Attraction and Repulsion: Unsupervised Domain Adaptive Graph Contrastive Learning Network

Man Wu, Shirui Pan, *Member, IEEE*, and Xingquan Zhu, *Senior Member, IEEE*

***Abstract*—Graph convolutional networks (GCNs) are important techniques for analytics tasks related to graph data. To date, most GCNs are designed for a single graph domain. They are incapable of transferring knowledge from/to different domains (graphs), due to the limitation in graph representation learning and domain adaptation across graph domains. This paper proposes a novel Graph Contrastive Learning Network (GCLN) for unsupervised domain adaptive graph learning. The key innovation is to enforce attraction and repulsion forces within each single graph domain, and across two graph domains. Within each graph, an attraction force encourages local patch node features to be similar to global representation of the entire graph, whereas a repulsion force will repel node features so they can separate network from its permutations (*i.e.* domain-specific graph contrastive learning). Across two graph domains, an attraction force encourages node features from two domains to be largely consistent, whereas a repulsion force ensures features are discriminative to differentiate graph domains (*i.e.* cross-domain graph contrastive learning). The within- and cross-domain graph contrastive learning is carried out by optimizing an objective function, which combines source classifier and target classifier loss, domain-specific contrastive loss, and cross-domain contrastive loss. As a result, feature learning from graphs is facilitated using knowledge transferred between graphs. Experiments on real-world datasets demonstrate that GCLN outperforms state-of-the-art graph neural network algorithms.**

***Index Terms*—Domain adaptive learning, graph contrastive learning, graph neural network, node classification.**

I. Introduction

DOMAIN adaptation is a type of transfer learning method that minimizes domain discrepancy to support knowledge transferring from source domains, which often have sufficient labeled information, to a target domain with unlabeled data. Because many domains share inherent correlation, domain adaptation learning has attracted extensive interests in Computer Vision (CV) [1] and Natural Language Processing (NLP) [2]. Nevertheless, applying domain adaptation to graph analytics,[1] like node classification across networks, has not been sufficiently investigated. When training node classification models, a frequently encountered dilemma is that the network to be classified (*i.e., target network*) may have little or no label information at all, whereas networks from similar or relevant domains (*i.e., source network*) may have sufficient label information. The *goal* of unsupervised graph domain adaptation learning [3] is to use rich labeled information from the source network to predict nodes in a target network into distinct classes, conditioned by that source network nodes are fully labeled but target network nodes have no label.

For years, domain adaptation research has been primarily concentrated on the CV and NLP, mainly because that vision and natural language are the two areas featured with many cross domain needs and applications. For networked data, existing solutions cannot be utilized to domain adaptive network classification, mainly because of two reasons: Firstly, these methods are developed for CV and NLP tasks, where samples (*e.g.* images) are independent and identically distributed (*i.i.d*), meaning that there is no requirement for model rotational invariance. Networks, however, contain nodes (*i.e.,* samples) and edges representing their dependency relationships. The unique data setting requires models to be rotation invariant because two graphs might have identical topology but are presented in different forms in adjacency matrix or node space (*i.e.* graph isomorphism), due to rotations or other variance. Secondly, most current domain adaptation models learn discriminative representation in a supervised way, with the loss value simply being related to the absolute position of each sample in the feature space. However, graph representation learning for node classification usually learns multi-purpose representations in an unsupervised way, which increases the optimization difficulty.

A few studies emerged recently to transfer knowledge between networks via domain adaptation [4]. By minimizing a maximum mean discrepancy (MMD) loss function, CDNE [4] learns node embeddings which are transferable for cross network classification. Nevertheless, it cannot exploit both network architecture and node content jointly, and thus it has limited model capabilities. In order to tackle the above limitation, many approaches are proposed to leverage network topology information for cross-network node classification. For example, AdaGCN [5] employs graph convolutional networks and adversarial learning to obtain node embeddings. UDA-GCN [3] combines domain classifiers and source/target classifiers for network representation learning on the target network to leverage knowledge from source networks.

Manuscript received September 22, 2021; revised January 24, 2022; accepted February 12, 2022. The work of Shirui Pan was supported by ARC Future Fellowship under Grant FT210100097. This work was supported by the US National Science Foundation under Grants IIS-1763452, CNS-1828181, and IIS-2027339. *(Corresponding author: Xingquan Zhu.)*

Man Wu and Xingquan Zhu are with the Department of Electrical Engineering and Computer Science, Florida Atlantic University, Boca Raton, FL 33431 USA (e-mail: mwu2019@fau.edu; xzhu3@fau.edu).

Shirui Pan is with Monash University, Melbourne, VIC 3800, Australia (e-mail: Shirui.Pan@monash.edu).

Digital Object Identifier 10.1109/TETCI.2022.3156044

[1]Networks and graphs are interchangeable terms in the paper when referring to graph structure data.

In summary, existing works mainly seek to improve cross-domain node classification by combining graph convolutional networks with an adversarial learning scheme, but they still fall short in addressing three levels of challenges below for domain adaptive graph learning:

Local vs. Global Consistency Within Single Domain: In a single graph, graph convolutional networks (GCNs) [6] utilize direct neighbour nodes (the local consistency) for knowledge embeddings, but largely overlook global consistency in the graph. However, the global consistency relationship is very significant. Specifically, an individual in a social network may be a member of some communities and may be affected by neighborhoods at different scales, ranging from close neighbors (*e.g.* families, friends) with local consistency relationship, to far neighbors (*e.g.* society, community) with global consistency relationship. Therefore, to learn a comprehensive node feature representation in the network, both local and global consistency relationships should be leveraged.

Intra- vs. Inter-Domain Consistency: When multiple graph domains are available, existing GNN-based methods cannot leverage information about high-level global factors of graphs effectively for representation learning of source and target domains. Specifically, these methods focus on learning node embedding features by preserving proximity and/or local information in the graphs. Recently, in order to generate more expressive node representations, the global properties of graphs are considered by some research work. For example, Deep Graph Infomax (DGI) [7] maximizes the mutual information between the global graph representation and the local patch representation to obtain better graph representations, which is the first work in this field and has performed well in many downstream tasks. However, most of these methods are used for single-domain graphs, and they cannot be applied to cross-domain graphs which require cross-domain node representation learning. Therefore, learning high-quality node representations needs to take intra- and inter-domain consistency into consideration so high-level global factors in cross-domain graphs can be leveraged for learning.

Domain Discrepancy and Adaptation: Existing domain adaptation methods for graph data typically use domain labels as supervised signals to train a classifier to model global distributions of source and target domains. Specifically, the domain classifier is trained with a Gradient Reversal Layer [8] to distinguish whether a node is from the source domain or the target domain. During this time, a source classifier is trained with source domain data for source domain classification. However, the target domain and its semantic information are largely ignored. Thus, to collaboratively learn domain-invariant semantic representations, we should consider not only the source domain information, but also the gap between domains as well as target domain information jointly in cross-domain learning.

In this paper, we propose a Graph Contrastive Learning Network (GCLN) to address the above challenges for unsupervised domain adaptive graph learning. Our main theme is to advocate a graph contrastive learning principle, which imposes an attraction force and a repulsion force to networks within a single domain and across different domains. In order to tackle local *vs.* global consistency within a single domain, we employ multiple networks, reflecting local and global consistency, to learn network embeddings, where an attraction force intends to make node features to be similar, and a repulsion force will repel node features so they can separate network from its permutations (*i.e.* domain-specific graph contrastive learning). In order to tackle inter-domain consistency and domain adaptation, cross-domain graph contrastive learning is introduced to use an attraction force to make node features from different networks to be similar, but a repulsion force will enforce node features to be discriminative for graph domains to be separable. By using a unified objective function to combine multiple losses, GCLN simultaneously enforces within- and cross-domain graph contrastive learning to learn network features for unsupervised domain adaptive graph node classification.

The main contributions of the paper, compared to existing research in the field, can be summarized as follows:

- A new graph contrastive learning framework is proposed to combine attraction and repulsion forces to ensure intra- and inter-domain consistency within a single network and across multiple networks. The new contrastive learning framework opens opportunities for knowledge transfering between networks from different domains.
- Our proposed model utilizes graph contrastive learning to learn node embedding features, by leveraging local and global information of every graph, as well as domain discrepancy across different graph domains. As a result, domain discrepancy for unsupervised cross-domain node classification is significantly reduced by using domain-invariant and semantic representations.
- Experiments are conducted on real-world datasets, and the results show that the proposed method outperforms all baseline approaches.

II. Related Work

In order to carry out domain adaptive learning for graphs, we employ graph networks and contrastive learning principles in our design. We now briefly review the closely related works in this section.

