Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Hierarchical graph contrastive domain adaptation for multi-source cross-network node classification

Chuanyun Lin[a], Xi Zhou[b], Xiao Shen[a,*]

[a] School of Computer Science and Technology, Hainan University, Haikou, China
[b] College of Tropical Agriculture and Forestry, Hainan University, Haikou, China

ARTICLE INFO

Keywords:
Graph Neural Network
Multi-source Cross-network Node Classification
Multi-source Domain Adaptation
Pseudo-labeling

ABSTRACT

The current research on cross-network node classification primarily focuses on transferring knowledge from a single source network to classify nodes in a target network. To allow for a comprehensive capture of the diversity in data distributions across different source networks, our work investigates a more practical and challenging problem known as Multi-Source Cross-Network Node Classification (MSCNNC). The goal of MSCNNC is to classify nodes in a target network by leveraging the complementary knowledge from multiple source networks. To address the MSCNNC problem, we propose a novel Multi-source Hierarchical Graph Contrastive Domain Adaptation (MHGCDA) model. Firstly, MHGCDA designs a transferability weight learning module to measure the fitness between each source network and the target network based on information entropy. The transferability weights can dynamically adjust the contribution of the knowledge from different source networks, thereby effectively mitigating negative transfer and optimizing the adaptation process for the target network. Secondly, MHGCDA conducts class-aware hierarchical graph contrastive domain adaptation to reduce intra-class domain divergence while enlarging inter-class domain discrepancy at both node and prototype levels. Specifically, the node-level and prototype-level graph contrastive domain adaptation are devised to align the node embeddings and prototype embeddings of the same class between each source network and the target network. Lastly, MHGCDA employs a novel positive and negative pseudo-labeling strategy to assign positive pseudo-labels to target nodes highly likely to belong to a specific class, and negative pseudo-labels to those highly unlikely to belong to a specific class. Such positive and negative pseudo-labeled nodes are iteratively incorporated into self-training, refining pseudo-labels to promote class-aware hierarchical graph contrastive domain adaptation. Extensive experiments on benchmark datasets demonstrate the effectiveness of MHGCDA on the MSCNNC problem.

1. Introduction

Node classification is one of the most important tasks in graph machine learning. With the continuous growth of graph-structured data and the high cost of graph data annotation (Hu, 2019), coupled with the success of domain adaptation in computer vision (CV) and natural language processing (NLP) (Gopalakrishnan, 2017), researchers have begun to investigate the problem of cross-network node classification (CNNC). The CNNC problem aims to transfer the knowledge learned from a source network with abundant node labels to assist node classification in a target network which lacks node labels (Shen, 2020). However, due to the inherent distribution shift between the source and target networks, directly applying the model trained on the labeled source network to the unlabeled target network would fail to obtain satisfactory performance. To tackle this, current research on CNNC (Dai, 2022; Shen, 2020; Shen, 2021; Wu et al., 2020a; Yang et al., 2022b; Zhang, 2019) typically combines graph neural networks (GNNs) with domain adaptation. On one hand, GNNs leverage the topological structures and node attributes to learn node embeddings for the source and target networks. On the other hand, domain adaptation is employed to alleviate domain discrepancy between the source and target networks.

Most existing CNNC methods (Dai, 2022; Shen, 2020; Shen, 2021; Shen, 2024b; Wu et al., 2020a; Yang et al., 2022b; Zhang, 2019) are designed for a single source network scenario, involving the knowledge transfer only from a single source network to a target network. However,

* Corresponding author.
E-mail addresses: cylin@hainanu.edu.cn (C. Lin), xzhou@hainanu.edu.cn (X. Zhou), xshen@hainanu.edu.cn (X. Shen).

https://doi.org/10.1016/j.eswa.2025.127900
Received 11 July 2024; Received in revised form 1 April 2025; Accepted 24 April 2025
Available online 27 April 2025

in real-world CNNC applications, there might be multiple source networks with rich label information. For example, in cross-domain movie recommendation, to predict the movie interests of users in a target social network (e.g. Douban), one can attempt to transfer the knowledge from multiple source networks (e.g. Flickr, YouTube, Netflix) where users have abundant labels indicating their movie interests. The existing CNNC methods might overly depend on the features and labels of a single source network, limiting the generalization ability on the target network, thus hindering effective handling of the CNNC scenario with multiple source networks. Recent advances in contrastive learning (Dong, 2024; Zhou, 2024) have demonstrated the capability to effectively integrate multi-source information and optimize feature distributions. Motivated by this, our work investigates how to take advantage of contrastive learning (Shen et al., 2023b) on multi-source graph domain adaptation, to integrate the information from multiple source networks to address CNNC.

In order to leverage complementary knowledge provided by different source networks to reduce the target network's overreliance on a single source network, we study a more realistic CNNC problem, i.e., multi-source CNNC (MSCNNC). As illustrated in Fig. 1, the MSCNNC problem contains multiple fully labeled source networks and an unlabeled target network, where each source network exhibits a different distribution from the target network. The goal of MSCNNC is to accurately classify nodes in the target network by transferring knowledge obtained from multiple source networks. Compared to single-source CNNC, the research on MSCNNC is still at an early stage that needs further exploration. To effectively address the MSCNNC problem, one needs to overcome the following challenges. First, to extend existing single-source CNNC methods to the multi-source scenario, a simple way is to combine all the source network data and treat them as a whole, or treat each source network equally during alignment with the target network. However, treating each source domain equally may lead to sub-optimal results in multi-source domain adaptation (Kang et al., 2020). Thus, the first challenge (**CH1**) is how to determine various transferability weights when transferring knowledge from different source networks to the target network, thereby enhancing the node classification performance in the target network and effectively avoiding negative transfer. Second, most existing CNNC methods just conduct class-agnostic domain alignment to mitigate the domain discrepancy at the domain level, while neglecting the class from which the samples are drawn. As a result, the target nodes might be misaligned with the source nodes of a different class, consequently degrading the node classification performance in the target network. Thus, the second challenge (**CH2**) is how to design the class-aware domain adaptation module to mitigate the intra-class domain discrepancy and explicitly enlarge the inter-class domain discrepancy, between each source network and the target network. Third, existing CNNC methods mainly utilize the supervised information only from the labeled source network to train the node classifier. Although previous self-training domain adaptation methods (Chen et al., 2011; Shen et al., 2019; Xu et al., 2018b) propose to exploit the pseudo-labels of the target domain to train the classifier, they just conduct positive pseudo-labeling to supervise the model that "the target sample is very likely to belong to a specific class". However, in the context of multi-source domain adaptation, exploiting negative pseudo-labeling (Kim et al., 2019; Rizve et al., 2021) to supervise the model that "the target sample is very unlikely to belong to a specific class" remains under-explored. Thus, the third challenge (**CH3**) is how to fully exploit the potential supervised information from the target network to

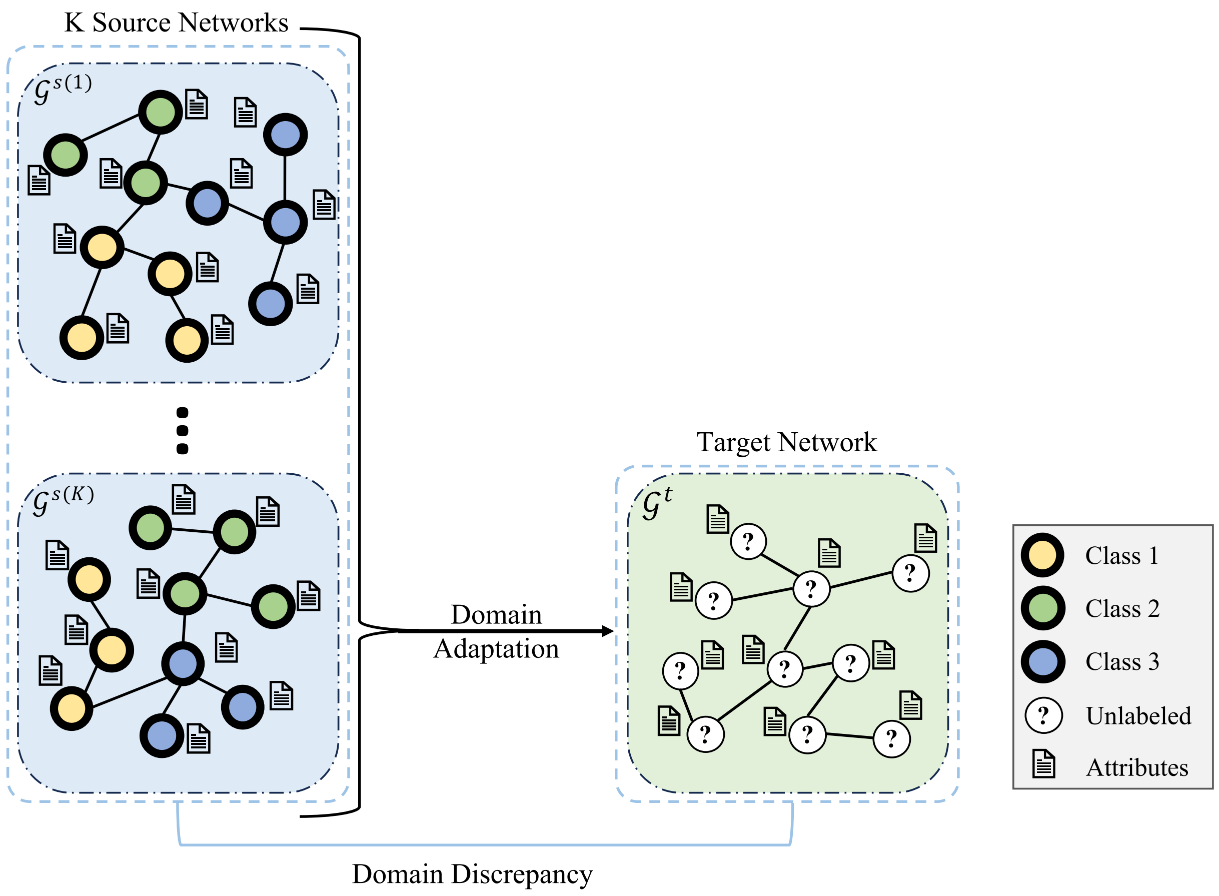

Fig. 1. Multi-source Cross-Network Node Classification. Given $K(K \geq 2)$ fully labeled source networks and a completely unlabeled target network, where domain discrepancies inherently exist between different networks, the goal of MSCNNC is to accurately classify nodes in the target network by transferring complementary knowledge from multiple source networks.

iteratively re-train the model, by taking advantage of both positive and negative pseudo-labeling.

To effectively address the MSCNNC problem, we propose a novel Multi-source Hierarchical Graph Contrastive Domain Adaptation (MHGCDA) model. Firstly, a shared-weight GNN encoder is employed to learn node embeddings for all the source networks and the target network. Then, multiple network-specific node classifiers are constructed for different source networks, where each classifier is trained upon the labeled information from each source network. To address **CH1**, the proposed MHGCDA model determines the transferability weight of each source network, according to the entropy of the target predictions of each network-specific node classifier. The entropy can reflect the confidence of a classifier on the target nodes, where a lower entropy indicates that the classifier is more certain about the target predictions. Inspired by (Kang et al., 2020), when the network-specific node classifier of a source network is more certain about the target nodes, the source network can be seen as more similar to the target network, and thus should be assigned with a larger transferability weight during domain alignment, and vice versa. To address **CH2**, the proposed MHGCDA model customizes a hierarchical graph contrastive domain adaptation module to conduct class-aware alignment between each source network and the target network, at both the node and prototype levels. Specifically, the node-level graph contrastive domain adaptation aims to map the nodes of the same class across networks close, while mapping those belonging to different classes across networks far apart in the embedding space. While the prototype-level graph contrastive domain adaptation ensures that the prototypes corresponding to the same class across networks are closer than the prototypes of different classes across networks. As a result, the intra-class domain discrepancy can be mitigated and the inter-class domain discrepancy can be enlarged at both the node and prototype levels. To address **CH3**, on one hand, we follow the conventional self-training domain adaptation methods (Chen et al., 2011; Shen et al., 2019) to assign positive pseudo-labels to the target nodes that are most likely to belong to each specific class, and adopt such positive pseudo-labeled target nodes as the supervised signals to iteratively re-train each network-specific node classifier. On the other hand, our work should be the first to adopt negative pseudo-labeling (Rizve et al., 2021) in the context of multi-source domain adaptation, by assigning negative pseudo-labels to the target nodes that are less likely to belong to each specific class. By taking such negative pseudo-labeled target nodes to iteratively re-train the network-specific node classifiers, the risk of providing incorrect supervised information can be decreased (Kim et al., 2019). In addition, the research on multi-source domain adaptation (Venkat et al., 2020) has found that enforcing an agreement among the domain-specific classifiers is conductive to aligning the domains in the latent space. Motivated by this, we select the target nodes for positive and negative pseudo-labeling only for which there is an agreement between multiple network-specific node classifiers. Specifically, a target node would be assigned with a positive (negative) pseudo-label of a specific class only when multiple network-specific node classifiers all predict that the probability of this node belonging to the specific class is higher (lower) than a threshold. We validate the proposed MHGCDA method on six real-world public networked datasets. Extensive experimental results demonstrate the effectiveness of the proposed MHGCDA on MSCNNC. Ablation studies are conducted to verify the contributions of key components in our MHGCDA framework.

Our main contributions are summarized as follows:

- Instead of treating different source networks equally during alignment with the target network, the proposed MHGCDA model assigns various transferability weights to reflect the importance of different source networks on the target network. Specifically, a larger transferability weight is assigned to the source network whose network-specific node classifier is more certain about the label prediction of the target nodes, so that the alignment between such a source network and the target network would be more emphasized.
- To get rid of class-agnostic domain alignment, the proposed MHGCDA customizes a class-aware hierarchical graph contrastive domain adaptation module to align each source network and the target network, at both the node and prototype levels. Then, the nodes (and prototypes) belonging to the same class across networks would be mapped closer than those corresponding to different classes across networks. As a result, the intra-class domain discrepancy can be mitigated and the inter-class domain discrepancy can be explicitly enlarged at both levels, so as to effectively enhance the node classification performance in the target network.
- Unlike conventional self-training domain adaptation methods which just utilize positive pseudo-labeling, the proposed MHGCDA is the first to exploit both positive and negative pseudo-labeling in the context of multi-source domain adaptation. Negative pseudo-labeling can reduce the risk of providing incorrect supervised information. Given the multi-source scenario, we assign positive and negative pseudo-labels to the target nodes, only when there is an agreement between multiple network-specific node classifiers. Such positive or negative pseudo-labeled target nodes would participate in iterative self-training, gradually optimizing network-specific node classifiers to ensure more accurate pseudo-labels to facilitate class-aware hierarchical graph contrastive domain adaptation. Extensive experimental results on cross-network benchmark datasets and ablation studies demonstrate the effectiveness of the proposed MHGCDA on the MSCNNC problem.

