Semi-supervised Domain Adaptation in Graph Transfer Learning

Ziyue Qiao[1,2,3] , **Xiao Luo**[2] , **Meng Xiao**[4,5] , **Hao Dong**[4,5] , **Yuanchun Zhou**[4,5] and **Hui Xiong**[2,3,†]
[1]Jiangmen Laboratory of Carbon Science and Technology, Jiangmen
[2]The Hong Kong University of Science and Technology (Guangzhou), Guangzhou
[3]Guangzhou HKUST Fok Ying Tung Research Institute, Guangzhou
[4]Computer Network Information Center, Chinese Academy of Sciences, Beijing
[5]University of Chinese Academy of Sciences, Beijing
{ziyuejoe, xiaoluopku, xiaomeng7890, donghcn}@gmail.com, zyc@cnic.cn, xionghui@ust.hk

Abstract

As a specific case of graph transfer learning, unsupervised domain adaptation on graphs aims for knowledge transfer from label-rich source graphs to unlabeled target graphs. However, graphs with topology and attributes usually have considerable cross-domain disparity and there are numerous real-world scenarios where merely a subset of nodes are labeled in the source graph. This imposes critical challenges on graph transfer learning due to serious domain shifts and label scarcity. To address these challenges, we propose a method named **S**emi-supervised **G**raph **D**omain **A**daptation (SGDA). To deal with the domain shift, we add adaptive shift parameters to each of the source nodes, which are trained in an adversarial manner to align the cross-domain distributions of node embedding, thus the node classifier trained on labeled source nodes can be transferred to the target nodes. Moreover, to address the label scarcity, we propose pseudo-labeling on unlabeled nodes, which improves classification on the target graph via measuring the posterior influence of nodes based on their relative position to the class centroids. Finally, extensive experiments on a range of publicly accessible datasets validate the effectiveness of our proposed SGDA in different experimental settings.

1 Introduction

In the real world, graphs have been gaining popularity for their ability to represent structured data. As a basic problem, node classification has been applied in a variety of scenarios, including social networks [Fan *et al.*, 2019; Ju *et al.*, 2023], academic networks [Kong *et al.*, 2019; Wu *et al.*, 2020b], and biological networks [Ingraham *et al.*, 2019; Wang *et al.*, 2018]. Graph transfer learning, which transfers knowledge from a labeled source graph to help predict the labels of nodes in a target graph with domain changes, has attracted a growing amount of interest recently. This problem is crucial due to the prevalence of unlabeled graphs in the real world and the anticipation of acquiring information from known domains.

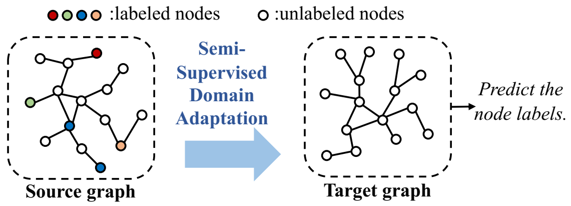


Figure 1: The semi-supervised domain adaptation on graphs.

Despite the significant progress made by graph transfer learning algorithms, they often assume that all nodes in the source graph are labeled. However, annotating the whole source graph becomes time-consuming and costly, particularly for large-scale networks. It is worth noting that recent semi-supervised node classification approaches can produce superior performance with a small number of node labels. This raises a natural problem, whether it is possible to use a small number of labeled data and a large amount of unlabeled data in the source network to infer label semantic information in the target graph with significant domain discrepancy. In a nutshell, this innovative application scenario is summarized as a semi-supervised domain adaptation on graphs.

Nonetheless, formalizing a semi-supervised domain adaptive framework for node classification remains a non-trivial task since it must address two basic issues: **Issue 1: How to overcome a significant domain shift cross graphs to give domain-invariant predictions?** The domain shift between the source and target graphs roughly lies in the following two views: graph topology and node attributes. For example, distinct graphs may have different link densities and substructure schemas, and hand-crafted node attributes from diverse sources could have significant biases individually. This brings more considerable domain shifts than traditional data. **Issue 2: How to mitigate the label scarcity for the classifier to give accurate and label-discriminative predictions?** Since the target graph is completely unlabeled, existing work only performs domain alignment, without considering the situation that the overall distributions may be aligned well but the class-level distributions may not match the classifier well. Even worse, only a subset of node labels is available on the

[†]Corresponding author.

source graph, impeding effective classifier learning. Thus, it is critical to leverage the graph topology to improve the discriminative ability of the classifier on the unlabeled nodes.