A. Graph Neural Networks

Graph neural networks (GNNs) [9], [10] are emerging methods for learning representations for nodes in graph structured data. By employing GNNs to encode nodes in a graph into latent representations, many downstream analyticts tasks can be easily performed, including node classification, link prediction, and clustering. GCN [6] is a seminar work which employs two graph convolutional layers to perform node classification in a semi-supervised setting in a graph, demonstrating high potential of graph convolutional networks (GCNs). GAT [11] is further proposed to improve the message passing of GCN, by automatically learning the weights of neighbors with an attention mechanism. GraphSage [12] is proposed to generalize the aggregation function of GCN, showing improved performance. While most GNNs embed the nodes into the Euclidean space which may not well capture the scale-free or hierarchical structure of networks. Recently HGCN [13] is proposed to overcome this problem by embedding nodes into the hyperbolic space.

Although GNNs achieve impressive performance in many tasks, many of them are primarily concerned with representation learning of nodes in a single network. Thus, embedding space drift [14] and distribution discrepancy [15] might occur when transferring the learned models across networks to deal with similar questions.

B. Contrastive Learning

Contrastive learning (CL) is a type of self-supervised learning technique [16] which aims to construct supervision information from the data, without using manual labels [17]. Typically, CL constructs a pair of instances and feeds them into a comparison module to learn the representation of data. Among existing CL techniques, mutual information estimation and maximization is a popular method [18].

Some recent works utilize the infomax principle [19] for unsupervised representation leaning in different domains [20]–[23]. For example, the Mutual Information Neural Estimation (MINE) method uses a statistical network as the classifier for distinguishing samples from the joint distribution and the product of marginals of two random variables, thus facilitating the process of learning mutual information estimation in deep networks. DIM [21] employs several infomax objectives to introduce structural information into the input patches. Deep Graph Infomax (DGI) [22] maximizes the mutual information between local patches of a graph and the high-level global information, which is the first work to apply infomax to graph structured data. Recently the method MERIT [24] was proposed to exploit multi-scale graph information from both local and global perspectives, then the MI is maximized between node representations from different views and networks. However, most of these methods are used for the domain-single graphs, and cannot consider the graphs of different domains. In order to go beyond the single domain, we propose a novel Graph Contrastive Learning Network (GCLN) for unsupervised domain adaption to learn node representations of different domains through preserving global properties of graphs, where a novel domain-specific graph contrastive learning module is designed for each domain-specific graph, and a novel cross-domain graph contrastive learning module is designed for different domains.

C. Cross-Domain Classification

As a subtopic of transfer learning, domain adaptation seeks to learn models that can transfer knowledge across domains that share the same label space [25]. In recent years, deep feature representation-based domain adaptation methods have gained lots of interest. There are three categories: reconstruction-based methods [26], discrepancy-based approaches [27], and adversarial-based methods [8][15]. To reduce the domain discrepancy [28] between domains, many methods employ an adversarial function as their learning objective. Specifically, DANN [8] is proposed to obtain domain invariant features by employing a gradient reversal layer in which gradients are back-propagated from the domain classifier in a minimax game between the domain classifier and the feature extractor. Zhang *et al.* [29] propose a novel model to enforce the distributional and structural similarities during the adaptation. Pilanci *et al.* [30] use graph matching as a domain discrepancy metric. Das *et al.* [31] treat the source and target domains as hyper-graphs and carry out a class-regularized hyper-graph matching using first-, second- and third-order similarities between the graphs. In [32], Wang *et al.* propose a novel cross-domain contrastive learning for unsupervised domain adaptation, by using contrastive self-supervised learning to align features to release the domain discrepancy between training and testing sets.

Domain adaptation has recently been used to graph-structured data [4], [5], [33]. In [4], CDNE is proposed to minimize the maximum mean discrepancy (MMD) loss to learn transferable node representations for cross network learning. However, network structures and node attributes cannot be modeled jointly, which may limit its effectiveness. Some studies [5], [33] learn the domain invariant node representation with graph convolutional networks and adversarial learning methods, for the task of cross-network node classification. These methods gain promising results. In [34], the deep multi-graph embedding (DMGE) [34] is proposed which learns cross-domain embedding by extracting a multiple graph from users' behaviors across different domains. A multi-graph neural network, which performs in an unsupervised manner, is further employed for the cross-domain representation learning. While the recent UDA-GCN [3] approach exploits local and global information for domain adaptation across networks, it does not fully exploit the intra- and inter-domain consistency. In contrast, we propose an end-to-end Unsupervised Domain Adaptive Graph Contrastive Learning Networks (GCLN) by jointly modeling intra- and inter-domain consistency, to improve the effectiveness of cross-domain node classification.

III. Problem Definition and Overall Framework

A. Problem Statement

1) Node Classification: In this paper, we aim to conduct the node classification on graphs. $G = (V, E, X, Y)$ represents a graph, where $V = \{v_i\}_{i=1,\ldots,N}$ is the set of vertices representing the nodes in the graph and $e_{i,j} = (v_i, v_j) \in E$ is an edge that represents relationship between node v_i and v_j. An adjacency matrix A can represent the topological structure of a graph G, with $A_{i,j} = 1$ if $(v_i, v_j) \in E$ or $A_{i,j} = 0$ otherwise. The content features associated with each node v_i are represented by $x_i \in X$. The label matrix of G is $Y \in \mathbb{R}^{N \times C}$, where N indicates the number of nodes in the graph G and C represents the number of categories of nodes in the graph. $Y^l_{(i)} = 1$ if a node $v_i \in V$ is linked to label l; else, $Y^l_{(i)} = 0$.

2) Source Domain Network: Given $G^s = (V^s, E^s, X^s, Y^s)$, G^s is a fully labeled source network, and V^s, E^s denote the nodes and edges of G^s, respectively. The label matrix of G^s is $Y^s \in \mathbb{R}^{N^s \times C}$, in which $N^s = |V^s|$ indicates the number of nodes in the graph G^s and C is the number of classes of nodes in the graph.

3) Target Domain Network: Furthermore, $G^t = (V^t, E^t, X^t)$ denotes the target network, and G^t is a completely unlabeled network. Here, V^t, E^t denote the nodes and edges of G^t, respectively.

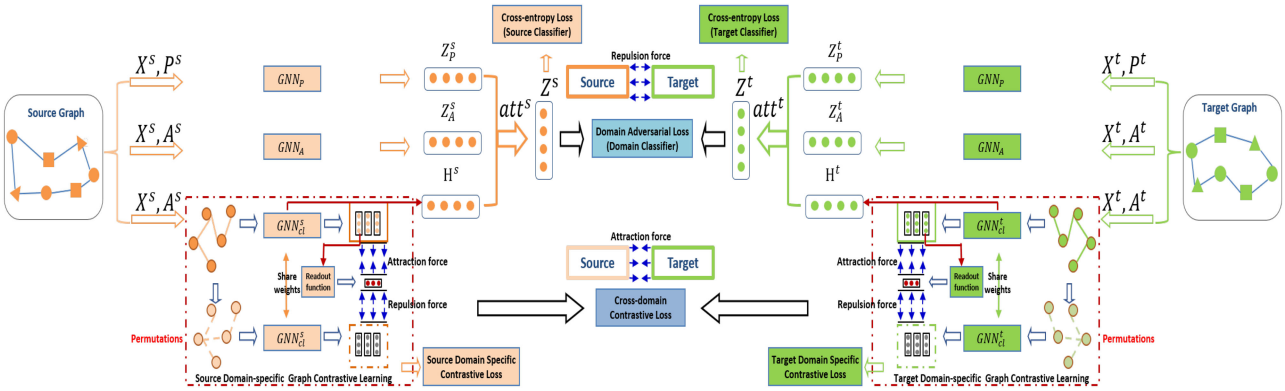


Fig. 1. Overall architecture of the proposed Graph Contrastive Learning Network (GCLN) for unsupervised domain adaptive graph node classification. The input consists of graphs from source domain (left) and target domain (right). GCLN consists of three main components: (1) Given adjacency matrices of source network A^s and target network A^t, GCLN first creates PPMI matrices P^s and P^t to capture global consistency for representation learning. (2) For each individual domain, *domain-specific graph contrastive learning* (detailed in Fig. 2) is carried out by encouraging local path features to be similar to global representation (attraction force), and maximizing the discrepancy between permuted graph and global representation of the entire graph (repulsion force). (3) Across source and target domains, *cross-domain graph contrastive learning* (detailed in Fig. 3) encourages node embedding features from different domains to be largely consistent (attraction force), and domain adversarial loss (domain classifier) intends to make sure that node representations can separate domains (repulsion force). Algorithm detailed in Section IV.

Unsupervised Domain Adaptation for Node Classification: Given a target network G^t in which nodes are all unlabeled, and a source network G^s where all nodes are labeled, the goal of unsupervised domain adaptation for node classification is to learn a model f from both G^t and G^s to classify the nodes in G^s with a maximum accuracy.