2. Related work

2.1. Graph neural networks

Graph Neural Networks (Wu et al., 2020b), as a specialized deep learning model for handling graph-structured data, have achieved significant breakthroughs in node classification. Graph Convolutional Network (GCN) (Kipf & Welling, 2016), which is motivated by a local first-order approximation of spectral graph convolutions, stands as a pivotal milestone in the field of GNNs. Graph Attention Network (GAT) (Veličković et al., 2017) employs a self-attention mechanism to learn adaptive weights for different neighboring nodes, enhancing information aggregation among the neighborhood. GraphSAGE (Hamilton et al., 2017) aggregates information obtained through local neighborhood sampling and can generate embeddings for unseen nodes during training. APPNP (Gasteiger et al., 2018) decouples feature transformation from message propagation, effectively alleviating the over-smoothing issue that GCN easily suffers from. Graph Isomorphism Network (Xu et al., 2018a) updates node embeddings by aggregating embeddings of neighboring nodes and obtains graph-level embedding through pooling operations. Graph Random Neural Network (Feng et al., 2020) extends graph data through random walks to capture more local structural information, and uses consistency regularization to constrain the consistency of label predictions in different data extensions. Dual Separated Attention-based Graph Neural Network (Shen et al., 2024a) separates self-representation learning from neighbor-representation learning by two feature extractors, so as to preserve both the commonality between connected nodes and the individuality of each node.

It is worth noting that the aforementioned GNNs primarily focus on node classification within a single network, while failing to tackle the node classification task across different networks.

2.2. Cross-network node classification

Current CNNC literatures commonly adopt the mainstream paradigm of combining GNNs and domain adaptation. CDNE (Shen, 2021) stands as a pioneering algorithm for the CNNC problem, it reduces domain-shifts through minimizing Maximum Mean Discrepancy (MMD), and captures network relationships by mapping closely connected nodes to have similar embeddings. ACDNE (Shen, 2020) blends adversarial domain adaptation with deep network embeddings to achieve network-invariant node representations. The deep network embedding module captures attributes' affinities and topological proximities among nodes using two feature extractors. AdaGCN (Dai, 2022) comprises semi-supervised learning and adversarial domain adaptation components. The former uses GCN to learn node embeddings, while the latter mitigates domain-shift by guiding adversarial domain adaptation based on Wasserstein distance. UDAGCN (Wu et al., 2020a) adopts dual-GCN to jointly utilize local and global consistency for feature aggregation, generating a unified representation for each node through an attention mechanism. RGDAL (Yang et al., 2022b) employs a GCN with constrained graph mutual information and adversarial learning to learn noise-resistant and domain-invariant graph representations. DM-GNN (Shen et al., 2023a) integrates feature-level and label-level message-passing GNN with conditional adversarial domain adaptation to learn network-transferable and label-discriminative node representations. TGAE (Wu et al., 2024) employs an attention mechanism to integrate local and global node information, learning network-transferable features by minimizing marginal and conditional distribution differences across networks. SEPA (Liu et al., 2024) constructs a prototype-based graph to capture class-wise semantics and introduces a domain discrepancy metric to align the source and target domains. GRADE (Wu, 2023) introduces a graph subtree discrepancy derivation of generalization error bounds for cross-network transfer learning based on source knowledge and inter-domain graph subtree differences. SGDA (Qiao et al., 2023) trains the adaptive shift parameter of each source node in an adversarial manner to align the cross-domain distribution of node embeddings, so that the node classifiers trained on labeled source nodes can be transferred to target nodes.

The aforementioned CNNC methods are primarily designed for a single source network scenario. While little attention has been paid to the scenario where multiple source networks with diverse data distributions exist in CNNC. MTGK (Yang et al., 2022a) proposes to learn transferable common subgraphs from multiple source networks to promote the task of node classification in the target network. However, it just conducts class-agnostic domain adaptation and does not utilize the potential supervised information from the target network for model training. Compared to single-source CNNC, the research on MSCNNC is still at an early stage and remains largely under-developed.

2.3. Multi-source domain adaptation

Multi-source domain adaptation (He et al., 2021) transfers the task knowledge from multiple labeled source domains to make predictions for an unlabeled target domain, where inherent domain shift exists between different domains. The mainstream single-source domain adaptation methods exhibit suboptimal performance when faced with the domain adaptation problem involving multiple source domains (Nguyen et al., 2021). In addition, in the presence of multiple source domains, determining which source domain can provide the best adaptation for the target task poses a challenge. M^{3}SDA (Peng et al., 2019) transfers knowledge learned from multiple labeled source domains to an unlabeled target domain by dynamically aligning feature distribution matrices. DCTN (Xu et al., 2018b) employs multi-path adversarial learning to minimize domain discrepancy, and fine-tunes feature extractors and classifiers using labeled data from multiple source domains and pseudo-labeled data from the target domain. SPS (Wang et al., 2022b) introduces a deep adaptation network that progressively assigns high-confidence pseudo-labels in the target domain using a self-taught learning approach to align conditional distributions. SImpAI (Venkat et al., 2020) trains a set of shared classifiers for alignment using labeled data from multiple source domains. Then, it generates pseudo-labels for target samples based on classifier consensus, and alternately trains the model on source and target domains until target consistency converges, so as to align different domains in the latent space. MSCAN (Kang et al., 2020) utilizes multi-source clustering integration to generate target label estimations, and assigns different weights for source-target alignment based on boundary-sensitive alignment. STEM (Nguyen et al., 2021) creates multi-source teacher-expert models by utilizing knowledge from multiple domain experts through a weighted ensemble strategy. MSDCN (Cai et al., 2022) maximizes the pairwise similarity over examples from multiple source domains and utilizes an alternative optimization strategy to train the ensemble multiple source classifiers to address the universal multi-source domain adaptation task. CDFA (Cai et al., 2023) designs a contrastive classifier network and a domain-specific learning network to learn domain-specific representations by separating redundant information from all representations, so as to improve transferability in both single-source and multi-source domain adaptation settings. DCADAN (Wang et al., 2025) employs a dynamic collaborative generator and a multi-source dynamic collaborative loss to enhance domain adaptation across multiple source domains. MIEM (Wen et al., 2023) aligns the source joint distribution and the target joint distribution simultaneously by estimating and minimizing mutual information in the latent feature space.

It is worth noting that existing multi-source domain adaptation methods are primarily designed for the fields of CV and NLP, based on the assumption that each sample is independent and identically distributed (i.i.d.) within each domain. However, due to the complex edges connecting nodes within each graph, the graph-structured data obviously violate such i.i.d. assumption. Thus, directly applying existing multi-source domain adaptation methods to MSCNNC might struggle to achieve satisfactory performance.

3. Proposed model

In this section, we first define the MSCNNC problem. Then, we provide an overview of MHGCDA framework and introduce each of the components of our framework in details.

3.1. Problem definition

In general, let $\mathscr{G} = (\mathscr{V}, \mathscr{E}, \mathbf{A}, \mathbf{X}, \mathbf{Y})$ represent a network composed of a set of nodes $\mathscr{V} = \{v_1, v_2 \dots v_n\}$, a set of edges $\mathscr{E} \subseteq \mathscr{V} \times \mathscr{V}$, an adjacency matrix $\mathbf{A} \in \{0,1\}^{n\times n}$, a node attribute matrix $\mathbf{X} \in \mathbb{R}^{n\times \mathbb{W}}$ and a node label matrix $\mathbf{Y} \in [0,1]^{n\times C}$, where n, C, $\mathbb{W}$ are the number of nodes, node classes and node attributes in $\mathscr{G}$, respectively. Specifically, $x_i \in \mathbb{R}^{\mathbb{W}}$ denotes the node attribute vector of v_i. $\mathrm{Y}_{ic} = 1$ if node v_i is associated with the class-label c; otherwise, $\mathrm{Y}_{ic} = 0$. $\mathrm{A}_{ij} = 1$ if v_i and v_j are connected in $\mathscr{G}$, i.e., $(v_i, v_j) \in \mathscr{E}$; otherwise, $\mathrm{A}_{ij} = 0$. In the context of MSCNNC, there are $K (K \geq 2)$ fully labeled source networks, represented as $\left\{\mathscr{G}^{s(k)}\right\}_{k=1}^{K} = \left\{\left(\mathscr{V}^{s(k)}, \mathscr{E}^{s(k)}, \mathbf{A}^{s(k)}, \mathbf{X}^{s(k)}, \mathbf{Y}^{s(k)}\right)\right\}_{k=1}^{K}$, and a completely unlabeled target network $\mathscr{G}^{t} = (\mathscr{V}^{t}, \mathscr{E}^{t}, \mathbf{A}^{t}, \mathbf{X}^{t})$. Each source network inherently has various data distribution with other source networks and the target network. The source networks $\left\{\mathscr{G}^{s(k)}\right\}_{k=1}^{K}$ and $\mathscr{G}^{t}$ all share the same node attributes and common label space. The MSCNNC problem aims to transfer the knowledge from multiple source networks $\left\{\mathscr{G}^{s(k)}\right\}_{k=1}^{K}$ to the target network $\mathscr{G}^{t}$, to promote the node classification performance on the target network. Frequently used notations are summarized in Table 1.

3.2. Overview of MHGCDA

Fig. 2 illustrates the model architecture of MHGCDA. Given multiple source networks $\left\{\mathscr{G}^{s(k)}\right\}_{k=1}^{K}$ and a target network $\mathscr{G}^{t}$, MHGCDA adopts a shared-weight GNN encoder f_h to generate node embeddings for each network. On the top of the GNN encoder, multiple unshared-weight

Table 1
Frequently Used Notations.

Notations	Descriptions
$\mathscr{G}^{s(k)}, \mathscr{G}^t$	k-th source network and target network
K	Number of source networks
v_i	i-th node in $\mathscr{G}$
$\boldsymbol{x}_i$	Node attribute vector of v_i
$\boldsymbol{y}_i$	Node label vector of v_i
$\boldsymbol{h}_i$	Node embedding vector of v_i
(v_i, v_j)	An edge connecting v_i and v_j in $\mathscr{G}$
$\boldsymbol{p}_c^{s(k)}$	The prototype of class c in the k-th source network
$\boldsymbol{p}_c^t$	The prototype of class c in the target network
f_h	GNN encoder
$f_y^{(k)}$	k-th network-specific node classifier
$\theta^h, \theta_y^{(k)}$	Learnable parameters of f_h and $f_y^{(k)}$
$\boldsymbol{Y}^{s(k)}$	Observed node label matrix of the k-th source network
$\widehat{\boldsymbol{Y}}^{t(k)}$	Predicted node label probability matrix of $\mathscr{G}^t$ by the k-th network-specific node classifier
$\widehat{\boldsymbol{Y}}^t$	Predicted node label probability matrix of $\mathscr{G}^t$ by ensemble predictions of all K network-specific node classifiers
$\overline{\overline{\boldsymbol{Y}}}^t$	Node pseudo label matrix of the target network
w^k	The transferability weight of k-th source network
τ_p, τ_n	The positive pseudo-labeling threshold and negative pseudo-labeling threshold
$\mathbb{B}^{s(k)}, \mathbb{B}^t$	Mini-batch sampled from the k-th source network and target network

network-specific node classifiers $\left\{f_y^{(k)}\right\}_{k=1}^K$ are added to adapt to various data distributions of different source networks. A transferability weight learning module is designed to measure the adaptation fitness of each source network $\mathscr{G}^{s(k)}$ to the target network based on the information entropy of each network-specific node classifier $f_y^{(k)}$. In addition, a class-aware hierarchical graph contrastive domain adaptation module is customized to mitigate the intra-class domain discrepancy and enlarge the inter-class domain discrepancy at both the node and prototype levels. On one hand, the node-level graph contrastive domain adaptation aims to maximize the consistency between the node embeddings belonging to the same class across networks. On the other hand, the prototype-level graph contrastive domain adaptation focuses on maximizing the mutual information between the prototype embeddings corresponding to the same class across networks. The node-level and prototype-level graph contrastive domain adaptation jointly yield better class-aware domain alignment between each pair of the source-target networks. Moreover, a positive and negative pseudo-labeling module is designed to assign positive and negative pseudo-labels to the nodes in the target network to indicate the presence and absence of a specific class in the target nodes. Then, the target nodes carrying positive or negative pseudo-labels would be employed to iteratively re-train the model, so as to exploit the potential supervised information from the target network.

3.3. Learning transferability weight for each source network

3.3.1. Graph neural network encoder

The proposed MHGCDA adopts a mini-batch training strategy to sample a source batch $\mathbb{B}^{s(k)} = \left\{v_i^{s(k)}\right\}_{i=1}^b$ from each of the source networks $\left\{\mathscr{G}^{s(k)}\right\}_{k=1}^K$ and a target batch $\mathbb{B}^t = \left\{v_i^t\right\}_{i=1}^b$ from the target network $\mathscr{G}^t$, where the batch size b is equal for each network, i.e., $b = \left|\mathbb{B}^{s(k)}\right| = |\mathbb{B}^t|, \forall 1 \le k \le K$. Then, for each mini-batch $\mathbb{B} \in \left\{\mathbb{B}^{s(k)}\right\}_{k=1}^K \cup \{\mathbb{B}^t\}$ of the source and target networks, we adopt a shared-weight GNN encoder to learn node embeddings. Here, we adopt the GAT (Veličković et al., 2017) as the GNN encoder $f_h(\cdot;\theta_h)$. Firstly, GAT learns the attention coefficient of v_j on v_i, for each edge (v_i, v_j) in the mini-batch $\mathbb{B}$, as:

$$\widehat{A}_{ij} = \frac{\exp\left(\text{LeakyReLU}\left(\mathbf{a}^T\left[\boldsymbol{W}_h\boldsymbol{x}_i \| \boldsymbol{W}_h\boldsymbol{x}_j\right]\right)\right)}{\sum_{v_l \in \{v_i\}\cup\mathscr{N}_i^{\mathbb{B}}} \exp\left(\text{LeakyReLU}\left(\mathbf{a}^T\left[\boldsymbol{W}_h\boldsymbol{x}_i \| \boldsymbol{W}_h\boldsymbol{x}_l\right]\right)\right)} \tag{1}$$

where $\mathscr{N}_i^{\mathbb{B}} = \{v_l | A_{il} = 1, v_l \in \mathbb{B}\}$ is a set of first-order neighbors of v_i in the mini-batch $\mathbb{B}$, and $[\cdot\|\cdot]$ denotes a concatenation operation. $\left\{\boldsymbol{W}_h \in \mathbb{R}^{d\times \mathbb{W}}, \mathbf{a} \in \mathbb{R}^{2d}\right\} = \theta_h$ denotes the learnable parameters of the GNN encoder, and d is the embedding dimension. $\widehat{A}_{ij}$ is the normalized attention coefficient of v_j on v_i, reflecting the importance of v_j on v_i relative to $\{v_i\} \cup \mathscr{N}_i^{\mathbb{B}}$.