In this paper, we address the aforementioned problems by developing a novel **S**emi-supervised **G**raph **D**omain **A**daptation model named SGDA, as shown in Figure 1. The core idea of SGDA is to transfer the label knowledge from the source graph to the target graph via adversarial domain transformation and improve the model's prediction on unlabeled target nodes via adaptive pseudo-labeling with posterior scores. Specifically, we add pointwise node mutual information into the graph encoder, enabling the exploration of high-order topological proximity to learn generalized node representations. In addition, we add shift parameters onto the source graph to align the distribution of cross-domain node embeddings, so as to the classifier trained on source node embedding can be used to predict the labels of the target node. We propose an adversarial domain transformation module to train the graph encoder and shift parameters. Furthermore, to address the label scarcity problem, we introduce pseudo labels to supervise the training of unlabeled nodes in both domains, which adaptively increases the training weights of nodes close to the pseudo labels' cluster centroids, thus making the model gives discriminative predictions on these unlabeled nodes. The main contributions of our method for the semi-supervised domain adaptation in graph transfer learning can be summarized as follows:

- To eliminate the domain shift cross graphs, we introduce the concept of shift parameters on the source graph encoding and propose an adversarial transformation module to learn domain-invariant node embeddings.
- To alleviate the label scarcity, we propose a novel pseudo-labeling method using posterior scores to supervise the training of unlabeled nodes, improving the discriminative ability of the model on the target graph.
- Extensive experiments on various graph transfer learning benchmark datasets demonstrate the superiority of our SGDA over state-of-the-art methods.

2 Related Works

Domain Adaptation. Domain adaptation aims to transfer semantic knowledge from a source domain to a target domain, which has various applications in computer vision [Zhang *et al.*, 2022; Yan *et al.*, 2022]. In the literature, current methods can be roughly categorized into two types, i.e., distance-based methods [Chang *et al.*, 2021; Li *et al.*, 2020; Zhang and Davison, 2021] and adversarial learning-based methods [Zhang *et al.*, 2018; Tzeng *et al.*, 2017; Volpi *et al.*, 2018]. Distance-based methods explicitly calculate the distribution distance between source and target domains and minimize them in the embedding space. Typical metrics for distribution difference include maximum mean discrepancy (MMD) [Chang *et al.*, 2021] and enhanced transport distance (ETD) [Li *et al.*, 2020]. Adversarial learning-based methods usually train a domain discriminator on top of the hidden embeddings and attempt to fuse it for domain alignment in an implicit fashion. Despite the enormous effectiveness of domain adaptation, these methods typically focus on image problems. We address the challenge of semi-supervised domain adaptation on graphs by exploiting the graph topology information to enhance the model performance.

Graph Transfer Learning. Graph transfer learning has been widely studied in recent years. Early models [Qiao *et al.*, 2022; Qiu *et al.*, 2020; Hu *et al.*, 2020] typically utilize source data to construct a graph model for a different but related task in the target data. The effectiveness of transfer learning on graphs has been widely validated in a multi-task learning paradigm. Thus, graph transfer learning alleviates the burden of collecting labels regarding new tasks. The recent focus has been transformed into the problem of domain adaptation on graphs. Typically, these methods [Guo *et al.*, 2022; Shen *et al.*, 2020a] combine graph model with domain adaption techniques. In particular, they produce domain-invariant node representations either implicitly confounding a domain discriminator using adversarial learning [Zhang *et al.*, 2021; Wu *et al.*, 2020a] or explicitly minimizing the distance [Shen *et al.*, 2020b] between representations in two domains. Still, most work establishes methods on graphs similar to those on images, without considering the complex structure of graphs or explicitly exploiting the graph topology information.

Semi-supervised Learning on Graphs. Semi-supervised learning on graphs refers to the node classification problem, where only a small subset of nodes are labeled. Graph neural networks (GNNs) such as GCN [Welling and Kipf, 2016], GraphSAGE [Hamilton *et al.*, 2017], and GAT [Veličković *et al.*, 2018] have achieved great success on these problem. These methods usually follow the paradigm of message passing where each node attains information from its connected neighbors, followed by an aggregation operation for node representation updating in a recursive fashion. Recently, a range of GNN methods have been proposed to enhance the model performance from the view of exploring augmentation [Wen *et al.*, 2022; Wang *et al.*, 2020], expanding continuous [Xhonneux *et al.*, 2020], adversarial learning [Xu *et al.*, 2022; Jin *et al.*, 2021] and etc [Qiao *et al.*, 2023; Tang *et al.*, 2021]. However, these methods usually focus on learning and evaluation from a single graph. By contrast, we investigate a novel graph transfer learning problem named semi-supervised domain adaptation on graphs in this paper.