B. Overall Framework

The crux of solving unsupervised domain adaption for networked data is in determining what should be shared and adapted across domains. Feature representation is an apt metaphor for domain comparison and adaption, therefore we introduce a Graph Contrastive Learning Network (GCLN) to release the domain gap and learn a low-dimensional common feature representations across domains, aiming to develop a classifier for node classification using cross-domain graphs. The following three components compose our framework, as indicated in Fig. 1:

- *Node Representation Learning:* We utilize two GCNs to capture local and global consistency within each graph for learning better feature representations.
- *Domain-specific and Cross-domain Graph Contrastive Learning:* We design a graph contrastive learning network to learn node representations for each domain-specific graph. Specifically, to generate more expressive node embeddings, the average mutual information (MI) between graph-level representations and local patch-level representations is maximized (attraction force), while the discrepancy between permuted graph and global representation of the whole graph is maximized (repulsion force). In addition, a cross-domain objective function is developed to incorporate the node representations across domains to learn high-quality embeddings.
- *Domain Adaptation Learning for Node Classification:* Three classifiers are trained using domain adaptive learning: 1) a source classifier is trained to minimize the classification loss on the nodes from the source domain; 2) a domain classifier is learned with a domain adversarial loss to distinguish samples from source or target domains; 3) a target classifier is trained by minimizing an entropy loss. This can better capture the semantic information of the target domain and enhance the performance of the model.

IV. Methodology

A. Node Representation Learning

The node representation learning module consists of a dual graph neural "networks" utilizing local and global information of the graph to learn semantic representations of nodes.

1) Local Consistency Network (GNN_A): We formulate the GNN_A as a type of feed-forward neural network to learn local consistency by directly using the GCN method proposed by [6]. Given the characteristic matrix X and adjacency matrix A of the graph, the output of the i-th hidden layer of network Z is denoted as follows:

$$GNN_A^{(i)}(X) = Z^{(i)} = \sigma\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{\frac{1}{2}}Z^{(i-1)}W^{(i)}\right), \quad (1)$$

where $\tilde{A} = A + I_n$ is the adjacent matrix with self-loops ($I_n \in \mathbb{R}^{n\times n}$ is the identity matrix), and $\tilde{D}_{i,i} = \sum_j \tilde{A}_{i,j}$. Accordingly, $\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{\frac{1}{2}}$ is the normalized adjacency matrix. $Z^{(i-1)}$ is the output of the $(i-1)$-th layer, and $Z^{(0)} = X$. $W^{(i)}$ are weight values of the neural network, and $\sigma(\cdot)$ is an activation function.

2) Global Consistency Network (GNN_P): In addition to the local network, we train another graph neural network GNN_P to capture the global consistency. Specifically, we construct a Point-wise mutual information matrix (PPMI) matrix $P \in \mathbb{R}^{N\times N}$, following previous work [3], [35]. The PPMI matrix is calculated with the random walks which model semantic similarities between nodes in a flexible way, thus it is able to traverse neighbors within long distance and capture global consistency. This global consistency network is given by (2) where P is the

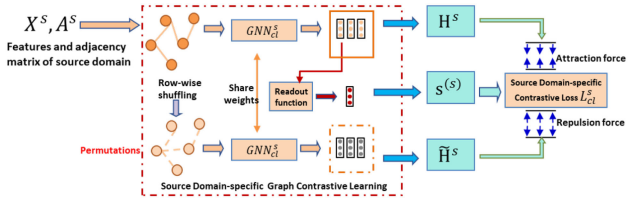


Fig. 2. Domain-specific graph contrastive learning module (source and target domains use same architecture). The attraction force encourages local path node features to be similar to global representation (by maximizing their mutual information). The repulsion force ensures node features can separate networks from its permutations (detailed in Section IV-B).

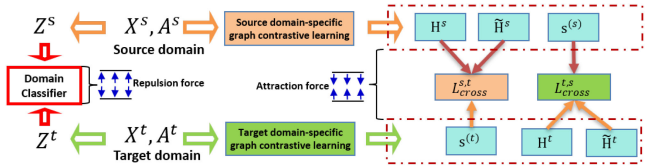


Fig. 3. Cross-domain graph contrastive learning module. The attraction force encourages node features from two domains to be largely consistent, whereas the repulsion force ensures the features are discriminative to differentiate graph domains (detailed in Section IV-B).

PPMI matrix and $D_{i,i} = \sum_j P_{i,j}$ for normalization.

$$GNN_P^{(i)}(X) = Z^{(i)} = \sigma\left(D^{-\frac{1}{2}} P D^{-\frac{1}{2}} Z^{(i-1)} W^{(i)}\right). \quad (2)$$

B. Domain-Specific & Cross-Domain Graph Contrastive Learning

Despite the fact that we can get node embeddings Z from node embedding module by using graph neural networks, this module often falls short to capture high-level global graph information that might be valuable for graph representation learning. As a result, we propose the graph contrastive learning network to exploit intra- and inter-domain consistency for domain adaptation.

1) Domain-Specific Graph Contrastive Learning: For domain-specific graph contrastive learning, our method intends to learn a low dimensional representation for each node by retaining the representations of both the global properties and the local graph patches.

Specifically, for source graph G^s and target graph G^t, we introduce a domain-specific graph neural encoder network GNN_{cl}, to generate domain-specific node embedding matrix H for their nodes. Taking source graph G^s in Fig. 2 as an example, given its corresponding feature matrix X^s and adjacency matrix A^s, the domain-specific encoder GNN_{cl}^s is designed as a single–layered GCN to learn the matrix H^s:

$$H^s = GNN_{cl}^s(X^s, A^s|W^s) = \sigma(\hat{D}^{-\frac{1}{2}} \hat{A}^s \hat{D}_t^{-\frac{1}{2}} X^s W^s), \quad (3)$$

where $\hat{A}^s = A^s + wI_n, I_n \in \mathbb{R}^{n\times n}$ is the identity matrix, n represents the number of nodes in the graph G^s, $D_i = \sum_j \hat{A}_{ij}^s$, W^s is a trainable weight matrix of the domain-specific graph encoder GNN_{cl}^s, and σ is the ReLU nonlinearity. Unlike conventional GCNs, we use a weight $w \in R$ to control the weight of the self-connections. A larger w indicates that the node itself is more significant in generating its embeddings, which reduces its neighboring nodes' importance.

In order to generate global graph property summary, we employ a non-linear $Readout()$ function in (4) to compute a representation $s^{(s)}$ summarizing global content of graph G^s.

$$s^{(s)} = Readout(H^s) = \sigma\left(\frac{1}{n}\sum_i^n h_i^s\right). \quad (4)$$

In (4), σ is the logistic Sigmoid function, and h_i^s denotes the i-th row vector of the matrix H^s.

Then, given the source domain-specific node embedding matrix H^s, and its graph-level summary representation $s^{(s)}$, we define a domain-specific contrastive loss as follows:

$$\mathcal{L}_{cl}^s = \frac{1}{2n^s}\left(\underbrace{\sum_i^{n^s} \log \mathcal{D}(h_i^s, s^{(s)})}_{\text{Attraction force}} + \underbrace{\sum_j^{n^s} \log\left(1 - \mathcal{D}(\widetilde{h}_j^s, s^{(s)})\right)}_{\text{Repulsion force}}\right), \quad (5)$$

where n^s denotes the number of nodes of source graph G^s, and $\mathcal{D}$ is a discriminator which scores patch summary representation pairs, *i.e.*, $(h_i^s, s^{(s)})$. Essentially, this loss function enables an attraction force to maximize mutual information of a local patch representation and the global graph representation, as well as a repulsion force to maximize the discrepancy between permuted graph patch representation and global graph representation.

Here, we use a simple bilinear scoring function:

$$\mathcal{D}(h_i^s, s^{(s)}) = \sigma((h_i^s)^T Q^s s^{(s)}), \quad (6)$$

where σ is the Sigmoid function, and $Q^s \in \mathbb{R}^{n\times n}$ is a trainable scoring matrix. To obtain the negative node embedding $\widetilde{h}_j^s$, we then corrupt the original content matrix by shuffling it in the row-wise manner, *i.e.*, $\widetilde{X}^s \longleftarrow X^s$, and reuse the encoder GNN_{cl}^s in (3):

$$\widetilde{H}^s = GNN_{cl}^s(\widetilde{X}^s, A^s|W^s) = \sigma(\hat{D}^{-\frac{1}{2}} \hat{A}^s \hat{D}^{-\frac{1}{2}} \widetilde{X}^s W^s). \quad (7)$$

After independently maximizing the average MI between local patches $h_1^s, h_2^s, \ldots, h_n^s$ and the graph-level summary $s^{(s)}$ of graph G^s, we can obtain source domain-specific node embedding matrix H^s to capture global information in G^s.