Next, the node embedding of v_i (i.e., $\mathbf{h}_i \in \mathbb{R}^d$) is computed by adaptively aggregating the embeddings of its neighbors in the mini-batch $\mathbb{B}$ as:

$$\boldsymbol{h}_i = \text{ELU}\left(\sum_{v_j \in \{v_i\}\cup\mathscr{N}_i^{\mathbb{B}}} \widehat{A}_{ij}\boldsymbol{W}_h\boldsymbol{x}_j\right) \tag{2}$$

The above process represents the learning of node embeddings using the GAT with one attention head. In order to obtain more robust node em-

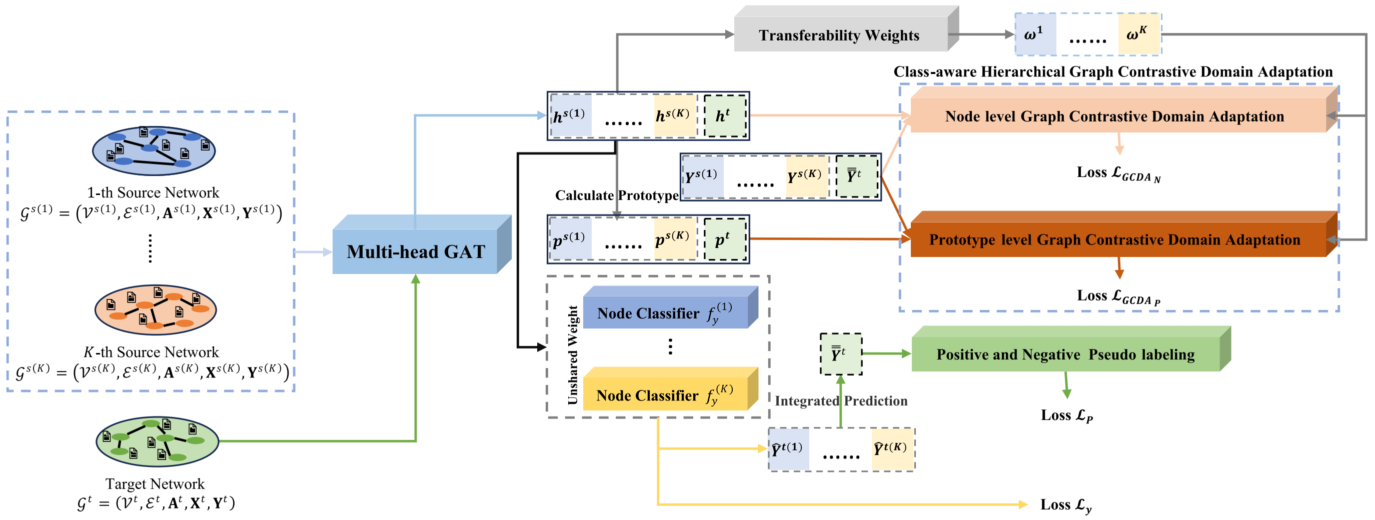
Transferability Weights
ω^1 ...... ω^K
Class-aware Hierarchical Graph Contrastive Domain Adaptation
$h^{s(1)}$ ...... $h^{s(K)}$ h^t
Node level Graph Contrastive Domain Adaptation
Loss $\mathcal{L}_{GCDA_N}$
$Y^{s(1)}$ ...... $Y^{s(K)}$ $\overline{\overline{Y}}^t$
Calculate Prototype
$p^{s(1)}$ ...... $p^{s(K)}$ p^t
Prototype level Graph Contrastive Domain Adaptation
Loss $\mathcal{L}_{GCDA_P}$
1-th Source Network
$\mathcal{G}^{s(1)} = (\mathcal{V}^{s(1)}, \mathcal{E}^{s(1)}, \mathbf{A}^{s(1)}, \mathbf{X}^{s(1)}, \mathbf{Y}^{s(1)})$
Multi-head GAT
K-th Source Network
$\mathcal{G}^{s(K)} = (\mathcal{V}^{s(K)}, \mathcal{E}^{s(K)}, \mathbf{A}^{s(K)}, \mathbf{X}^{s(K)}, \mathbf{Y}^{s(K)})$
Unshared Weight
Node Classifier $f_y^{(1)}$
Node Classifier $f_y^{(K)}$
$\overline{\overline{Y}}^t$
Positive and Negative Pseudo labeling
Loss $\mathcal{L}_P$
Integrated Prediction
$\widehat{Y}^{t(1)}$ ...... $\widehat{Y}^{t(K)}$
Target Network
$\mathcal{G}^t = (\mathcal{V}^t, \mathcal{E}^t, \mathbf{A}^t, \mathbf{X}^t, \mathbf{Y}^t)$
Loss $\mathcal{L}_y$

Fig. 2. The model architecture of MHGCDA, which contains a shared-weight GNN encoder along with multiple unshared-weight network-specific node classifiers, a transferability weight learning module, a class-aware hierarchical graph contrastive domain adaptation module at both node and prototype levels, and a positive and negative pseudo-labeling module.

beddings, we adopt the multi-head GAT, following (Veličković et al., 2017). To aggregate information from higher-order neighbors, we further stack $\mathbb{L}$-layer multi-head GAT.

3.3.2. *Network-specific node classifiers*

A paradigm of multi-source domain adaptation is to learn a shared feature extractor, along with multiple network-specific classifiers, and obtain a weighted ensemble prediction for the target samples (Venkat et al., 2020). Accordingly, the proposed MHGCDA learns K network-specific node classifiers $\left\{f_y^{(k)}\right\}_{k=1}^{K}$ for different source networks, where each k-th network-specific node classifier $f_y^{(k)}$ is employed to fit the data distribution corresponding to each source network $\mathcal{G}^{s(k)}$. Here, we adopt a single-layer GAT to construct a node classifier on the GNN encoder. Specifically, for the k-th source network $\mathcal{G}^{s(k)}$, the label prediction of v_i is computed by the k-th network-specific node classifier $f_y^{(k)}$ as:

$$\widehat{y}_i^{s(k)} = Softmax\left(\sum_{v_j^{s(k)}\in\left\{v_i^{s(k)}\right\}\cup\mathcal{N}_i^{\mathbb{B}^{s(k)}}} \widetilde{A}_{ij}^{s(k)}\boldsymbol{W}_y^{(k)}\boldsymbol{h}_j^{s(k)}\right) \tag{3}$$

where $\mathcal{N}_i^{\mathbb{B}^{s(k)}}$ is a set of first-order neighbors of $v_i^{s(k)}$ in the source mini-batch $\mathbb{B}^{s(k)}$. $\widetilde{A}_{ij}^{s(k)}$ is the normalized attention coefficient of $v_j^{s(k)}$ on $v_i^{s(k)}$ relative to $\left\{v_i^{s(k)}\right\}\cup\mathcal{N}_i^{\mathbb{B}^{s(k)}}$, which can be similarly learned as in Eq. (1). $\widehat{y}_i^{s(k)}\in\mathbb{R}^C$ is the label prediction of $v_i^{s(k)}$ over C categories. $\boldsymbol{W}_y^{(k)}$ is the weight matrix to transform the node embedding $\boldsymbol{h}_j^{s(k)}$ into the output label space. Eq. (3) generates the label predictions by the GAT with one attention head. It is feasible to generate more robust label predictions by averaging over multiple attention heads (Veličković et al., 2017). For a more concise representation, we denote all the learnable parameters of the k-th network-specific node classifier $f_y^{(k)}$ as $\theta_y^{(k)}$.

Next, the Softmax cross-entropy loss is employed to compute the node classification loss of the k-th source network $\mathcal{G}^{s(k)}$, based on the labeled nodes in the source mini-batch $\mathbb{B}^{s(k)}$, as:

$$\mathcal{L}_y^k = -\frac{1}{b}\sum_{c=1}^{C}\sum_{v_i^{s(k)}\in\mathbb{B}^{s(k)}} Y_{ic}^{s(k)}\log\widehat{Y}_{ic}^{s(k)} \tag{4}$$

where $\widehat{Y}_{ic}^{s(k)}$ is the predicted probability that $v_i^{s(k)}$ is labeled as class c. Minimizing Eq. (4) reduces the source risk of k-th source network $\mathcal{G}^{s(k)}$, yielding label-discriminative node embeddings for $\mathcal{G}^{s(k)}$.

3.3.3. *Transferability weights based on information entropy*

In multi-source domain adaptation, treating each source domain equally during alignment to the target data would result in the sub-optimal performance (Kang et al., 2020). To compute transferability weights for different source networks in the context of MSCNNC, our work is inspired by the findings in (Kang et al., 2020) that the information entropy of each domain-specific classifier can reflect the fitness of the corresponding source domain to the target data. Specifically, the source network whose network-specific node classifier has smaller information entropy can be seen as better fitted and more similar to the target network, thus, such a source network should be assigned with a larger transferability weight to make its alignment to the target network stronger.

Firstly, we employ the k-th network-specific node classifier $f_y^{(k)}$ to generate the label prediction for each target node $v_i^t\in\mathbb{B}^t$, similar to Eq. (3):

$$\widehat{y}_i^{t(k)} = f_y^{(k)}\left(\left\{\boldsymbol{h}_j^t|v_j^t\in\left\{v_i^t\right\}\cup\mathcal{N}_i^{\mathbb{B}^t}\right\};\theta_y^{(k)}\right) \tag{5}$$

where $\widehat{y}_i^{t(k)}\in\mathbb{R}^C$ denotes the target label prediction of v_i^t over C categories generated by $f_y^{(k)}$.

Then, we compute the fitness of each k-th source network $\mathcal{G}^{s(k)}$ to the target network $\mathcal{G}^t$ by the inverse of information entropy of the k-th network-specific node classifier $f_y^{(k)}$, as follows:

$$\eta^k = \frac{1}{\sum_{v_i^t\in\mathbb{B}^t}\mathcal{H}\left(\widehat{y}_i^{t(k)}\right)} \tag{6}$$

$$\mathcal{H}\left(\widehat{y}_i^{t(k)}\right) = -\sum_{c=1}^{C}\widehat{Y}_{ic}^{t(k)}\log\left(\widehat{Y}_{ic}^{t(k)}\right) \tag{7}$$

where η^k is the fitness of the k-th source network $\mathcal{G}^{s(k)}$ to $\mathcal{G}^t$, $\widehat{Y}_{ic}^{t(k)}$ denotes the predicted probability of v_i^t belonging to class c, predicted by the k-th network-specific node classifier $f_y^{(k)}$. $\mathcal{H}(\cdot)$ is the Shannon entropy which reflects the confidence of network-specific node classifier on the target nodes, where the lower the entropy is, the more certain the network-specific node classifier. When the network-specific node classifier $f_y^{(k)}$ is more certain about the target nodes, it can be assumed that the fitness between the k-th source network $\mathcal{G}^{s(k)}$ and the target network $\mathcal{G}^t$ is higher, and vice versa.

The fitness η^k of the k-th source network $\mathcal{G}^{s(k)}$ is normalized across the fitness of all pairs of source-target network alignment, to obtain the transferability weight w^k of each k-th source network $\mathcal{G}^{s(k)}$, as:

$$w^k = \frac{\eta^k}{\sum_{k=1}^{K}\eta^k} \tag{8}$$

where a higher transferability weight w^k indicates the greater importance of adaptation between $\mathcal{G}^{s(k)}$ and $\mathcal{G}^t$, i.e., the alignment of the k-th source network to the target network should be more emphasized.

Next, the node classification loss $\mathcal{L}_y^k$ on each source network $\mathcal{G}^{s(k)}$ is weighted by the corresponding transferability weight w^k to obtain an overall node classification loss $\mathcal{L}_y$, as:

$$\mathcal{L}_y = \sum_{k=1}^{K} w^k\mathcal{L}_y^k \tag{9}$$

According to the theory of multi-source domain adaptation (Mansour et al., 2012), a distribution weighted combination rule to ensemble multiple domain-specific classifiers can benefit from favorable theoretical guarantees. Motivated by this, we ensemble all the network-specific node classifiers $\left\{f_y^{(k)}\right\}_{k=1}^{K}$ to obtain the final label prediction of each target node, where each k-th network-specific node classifier is weighted by the corresponding transferability weight w^k, as:

$$\widehat{y}_i^t = \sum_{k=1}^{K} w^k\widehat{y}_i^{t(k)} \tag{10}$$

where $\widehat{y}_i^t\in\mathbb{R}^C$ represents the final label prediction of v_i^t over C categories.

