.244$, $p < 0.05$, $\eta_p^2 = 0.362$; learning stage × probability pair interaction: $F(3,33) = 3.623$, $p < 0.05$, $\eta_p^2 = 0.248$]. Simple effects analysis showed a marginally significant difference among probability pairs after learning acquisition [$F(3,9) = 3.250$, $p = 0.074$, $\eta_p^2 = 0.520$]. Specifically, there were (marginally) significant differences between 0_1/3 and 0_2/3 ($p = 0.089$), 1/3_2/3 and 1/3_1 ($p = 0.073$), and 0_1/3 and 1/3_1 ($p < 0.05$). Additionally, the data of the 1/3 and 2/3 probability differences after rule acquisition were averaged respectively. A paired sample *t*-test showed a significant difference between 1/3 ($M = 498.20$ ms, $SD = 176.54$ ms) and 2/3 ($M = 419.97$ ms, $SD = 135.34$ ms) probability differences [$t(11) = 3.406$, $p < 0.05$, Cohen's $d = 0.24$]. After successful rule learning, the low flexibility subjects responded faster to stimuli with a higher probability difference.

DISCUSSION

In the present study, learners were grouped into high and low cognitive flexibility groups based on their performance on the number-letter switching task. The learning characteristics of the two groups in a rule task with probabilistic cues were preliminarily explored. Behavioral data analysis showed that the differences between the two groups are mainly manifested in the following three points: (1) The high CF group showed a higher rate of rule acquisition, which partially verified our hypothesis that high cognitive flexibility learners would show more accurate rule acquisition during rule learning. However, the two groups showed very similar average rule acquisition points, the high flexibility group did not show faster rule acquisition as predicted. (2) The high flexibility group showed significantly higher accuracy than the low flexibility group after rule acquisition. (3) After rule learning, the low-flexibility group showed significantly different response latencies across the four probabilistic pairing conditions, while the high-flexibility group did not show such differences.

The rule acquisition speed of the high flexibility group was not faster than that of the low flexibility group, which may be related to the reward feedback provided in block 2. In order to reduce task difficulty, the exact coin position was shown to the learners during the second block (for reward feedback, the coin was shown under the red block on the dominant side; for non-reward feedback, the coin was shown under the white block). This feedback aided the subjects in identifying the existence of a "dominant side." Combining this knowledge with the relationship between reward and red blocks, the subjects could identify the basis for response more easily. This design reduces the difficulty of the task (Bellebaum and Daum, 2008) which has been proved in previous study. However, we guess this design of exact coin position in block 2 may have weakened the inter-group differences of the high and low flexibility groups to some extent. It is possible to find the correct basis of reaction from the specific position of the coin for both high and low groups, and the average learning curves in **Figure 5** also supported this. The cue of exact location of the coin provided participants with a shortcut to the task, which was open to both groups.

Even if the involvement of block design made the task became easier, the overall rule learning rate of the task in present study was 51.3% [(21 + 12)/(39 + 37)]. That is, only half of the participants identified the correct rule, which is lower compared with previous studies using this paradigm [66.7% (18/27)] (Bellebaum and Daum, 2008). This may be related to differences in participant cognitive flexibility levels: in Bellebaum's study the level of flexibility was not a key factor of interest and so the overall level of cognitive flexibility in their sample is unknown. In the present study, 48.7% of subjects had low flexibility. This relatively large proportion of low flexibility subjects may explain the overall lower learning rate in our study.

After rule learning, the accuracy of the high flexibility group was higher than that of the low flexibility group for the 1/3 and 2/3 probability differences. Especially for the 1/3 probability difference, the advantage of the high flexibility group was more pronounced. This finding may indicate that the high-flexibility group used the response criterion correctly more frequently and that this group may have a greater mastery and confidence surrounding rule learning. Using the Iowa Gambling task, Dong et al. (2016) showed that people with high flexibility showed explicit knowledge of task rules whereas the low group did not, which is consistent with the higher response accuracy of the high flexibility group in this study. Their research also showed that the lower P300 amplitude of the low flexibility group in the stimulus selection evaluation stage might be due to the lower cognitive and abstract generalization abilities or working memory abilities of the low flexibility group. The present findings extend the advantage of the high cognitive flexibility group to probabilistic reward learning, that is, the high flexibility group could distinguish stimuli with little differences in probability more effectively at the later stage of learning. However, what are the differences in the internal learning processes that result in group differences? Müller et al. (2014) suggests that differences in cognitive flexibility among individuals is related to many factors, including gray matter volume of the right anterior insula, the functional connection between the bilateral anterior insula and the midcingulate cortex/supplementary motor areas, and the degree of impulsivity according to the Big Five personality traits. Different factors exert unique effects on cognitive flexibility. Research across multiple paradigms and using various methodologies (i.e., fMRI, EEG) is needed to further understand the mechanisms underlying cognitive flexibility.