Likewise, for target graph G^t, we can obtain target domain-specific node embedding matrix H^t that learns global information in G^t by minimizing target domain-specific contrastive loss $\mathcal{L}_{cl}^s$. Finally, we combine these domain-specific contrastive losses into a single loss $\mathcal{L}_{cl}^{specific}$ as follows:

$$\mathcal{L}_{cl}^{specific} = \mathcal{L}_{cl}^s + \mathcal{L}_{cl}^t. \quad (8)$$

2) Cross-Domain Graph Contrastive Learning: In order to measure local and global information across different domains, a cross-domain contrastive objective function is designed to integrate embeddings across domains to jointly learn high-quality representations. Fig. 3 illustrates the overview structure of cross-domain graph contrastive learning. The motivation is to make node representations of the source domain largely consistent

with that of the target domain (the attraction force to the right), but also make the node presentations sufficiently discriminative for domain classifier to separate source *vs.* target domains (the repulsion force to the left).

For each graph including the source graph G^s and the target graph G^t, we first obtain corresponding domain-specific node embedding matrix H, graph-level summary embedding s and negative node embedding matrix $\widetilde{H}$ based on previous domain-specific graph contrastive learning network. Then, we try to enhance domain-specific node embedding matrix H in each domain by leveraging the global summary information from the other domain. In order to do that, we design novel cross-domain contrastive losses $\mathcal{L}^{s,t}_{cross}$ and $\mathcal{L}^{t,s}_{cross}$ as follows:

$$\mathcal{L}^{s,t}_{cross} = \frac{1}{2n^s}\left(\sum_i^{n^s} \log \mathcal{D}(h^s_i, s^{(t)}) + \sum_j^{n^s} \log\left(1 - \mathcal{D}(\widetilde{h}^s_j, s^{(t)})\right)\right), \tag{9}$$

$$\mathcal{L}^{t,s}_{cross} = \frac{1}{2n^t}\left(\sum_i^{n^t} \log \mathcal{D}(h^t_i, s^{(s)}) + \sum_j^{n^t} \log\left(1 - \mathcal{D}(\widetilde{h}^t_j, s^{(s)})\right)\right), \tag{10}$$

where n^s and n^t represent the number of nodes of the source graph G^s and the target graph G^t, respectively. As a result, the attraction force of the cross-domain contrastive learning is defined by the cross-domain contrastive loss $\mathcal{L}_{cross}$ in (11), which tries to make source and target node features to be largely consistent.

$$\mathcal{L}^{cross}_{cl} = \mathcal{L}^{s,t}_{cross} + \mathcal{L}^{t,s}_{cross}. \tag{11}$$

The repulsion force of the cross-domain contrastive learning is later defined in (15), which intends to make node presentations sufficiently discriminative to separate source *vs.* target domains.

We combine domain-specific contrastive losses and cross-domain contrastive loss as the final objective function $\mathcal{L}_{cl}$ of graph contrastive learning module.

$$\mathcal{L}_{cl} = \mathcal{L}^{specific}_{cl} + \mathcal{L}^{cross}_{cl}. \tag{12}$$

Note that, we use the same dimension d in the source-domain graph and target-domain graph to simplify model adjustment.

C. Attention Mechanism

The node embedding module aims to generate four embeddings: Z^s_A, Z^s_P for source graph and Z^t_A, Z^t_P for target graph. In addition, by leveraging the cross-domain graph contrastive learning module, we obtain two embeddings denoted as H^s for source graph and H^t for target graph. A unified representation is obtained by aggregating embeddings from different graphs. Specifically, aiming to learn a unified representation, we use an attention scheme, following [3], to capture the significance of all embeddings from the local, global consistency and graph contrastive learning networks in each domain, for the sake that the contributions of their embeddings are different.

D. Domain Adaptive Learning for Cross-Domain Node Classification

To better classify nodes in the target network, a knowledge transfer from different domains is enabled. Our proposed model consists of an adversarial module, a source classifier, a target classifier as well as domain-specific and cross-domain graph contrastive learning, which can learn discriminative and domain-invariant node embeddings.

The overall objective function can defined as follows:

$$\mathcal{L}(Z^s, Y^s, Z^t) = \mathcal{L}_S + \gamma_1 \mathcal{L}_{DA} + \gamma_2 \mathcal{L}_T + \gamma_3 \mathcal{L}_{cl} \tag{13}$$

In (13), γ_1, γ_2, and γ_3 are the balance factors. Their values all range from 0 to 1. Specifically, γ_1 is the balance factor of the loss for the domain classifier, γ_2 regulates loss for the target classifier, and γ_3 balances the loss of the graph contrastive learning module. The $\mathcal{L}_{cl}$ represents the graph contrastive learning loss. The $\mathcal{L}_S$ denotes the loss of the source classifier. Similarly, $\mathcal{L}_{DA}$ is the loss for the domain classifier, and $\mathcal{L}_T$ is the loss for the target classifier. The definitions of respective loss terms are detailed as follows.

1) Source Classifier Loss: A cross-entropy loss is used in the source domain as the source classifier loss $\mathcal{L}_S(f_s(Z^s), Y^s)$:

$$\mathcal{L}_S(f_s(Z^s), Y^s) = -\frac{1}{N_s}\sum_{i=1}^{N_s} y_i \log(\hat{y}_i), \tag{14}$$

where y_i is the ground-truth label of the i-th node in the source domain, and the classification prediction for the i-th source labeled node v^s_i is denote as $\hat{y}_i$.

2) Domain Classifier Loss: To separate the two domains (*i.e., source domain network G^s and target domain network G^t*), we design a domain classifier loss $\mathcal{L}_{DA}(Z^s, Z^t)$ to act as a repulsion force for node representation from the two domains to be discriminative for separation. To achieve the goal, with the help of adversarial learning, a domain classifier $f_d(Q_\lambda(Z^s, Z^t); \theta_D)$ parameterized by θ_D is learned to discriminate if the node comes from G^s or G^t.

On one hand, with source classifier f_s via minimizing (14), we can classify each node into the correct class. On the other hand, in order to make node representations from different domains dissimilar, we use a domain classifier which can differentiate if the node comes from G^t or G^s. Here, with the aid of adversarial training, we use Gradient Reversal Layer (GRL) [8] and minimize the following domain classifier loss:

$$\mathcal{L}_{DA} = -\frac{1}{N^s + N^t}\sum_{i=1}^{N^s+N^t} m_i \log(\hat{m}_i) + (1 - m_i)\log(1 - \hat{m}_i), \tag{15}$$

In (15), $m_i \in \{0, 1\}$ is the ground truth of node i. $\hat{m}_i$ denotes the domain prediction result for the node i in the source domain or target domain.

3) Target Classifier Loss: In the target domain, we use an entropy loss as target classifier loss, as defined as follows:

$$\mathcal{L}_T(f_t(Z^t)) = -\frac{1}{N^t}\sum_{i=1}^{N^t} \hat{y}_i \log(\hat{y}_i), \tag{16}$$

where $\hat{y}_i$ is the result of classification prediction for the i-th node v_i^t in target domain.

By minimizing the objective function in (13), $\mathcal{L}_S(Z^s, Y^s)$, $\mathcal{L}_{DA}(Z^s, Z^t)$, $\mathcal{L}_T(Z^t)$ and $\mathcal{L}_{cl}$ can be jointly optimized. Furthermore, we utilize the standard backpropagation algorithms to optimize all parameters.

E. Algorithm Description

Algorithm 1 shows the detailed algorithmic process. Given a source graph $G^s = (V^s, E^s, X^s, Y^s)$ and a target graph $G^t = (V^t, E^t, X^t)$, our goal is to learn node embeddings for source and target graphs Z^s and Z^t, respectively.

Firstly, a dual graph convolutional network is utilized to learn local and global consistency relationship of each graph (Steps 2-7). Then, we utilize domain-specific and cross-domain graph contrastive learning networks to learn node representations of different domains by preserving global properties of corresponding graphs (Steps 8-10). Specifically, X^s X^t denote the original input, Z_A^s, Z_P^s, H^s denotes the output for source domain and Z_A^t, Z_P^t, H^t for target domain. Then an attention mechanism is introduced to the output of each domain to obtain the final node representation output Z^s and Z^t (Step 11). Finally, to release the domain discrepancy for cross-domain node classification (Steps 12-14), GCLN maximally utilizes the domain and label information to learn domain-invariant and semantic representations effectively by using the source classifier, domain classifier and target classifier.

F. Time Complexity Analysis

The key parts of our model include node representation learning (Local Consistency Network (GNN_A) and Global Consistency Network (GNN_P)), and the graph contrastive learning component. Regarding the node representation learning network, for the source domain, given a source graph with n^s nodes and m^s edges, the time complexity for the graph convolution of GCN is $\mathcal{O}(m^s)$ for sparse adjacent matrix (*i.e.* a sparse network). Furthermore, instead of the adjacency matrix, the point-wise mutual information (PPMI) matrix is utilized for propagation. Because the PPMI matrix is not necessarily a sparse matrix, its complexity is akin to the complexity of a dense matrix: $\mathcal{O}((n^s)^2)$. Furthermore, because our local and global GCNs mainly consist of two matrices, the sparse adjacency matrix and a dense PPMI matrix from the source domain and target domain, the overall time complexity of the node representation learning network is $\mathcal{O}(m^s + (n^s)^2 + m^t + (n^t)^2)$.