3.4. *Class-aware hierarchical graph contrastive domain adaptation*

Existing CNNC methods mainly conduct class-agnostic domain alignment, which might lead to misalignment of class-conditional distributions of node embeddings between the source and target networks. Although a few CNNC methods adopt the class-conditional MMD (Shen, 2021) or class-conditional adversarial domain adaptation (Shen et al., 2023a) to conduct class-aware domain alignment, they just focus on mitigating the intra-class domain discrepancy. However, the inter-class domain discrepancy, which is beneficial for enhancing the domain

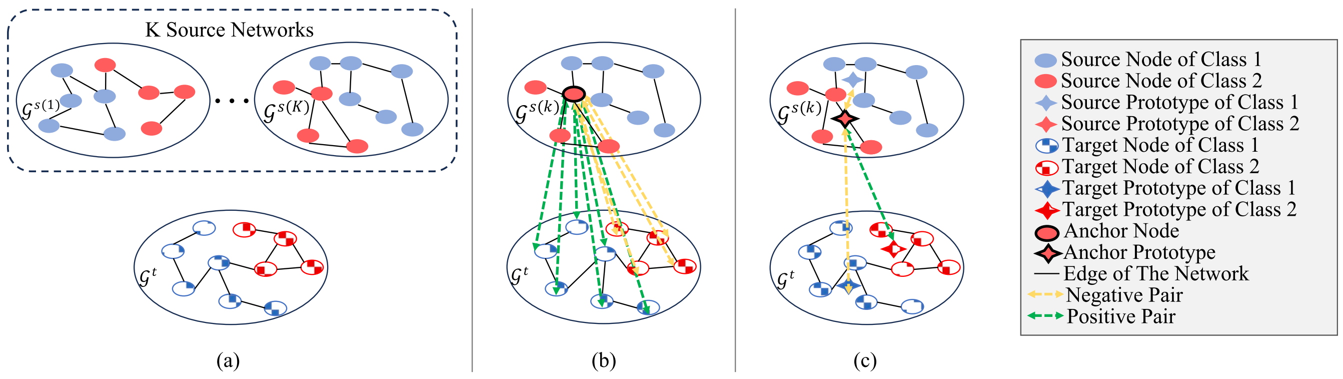

Fig. 3. Illustration of positive and negative pairs in class-aware hierarchical graph contrastive domain adaptation. Different colors correspond to different node classes. Circles and quadrangles denote nodes and prototypes, respectively, and circles and quadrangles with black contour lines denote anchor node and anchor prototype, respectively. (a) The input K labeled source networks $\left\{\mathscr{G}^{s(k)}\right\}_{k=1}^{K}$ and an unlabeled target network $\mathscr{G}^t$. (b) Positive and negative pairs between each pair of the source-target network, i.e., $\mathscr{G}^{s(k)}$ and $\mathscr{G}^t$, in node-level graph contrastive domain adaptation. The anchor node forms positive pairs with the nodes of the same class from different networks, and forms negative pairs with the nodes of different classes from different networks. (c) Positive and negative pairs between $\mathscr{G}^{s(k)}$ and $\mathscr{G}^t$ in prototype-level graph contrastive domain adaptation. The anchor prototype forms a positive pair with the prototype corresponding to the same class in different network, and forms negative pairs with the prototypes of different classes within the same network and from different networks.

adaptation performance, has been ignored by such class-conditional MMD or class-conditional adversarial domain adaptation methods (Kang et al., 2020). To address this, a line of contrastive domain adaptation methods (Kang et al., 2020; Shao et al., 2024; Singh, 2021; Wang et al., 2022a) has been proposed to apply contrastive learning to explicitly model both intra-class domain discrepancy and inter-class domain discrepancy. However, these contrastive domain adaptation methods define positive and negative pairs for contrastive learning (Shen et al., 2023b) between the source and the target domains at a single level, i.e., either sample-level (Kang et al., 2020; Wang et al., 2022a), or centroid-level (Shao et al., 2024; Singh, 2021).

Instead, the proposed MHGCDA customizes a class-aware hierarchical graph contrastive domain adaptation module to reduce the domain shift between each pair of the source-target networks, i.e., $\mathscr{G}^{s(k)}$ and $\mathscr{G}^t$, by defining positive and negative pairs for contrastive learning at both the node and prototype levels. The node-level (prototype-level) graph contrastive domain adaptation forces intra-class distance to be smaller than inter-class distance for nodes (prototypes) from different networks. As a result, the proposed MHGCDA can mitigate the intra-class domain discrepancy and enlarge the inter-class domain discrepancy at both the node and prototype levels.

3.4.1. *Pseudo-labels based on classifier agreement*

Note that in the context of the MSCNNC problem, only nodes in each source network $\mathscr{G}^{s(k)}$ are labeled, while all nodes in the target network $\mathscr{G}^t$ are unlabeled. To perform class-aware hierarchical graph contrastive domain adaptation, pseudo-labels need to be generated for nodes in the target network $\mathscr{G}^t$. Motivated by (Venkat et al., 2020), the proposed MHGCDA assigns pseudo-labels to the target network according to “classifier agreement”. Specifically, when multiple network-specific node classifiers $\left\{f_y^{(k)}\right\}_{k=1}^{K}$ all predict the same class c for a target node v_i^t, then a pseudo-label of class c would be assigned to v_i^t, i.e., $\overline{\overline{Y}}_{ic}^t = 1$; whereas if the label predictions of multiple network-specific node classifiers are not consistent towards v_i^t, then v_i^t remains unlabeled, as:

$$\overline{\overline{Y}}_{ic}^t = \prod_{k=1}^{K} \mathbb{I}\left[\underset{j}{argmax}\left(\widehat{Y}_{ij}^{t(k)}\right) = c\right] \tag{11}$$

where $\mathbb{I}(\bullet)$ is the indicator function that returns 1 when the condition is true, else returns 0.

Utilizing classifier agreement can filter out noisy label predictions on the target nodes, yielding more accurate target pseudo-labels to facilitate class-aware hierarchical graph contrastive domain adaptation. In addition, more agreement would be reached by multiple network-specific node classifiers as iterative model training, thus more target nodes can gradually acquire pseudo-labels.

3.4.2. *Node-level graph contrastive domain adaptation*

MHGCDA conducts node-level graph contrastive domain adaptation to align each pair of the source-target networks, i.e., $\mathscr{G}^{s(k)}$ and $\mathscr{G}^t$, from the perspective of nodes, by defining positive and negative pairs based on the label consistency among nodes from different networks.

Fig. 3 (b) illustrates the definition of positive and negative pairs between $\mathscr{G}^{s(k)}$ and $\mathscr{G}^t$ in the node-level graph contrastive domain adaptation. Since MHGCDA adopts a mini-batch training strategy, we define positive and negative pairs between nodes in the mini-batches sampled from $\mathscr{G}^{s(k)}$ and $\mathscr{G}^t$ (i.e., $\mathbb{B}^{s(k)}$ and $\mathbb{B}^t$). Specifically, choosing the embedding of i-th node in the source mini-batch $\mathbb{B}^{s(k)}$ as the anchor (i.e. $\boldsymbol{h}_i^{s(k)}$), it forms positive pairs with the embedding of each node of the same class in the target mini-batch $\mathbb{B}^t$, and forms negative pairs with the embedding of each node from different classes in the target mini-batch $\mathbb{B}^t$. Let $\mathbb{P}\left(\boldsymbol{h}_i^{s(k)}\right) = \left\{\boldsymbol{h}_j^t | v_j^t \in \mathbb{B}^t, \overline{\overline{Y}}_{jc}^t = Y_{ic}^{s(k)} = 1, \exists c \in \{1, \cdots, C\}\right\}$ denote a set of positive samples and $\mathbb{N}\left(\boldsymbol{h}_i^{s(k)}\right) = \left\{\boldsymbol{h}_j^t | v_j^t \in \mathbb{B}^t, \boldsymbol{h}_j^t \notin \mathbb{P}\left(\boldsymbol{h}_i^{s(k)}\right), \overline{\overline{Y}}_{jc}^t = 1, \exists c \in \{1, \cdots, C\}\right\}$ denote a set of negative samples of the anchor $\boldsymbol{h}_i^{s(k)}$. Note that $\boldsymbol{h}_j^t$ would be selected as a negative sample of the anchor $\boldsymbol{h}_i^{s(k)}$, only if the target node v_j^t has been assigned with a pseudo-label indicating its class is different from $v_i^{s(k)}$. That is, the target nodes not assigned with any pseudo-labels according to classifier agreement would not be selected as a negative sample during node-level graph contrastive domain adaptation. This can alleviate false negative samples in node-level graph contrastive domain adaptation, consequently yielding more accurate class-aware alignment of node embeddings across networks.

According to the aforementioned positive and negative pairs, the node-level graph contrastive domain adaptation loss associated with anchor $\boldsymbol{h}_i^{s(k)}$ is defined by extending the normalized temperature-scaled cross-entropy (NT-Xent) loss (Chen et al., 2020) as:

$$\mathscr{L}_{GCDA_N}\left(\boldsymbol{h}_i^{s(k)}\right) = -log\frac{\frac{1}{\left|\mathbb{P}\left(\boldsymbol{h}_i^{s(k)}\right)\right|}\sum_{\boldsymbol{h}_j^t\in\mathbb{P}\left(\boldsymbol{h}_i^{s(k)}\right)}\exp\left(\text{sim}\left(\boldsymbol{h}_i^{s(k)},\boldsymbol{h}_j^t\right)/\tau\right)}{\underbrace{\sum_{\boldsymbol{h}_j^t\in\mathbb{P}\left(\boldsymbol{h}_i^{s(k)}\right)}\exp\left(\text{sim}\left(\boldsymbol{h}_i^{s(k)},\boldsymbol{h}_j^t\right)/\tau\right)}_{\text{Positive pairs}}+\underbrace{\sum_{\boldsymbol{h}_l^t\in\mathbb{N}\left(\boldsymbol{h}_i^{s(k)}\right)}\exp\left(\text{sim}\left(\boldsymbol{h}_i^{s(k)},\boldsymbol{h}_l^t\right)/\tau\right)}_{\text{Negative pairs}}} \tag{12}$$

where $\text{sim}(u,v)=\frac{u^Tv}{\|u\|_2\|v\|_2}$ denotes the cosine similarity function and τ represents the temperature coefficient. $\left|\mathbb{P}\left(\boldsymbol{h}_i^{s(k)}\right)\right|$ is the number of positive samples of the anchor $\boldsymbol{h}_i^{s(k)}$. Optimizing Eq. (12) would pull positive pairs (i.e., cross-network nodes with the same class label) closer in the embedding space, while pushing negative pairs (i.e., cross-network nodes with different class labels) farther apart. As a result, the intra-class distance would be smaller than the inter-class distance for nodes from different networks. Mitigating the intra-class distance can compact the node embeddings within a class. While enlarging the inter-class distance can push the node embeddings of different classes further away from the decision boundary so as to better separate nodes from different classes, which is indeed conductive for the downstream node classification task.

In Eq. (12), we consider the embedding of the node from the source mini-batch $\mathbb{B}^{s(k)}$ as the anchor. Alternatively, we can take the embedding of the j-th node in the target mini-batch $\mathbb{B}^t$ as the anchor (i.e. $\boldsymbol{h}_j^t$) and similarly define its positive samples as the embeddings of cross-network nodes sharing the same class label $\mathbb{P}\left(\boldsymbol{h}_j^t\right)=\left\{\boldsymbol{h}_i^{s(k)}|v_i^{s(k)}\in\mathbb{B}^{s(k)},\overline{\overline{Y}}_{jc}^t=Y_{ic}^{s(k)}=1,\exists c\in\{1,\cdots,C\}\right\}$ and its negative samples as the embeddings of cross-network nodes belonging to different classes $\mathbb{N}\left(\boldsymbol{h}_j^t\right)=\left\{\boldsymbol{h}_i^{s(k)}|v_i^{s(k)}\in\mathbb{B}^{s(k)},\boldsymbol{h}_i^{s(k)}\notin\mathbb{P}\left(\boldsymbol{h}_j^t\right)\right\}$. Then, we can compute $\mathscr{L}_{GCDA_N}\left(\boldsymbol{h}_j^t\right)$ similar to Eq. (12). By selecting the embedding of each node in the source mini-batch $\mathbb{B}^{s(k)}$ and the target network $\mathbb{B}^t$ as the anchor, the node-level graph contrastive domain adaptation loss between $\mathscr{G}^{s(k)}$ and $\mathscr{G}^t$ is computed as:

$$\mathscr{L}_{GCDA_N}\left(\mathscr{G}^{s(k)},\mathscr{G}^t\right)=\frac{1}{\left|\mathbb{B}^{s(k)}\right|}\sum_{i=1}^{\left|\mathbb{B}^{s(k)}\right|}\mathscr{L}_{GCDA_N}\left(\boldsymbol{h}_i^{s(k)}\right)+\frac{1}{\left|\mathbb{B}^t\right|}\sum_{j=1}^{\left|\mathbb{B}^t\right|}\mathscr{L}_{GCDA_N}\left(\boldsymbol{h}_j^t\right) \tag{13}$$

Minimizing Eq. (13) mitigates the domain shift among nodes of the same category while maximizing the domain shift among nodes of different categories, so as to ensure the alignment of the class-conditional distributions of nodes between the k-th source network $\mathscr{G}^{s(k)}$ and the target network $\mathscr{G}^t$. Then, computing the node-level graph contrastive domain adaptation loss for each pair of the source-target networks, i.e., $\mathscr{G}^{s(k)}$ and $\mathscr{G}^t$, $\forall 1\le k\le K$, and weighted by the corresponding transferability weight w^k of each k-th source network in Eq. (8), the overall node-level graph contrastive domain adaptation loss is defined as:

$$\mathscr{L}_{GCDA_N}=\sum_{k=1}^{K}w^k\mathscr{L}_{GCDA_N}\left(\mathscr{G}^{s(k)},\mathscr{G}^t\right) \tag{14}$$

3.4.3. *Prototype-level graph contrastive domain adaptation*

Next, MHGCDA conducts the prototype-level graph contrastive domain adaptation to align the class-conditional distribution between each source-target network pair, i.e., $\mathscr{G}^{s(k)}$ and $\mathscr{G}^t$, by defining positive and negative pairs for contrastive learning between prototypes. Prototypes play a crucial role as compact representations of class-specific features. Since MHGCDA adopts the mini-batch training strategy, the prototype representation of each class is updated based on the embeddings of nodes belonging to the specific class in the mini-batch. Specifically, the prototype of class c in the k-th source network $\mathscr{G}^{s(k)}$ is computed as the mean of the embeddings of all nodes in the source mini-batch $\mathbb{B}^{s(k)}$ that are labeled as class c, i.e., $\boldsymbol{p}_c^{s(k)}=\frac{\sum_{i=1}^{\left|\mathbb{B}^{s(k)}\right|}Y_{ic}^{s(k)}\boldsymbol{h}_i^{s(k)}}{\sum_{i=1}^{\left|\mathbb{B}^{s(k)}\right|}Y_{ic}^{s(k)}}$. Similarly, the prototype of class c in the target network $\mathscr{G}^t$ is computed based on the embeddings of the nodes whose pseudo-labels are class c in the target mini-batch $\mathbb{B}^t$ as $\boldsymbol{p}_c^t=\frac{\sum_{j=1}^{|\mathbb{B}^t|}\overline{\overline{Y}}_{jc}^t\boldsymbol{h}_j^t}{\sum_{j=1}^{|\mathbb{B}^t|}\overline{\overline{Y}}_{jc}^t}$. It is worth noting that the target prototypes are computed based on the pseudo-labels. To guarantee accurate target prototype extraction, it is required to obtain reliable and accurate pseudo-labels. In the proposed MHGCDA, we assign pseudo-labels to the target nodes according to "classifier agreement", that is, only when all network-specific node classifiers predict the same class-label for a target node, then the corresponding pseudo-label can be assigned to the target node. Such "classifier agreement" strategy guarantees reliable pseudo-labels to promote accurate target prototype extraction.