There were significant differences among the probabilistic pairs for the low flexibility group. The differences were mainly due to faster responses to large probability differences (2/3) than to small probability differences (1/3) in the post-learning stage. However, the high flexibility group did not change significantly with the change of probability pairs. For the high flexibility group, although the number of red blocks on the left and right sides changed constantly according to the settings of the experimental conditions, high flexibility subjects may approach all conditions by applying a unified 'framework,' or response

basis, with all conditions being parallel parts of that framework. For the low flexibility learners, they may not have employed such a framework. Their response basis may vary for each probability condition even after entering the post-learning stage. This may be evidenced by the low flexibility learners showing slower response latencies for the 1/3 probability difference condition that were not accompanied by accuracy levels comparable to that of the 2/3 condition (accuracy is significantly lower in the 1/3 probability difference for the low flexibility group). It is perhaps due to confusion about the response basis in the 1/3 probability difference condition that the latency under this condition is longer and the accuracy is lower. In fact, this condition appeared to be particularly difficult for the low flexibility subjects. This may be due to two reasons: (1) A greater number of learners in the low group did not identify the correct response criteria (rule) applicable to all conditions of the task during learning. (2) Because the probability attributes of the rules were not indicated before the task, and people generally tend to search for a simple 'stimulus-response' connection (i.e., correct response = reward), it may be a greater challenge for the low cognitive flexibility group to realize and accept a probability-based reward pattern.

Cognitive flexibility involves explicit and implicit forms of processing (Fujino et al., 2017), and the flexible goal achievement is not fully conscious (Custers and Aarts, 2010). Study has shown that there are different areas of brain activity in individuals with rationality-based explicit aspect of flexibility and experience-based implicit aspect of flexibility, and there is strong connection between them (Fujino et al., 2017). In present study, subjects were required to describe the founded rule after the task. Most of the subjects who had reached the acquisition level behaviorally could describe the correct rule while others report being unclear or not always following the same rule, not all the subjects who had learned the rule could grasp the rule consciously. However, the objective separation of implicit and explicit parts in probabilistic rule learning was not yet realized. Meanwhile, for the confidence of the rule, learner group had significant higher confidence score than non-learner group. When subjects were further divided into high CF – learner, high CF – non-learner, low CF – learner and low CF – non-learner group, only high CF – learners' confidence score was significantly higher than that of low CF – non-learners. No matter high CF – learner vs. high CF – non-learner group or low CF – learner vs. low CF – non-learner group, the confidence scores were not significant. This was because although some subjects had not learned the rules objectively, they thought the rule had been found was correct and gave relatively high confidence scores.

As mentioned before, the study has reached some conclusions about differences between high and low CF learners in probabilistic rule learning. And, there are some issues not taken into account and deserve further attention. Firstly, the sample was imbalanced by sex, and most of the subjects were mainly women. The results of this study cannot exclude the effect of potential gender differences. Secondly, it is possible that the difference of two groups in acquisition speed of rule may be disguised as the setting of the task, which depends on the use of other probabilistic rule tasks for further investigation. Thirdly, the differences between groups were in behavioral, post-learning stage in detail. An unresolved issue concerns the question as to what is the cause of the differences before the rule is acquired. Further detailed analysis of the learning process using tools such as ERP will be helpful. Finally, in present study, the artificial 80% response accuracy rate was used as the cut-off point for the acquisition of rule, and the six-point scale also used to evaluate the learners' certainty of rule after the completion of the coin-searching task. Whether there are some objective and implicit indicators of rule acquisition in probabilistic rule learning is an interesting point for further study.

CONCLUSION

This study preliminarily confirms that there are significant differences in learning outcomes between high and low cognitive flexibility learners in probabilistic rule learning. In short, cognitive flexibility is associated with the performance after the rule acquisition during the probabilistic rule task. The deep-rooted reasons for these differences need to be further explored by using other experimental techniques.

DATA AVAILABILITY STATEMENT

The datasets generated for this study are available on request to the corresponding author.

ETHICS STATEMENT

The study protocol was reviewed and approved by the Ethical Committee of Soochow University. The participants provided their written informed consent to participate in this study.

AUTHOR CONTRIBUTIONS

XF contributed to experimental design, data collection, data analysis, and manuscript writing. GP contributed to manuscript writing. WF contributed to data analysis and manuscript writing. CF contributed to experimental design.

FUNDING

The work was supported by the Humanities and Social Sciences of Ministry of Education Planning Fund of China (No. 17YJA880019).

ACKNOWLEDGMENTS

We thank all the participants took part in the experiment.

REFERENCES

Bellebaum, C., and Daum, I. (2008). Learning-related changes in reward expectancy are reflected in the feedback-related negativity. *Eur. J. Neurosci.* 27, 1823–1835. doi: 10.1111/j.1460-9568.2008.06138.x

Buchsbaum, B. R., Greer, S., Chang, W.-L., and Berman, K. F. (2005). Meta-analysis of neuroimaging studies of the wisconsin card−sorting task and component processes. *Huma Brain Mapp.* 25, 35–45. doi: 10.1002/hbm.20128

Craig, S., Lewandowsky, S., and Little, D. R. (2011). Error discounting in probabilistic category learning. *J. Exp. Psychol. Learn. Mem. Cogn.* 37, 673–687. doi: 10.1037/a0022473