Next, the computation of the graph contrastive learning component mainly consists of domain-specific encoder network (GNN_{cl}), domain-specific contrastive loss, (8), and cross-domain contrastive loss, (11). Specifically, about the domain-specific encoder network (GNN_{cl} for source and target domain, its complexity is $\mathcal{O}(2 * m^s + 2 * m^t)$. For domain-specific contrastive loss, its complexity is $\mathcal{O}(2 * n^s + 2 * n^t)$. For cross-domain contrastive loss, the complexity is also $\mathcal{O}(2 * n^s + 2 * n^t)$. Therefore, the overall time complexity of the graph contrastive learning component is $\mathcal{O}(2\ m^s + 2\ m^t + 4 * n^s + 4 * n^t)$.

Algorithm 1: GCLN: Graph Contrastive Learning Network for Unsupervised Domain Adaptive Node Classification.

Require:
Source domain network: $G^s = (V^s, E^s, X^s, Y^s)$;
Target domain network: $G^t = (V^t, E^t, X^t)$;
The numbers of GCN layers: L;
Ensure:
$[Z^s, Z^t]$: learned output embeddings
$[f_s, f_d, f_t]$: learned source classifier, domain classifier, and target classifier, respectively;

1: **while** not convergence **do**
2: **for** i=1 to L **do**
3: $Z_A^i \leftarrow Conv_A^{(i)}(Z^{(i-1)})$
4: $Z_P^i \leftarrow Conv_P^{(i)}(Z^{(i-1)})$
5: **if** $i == 1$ **then**
6: $Z^{(i-1)} \leftarrow X$
7: $[Z_A^s, Z_P^s, Z_A^t, Z_P^t] \leftarrow$ Obtain embeddings Z_A^s, Z_P^s for source domain nodes and Z_A^t, Z_P^t for target domain nodes.
8: $H^s \leftarrow GNN_{cl}^s(X^s, A^s|W^s)$
9: $H^t \leftarrow GNN_{cl}^t(X^t, A^t|W^t)$
10: $[H^s, H^t] \leftarrow$ Learn two domain-specific node embedding matrices following (7)
11: $[Z^s, Z^t] \leftarrow$ Learn embedding features for source and target domains, respectively, using the attention mechanism.
12: $f_s \leftarrow$ Update source classifier from Z^s and Y^s using (14)
13: $f_d \leftarrow$ Update domain classifier from Z^s and Z^t using (15)
14: $f_t \leftarrow$ Update target classifier from Z^t using (16)
15: Back-propagate loss gradient from Z^s, Z^t and Y^s using (13)
16: Update weights
17: *Terminate* if early stopping criteria satisfied.

In summary, the complexity of GCLN is $\mathcal{O}(3\ m^s + (n^s)^2 + 3\ m^t + (n^t)^2 + 4n^s + 4n^t)$. Because the most dense network is the one with each node connecting to all other nodes, we have $m^s \leq (n^s)^2$ and $m^t \leq (n^t)^2$ for general networks. Thus, the asymptotic complexity of GCLN is bounded by $\mathcal{O}((n^s)^2 + (n^t)^2)$.

V. EXPERIMENTS

A. Benchmark Datasets

We conduct multiple experiments on DBLPv8, ACMv9, and Citationv2, which are three paper citation networks [3], [36] and their statistics are reported in Table I. In our experiments, these three datasets (DBLPv8, ACMv9, and Citationv2) are extracted from different sources (DBLP, ACM and Microsoft Academic Graph respectively). In addition, for each dataset, we select papers from different time periods. Following [3], each dataset is regarded as an undirected network and each edge denotes the relationship (*i.e.*, the citation relationship between two papers).

TABLE I
STATISTICS OF THE EXPERIMENTAL DATASETS

Dataset	# of Nodes	# of Edges	# of Features	# of Labels
DBLPv8	5,578	7,341	7,537	6
ACMv9	7,410	11,135	7,537	6
Citationv2	4,715	6,717	7,537	6

Specifically, in the experiments, we have six categories, which are "Database", "Data mining" "Artificial intelligent", Computer vision", "Information Security" and "High Performance Computing". We classify each paper to the corresponding category based on its research topic. To evaluate the performance of our model, we conduct node classification tasks on these different network domains (we use D, A, C to represent DBLPv8 domain, ACMv9 domain, and Citationv2 domain, respectively). In the experiments, we carry out six transfer learning tasks, which are D $\rightarrow$ C, A $\rightarrow$ C, D $\rightarrow$ A, C $\rightarrow$ A, C $\rightarrow$ D and A $\rightarrow$ D.

B. Baselines

To demonstrate the effectiveness of our proposed model, we compare our method with the following baselines, including the single-domain node classification methods and cross-domain node classification methods.

Single-domain node classification methods:

- DeepWalk [37]: DeepWalk designs a random walk sampling strategy to obtain the low-dimensional representation for each node, which is a classic single domain graph representation method.
- LINE [38]: LINE is a classic network embedding method designed to learn the undirected network representation. LINE takes the first-order and second-order proximities of each graph into consideration to model node co-occurrence probabilities and node conditional probabilities.
- GraphSAGE [12]: GraphSAGE is an inductive framework that leverages node feature information (*e.g.*, text attributes) to efficiently generate node embeddings for target network node classification.
- DNN: DNN is a node-based multi-layer perceptron (MLP) that employs just node features.
- GCN [6]: GCN is a graph-based deep convolutional network that incorporates network architecture, node features, and observed labels into the learning framework to learn embedding features for nodes.

For these methods, we train a classifier from the embedding of the source domain and use the classifier to make predictions on the target domain.

Cross-domain node classification models with adaption:

- DGRL [8]: In DGRL, the feature generator is a 2-layer perceptron and a gradient reverse layer (GRL) is added to learn better representations for domain classification.
- AdaGCN [5]: AdaGCN utilizes the GCN as the feature generator and it also uses the gradient reverse layer (GRL) to conduct domain classification.
- UDA-GCN [3]: UDA-GCN employs a dual graph convolutional network as the feature generator to preserve the local and global consistency for the node representation learning, and the domain classification is conducted by adding a gradient reverse layer (GRL).

These are current state-of-the-art models which enable joint training between source and target domains for domain adaptive network classification.

C. Experimental Settings

We use Pytorch [39] as our deep learning framework and utilize Adam as an optimizer. Following the evaluation strategy in the unsupervised domain adaptation [3], [5], best results of all approaches are reported by grid search on the hyperparameter space. We set the learning rate to $1e^{-4}$ for each method. For some deep approaches (*i.e.* GCN, AdaGCN, and UDA-GCN), they all contain two hidden layers ($L = 2$) with a structure of $128 - 16$ for both source and target networks. To be consistent, our proposed model GCLN, GNN_A module and GNN_P module have the same hidden layers configuration as above, and domain-specific graph neural encoder module GNN^s_{cl} and GNN^t_{cl} contain a single layer with $dim = 16$, respectively. Here, the final dimension of node output embedding is set to 16.

For DeepWalk and LINE, we first learn node embeddings and train a classifier with the labels of data in the source domain. The dimension of node embeddings for these methods are the same, which are set to 128. DNN and DGRL have similar parameter settings with GCN and AdaGCN, respectively. The adaptation rate λ is set according to the following schedule: $\lambda = \min(\frac{2}{1+\exp(-10p)} - 1, 0.1)$, and the p changes from 0 to 1 during training process as in [8]. We set balance parameters γ_1, γ_2 and γ_3 to 1, 0.8, and 0.6, respectively. The dropout rate for each GNN layer is set to 0.3.

D. Cross-Domain Classification Results

To show the performance of each approach compared with our proposed model, we list the results of different methods in Table II. Based on the results, we have observations as follows:

1) GCN and GraphSAGE, the graph-based approaches perform better than DeepWalk and LINE, which indicates that the graph convolutional neural networks have competitive advantages over the traditional two-step network embedding methods. The reason may be that the traditional methods like DeepWalk and LINE lack the ability to encode both the local graph structure and node features at the same time for cross domain node classification tasks.
2) DGRL, AdaGCN, UDA-GCN obtain better performances than DNN and GCN (single-domain node classification approaches), confirming that the introduction of domain loss can assist in improving the model performance in cross-domain node classification tasks. In addition, UDA-GCN outperforms AdaGCN, indicating the effectiveness of the global consistency relation and target classifier loss.
3) The proposed GCLN model achieves the best performance among all baseline methods on these six transfer learning tasks. It confirms that the proposed GCLN combines attraction and repulsion force to ensure intra- and inter-domain consistency within a single network and across multiple networks, where the latent representation of each node can be deeply explored and the distribution gap across different domains can be largely decreased.