Fig. 3(c) illustrates the definition of positive and negative pairs between $\mathscr{G}^{s(k)}$ and $\mathscr{G}^t$ in the prototype-level graph contrastive domain adaptation. By selecting the prototype of class c in $\mathscr{G}^{s(k)}$ as the anchor (i.e. $\boldsymbol{p}_c^{s(k)}$), it forms a positive pair with the prototype of the same class in $\mathscr{G}^t$ (i.e., $\boldsymbol{p}_c^t$). While it forms negative pairs with the prototypes of different classes in both $\mathscr{G}^{s(k)}$ (i.e., $\{\boldsymbol{p}_l^{s(k)}|l\neq c,l\in\{1,\cdots,C\}\}$) and $\mathscr{G}^t$ (i.e., $\{\boldsymbol{p}_l^t|l\neq c,l\in\{1,\cdots,C\}\}$).

According to the aforementioned prototype-level positive and negative pairs, the prototype-level graph contrastive domain adaptation loss associated with anchor $\boldsymbol{p}_c^{s(k)}$ is defined using the modified NT-Xent loss (Chen et al., 2020) as:

$$\mathscr{L}_{GCDA_P}\left(\boldsymbol{p}_c^{s(k)}\right)=-log\frac{\exp\left(\text{sim}\left(\boldsymbol{p}_c^{s(k)},\boldsymbol{p}_c^t\right)/\tau\right)}{\underbrace{\exp\left(\text{sim}\left(\boldsymbol{p}_c^{s(k)},\boldsymbol{p}_c^t\right)/\tau\right)}_{\text{Positive pair}}+\underbrace{\sum_{l=1}^{C}\mathbb{I}(l\neq c)\left(\exp\left(\text{sim}\left(\boldsymbol{p}_c^{s(k)},\boldsymbol{p}_l^{s(k)}\right)/\tau\right)+\exp\left(\text{sim}\left(\boldsymbol{p}_c^{s(k)},\boldsymbol{p}_l^t\right)/\tau\right)\right)}_{\text{Negative pairs}}} \tag{15}$$

where $\mathbb{I}(\bullet)$ is the indicator function that returns 1 when the condition is true, else returns 0. On one hand, minimizing Eq. (15) maximizes the agreement between positive pairs (i.e. cross-network prototypes corresponding to the same class), so as to mitigate the inter-class domain discrepancy by reducing the discrepancy between the clusters of the same category across networks. On the other hand, minimizing Eq. (15) minimizes the agreement between negative pairs (i.e., prototypes of different classes within a network and from different networks). As a result, the inter-class distance within a network and across networks can be explicitly enlarged, yielding more label-discriminative and class-conditional network-invariant node embeddings to allow for effective transfer of knowledge between the source and target networks.

Then, by selecting the prototype of class c in $\mathscr{G}^t$ as the anchor (i.e. $\boldsymbol{p}_c^t$), the loss $\mathscr{L}_{GCDA_P}(\boldsymbol{p}_c^t)$ can be calculated in a manner similar to Eq. (15). By selecting the prototype of each class in $\mathscr{G}^{s(k)}$ and $\mathscr{G}^t$ as the anchor, the prototype-level graph contrastive domain adaptation loss between $\mathscr{G}^{s(k)}$ and $\mathscr{G}^t$ is computed as:

$$\mathscr{L}_{GCDA_P}\left(\mathscr{G}^{s(k)}, \mathscr{G}^t\right) = \frac{1}{C}\sum_{c=1}^{C}\left(\mathscr{L}_{GCDA_P}\left(\boldsymbol{p}_c^{s(k)}\right) + \mathscr{L}_{GCDA_P}\left(\boldsymbol{p}_c^t\right)\right) \tag{16}$$

Similar to Eq. (14), computing the prototype-level graph contrastive domain adaptation loss for each pair of the source-target networks $\left(\mathscr{G}^{s(k)}, \mathscr{G}^t\right)_{k=1}^K$, and weighted by the transferability weight w^k of each k-th source network $\mathscr{G}^{s(k)}$, the overall prototype-level graph contrastive domain adaptation loss can be obtained as:

$$\mathscr{L}_{GCDA_P} = \sum_{k=1}^{K} w^k \mathscr{L}_{GCDA_P}\left(\mathscr{G}^{s(k)}, \mathscr{G}^t\right) \tag{17}$$

3.5. *Positive and negative pseudo-labeling*

In the aforementioned process, the network-specific node classifiers are trained based on the observed labels from the source networks. To further exploit the potential supervised signals from the target network to train the network-specific node classifiers, MHGCDA introduces a positive and negative pseudo-labeling module to assign pseudo-labels to the nodes in the target network. Fig. 4 illustrates the idea of positive and negative pseudo-labeling. As shown in Fig. 4, it assigns corresponding positive pseudo-labels to nodes in the target network that are highly likely to belong to a specific class, and assigns corresponding negative pseudo-labels to nodes in the target network that are highly unlikely to belong to a specific class.

On one hand, a target node v_j^t in $\mathbb{B}^t$ would be assigned with a positive pseudo-label of class c, if and only if all the K network-specific node classifiers predict that v_j^t is highly likely to belong to class c, i.e., the corresponding predicted probabilities by all the K network-specific node classifiers are greater than a positive pseudo-labeling threshold τ_p, $\left\{\widehat{Y}_{jc}^{t(k)} > \tau_p, \forall 1 \le k \le K\right\}$. Then, the target nodes with positive pseudo-labels would be employed to iteratively re-train each k-th network-specific node classifier in a self-training manner by optimizing the positive pseudo-labeling loss, as:

$$\mathscr{L}_{pl}\left(\widehat{\boldsymbol{Y}}^{t(k)}\right) = -\sum_{j=1}^{|\mathbb{B}^t|}\sum_{c=1}^{C}\frac{1}{N_c^{pl}}\left[\left(\prod_{k=1}^{K}\mathbb{I}\left[\widehat{Y}_{jc}^{t(k)} > \tau_p\right]\mathbb{I}\left[\underset{l}{argmax}\left(\widehat{Y}_{jl}^{t(k)}\right)\right.\right.\right.$$
$$\left.\left.\left. = c\right]\right) log\left(\widehat{Y}_{jc}^{t(k)}\right)\right] \tag{18}$$

where $N_c^{pl} = \sum_{j=1}^{|\mathbb{B}^t|}\prod_{k=1}^{K}\mathbb{I}\left[\widehat{Y}_{jc}^{t(k)} > \tau_p\right]\mathbb{I}\left[argmax_l\left(\widehat{Y}_{jl}^{t(k)}\right) = c\right]$ is the number of the target nodes in $\mathbb{B}^t$ which have been assigned with the positive pseudo-label of class c. The positive pseudo-label supervises the model that "the target node belongs to the corresponding class". Thus, by optimizing the positive pseudo-labeling loss, if a target node v_j^t is assigned with a positive pseudo-label of class c, then the predicted probability $\widehat{Y}_{jc}^{t(k)}$ would be optimized to be 1.

On the other hand, if all the network-specific node classifiers predict that v_j^t is highly unlikely to belong to class c, i.e., the corresponding predicted probabilities are lower than a negative pseudo-labeling threshold τ_n for all the K network-specific node classifiers, then v_j^t would be assigned with a negative pseudo-label of class c. The target nodes assigned with negative pseudo-labels would be also employed to iteratively re-train each k-th network-specific node classifier, by optimizing the following negative pseudo-labeling loss:

$$\mathscr{L}_{nl}\left(\widehat{\boldsymbol{Y}}^{t(k)}\right) = -\sum_{j=1}^{|\mathbb{B}^t|}\sum_{c=1}^{C}\frac{1}{N_c^{nl}}\left[\left(\prod_{k=1}^{K}\mathbb{I}\left[\widehat{Y}_{jc}^{t(k)} < \tau_n\right]\mathbb{I}\left[\underset{l}{argmax}\left(\widehat{Y}_{jl}^{t(k)}\right)\right.\right.\right.$$
$$\left.\left.\left. \neq c\right]\right) log\left(1 - \widehat{Y}_{jc}^{t(k)}\right)\right] \tag{19}$$

where $N_c^{nl} = \sum_{j=1}^{|\mathbb{B}^t|}\prod_{k=1}^{K}\mathbb{I}\left[\widehat{Y}_{jc}^{t(k)} < \tau_n\right]\mathbb{I}\left[arg\,max_l\left(\widehat{Y}_{jl}^{t(k)}\right) \neq c\right]$ denotes the number of target nodes assigned with negative pseudo-label of class c in $\mathbb{B}^t$. In contrast to positive pseudo-label, the negative pseudo-label supervises the model that "the target node does not belong to the corresponding class". By optimizing the negative pseudo-labeling loss, if a target node v_j^t is assigned with a negative pseudo-label of class c, then the predicted probability $\widehat{Y}_{jc}^{t(k)}$ would be optimized to be 0, which in turn increases the predicted probabilities of other classes except class c.

Next, weighting the positive and negative pseudo-labeling losses of each k-th network-specific node classifier according to the transferability weight w^k of each source network, the total pseudo-labeling loss is defined as:

$$\mathscr{L}_P = \sum_{k=1}^{K} w^k\left(\mathscr{L}_{pl}\left(\widehat{\boldsymbol{Y}}^{t(k)}\right) + \mathscr{L}_{nl}\left(\widehat{\boldsymbol{Y}}^{t(k)}\right)\right) \tag{20}$$

By optimizing the overall pseudo-labeling loss, an increasing number of positive and negative pseudo-labeled target nodes can participate in the iterative self-training process to re-train the network-specific node classifiers. Therefore, the proposed MHGCDA can effectively leverage the potential supervisory information from more target nodes, gradually generating more accurate target pseudo-labels to promote class-aware hierarchical graph contrastive domain adaptation.

3.6. *Overall loss of MHGCDA*

The overall loss function of MHGCDA is given by integrating the node classification loss $\mathscr{L}_Y$, the node-level graph contrastive domain adaptation loss $\mathscr{L}_{GCDA_N}$, the prototype-level graph contrastive domain adaptation loss $\mathscr{L}_{GCDA_P}$, and the pseudo-labeling loss $\mathscr{L}_P$, as:

Algorithm 1: MHGCDA

```
Input: K source network $\mathscr{G}^{s(k)} = \left(\mathscr{V}^{s(k)}, \mathscr{E}^{s(k)}, \mathbf{A}^{s(k)}, \mathbf{X}^{s(k)}, \mathbf{Y}^{s(k)}\right)$ and the target
    network $\mathscr{G}^t = (\mathscr{V}^t, \mathscr{E}^t, \mathbf{A}^t, \mathbf{X}^t)$; trade-off parameters $\phi$, $\vartheta$ and $\psi$.
1   Initialize learnable parameters $\theta_h, \left\{\theta_y^{(k)}\right\}_{k=1}^K$;
2   while not max epoch do
3       while not max iteration do
4           Sample source mini-batch $\left\{\mathbb{B}^{s(k)}\right\}_{k=1}^K$ from $\left\{\mathscr{G}^{s(k)}\right\}_{k=1}^K$ and sample target
            mini-batch $\mathbb{B}^t$ from the $\mathscr{G}^t$;
5           Learn multi-source cross-network node embeddings $\left\{\boldsymbol{h}_i^{s(k)}\right\}_{i=1}^{|\mathbb{B}^{s(k)}|}$ of
            $\left\{\mathbb{B}^{s(k)}\right\}_{k=1}^K$ and $\left\{\boldsymbol{h}_j^t\right\}_{j=1}^{|\mathbb{B}^t|}$ of $\mathbb{B}^t$ via the GNN encoder $f_h$ in Eqs. (1)–(2);
```

(continued on next page)

(*continued*)

```
Algorithm 1: MHGCDA
6       Compute the transferability weight $w^k$ of each $k$-th source network based on the information entropy of the corresponding network-specific node classifier $\left\{\mathscr{H}\left(f_y^{(k)}\left(\boldsymbol{h}_j^t\right)\right)\right\}_{j=1}^{|\mathbb{B}^t|}$ in Eqs. (5)–(8);
7       Compute node classification loss $\mathscr{L}_y$ based on $\left\{\left(f_y^{(k)}\left(\boldsymbol{h}_i^{s(k)}, Y_i^{s(k)}\right)\right\}_{i=1}^{|\mathbb{B}^{s(k)}|}$ in Eqs. (3)–(4) and Eq. (9);
8       Conduct node-level graph contrastive domain adaptation on $\left\{\boldsymbol{h}_i^{s(k)}, \boldsymbol{h}_j^t, Y_i^{s(k)}, \overline{\overline{Y}}_j^t\right\}_{i,j=1}^{b}$ and compute node-level graph contrastive domain adaptation loss $\mathscr{L}_{GCDA_N}$ in Eqs. (12)–(14);
9       Conduct prototype-level graph contrastive domain adaptation on $\left\{\left\{\boldsymbol{p}_c^{s(k)}\right\}_{c=1}^{C}, \left\{\boldsymbol{p}_c^t\right\}_{c=1}^{C}\right\}$ and compute prototype-level graph contrastive domain adaptation loss $\mathscr{L}_{GCDA_P}$ in Eqs. (15)–(17);
10      Compute pseudo-labeling loss $\mathscr{L}_P$ based on $\left\{\widehat{Y}_j^{t(k)}\right\}_{j=1}^{|\mathbb{B}^t|}$ in Eqs. (18)–(20);
11      Backpropagate and update learnable parameters $\theta_h$, $\left\{\theta_y^{(k)}\right\}_{k=1}^{K}$ to optimize Eq. (21);
12    end while
13    Update the pseudo-label $\overline{\overline{\boldsymbol{Y}}}^t$ of the target network using the integrated prediction result $\widehat{\boldsymbol{Y}}^{t(k)}$ from network-specific node classifiers $\left\{f_y^{(k)}\right\}_{k=1}^{K}$ in Eq. (11);
14  end while
15  Predict the node label matrix $\widehat{\boldsymbol{Y}}^t$ of the target network by weighting the prediction $\widehat{\boldsymbol{Y}}^{t(k)}$ from all the network-specific node classifiers $\left\{f_y^{(k)}\right\}_{k=1}^{K}$ in Eq. (10).
Output: Predicted node label matrix of the target network $\widehat{\boldsymbol{Y}}^t$.
```

$$\mathscr{L} = \sum_{k=1}^{K} w^k \mathscr{L}_y + \phi \sum_{k=1}^{K} w^k \mathscr{L}_{GCDA_N} + \vartheta \sum_{k=1}^{K} w^k \mathscr{L}_{GCDA_P} + \psi \sum_{k=1}^{K} w^k \mathscr{L}_P \tag{21}$$

where ϕ, ϑ, and ψ are the trade-off parameters to balance the effects of different losses. Algorithm 1 illustrates the training process of MHGCDA.

Table 2
Statistics of the Real-world Networked Datasets.