Custers, R., and Aarts, H. (2010). The unconscious will: how the pursuit of goals operates outside of conscious awareness. *Science* 329, 47–50. doi: 10.1126/science.1188595

Dajani, D. R., and Uddin, L. Q. (2015). Demystifying cognitive flexibility: implications for clinical and developmental neuroscience. *Trends Neurosci.* 38, 571–578. doi: 10.1016/j.tins.2015.07.003

Darby, K. P., Castro, L., Wasserman, E. A., and Sloutsky, V. M. (2018). Cognitive flexibility and memory in pigeons, human children, and adults. *Cognition* 177, 30–40. doi: 10.1016/j.cognition.2018.03.015

Dehaene, S., Meyniel, F., Wacongne, C., Wang, L., and Pallier, C. (2015). The neural representation of sequences: from transition probabilities to algebraic patterns and linguistic trees. *Neuron* 88, 2–19. doi: 10.1016/j.neuron.2015.09.019

Diamond, A. (2013). Executive functions. *Annu. Rev. Psychol.* 64, 135–168. doi: 10.1146/annurev-psych-113011-143750

Dong, X., Du, X., and Qi, B. (2016). Conceptual knowledge influences decision making differently in individuals with high or low cognitive flexibility: an ERP study. *PLoS One* 11:e0158875. doi: 10.1371/journal.pone.0158875

Fujino, J., Tei, S., Jankowski, K. F., Kawada, R., Murai, T., and Takahashi, H. (2017). Role of spontaneous brain activity in explicit and implicit aspects of cognitive flexibility under socially conflicting situations: a resting-state fMRI study using fractional amplitude of low-frequency fluctuations. *Neuroscience* 367, 60–71. doi: 10.1016/j.neuroscience.2017.10.025

Hartstra, E., Oldenburg, J. F. E., Leijenhorst, L. V., Rombouts, S. A., and Crone, E. A. (2010). Brain regions involved in the learning and application of reward rules in a two-deck gambling task. *Neuropsychologia* 48, 1438–1446. doi: 10.1016/j.neuropsychologia.2010.01.012

Klayman, J., and Ha, Y.-W. (1989). Hypothesis testing in rule discovery: strategy, structure, and content. *J. Exp. Psychol. Learn. Mem. Cogn.* 15, 596–604. doi: 10.1037/0278-7393.15.4.596

Lange, F., Kip, A., Klein, T., Müller, D., Seer, C., and Kopp, B. (2018). Effects of rule uncertainty on cognitive flexibility in a card-sorting paradigm. *Acta Psychol.* 190, 53–64. doi: 10.1016/j.actpsy.2018.07.002

Lange, F., Seer, C., and Kopp, B. (2017). Cognitive flexibility in neurological discorders: cognitive components and event-related potentials. *Neurosci. Biobehav. Rev.* 83, 496–507. doi: 10.1016/j.neubiorev.2017.09.011

Lehto, J. E., and Elorinne, E. (2003). Gambling as an executive function task. *Appl. Neuropsychol.* 10, 234–238. doi: 10.1207/s15324826an1004_5

Liu, Z., Braunlich, K., Wehe, H. S., and Seger, C. A. (2015). Neural networks supporting switching, hypothesis testing, and rule application. *Neuropsychologia* 77, 19–34. doi: 10.1016/j.neuropsychologia.2015.07.019

Müller, V. I., Langner, R., Cieslik, E. C., Rottschy, C., and Eickhoff, S. B. (2014). Interindividual differences in cognitive flexibility: influence of gray matter volume, functional connectivity and trait impulsivity. *Brain Struct. Funct.* 220, 2401–2414. doi: 10.1007/s00429-014-0797-6

Rogers, R., and Monsell, S. (1995). Costs of a predictable swith between simple cognitive tasks. *J. Exp. Psychol. Gen.* 124, 207–231. doi: 10.1037/0096-3445.124.2.207

Stad, F. E., Heijningen, C. J. M. V., Wiedl, K. H., and Resing, W. C. M. (2018). Predicting school achievement: differential effects of dynamic testing measures and cognitive flexibility for math performance. *Learn. Individ. Differ.* 67, 117–125. doi: 10.1016/j.lindif.2018.07.006

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.

Advantages of publishing in Frontiers

OPEN ACCESS
Articles are free to read for greatest visibility and readership

FAST PUBLICATION
Around 90 days from submission to decision

HIGH QUALITY PEER-REVIEW
Rigorous, collaborative, and constructive peer-review

TRANSPARENT PEER-REVIEW
Editors and reviewers acknowledged by name on published articles

Frontiers
Avenue du Tribunal-Fédéral 34
1005 Lausanne | Switzerland

Visit us: www.frontiersin.org
Contact us: frontiersin.org/about/contact

REPRODUCIBILITY OF RESEARCH
Support open data and methods to enhance research reproducibility

DIGITAL PUBLISHING
Articles designed for optimal readership across devices

FOLLOW US
@frontiersin

IMPACT METRICS
Advanced article metrics track visibility across digital media

EXTENSIVE PROMOTION
Marketing and promotion of impactful research

LOOP RESEARCH NETWORK
Our network increases your article's readership