TABLE II
CLASSIFICATION ACCURACY COMPARISONS ON SIX CROSS-DOMAIN TASKS

Methods	C→D	A→D	D→C	A→C	D→A	C→A	**Average**
DeepWalk	0.1397	0.1798	0.2840	0.2284	0.3649	0.2005	0.2329
LINE	0.2216	0.1972	0.2539	0.2848	0.4117	0.1895	0.2598
GraphSAGE	0.6151	0.7228	0.6312	0.4333	0.6961	0.4413	0.5900
DNN	0.4035	0.4279	0.5065	0.3832	0.5904	0.3669	0.4464
GCN	0.6250	0.6486	0.6259	0.4327	0.6945	0.4474	0.5790
DGRL	0.4229	0.4303	0.5018	0.3877	0.5947	0.3799	0.4529
AdaGCN	0.6388	0.7142	0.6399	0.4448	0.7045	0.4494	0.5986
UDA-GCN	0.7182	0.7341	0.7281	0.4770	0.7617	0.4603	0.6466
GCLN	**0.7311**	**0.7720**	**0.7383**	**0.5069**	**0.7717**	**0.4827**	**0.6671**

TABLE III
CLASSIFICATION ACCURACY COMPARISONS BETWEEN GCLN VARIANTS ON SIX CROSS-DOMAIN TASKS

Methods	C→D	A→D	D→C	A→C	D→A	C→A	**Average**
GCLN$\neg d$	0.6730	0.6909	0.7247	0.4694	0.7532	0.4667	0.6296
GCLN$\neg t$	0.6848	0.7510	0.7012	0.4785	0.7405	0.4229	0.6298
GCLN$\neg ds$	0.7139	0.7320	0.6971	0.4897	0.7459	0.4718	0.6417
GCLN$\neg cd$	0.6832	0.7456	0.7164	0.4780	0.7215	0.4545	0.6332
GCLN$\neg p$	0.7062	0.7257	0.6899	0.4479	0.7244	0.4112	0.6176
GCLN	**0.7311**	**0.7720**	**0.7383**	**0.5069**	**0.7717**	**0.4827**	**0.6671**

E. Analysis of GCLN Components

Because GCLN has multiple key components, in this section, we compare several variants of GCLN with respect to the following aspects to demonstrate the effectiveness of GCLN. In particular, the following variants of GCLN are compared.

- GCLN$\neg d$: A variant of GCLN with the domain classifier loss ($\mathcal{L}_{DA}$) of GCLN being removed.
- GCLN$\neg t$: A variant of GCLN with the target classifier loss ($\mathcal{L}_T$) of UDA-GCL being removed.
- GCLN$\neg ds$: A variant of GCLN with the domain-specific contrastive loss ($\mathcal{L}_{cl}^{specific}$) of GCLN being removed.
- GCLN$\neg cd$: A variant of GCLN with the cross-domain contrastive loss ($\mathcal{L}_{cl}^{cross}$) of GCLN being removed.
- GCLN$\neg p$: A variant of GCLN with the Global Consistency Network (GNN_P) being removed.

The ablation study results are shown in Table III.

1) Impact of Domain-Adversarial Loss: To verify the effectiveness of the domain-adversarial loss, we compare GCLN model and GCLN$\neg d$ in Table III, from which we can find out that the GCLN model performs significantly better than GCLN$\neg d$. This shows that the proposed model can learn a superior representation for nodes from different domains by using the domain-adversarial loss.

2) Impact of the Target Classifier Loss: A variant model GCLN$\neg t$ is used to show the effectiveness of the target classifier loss. The only difference between GCLN$\neg t$ and GCLN is that GCLN$\neg t$ does not use target domain information which is also the core information in cross-domain learning. Table III shows that node classification performance on both datasets is raised when using the target information, indicating the effectiveness of the target classifier loss.

3) Impact of the Domain-Specific Contrastive Loss: The domain-specific graph contrastive learning module learns node embeddings by considering local and global information of each domain-specific graph. From Table III, our GCLN outperforms the GCLN$\neg ds$, showing that the importance of relationships between local and global information can be better captured by the domain-specific contrastive loss, which can enhance node representations of source and target domains.

4) Impact of the Cross-Domain Contrastive Loss: The cross-domain contrastive graph learning module focuses on modeling connections between local and global information cross different domains to facilitate them to learn high-quality node representations. The performance of GCLN is superior to GCLN$\neg cd$, indicating the effectiveness of this cross-domain contrastive loss.

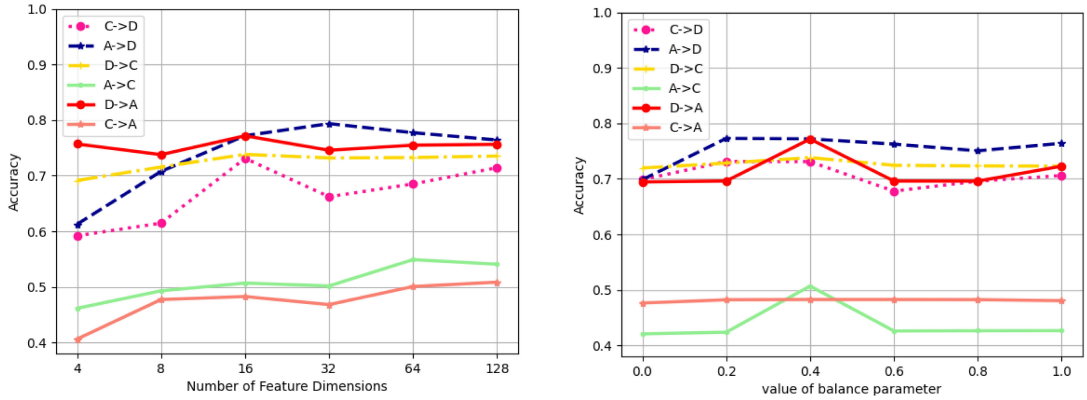
5) Impact of the Global Consistency Network Module: To investigate the effectiveness of the global consistency network used in the proposed model, we compare GCLN with GCLN$\neg p$ (*i.e.*, the version without global consistency network). The results show that the GCLN model performs better than GCLN$\neg p$, which confirms the superiority of the global consistency network.

F. Parameter Analysis

1) Impact of Feature Dimensions for Node Representations Z^s and Z^t: We set the same dimensions d for source output feature Z^s and target output feature Z^t, and vary d from 4 to 128. We report the results for cross domain node classification on six datasets in Fig. 4(a). We observed that when d is increased from 4 to 128, the performance for the node classification in the target domain also increases. Moreover, when d increases from 16 to 128, there are only slight differences.

The results indicate that with sufficient feature dimensions (*e.g.*, $d \geq 16$), the performance of our algorithm is relatively stable.

2) Impact of Global Consistency Network With Contrastive Learning: In order to learn more expressive node representations for source and target domain by capturing global graph property, we leverage a graph contrastive learning module to achieve representation learning on each domain-specific graph. Specially, we only utilize the original topological structure (adjacency matrix A) of each domain-specific graph as input to train each domain-specific graph neural encoder network



(a) embedding feature dimensions (b) balance parameter γ_3

Fig. 4. (a) Impact of feature dimensions of node output embeddings; (b) Impact of balance parameter γ_3 of the graph contrative learning loss.

TABLE IV
CLASSIFICATION ACCURACY COMPARISONS BETWEEN GCLN AND GCLN_G (THE CONTRASTIVE LEARNING WITH THE PPMI MATRIX ON TWO DOMAINS) ON SIX CROSS-DOMAIN TASKS

Methods	C→D	A→D	D→C	A→C	D→A	C→A	**Average**
GCLN_G	0.7017	0.7648	0.7043	0.4978	0.7343	0.4722	0.6458
GCLN	**0.7311**	**0.7720**	**0.7383**	**0.5069**	**0.7717**	**0.4827**	**0.6671**

GNN_{cl}, but ignore PPMI matrix P which is used to ensure global consistency for training GNN encoder GNN_p effectively. Intuitively, PPMI matrix P may also improve performance of GNN encoder GNN_{cl} in our graph contrastive learning module. In this part, we aim to explore the impact of PPMI matrix P on the graph contrastive learning module, and conduct extensive experiments by using the corresponding PPMI matrix P instead of the adjacency matrix A, to train each domain-specific graph neural encoder network GNN_{cl}, denoted as GCLN_G. The results of six cross domain node classification tasks are reported in Table IV.

From the results, the performance of GCLN_G model is worse than our proposed model in all cases (it is nearly the same as UDA-GCN), demonstrating that global consistency information of PPMI matrix P has no positive effect on our graph contrastive learning module for learning more expressive representations. The worse results of GCLN_G model may be due to conflicts between its global consistency information and global summary information of our graph contrastive learning module, which may lead to the worse effect of node representation learning.

3) Impact of Balance Parameter γ_3 of the Graph Contrative Learning Loss: We analyze the sensitivity of balance for parameter γ_3 in (13) by fixing $\gamma_1 = 1.0, \gamma_2 = 0.8$ on cross-domain datasets (γ_1 and γ_2 are reported best parameters for UDA-GCN [3]). We change the values of γ_3 from 0 to 1.0 and the results on six cross-domain tasks are shown in Fig. 4(b). From the results, we get the best score when $\gamma_3 = 0.6$ on all cross-domain tasks. This demonstrates that graph contrastive learning module is able to adapt to the cross-domain data for obtaining more expressive representations and further enhance the effect of objective classifiers.