Citation Network	Dataset	# Nodes	# Edges	# Node Attributes	# Node Labels
Citation Network I	ACMv9 (A_I)	9360	15,602	6775	5
	Citationv1 (C_I)	8935	15,113		
	DBLPv7 (D_I)	5484	8130		
Citation Network II	ACMv8 (A_{II})	8806	17,661	7786	9
	Citationv1 (C_{II})	9737	14,054		
	DBLPv4 (D_{II})	8653	12,967		

3.7. Complexity Analysis

The time complexity of the GNN encoder on the source mini-batches $\left\{\mathbb{B}^{s(k)}\right\}_{k=1}^{K}$ and the target mini-batch $\mathbb{B}^t$ is $O\left(d\left(b\mathbb{w} + \left(\left|\mathscr{E}^{\mathbb{B}^t}\right| + \sum_{k=1}^{K}\left|\mathscr{E}^{\mathbb{B}^{s(k)}}\right|\right)\right)\right)$, where $b = \left|\mathbb{B}^{s(k)}\right| = |\mathbb{B}^t|$ is the batch size, $\left|\mathscr{E}^{\mathbb{B}^t}\right|$ and $\left|\mathscr{E}^{\mathbb{B}^{s(k)}}\right|$ are the numbers of edges in $\mathbb{B}^t$ and $\mathbb{B}^{s(k)}$, $\mathbb{w}$ is the number of node attributes, d is the number of dimensions of node embeddings, and K is the number of source networks. The time complexity of the network-specific node classifiers on $\left\{\mathbb{B}^{s(k)}\right\}_{k=1}^{K}$ and $\mathbb{B}^t$ is $O(Kbd)$. The time complexity of the node-level graph contrastive domain adaptation across $\left\{\mathbb{B}^{s(k)}\right\}_{k=1}^{K}$ and $\mathbb{B}^t$ is $O\left(Kb^2d\right)$, and the time complexity of prototype-level graph contrastive domain adaptation is $O(KC^2d)$, where C represents the number of node categories. The time complexity of the pseudo-labeling module is $O(KbC)$. Since $\left|\mathbb{B}^{s(k)}\right| \le b^2$, $\left|\mathscr{E}^{\mathbb{B}^t}\right| \le b^2$, the time complexity of MHGCDA on $\left\{\mathbb{B}^{s(k)}\right\}_{k=1}^{K}$ and $\mathbb{B}^t$ is $O\left(db\mathbb{w} + Kb^2d + KC^2d + KbC\right)$. In addition, $\left\{\mathscr{G}^{s(k)}\right\}_{k=1}^{K}$ and $\mathscr{G}^t$ have a maximum number of batches of n^{st}/b, where $n^{st} = \max\left(\left\{n^{s(k)}\right\}_{k=1}^{K}, n^t\right)$, $n^{s(k)}$ and n^t denote the number of nodes in $\mathscr{G}^{s(k)}$ and $\mathscr{G}^t$, respectively.

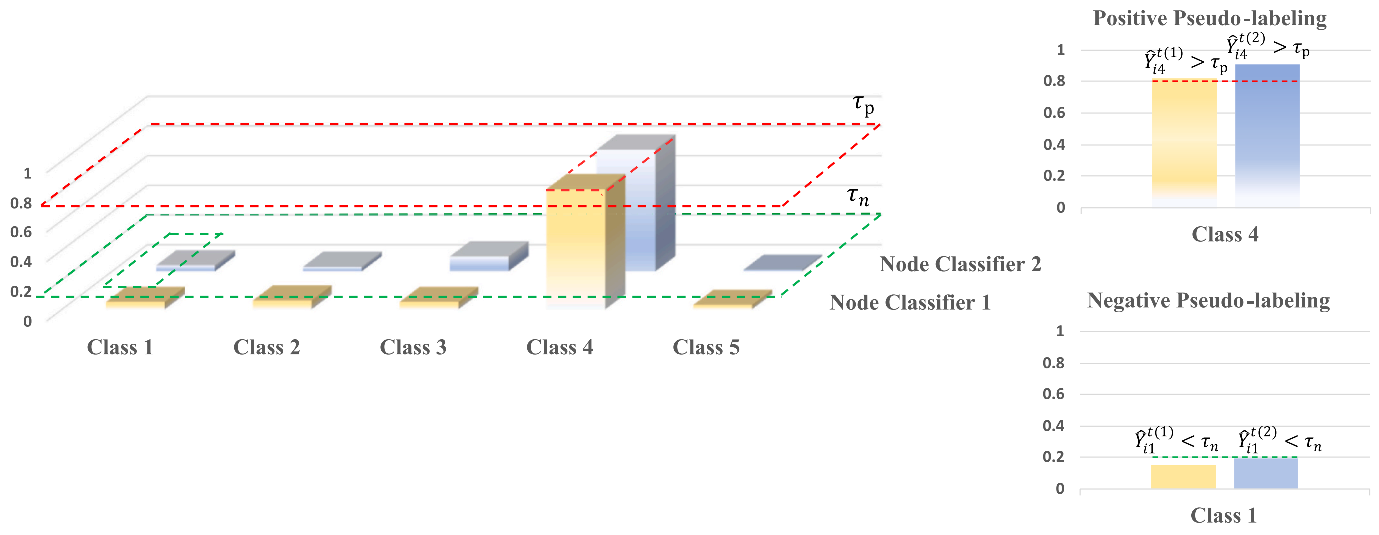

Fig. 4. Illustration of the positive and negative pseudo-labeling module. Yellow indicates the predicted results of the first network-specific node classifier, blue represents the predicted probabilities of the second network-specific node classifier. $\widehat{Y}_{ic}^{t(k)}$ denotes the predicted probability by the k-th network-specific node classifier that node v_i belongs to class c. Positive pseudo-label of class c would be assigned to node v_i if the predicted probabilities of node v_i belonging to class c by all network-specific node classifiers are greater than τ_p. Negative pseudo-label of class c would be assigned to node v_i if the predicted probabilities of node v_i belonging to class c by all network-specific node classifiers are lower than τ_n. $\widehat{Y}_{i4}^{t(1)} > \tau_p$ and $\widehat{Y}_{i4}^{t(2)} > \tau_p$ imply that the predicted probability of node v_i belonging to class 4 is sufficiently high by all the network-specific node classifiers. While $\widehat{Y}_{i1}^{t(1)} < \tau_n$ and $\widehat{Y}_{i1}^{t(2)} < \tau_n$ mean that the predicted probability of v_i belonging to class 1 is sufficiently low by all the network-specific node classifiers. Thus, node v_i would be assigned with a positive pseudo-label of class 4 and a negative pseudo-label of class 1.

Thus, the total time complexity of MHGCDA is $O\left(n^{st}\left(d\mathbb{W} + Kbd + \frac{KC^2d}{b} + KC\right)\right)$.

4. Experiments

In this section, we conducted extensive experiments on six real-world citation networks to investigate the following three research questions (RQs):

RQ1: How does the performance of MHGCDA compared to state-of-the-art baseline methods?

RQ2: How do different model variants of MHGCDA impact the performance?

RQ3: How do hyper-parameters affect the performance of MHGCDA?

4.1. Experimental Setup

4.1.1. Dataset

In our experiments, we conducted experiments on two sets of citation networked datasets, referred to as Citation Network I and Citation Network II. Citation Network I (Shen, 2021) includes three benchmark graph datasets widely used to evaluate the CNNC tasks, i.e., ACMv9, Citationv1, and DBLPv7. Citation Network II (Shen et al., 2025) comprises another three newly constructed CNNC benchmark datasets, i.e., ACMv8, Citationv1, and DBLPv4. The statistics of each dataset was provided in Table 2 Each citation dataset was modeled as an undirected network in our experiments, where each node represents a research paper, and each edge indicates a citation relationship between two papers. Node attributes were extracted from the keywords in the paper titles. The node class represents the research topic of the paper. Within each set of citation networked datasets (i.e. Citation Network I and Citation Network II), one network is chosen as the target network $\mathscr{G}^t$, while the remaining K networks serve as the source networks $\left\{\mathscr{G}^{s(k)}\right\}_{k=1}^{K}$. Specifically, in Citation Network I, we can construct three MSCNNC tasks, i.e., $\mathrm{C_I D_I} \rightarrow \mathrm{A_I}$, $\mathrm{A_I D_I} \rightarrow \mathrm{C_I}$ and $\mathrm{A_I C_I} \rightarrow \mathrm{D_I}$, where $\mathrm{A_I}$, $\mathrm{C_I}$ and $\mathrm{D_I}$ correspond to ACMv9, Citationv1, and DBLPv7 in Citation Network I, respectively. Similarly, in Citation Network II, we can create three MSCNNC tasks: $\mathrm{C_{II} D_{II}} \rightarrow \mathrm{A_{II}}$, $\mathrm{A_{II} D_{II}} \rightarrow \mathrm{C_{II}}$ and $\mathrm{A_{II} C_{II}} \rightarrow \mathrm{D_{II}}$, where $\mathrm{A_{II}}$, $\mathrm{C_{II}}$ and $\mathrm{D_{II}}$ represent ACMv8, Citationv1, and DBLPv4 in Citation Network II, respectively.

4.1.2. Baselines

We validate the effectiveness of MHGCDA by comparing it with existing single-source CNNC methods and multi-source domain adaptation methods, following three evaluation standards defined in the multi-source domain adaptation literature (Venkat et al., 2020): (1) **Single Best**: the best performance of single-source CNNC methods across all source networks; (2) **Single Combine**: merging all source networks into one source network to perform single-source CNNC; (3) **Multi-source**: adaptation from all source networks to the target network.

In the Single Best and Single Combine evaluation standards, we compared MHGCDA with the single-source CNNC methods. **ASN** (Zhang et al., 2021) employs dual-GCN to learn node embeddings, combining deep network embeddings with adversarial domain adaptation to reduce cross-domain distribution differences. **AdaGCN** (Dai, 2022) generates discriminative node embeddings based on information in the source and target networks, reducing domain adaptation through Wasserstein distance-guided adversarial domain adaptation. **UDAGCN** (Wu et al., 2020a) utilizes global and local information to generate node embeddings through a dual-graph convolutional network component, leveraging GRL-based adversarial domain adaptation (Ganin et al., 2016) to reduce domain differences. **GRADE** (Wu et al., 2023) is a node–node GCL method that learns node representations through corruption to generate two views and maximizes the consistency of node representations between views.

For the Multi-source evaluation standard, we selected two multi-source domain adaptation methods as baselines. **SImpAI** (Venkat et al., 2020) generates pseudo-labels for target samples based on classifier protocols and leverages labeled data from multiple source domains to train shared classifiers for alignment. **MIEM** (Wen et al., 2023) aligns both source joint distributions and target joint distribution by estimating and minimizing mutual information in the network latent feature space. Note that SImpAI and MIEM were originally developed in the CV domain. To adjust them to better process graph structured data in MSCNNC, we further constructed two stronger baselines by replacing their feature extractors with the GAT encoder, and denoted them as **SImpAI (GAT)** and **MIEM (GAT)**.

We adopted Micro-F1 and Macro-F1 to evaluate the performance of MSCNNC, following previous CNNC literatures (Dai, 2022; Shen, 2020; Shen, 2021). In our experiments, for each task, we run each comparing algorithm five times with different random initializations and report the average scores of Micro-F1 and Macro-F1.

4.1.3. Implementation details

The proposed MHGCDA[1] was implemented using PyTorch 1.13.0 and Deep Graph Library (DGL) 0.6.1. We trained MHGCDA for 100 epochs using the Adam optimizer. Following (Shen, 2020), we decayed the learning rate following the formula $\mu_p = \frac{\mu_0}{(1+10p)^{0.75}}$, where μ_0 is the initial learning rate chosen from $\{0.01, 0.001\}$, and p linearly progresses from 0 to 1 during training. The batch size b was chosen from $\{2000, 4000, 5000, 6000\}$. A ℓ_2-norm regularization was applied to the trainable parameters to prevent overfitting, and weight decay was selected from $\{0.01, 0.001, 0.0001\}$. The number of attention heads K_h in the network-specific node classifier f_h was chosen from $\{16, 32\}$, the number of attention heads K_y in the GNN encoder f_y was chosen from $\{1, 2, 3\}$, the number of embedding dimensions d for each attention head was chosen from $\{8, 16\}$, and the number of layers $\mathbb{L}$ in the GNN encoder was chosen from $\{1, 2, 3\}$. The weight of the node-level graph contrastive domain adaptation loss ϕ and the temperature coefficient τ were set to 1. The weight of the pseudo-labeling loss ψ was chosen from $\{1, 0.1, 0.001\}$ and the weight of prototype-level graph contrastive domain adaptation loss ϑ was chosen from $\{1, 0.1\}$. The threshold τ_p for positive pseudo-labeling was chosen from $\{0.8, 0.9\}$, and the threshold τ_n for negative pseudo-labeling was chosen from $\{0.1, 0.2, 0.3\}$. The hyper-parameters of the single-source CNNC baselines were set to the optimal values according to their papers. The hyper-parameter settings of the GAT encoder in the multi-source domain adaptation baselines were consistent with our MHGCDA for fair comparisons.

4.2. Performance Comparison (RQ1)

We presented the experimental results of all comparative methods on Citation Network I and Citation Network II datasets in Tables 3 and 4 respectively. We have the following observations:

Firstly, the results of the Single Combine baselines are generally better than those of the Single Best baselines. This reflects the fact that migrating knowledge from multiple source networks to the target network can achieve better results in the MSCNNC task as opposed to using the knowledge from a single source network.

Secondly, in the multi-source evaluation standard, the performance of the multi-source domain adaptation methods without adding the GAT component are significantly worse than those with the GAT component. This is attributed to the fact that the multi-source domain adaptation methods were originally designed for image data in the CV domain and do not consider the connections between nodes in the graph-structured

[1] The code and datasets are available at https://github.com/SHL185416/MHGCDA.

Table 3
F1 Scores of MSCNNC on the Citation Network I Datasets. The Best Results among All Comparison Algorithms are Displayed in Bold. The Numbers in Parentheses Are the Standard Deviations Over 5 Random Initializations.