3 Problem Definition

The *source graph* is expressed as $G^s = \{A^s, X^s\}$ with $\mathcal{V}^{s,l}, \mathcal{V}^{s,u}$, and $Y^{s,l}$, where $A^s \in \mathbb{R}^{N^s \times N^s}$ is the adjacency matrix and N^s is the number of nodes in G^s. $A^s_{ij} = 1$ if there is an edge between nodes n_i and n_j, otherwise, $A^s_{ij} = 0$. $X^s \in \mathbb{R}^{N^s \times d}$ is the attribute matrix, where d is the dimension of node attributes. $\mathcal{V}^{s,l}$ is the labeled node set, $y_i \in \mathbb{R}^C$ is the ground-truth of the node $n_i \in \mathcal{V}^{s,l}$ where C is the number of classes and the k-th element $y_{i,k} = 1$ if the i-th node belong to the k-th class and otherwise $y_{i,k} = 0$. $\mathcal{V}^{s,u}$ is rest unlabeled node set in G^s, i.e, $|\mathcal{V}^{s,l}| + |\mathcal{V}^{s,u}| = N^s$. $|\mathcal{V}^{s,l}|$ is much fewer than $|\mathcal{V}^{s,u}|$ due to the expensive labeling cost.

The *target graph* is expressed as $G^t = \{A^t, X^t\}$ with node set $\mathcal{V}^t$, where $A^t \in \mathbb{R}^{N^t \times N^t}$ is adjacency matrix and N^t is number of nodes in G^t. $X^t \in \mathbb{R}^{N^t \times d}$ is the node attribute

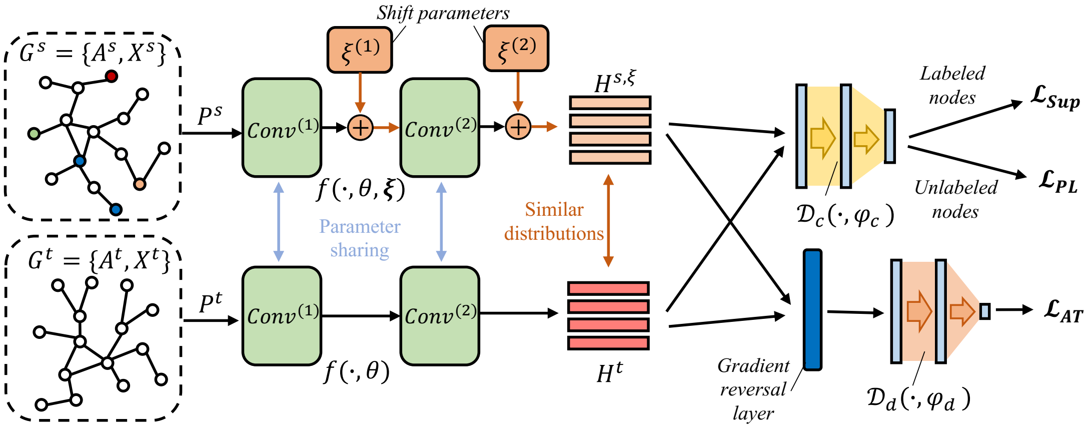


Figure 2: The framework of SGDA. The source graph and target graph with reconstructed high-order topologies P^s and P^t are fed into a two-layer graph convolutional network to generate generalized node embeddings, where source graph are added with shift parameters $\boldsymbol{\xi}$ to promote distribution alignment. Three losses $\mathcal{L}_{Sup}$, $\mathcal{L}_{AT}$, and $\mathcal{L}_{PL}$ perform supervised learning, domain adversarial transformation via shifting, and pseudo-labeling with posterior scores, respectively.

matrix. Note that the attribute set of the source and target graphs may have certain differences. However, one can create a union attribute set between them to align the dimension.

The problem of *semi-supervised domain adaptation on graphs* is given the source graph G^s with limited labels and the target graph G^t completely unlabeled, and they have certain domain discrepancy on the data distributions but share the same label space. The goal is to learn a model to accurately predict the node classes in the target graph with the assistance of the partially labeled source graph.

4 Methodology

As shown in Figure 2, our SGDA consists of three modules as below: (1) *Node Embedding Generalization.* To sufficiently explore high-order structured information in both graphs to learn generalized node representations; (2) *Adversarial Transformation.* To eliminate serious domain discrepancy between the source graph and the target graph, we introduce adaptive distribution shift parameters to the source graph, which are trained in an adversarial manner with regard to a domain discriminator. Therefore, the source graph is equipped with the target distribution. (3) *Pseudo-Labeling with Posterior Scores.* To alleviate the label scarcity, we propose pseudo-labeling loss on all unlabeled nodes cross-domain, which improves classification on the target graph via measuring the influence of nodes adaptively based on their relative position to the class centroid.