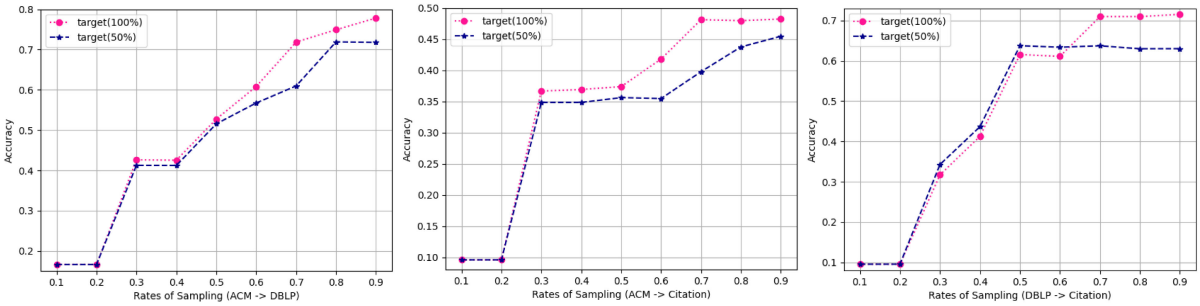
4) Visualization: It is an important application that uses network representation to create meaningful visualizations. We visualize the learned feature representations for the target domain dataset. For simplicity, to show the effectiveness of GCLN, we only visualize two learned feature representations in the ACM → DBLP. Specifically, we visualize the feature embedding into a 2-D vector space by using the T-distributed Stochastic Neighbor Embedding (t-SNE) [40] method. Fig. 5 shows the visualization results of different approaches. We can observe that it is not very meaningful to visualize embeddings of DNN, as many nodes that belong to the same category are not grouped together and many clusters overlap. GraphSAGE and UDA-GCN achieve better visualization results compared with DNN. The embedding obtained by these algorithms result in more meaningful visualization. However, it is still hard to find the boundaries of most clusters. For the proposed GCLN, the clustering results are clearer and the clusters are well separated. This observation indicates that GCLN can produce much more meaningful layout for networks compared with other models.

5) Sample Complexity: We conduct experiments and report accuracy of our model in terms of data sampling rates in Fig. 6 for different domain learning tasks including ACM->DBLP (A->D), ACM->Citation (A->C) and DBLP->Citation (D->C). When sampling rate of source domain rises from 0.1 to 0.9, accuracy increases due to more training data from source domain being used. For different sampling rate of target domain (100% and 50%), they behave differently, showing that data of target domain has different influences on the performance of our model.

6) Different Loss Functions Evolve Through the Different Epochs: We show the result curves of different losses as the epoch changes from 0 to 200 in Fig. 7. From results, we can conclude that the losses fluctuate greatly due to the lack of data training at the beginning, but eventually they will be stable, indicating the convergence of our model. In addition, with the

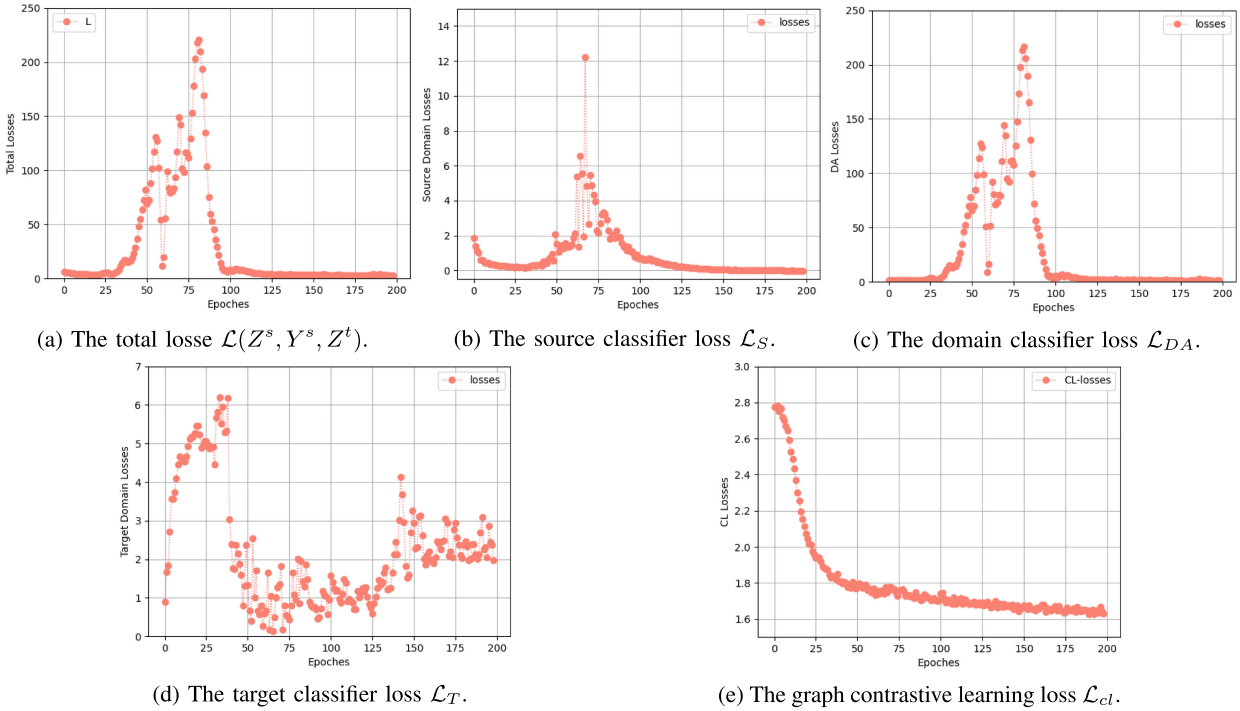
(a) DNN (b) GraphSAGE (c) UDA-GCN (d) GCLN

Fig. 5. Visualization of the embedding features learned from different methods using t-SNE [40]. The source domain is ACM and the target domain is DBLP. Each plot shows the learned embedding for the target domain dataset (DBLP), which has six classes. Nodes are color-coded using their class labels.



(a) ACM->DBLP (b) ACM->Citation (c) DBLP->Citation

Fig. 6. Impacts of data sampling rates for different domain adaptation learning tasks including ACM->DBLP (A->D), ACM->Citation (A->C) and DBLP->Citation (D->C).



(a) The total losse $\mathcal{L}(Z^s, Y^s, Z^t)$. (b) The source classifier loss $\mathcal{L}_S$. (c) The domain classifier loss $\mathcal{L}_{DA}$.

(d) The target classifier loss $\mathcal{L}_T$. (e) The graph contrastive learning loss $\mathcal{L}_{cl}$.

Fig. 7. Impacts of different loss functions for different epochs including the total losse $\mathcal{L}(Z^s, Y^s, Z^t)$, the source classifier loss $\mathcal{L}_S$, the domain classifier loss $\mathcal{L}_{DA}$, the target classifier loss $\mathcal{L}_T$, and the graph contrastive learning loss $\mathcal{L}_{cl}$.

increase of training epochs, each losses decrease and tend to be stable, which also illustrates the usefulness of these losses to our model.

VI. Conclusion

In this paper, we studied unsupervised graph domain adaptation learning. We argued that transferring knowledge between graph domains is challenging in both graph representation learning and domain adaptation, and proposed a novel method, Unsupervised Domain Adaptive Graph Contrastive Learning Network (GCLN) to enable knowledge transferring between graphs via domain adaptation. To learn more expressive representations of nodes for source domain and target domain by capturing global information of graphs, we designed a graph contrastive learning framework to facilitate representation learning on each domain-specific graph (source and target domain) by maximizing the mutual information between patch-level representation and graph-level representation of the graph data. In addition, we developed a cross-domain objective function to jointly learn the node embeddings from different domains, so that the data from one domain can enhance another for the representation learning. By optimizing a cross entropy loss function for the node classification from the source domain, a domain adversarial loss function to distinguish nodes from different domains, and an entropy loss function to exploit the target domain information, we can reduce the discrepancy between domains and further enhance the task of domain adaptation. Extensive experiments on three real-world graph datasets and demonstrated that our proposed method GCLN outperforms state-of-the-art algorithms for node classification in the unsupervised cross network settings.

References

[1] M. Long *et al.*, "Transfer sparse coding for robust image representation," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 407–414.

[2] J. Jiang and C. Zhai, "Instance weighting for domain adaptation in NLP," in *Proc. 45th Annu. Meeting Assoc. Comput. Linguistics*, 2007, pp. 264–271.

[3] M. Wu, S. Pan, C. Zhou, X. Chang, and X. Zhu, "Unsupervised domain adaptive graph convolutional networks," in *Proc. Web Conf.*, 2020, pp. 1457–1467.