Task	F1 (%)	Single Best				Single Combine				Multi-Source				
		ASN	AdaGCN	UDAGCN	GRADE	ASN	AdaGCN	UDAGCN	GRADE	MIEM	SImpAI	MIEM (GAT)	SImpAI (GAT)	MHGCDA
$C_ID_I \rightarrow A_I$	Micro-F1	70.54 (3.15)	59.52 (3.23)	67.27 (1.04)	67.36 (0.38)	69.23 (2.92)	63.97 (1.11)	68.87 (3.44)	68.00 (0.58)	52.18 (0.36)	42.44 (1.25)	53.09 (0.40)	67.78 (0.32)	**73.89** (0.52)
	Macro-F1	71.02 (3.37)	58.65 (4.98)	66.84 (2.09)	66.97 (0.59)	66.13 (5.41)	64.47 (1.27)	69.50 (4.29)	68.07 (0.92)	50.33 (0.62)	37.06 (2.41)	50.52 (0.58)	66.13 (0.52)	**75.10** (0.51)
$A_ID_I \rightarrow C_I$	Micro-F1	75.99 (3.83)	71.50 (2.16)	70.01 (2.62)	72.21 (0.69)	80.37 (0.43)	62.23 (3.29)	77.54 (0.68)	74.49 (0.59)	55.27 (1.60)	46.30 (3.11)	58.55 (0.57)	72.97 (3.95)	**83.49** (0.34)
	Macro-F1	71.07 (3.27)	69.07 (2.34)	65.92 (3.20)	69.24 (0.74)	77.00 (0.54)	61.06 (3.42)	76.11 (0.55)	72.21 (0.59)	53.15 (1.24)	41.89 (3.91)	55.33 (0.59)	69.57 (2.73)	**82.57** (0.35)
$A_IC_I \rightarrow D_I$	Micro-F1	74.77 (1.63)	71.36 (1.82)	74.47 (0.47)	70.76 (0.88)	73.86 (1.41)	68.74 (0.87)	73.93 (0.17)	70.03 (0.72)	56.77 (0.62)	46.09 (3.78)	62.31 (0.74)	70.81 (3.07)	**76.32** (0.57)
	Macro-F1	71.20 (2.89)	68.82 (1.63)	73.16 (0.37)	67.45 (0.65)	70.20 (1.90)	65.40 (0.82)	73.23 (0.20)	67.40 (1.01)	53.79 (0.83)	40.98 (4.49)	58.90 (0.60)	68.09 (2.07)	**74.93** (0.66)
Avg	Micro-F1	73.77 (2.87)	67.46 (2.40)	70.58 (1.38)	70.11 (0.65)	74.49 (1.59)	64.98 (1.76)	73.45 (1.43)	70.84 (0.63)	54.74 (0.86)	44.94 (2.71)	57.98 (0.57)	70.52 (2.45)	**77.90** (0.47)
	Macro-F1	71.10 (3.18)	65.51 (2.98)	68.64 (1.89)	67.89 (0.66)	71.11 (2.62)	63.64 (1.84)	72.95 (1.68)	69.23 (0.84)	52.42 (0.89)	39.98 (3.60)	54.91 (0.59)	67.93 (1.77)	**77.53** (0.50)

Table 4
F1 Scores of MSCNNC on the Citation Network II Datasets. The Best Results among All Comparison Algorithms are Displayed in Bold. The Numbers in Parentheses Are the Standard Deviations Over 5 Random Initializations.

Task	F1 (%)	Single Best				Single Combine				Multi-Source				
		ASN	AdaGCN	UDAGCN	GRADE	ASN	AdaGCN	UDAGCN	GRADE	MIEM	SImpAI	MIEM (GAT)	SImpAI (GAT)	MHGCDA
$C_{II}D_{II} \rightarrow A_{II}$	Micro-F1	64.18 (1.44)	50.59 (1.75)	60.20 (0.28)	57.03 (0.74)	65.26 (2.04)	62.28 (0.35)	66.00 (0.30)	60.50 (0.74)	45.59 (0.20)	40.42 (0.82)	47.30 (2.25)	62.85 (0.49)	**69.33** (0.39)
	Macro-F1	63.54 (1.81)	49.21 (2.03)	59.60 (0.31)	54.73 (1.40)	64.37 (2.91)	62.60 (0.60)	66.46 (0.40)	60.05 (0.74)	44.30 (0.23)	37.64 (1.63)	45.74 (2.55)	62.59 (0.49)	**69.93** (0.47)
$A_{II}D_{II} \rightarrow C_{II}$	Micro-F1	65.27 (0.13)	55.38 (1.95)	60.87 (0.60)	58.38 (0.79)	67.11 (0.30)	57.77 (0.48)	64.40 (0.50)	62.11 (0.38)	52.32 (0.23)	42.90 (2.86)	54.05 (1.09)	64.31 (1.84)	**70.24** (0.35)
	Macro-F1	63.76 (0.24)	53.46 (1.97)	60.31 (0.58)	55.31 (1.38)	65.46 (0.24)	56.64 (0.13)	63.04 (0.56)	60.95 (0.33)	50.32 (0.41)	39.90 (3.07)	52.25 (1.16)	62.88 (0.96)	**69.14** (0.25)
$A_{II}C_{II} \rightarrow D_{II}$	Micro-F1	63.68 (0.93)	52.26 (1.03)	60.50 (0.26)	61.13 (1.22)	66.66 (0.50)	61.51 (1.54)	62.86 (0.43)	64.02 (0.70)	52.93 (0.56)	43.49 (3.19)	59.61 (2.69)	59.47 (1.00)	**72.05** (0.45)
	Macro-F1	60.03 (1.08)	49.85 (1.68)	57.25 (0.24)	57.15 (1.45)	63.74 (0.55)	58.83 (1.43)	60.80 (0.38)	61.67 (0.53)	50.49 (0.28)	40.62 (3.32)	57.13 (2.68)	56.46 (1.00)	**68.82** (0.38)
Average	Micro-F1	64.38 (0.83)	52.74 (1.58)	60.52 (0.38)	58.85 (0.92)	66.34 (0.95)	60.52 (0.79)	64.42 (0.41)	62.21 (0.61)	50.28 (0.33)	42.27 (2.29)	53.65 (2.01)	63.00 (1.09)	**70.54** (0.39)
	Macro-F1	62.44 (1.04)	50.84 (1.89)	59.05 (0.38)	55.73 (1.41)	64.52 (1.23)	59.36 (0.72)	63.43 (0.45)	60.89 (0.53)	48.37 (0.30)	39.39 (2.67)	51.70 (2.13)	61.40 (1.03)	**69.29** (0.36)

data. However, it has been widely acknowledged that taking advantage of the topological structure is indispensable for succeeding in CNNC (Dai, 2022; Shen, 2020; Shen, 2021).

The proposed MHGCDA significantly outperforms other baselines across six MSCNNC tasks. Firstly, MHGCDA surpasses all the Single Best baselines. For instance, on the $C_ID_I \rightarrow A_I$ task, MHGCDA achieves an improvement of 3.35 % in Micro-F1 and 4.08 % in Macro-F1 over the best Single Best baseline ASN. This significant improvement can be attributed to the ability of MHGCDA to integrate complementary knowledge from multiple source networks. Secondly, MHGCDA outperforms all the Single Combine baselines. It is worth noting that unlike the Single Combine baselines that treat all source networks equally, MHGCDA calculates various transferability weights for different source networks based on the information entropy of the corresponding network-specific node classifiers, thus enabling adaptive adjustment of the knowledge transferred from different source networks. This adaptive weighting strategy allows MHGCDA to prioritize the knowledge transferred from more relevant source networks, leading to a more effective knowledge transfer process. Thirdly, MHGCDA still significantly outperforms the state-of-the-art multi-source domain adaptation baselines integrated with the GNN encoder. This might be attributed to two novel designs in MHGCDA. On one hand, a hierarchical class-aware graph contrastive domain adaptation module is devised to mitigate the intra-class domain discrepancy and enlarge the inter-class domain discrepancy at both the node and prototype levels. On the other hand, a positive and negative pseudo-labeling module is devised to assign positive (negative) pseudo-labels to the target nodes that are highly likely (unlikely) to belong to a specific category. This allows more unlabeled target nodes to participate in the iterative self-training process, yielding more reliable pseudo-labels gradually to promote class-aware graph

Table 5
Micro-F1 and Macro-F1 of MHGCDA variants. The Highest Micro −F1 and Macro-F1 among all Model Variants are shown in Boldface.

Model Variants	Metrics (%)	Citation I			Citation II		
		$C_ID_I \rightarrow A_I$	$A_ID_I \rightarrow C_I$	$A_IC_I \rightarrow D_I$	$C_{II}D_{II} \rightarrow A_{II}$	$A_{II}D_{II} \rightarrow C_{II}$	$A_{II}C_{II} \rightarrow D_{II}$
MHGCDA	Micro-F1	**73.89**	**83.49**	**76.32**	**69.33**	**70.24**	**72.05**
	Macro-F1	**75.10**	**82.57**	**74.93**	**69.93**	**69.14**	**68.82**
w/o Node-level Graph Contrastive Domain Adaptation Loss $\mathcal{L}_{GCDA_N}$	Micro-F1	73.75	83.48	76.06	68.54	69.76	71.85
	Macro-F1	74.95	82.56	74.53	69.04	68.61	68.55
w/o Prototype-level Graph Contrastive Domain Adaptation Loss $\mathcal{L}_{GCDA_P}$	Micro-F1	66.44	81.64	73.23	67.03	70.17	60.57
	Macro-F1	58.35	80.16	67.39	67.29	69.06	54.91
w/o Node Classification Loss $\mathcal{L}_y$	Micro-F1	17.85	17.26	24.14	11.84	36.11	33.73
	Macro-F1	06.00	05.72	07.60	04.76	21.34	27.37
w/o Pseudo-labeling Loss $\mathcal{L}_P$	Micro-F1	67.65	72.11	74.87	68.61	69.70	69.81
	Macro-F1	74.95	65.01	69.55	69.16	68.54	66.96
w/o Learning Transferability Weight	Micro-F1	73.29	82.19	75.71	67.38	69.99	71.87
	Macro-F1	74.09	81.06	74.08	67.90	68.86	68.76

contrastive domain adaptation.

4.3. Ablation study (RQ2)

In this section, we conducted extensive ablation studies to validate the contribution of each component in the proposed MHGCDA, the results are presented in Table 5. Firstly, the absence of either node-level or prototype-level graph contrastive domain adaptation loss leads to a performance decline in MHGCDA. This demonstrates that the hierarchical graph contrastive domain adaptation, which alleviates intra-class domain discrepancy and amplifies inter-class domain discrepancy at both node and prototype levels, is essential for aligning the class-conditional distributions of node embeddings in the MSCNNC problem.

Secondly, the performance of the variant without the node classification loss is significantly inferior to MHGCDA. This is because the node classification loss is used to minimize the source risk and learn label-

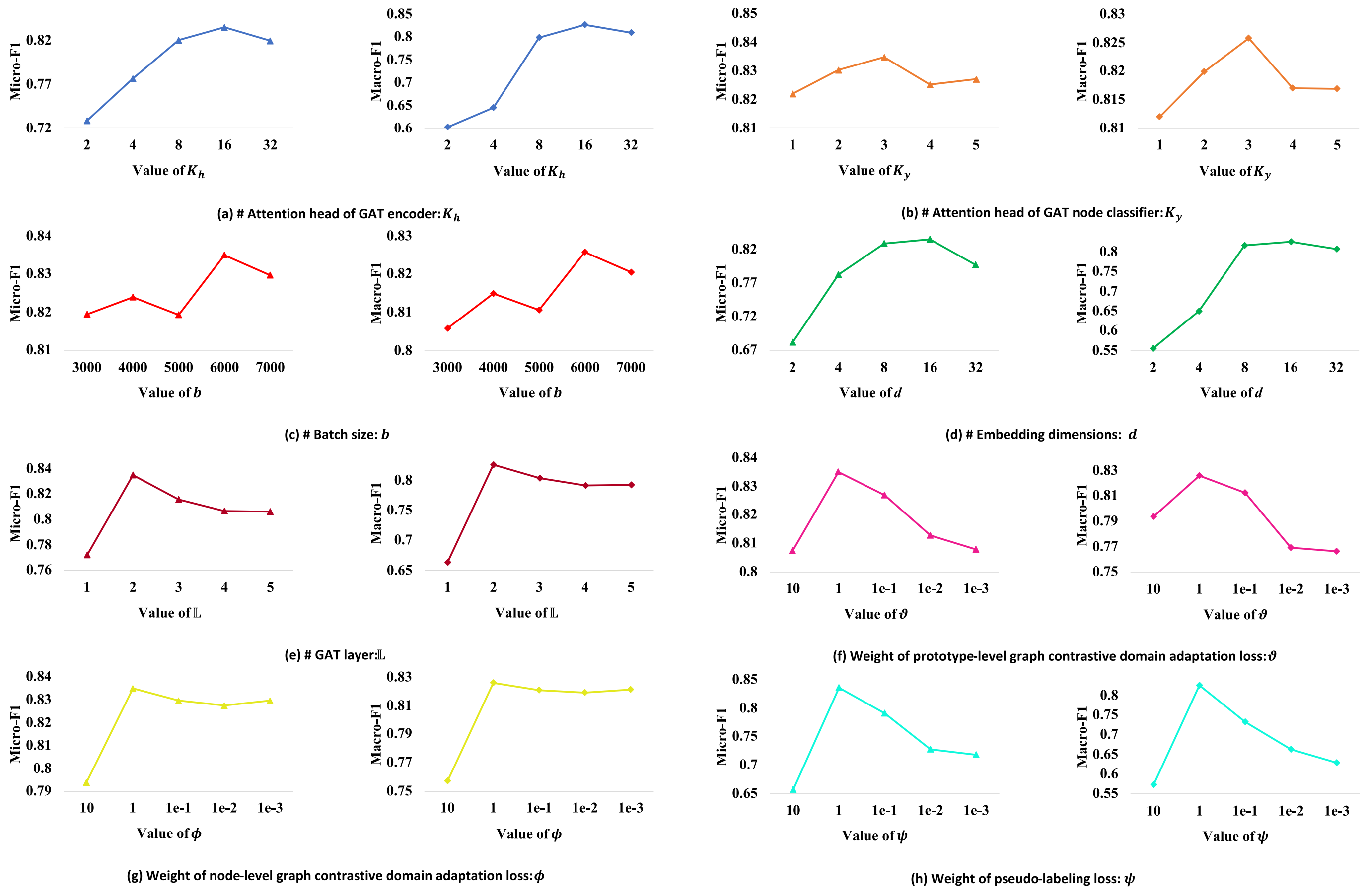

Fig. 5. Parameter sensitivity of MHGCDA on the representative task $A_ID_I \rightarrow C_I$.

discriminative node embeddings based on the observed labels from different source networks, which is indispensable for the MSCNNC problem.

Thirdly, the variant without the pseudo-labeling loss performs noticeably worse than MHGCDA. This is because positive and negative pseudo-labeling plays a crucial role in effectively utilizing the potential supervisory information from target nodes that are highly likely or highly unlikely to belong to specific categories. By iteratively re-training the model with the target nodes carrying positive or negative pseudo-labels, the proposed MHGCDA can gradually obtain more accurate pseudo-labels to guide the class-aware hierarchical graph contrastive domain adaptation.

Lastly, without learning transferability weights, MHGCDA would no longer consider the fitness between each source network and the target network. Instead, it would treat each source network equally. This contradicts the distribution-weighted assumption for multi-source domain adaptation (Mansour et al., 2008), as the fitness of individual source network to target network data may vary significantly at different stages of model training. Thus, treating each source network equally would result in worse performance in the MSCNNC task.

4.4. *Parameter sensitivity (RQ3)*

Next, we investigate the sensitivity of the hyper-parameters $K_h, K_y, b, d, \mathbb{L}, \phi, \vartheta, \psi$ on the performance of MHGCDA on the representative task $\mathrm{A_I D_I} \rightarrow \mathrm{C_I}$. The hyper-parameter K_h represents the number of attention heads in the GAT encoder. As shown in Fig. 5(a), $K_h = 16$ achieves the best scores, indicating that the multi-head attention mechanism indeed benefits the model in learning robust node embeddings. The hyper-parameter K_y represents the number of attention heads in the network-specific GAT node classifier. As shown in Fig. 5(b), $K_y = 3$ obtains the best scores, again confirming that the multi-head attention mechanism is advantageous for learning robust label prediction. The batch size is indicated by the hyper-parameter b. As shown in Fig. 5(c), MHGCDA obtains the best performance when b is 6000. The reason may be that a large batch size can obtain a more complete graph structure than a small batch size. The hyper-parameter d denotes the dimension of node embedding, as shown in Fig. 5(d), better scores are obtained when d is 16. The hyper-parameter $\mathbb{L}$ is the number of layers of the GAT encoder. From Fig. 5(e), we can observe that a deeper GAT usually yields better performance compared to a single-layer GAT. This suggests that aggregating information from higher-order neighbors is beneficial for node classification. While too many GAT layers may degrade the performance, this may be due to the over-smoothing problem. The hyper-parameters ϑ, ϕ, ψ are the weights of the prototype-level graph contrastive domain adaptation loss, the node-level graph contrastive domain adaptation loss, and the pseudo-labeling loss, respectively. As shown in Fig. 5(f), 5(g), and 5(h), MHGCDA achieves the highest Micro-F1 and Macro-F1 scores when these three trade-off parameters are set to 1.

5. Conclusions

We propose a novel MHGCDA model to address the challenging MSCNNC problem. Instead of treating each source network equally during alignment with the target network, the proposed MHGCDA model learns various transferability weights for different source networks based on the information entropy. A hierarchical class-aware graph contrastive domain adaptation module is designed in MHGCDA to alleviate domain shifts at both node and prototype levels. In the target network, MHGCDA assigns positive pseudo-labels to the nodes highly likely belonging to a specific class and negative pseudo-labels to the nodes highly unlikely belonging to a specific class. As the model trains, more pseudo-labeled target nodes gradually participate in iterative self-training, producing accurate pseudo-labels to promote class-aware domain alignment. Extensive experimental results on cross-network benchmark datasets demonstrate that the proposed MHGCDA exhibits superior performance in the MSCNNC task.

However, several factors might impact the validity of the proposed MHGCDA. First, the performance of MHGCDA depends heavily on the quality of the labeled data in the source networks. If low confidence or incorrectly labeled nodes are included in the source networks, then utilizing such labeled nodes for training may result in negative transfer. Thus, instead of utilizing all the labeled nodes in the source networks for training, it is promising to make more exploration to select only high confidence labeled nodes from the source networks for training. Additionally, in the current MHGCDA model, the thresholds for positive and negative pseudo-labeling are manually selected, which might be sensitive to different datasets. Future work will focus on designing an adaptive thresholding mechanism to avoid manual selection.

CRediT authorship contribution statement

Chuanyun Lin: Methodology, Software, Writing – original draft. **Xi Zhou:** Investigation, Validation, Writing – review & editing, Funding acquisition. **Xiao Shen:** Conceptualization, Writing – review & editing, Supervision, Funding acquisition.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: [Xiao Shen reports financial support was provided by Hainan Provincial Natural Science Foundation of China, National Natural Science Foundation of China, the Innovation Platform for “New Star of South China Sea” of Hainan Province, the Specific Research Fund of The Innovation Platform for Academicians of Hainan Province, and the Research Start-up Fund of Hainan University].

Acknowledgments

This work was supported in part by Hainan Provincial Natural Science Foundation of China (No. 625RC714), National Natural Science Foundation of China (No. 62362020, No. 62102124), the Innovation Platform for “New Star of South China Sea” of Hainan Province (No. NHXXRCXM202306), the Specific Research Fund of The Innovation Platform for Academicians of Hainan Province (No. YSPTZX202410), and the Research Start-up Fund of Hainan University (No. KYQD(ZR)-22016).

Data availability

Data will be made available on request.

References

Cai, Z., Zhang, D., Zhang, T., Hu, C., & Jing, X.-Y. (2023). Single-/multi-source domain adaptation via domain separation: A simple but effective method. *Pattern Recognition Letters, 174*, 124–129.

Cai, Z., Zhang, T., Ma, F., & Jing, X.-Y. (2022). Dual contrastive universal adaptation network for multi-source visual recognition. *Knowledge-Based Systems, 254*, Article 109632.

Chen, M., Weinberger, K. Q., & Blitzer, J. (2011). Co-training for domain adaptation. Advances in neural information processing systems (pp. 2456-2464).

Chen, T., Kornblith, S., Norouzi, M., & Hinton, G. (2020). A simple framework for contrastive learning of visual representations. International Conference on Machine Learning (pp. 1597-1607).

Dai, Q., Wu, X.-M., Xiao, J., Shen, X., & Wang, D. (2022). Graph transfer learning via adversarial domain adaptation with graph convolution. *IEEE Transactions on Knowledge Data Engineering, 35*(5), 4908–4922.

Dong, Y., Jiang, H., Wang, X., Mu, M., & Jiang, W. (2024). An interpretable multiscale lifting wavelet contrast network for planetary gearbox fault diagnosis with small samples. *Reliability Engineering System Safety, 251*, Article 110404.

Feng, W., Zhang, J., Dong, Y., Han, Y., Luan, H., Xu, Q., Yang, Q., Kharlamov, E., & Tang, J. (2020). Graph random neural networks for semi-supervised learning on graphs. Advances in neural information processing systems (pp. 22092-22103).

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., March, M., & Lempitsky, V. (2016). Domain-adversarial training of neural networks. *Journal of Machine Learning Research, 17*(59), 1–35.

Gasteiger, J., Bojchevski, A., & Günnemann, S. (2018). Predict then propagate: Graph neural networks meet personalized pagerank. International Conference on Learning Representations.

Gopalakrishnan, K., Khaitan, S. K., Choudhary, A., & Agrawal, A. (2017). Deep convolutional neural networks with transfer learning for computer vision-based data-driven pavement distress detection. *Construction building materials, 157*, 322–330.

Hamilton, W. L., Ying, R., & Leskovec, J. (2017). Inductive representation learning on large graphs. Advances in neural information processing systems (pp. 1025-1035).

He, J., Jia, X., Chen, S., & Liu, J. (2021). Multi-source domain adaptation with collaborative learning for semantic segmentation. Proceedings of The IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 11008-11017).

Hu, W., Liu, B., Gomes, J., Zitnik, M., Liang, P., Pande, V., & Leskovec, J. (2019). Strategies for pre-training graph neural networks. *International Conference on Learning Representations*.

Kang, G., Jiang, L., Wei, Y., Yang, Y., & Hauptmann, A. (2020). Contrastive adaptation network for single-and multi-source domain adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 44*(4), 1793–1804.

Kim, Y., Yim, J., Yun, J., & Kim, J. (2019). Nlnl: Negative learning for noisy labels. Proceedings of The IEEE/CVF International Conference on Computer Vision (pp. 101-110).

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

Liu, M., Zhang, Z., Ma, N., Gu, M., Wang, H., Zhou, S., & Bu, J. (2024). Structure enhanced prototypical alignment for unsupervised cross-domain node classification. *Neural Networks, 177*, Article 106396.

Mansour, Y., Mohri, M., & Rostamizadeh, A. (2008). Domain adaptation with multiple sources. *Advances in neural information processing systems*.

Mansour, Y., Mohri, M., & Rostamizadeh, A. (2012). Multiple source adaptation and the rényi divergence. Proceedings of The Twenty-Fifth Conference on Uncertainty in Artificial Intelligence (pp. 367-374).

Nguyen, V.-A., Nguyen, T., Le, T., Tran, Q. H., & Phung, D. (2021). Stem: An approach to multi-source domain adaptation with guarantees. Proceedings of The IEEE/CVF International Conference on Computer Vision (pp. 9352-9363).

Peng, X., Bai, Q., Xia, X., Huang, Z., Saenko, K., & Wang, B. (2019). Moment matching for multi-source domain adaptation. Proceedings of The IEEE/CVF International Conference on Computer Vision (pp. 1406-1415).

Qiao, Z., Luo, X., Xiao, M., Dong, H., Zhou, Y., & Xiong, H. (2023). Semi-supervised domain adaptation in graph transfer learning. Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence (pp. 2279-2287).

Rizve, M. N., Duarte, K., Rawat, Y. S., & Shah, M. (2021). In defense of pseudo-labeling: An uncertainty-aware pseudo-label selection framework for semi-supervised learning. International Conference on Learning Representations.

Shao, M., Xue, P., Zhou, X., & Shen, X. (2024). Contrastive domain-adaptive graph selective self-training network for cross-network edge classification. *Pattern Recognition, 152*, Article 110448.

Shen, X., Chen, Z., Pan, S., Zhou, S., Yang, L. T., & Zhou, X. (2025). Open-Set Cross-Network Node Classification via Unknown-Excluded Adversarial Graph Domain Alignment. Proceedings of the AAAI Conference on Artificial Intelligence (pp. 20398–20408).

Shen, X., Choi, K.-S., & Zhou, X. (2024a). Dual separated attention-based graph neural network. *Neurocomputing, 599*, Article 128106.

Shen, X., Dai, Q., Chung, F.-l., Lu, W., & Choi, K.-S. (2020). Adversarial deep network embedding for cross-network node classification. Proceedings of the AAAI Conference on Artificial Intelligence (pp. 2991-2999).

Shen, X., Dai, Q., Mao, S., Chung, F.-L., & Choi, K.-S. (2021). Network together: Node classification via cross-network deep network embedding. *IEEE Transactions on Neural Networks Learning Systems, 32*(5), 1935–1948.

Shen, X., Mao, S., & Chung, F.-L. (2019). Cross-network learning with fuzzy labels for seed selection and graph sparsification in influence maximization. *IEEE Transactions on Fuzzy Systems, 28*(9), 2195–2208.

Shen, X., Pan, S., Choi, K.-S., & Zhou, X. (2023a). Domain-adaptive message passing graph neural network. *Neural Networks, 164*, 439–454.

Shen, X., Shao, M., Pan, S., Yang, L. T., & Zhou, X. (2024b). Domain-adaptive graph attention-supervised network for cross-network edge classification. *IEEE Transactions on Neural Networks and Learning Systems, 35*(12), 17842–17855.

Shen, X., Sun, D., Pan, S., Zhou, X., & Yang, L. T. (2023b). Neighbor contrastive learning on learnable graph augmentation. Proceedings of the AAAI Conference on Artificial Intelligence (pp. 9782-9791).

Singh, A. (2021). Clda: Contrastive learning for semi-supervised domain adaptation. Advances in neural information processing systems (pp. 5089-5101).

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., & Bengio, Y. (2017). Graph attention networks. *International Conference on Learning.* Representations.

Venkat, N., Kundu, J. N., Singh, D., & Revanur, A. (2020). Your classifier can secretly suffice multi-source domain adaptation. Advances in neural information processing systems (pp. 4647-4659).

Wang, R., Wu, Z., Weng, Z., Chen, J., Qi, G.-J., & Jiang, Y.-G. (2022a). Cross-domain contrastive learning for unsupervised domain adaptation. *IEEE Transactions on Multimedia*.

Wang, X., Jiang, H., Mu, M., & Dong, Y. (2025). A dynamic collaborative adversarial domain adaptation network for unsupervised rotating machinery fault diagnosis. *Reliability Engineering System Safety, 255*, Article 110662.

Wang, Z., Zhou, C., Du, B., & He, F. (2022b). Self-paced Supervision for Multi-Source Domain Adaptation. Proceedings of The Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22.

Wen, L., Chen, S., Xie, M., Liu, C., & Zheng, L. (2023). Training Multi-source Domain Adaptation Network by Mutual Information Estimation and Minimization. *Neural Networks, 171*, 353–361.

Wu, H., Tian, L., Wu, Y., Zhang, J., Ng, M. K., & Long, J. (2024). Transferable graph auto-encoders for cross-network node classification. *Pattern Recognition, 150*, Article 110334.

Wu, J., He, J., & Ainsworth, E. (2023). Non-iid transfer learning on graphs. Proceedings of The AAAI Conference on Artificial Intelligence (pp. 10342-10350).

Wu, M., Pan, S., Zhou, C., Chang, X., & Zhu, X. (2020a). Unsupervised domain adaptive graph convolutional networks. Proceedings of The Web Conference 2020 (pp. 1457-1467).

Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., & Philip, S. Y. (2020b). A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks Learning Systems, 32*(1), 4–24.

Xu, K., Hu, W., Leskovec, J., & Jegelka, S. (2018a). How powerful are graph neural networks? *International Conference on Learning.* Representations.

Xu, R., Chen, Z., Zuo, W., Yan, J., & Lin, L. (2018b). Deep cocktail network: Multi-source unsupervised domain adaptation with category shift. Proceedings of The IEEE Conference on Computer Vision and Pattern Recognition (pp. 3964-3973).

Yang, H., He, H., Zhang, W., & Bai, Y. (2022a). MTGK: Multi-source cross-network node classification via transferable graph knowledge. *Information Sciences, 589*, 395–415.

Yang, S., Cai, B., Cai, T., Song, X., Jiang, J., Li, B., & Li, J. (2022b). Robust cross-network node classification via constrained graph mutual information. *Knowledge-Based Systems, 257*, Article 109852.

Zhang, X., Du, Y., Xie, R., & Wang, C. (2021). Adversarial separation network for cross-network node classification. Proceedings of The 30th ACM International Conference on Information & Knowledge Management (pp. 2618-2626).

Zhang, Y., Song, G., Du, L., Yang, S., & Jin, Y. (2019). Dane: Domain adaptive network embedding. Proceedings of The International Joint Conference on Artificial Intelligence (pp. 4362-4368).

Zhou, X., Yang, J., Luo, Y., & Shen, X. (2024). HNCGAT: A method for predicting plant metabolite–protein interaction using heterogeneous neighbor contrastive graph attention network. *Briefings in bioinformatics, 25*(5), Article bbae397.