[4] X. Shen, Q. Dai, S. Mao, F. L. Chung, and K. S. Choi, "Network Together: Node Classification via Cross network Deep Network Embedding," *IEEE Trans. Neural. Netw. Learn. Syst.*, vol. 32, no. 5, pp. 1935–1948, Jun. 2021.

[5] X. Shen, Q. Dai, F. L. Chung, W. Lu, and K. S. Choi, "Adversarial Deep Network Embedding for Cross-Network Node Classification," in *Proc. 34th AAAI Conf. Artif. Intell.*, 2020, pp. 2991–2999.

[6] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. Int. Conf. Learn. Representations*, 2017, pp. 1–14.

[7] P. Velickovic, W. Fedus, W. L. Hamilton, P. Liò, Y. Bengio, and R. D. Hjelm, "Deep graph infomax," in *Proc. Int. Conf. Learn. Representations*, 2019, pp. 1–17.

[8] Y. Ganin *et al.*, "Domain-adversarial training of neural networks," *J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 2096–2030, 2016.

[9] Z. Wu *et al.*, "A comprehensive survey on graph neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 1, pp. 4–24, Jan. 2021.

[10] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang, "Beyond low-pass filtering: Graph convolutional networks with automatic filtering," 2021, *arXiv:2107.04755*.

[11] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," in *Proc. Int. Conf. Learn. Representations*, 2018, pp. 1–12.

[12] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proc. Neural Informat. Process. Syst.*, 2017, pp. 1024–1034.

[13] Q. Liu, M. Nickel, and D. Kiela, "Hyperbolic graph neural networks," in *Proc. Neural Informat. Process. Syst.*, 2019, pp. 8228–8239.

[14] L. Du, Y. Wang, G. Song, Z. Lu, and J. Wang, "Dynamic network embedding: An extended approach for skip-gram based network embedding," in *Proc. Int. Joint Conf. Artif. Intell.*, 2018, pp. 2086–2092.

[15] E. Tzeng *et al.*, "Adversarial discriminative domain adaptation," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 7167–7176.

[16] Y. Liu, S. Pan, M. Jin, C. Zhou, F. Xia, and P. S. Yu, "Graph self-supervised learning: A survey," 2021, *arXiv:2103.00111*.

[17] A. Jaiswal, A. R. Babu, M. Z. Zadeh, D. Banerjee, and F. Makedon, "A survey on contrastive self-supervised learning," *Technologies*, vol. 9, no. 1, pp. 1–20, Feb. 2021.

[18] M. I. Belghazi *et al.*, "Mutual information neural estimation," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 530–539.

[19] R. Linsker, "Self-organization in a perceptual network," *Computer*, vol. 21, no. 3, pp. 105–117, 1988.

[20] M. Ravanelli and Y. Bengio, "Learning speaker representations with mutual information," in *Proc. Interspeech*, 2019, pp. 1153–1157.

[21] R. D. Hjelm *et al.*, "Learning deep representations by mutual information estimation and maximization," in *Proc. Int. Conf. Learn. Representations*, 2019, pp. 1–24.

[22] P. Velickovic, W. Fedus, W. L. Hamilton, P. Liò, Y. Bengio, and R. D. Hjelm, "Deep graph infomax," in *Proc. Int. Conf. Learn. Representations*, 2019, pp. 1–17.

[23] Y. Liu, Z. Li, S. Pan, C. Gong, C. Zhou, and G. Karypis, "Anomaly detection on attributed networks via contrastive self-supervised learning," *IEEE Trans. Neural Netw. Learn. Syst.*, to be published, doi: 10.1109/TNNLS.2021.3068344.

[24] M. Jin, Y. Zheng, Y. Li, C. Gong, C. Zhou, and S. Pan, "Multi-scale contrastive siamese networks for self-supervised graph representation learning," in *Proc. 13th Int. Joint Conf. Artif. Intell.*, 2021, pp. 1477–1483.

[25] Y. Mansour, M. Mohri, and A. Rostamizadeh, "Domain adaptation: Learning bounds and algorithms," in *Proc. 22nd Conf. Learn. Theory*, COLT 2009, pp. 1–12.

[26] F. Zhuang *et al.*, "Supervised representation learning: Transfer learning with deep autoencoders," in *Proc. Int. Joint Conf. Artif. Intell.*, 2015, pp. 4119–4125.

[27] E. Tzeng *et al.*, "Deep domain confusion: Maximizing for domain invariance," 2014, *arXiv:1412.3474*.

[28] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *Proc. 32nd Int. Conf. Mach. Learn.*, 2015, pp. 1180–1189.

[29] Y. Zhang, S. Miao, and R. Liao, "Structural domain adaptation with latent graph alignment," in *Proc. 25th IEEE Int. Conf. Image Process.*, 2018, pp. 3753–3757.

[30] M. Pilanci and E. Vural, "Domain adaptation on graphs by learning aligned graph bases," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 2, pp. 587–600, Feb. 2020.

[31] D. Das and C. G. Lee, "Unsupervised domain adaptation using regularized hyper-graph matching," in *Proc. 25th IEEE Int. Conf. Image Process.*, 2018, pp. 3758–3762.

[32] R. Wang, Z. Wu, Z. Weng, J. Chen, G. Qi, and Y. Jiang, "Cross-domain contrastive learning for unsupervised domain adaptation," 2021, *arxiv:2106.05528*.

[33] Y. Zhang, G. Song, L. Du, S. Yang, and Y. Jin, "DANE: Domain adaptive network embedding," in *Proc. Int. Joint Conf. Artif. Intell.*, 2019, pp. 4362–4368.

[34] Y. Ouyang, B. Guo, X. Tang, X. He, J. Xiong, and Z. Yu, "Learning cross-domain representation with multi-graph neural network," 2019, *arXiv:1905.10095*.

[35] C. Zhuang and Q. Ma, "Dual graph convolutional networks for graph-based semi-supervised classification," in *Proc. World Wide Web Conf.*, 2018, pp. 499–508.

[36] J. Tang, J. Zhang, L. Yao, J. Li, L. Zhang, and Z. Su, "ArnetMiner: Extraction and mining of academic social networks," in *Proc. 14th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2008, pp. 990–998.

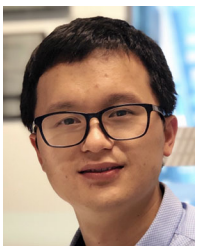
[37] B. Perozzi, R. Al-Rfou, and S. Skiena, "DeepWalk: Online learning of social representations," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2014, pp. 701–710.

[38] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "LINE: Large-scale information network embedding," in *Proc. Int. World Wide Web Conf. Steering Committee*, 2015, pp. 1067–1077.

[39] A. Paszke *et al.*, "Automatic differentiation in PyTorch," in *Proc. Neural Informat. Process. Syst. Workshop*, 2017, pp. 1–4.

[40] L. v. d. Maaten and G. Hinton, "Visualizing data using T-SNE," *J. Mach. Learn. Res.*, vol. 9, no. Nov., pp. 2579–2605, 2008.

Man Wu is currently working toward the Ph.D. degree in computer science with the Department of Electrical Engineering and Computer Science, Florida Atlantic University, Boca Raton, FL, USA. Her research interests include machine learning, graph, and network analysis, with an emphasis on understanding and modeling complicated relational structures, graphs, and networks.



Shirui Pan (Member, IEEE) received the Ph.D. degree in information technology from the University of Technology Sydney, Ultimo, NSW, Australia. From 2021 to 2025, he is currently a Senior Lecturer and an ARC Future Fellow with the Department of Data Science and AI, Faculty of IT, Monash University, Melbourne, VIC, Australia. His research interests include data science and AI. He has authored or coauthored more than 100 research papers in top-tier journals and conferences, including the TPAMI, TKDE, TNNLS, ICML, NeurIPS, KDD, AAAI, IJCAI, WWW, CVPR, and ICDM. He was the recipient of the Best Student Paper Award of IEEE ICDM 2020. He is recognised as one of the AI 2000 AAAI/IJCAI Most Influential Scholars in Australia, in 2021.

Xingquan Zhu (Senior Member, IEEE) received the Ph.D. degree in computer science from Fudan University, Shanghai, China. He is currently a Full Professor with the Department of Electrical Engineering and Computer Science, Florida Atlantic University, Boca Raton, FL, USA. His research interests mainly include data analytics, machine learning, and bioinformatics. From 2000, he has authored or coauthored more than 280 refereed journal and conference papers in these areas, including four best paper awards, such as PAKDD-21, IRI-18, PAKDD-13, and ICTAI-05, and three best student paper awards, such as ICDM-20, ICKG-20, and ICPR-12. He was an Associate Editor for the IEEE TRANSACTION ON KNOWLEDGE AND DATA ENGINEERING during 2008–2012 and 2014–2021. Since 2017, he has been an Associate Editor for the *ACM Transaction on Knowledge Discovery*.