4.1 Node Embedding Generalization

Considering that the model needs to perform the cross-domain transfer and the labels are limited for the classification task, learning generalized node embeddings is critical for such a domain adaptation procedure. In view of this, we compute the positive pointwise mutual information [Zhuang and Ma, 2018] between nodes to fully explore high-order unlabeled graph topology information and use the graph convolutional network [Welling and Kipf, 2016] to encode nodes into generalized low-dimensional embeddings.

Given a graph $G = \{A, X\}$ with the adjacency matrix $A \in \mathbb{R}^{N\times N}$, We use the random walk to sample a set of paths on A and obtain a co-occurrence frequency matrix $F \in \mathbb{R}^{N\times N}$, where F_{ij} counts the times of the node n_j occurs within a predefined window in node n_i's context. Then, the positive mutual information between nodes is computed by:

$$\mathbb{P}_{ij} = \frac{F_{ij}}{\sum_{i,j} F_{ij}}, \ \mathbb{P}_i = \frac{\sum_j F_{ij}}{\sum_{i,j} F_{ij}}, \ \mathbb{P}_j = \frac{\sum_i F_{ij}}{\sum_{i,j} F_{ij}}, \\ P_{ij} = max\{log(\frac{\mathbb{P}_{ij}}{\mathbb{P}_i \times \mathbb{P}_j}), 0\}, \tag{1}$$

where $\mathbb{P}_{ij}$ is the probability of the node n_j occurring in the context of the node n_i. $\mathbb{P}_i$ and $\mathbb{P}_j$ are the probability of the node n_i as the anchor and node n_j as the context, respectively. P_{ij} is the positive mutual information between n_i and n_j, which reflects the high-order topological proximity between nodes, as it assumes that if two nodes have high-frequency co-occurrence, P_{ij} should be greater than if they are expected independent. i.e., $\mathbb{P}_{ij} > \mathbb{P}_i \times \mathbb{P}_j$. We can obtain a mutual information matrix P as the new adjacency matrix of G. Then, the l-th graph convolutional layer $Conv^{(l)}(\cdot)$ is defined as:

$$\begin{aligned} H^{(l)} &= Conv^{(l)}(P, H^{(l-1)}), \\ &= \sigma(D^{-\frac{1}{2}}\widetilde{P}D^{-\frac{1}{2}}H^{(l-1)}W^{(l)}), \end{aligned} \tag{2}$$

where $\sigma(\cdot)$ denotes an activation function. $\widetilde{P} = P + I$ where I is the identity matrix and D is the diagonal degree matrix of P (i.e., $D_{ii} = \sum_j \widetilde{P}_{ij}$). $W^{(l)}$ is the l-th layer weight matrix. $H^{(l)}$ is the l-th layer hidden output and $H^{(0)} = X$. Finally, we can build the backbone of our method by stacking L layers of graph convolutional networks in Equation 2, expressed as $f(G; \theta)$, where θ is the model parameters.

4.2 Adversarial Transformation via Shifting

Usually, the general learning objective of the domain adaptation is to train a feature encoder to eliminate the distribution discrepancy between the source domain and the target domain and then generate embeddings with similar distribution on both domains. Therefore, the classifier learned on the source domain can be adapted to the target domain. Most methods [Zhang *et al.*, 2021; Xiao *et al.*, 2022] attempt to match embedding space distributions by optimizing the feature encoder itself. However, graphs with non-Euclidean topology usually have more considerable input disparity than traditional data. Only using the parameters in the encoder (e.g., GNNs) may be insufficient to shift the distributions finely. Performing transition by adding trainable parameters (e.g., transport, perturbation) on input spaces has been proven to be effective in shifting one distribution to another one [Jiang *et al.*, 2020; Li *et al.*, 2020]. Be aware that, we proposed an adversarial transformation module, which aims to add shift parameters on the source graph to modify its distribution and use adversarial learning to train both the graph encoder and shift parameters to align the cross-domain distributions.

Specifically, given the source graph $G^s = \{A^s, X^s\}$ and the target graph $G^t = \{A^t, X^t\}$, we first add the shift parameters $\boldsymbol{\xi}$ onto the source graph and obtain the shifted source node embeddings $H^{s,\xi} = f(G^s; \theta, \boldsymbol{\xi})$ with the distribution $\mathcal{H}^{s,\xi}$, where $H^{s,\xi} \in \mathbb{R}^{N^s \times h}$ and h is the output dimension. Meanwhile the target node embeddings are obtained by $H^t = f(G^t; \theta)$ with the distribution $\mathcal{H}^t$, where $H^t \in \mathbb{R}^{N^t \times h}$. The optimization objective is to make the distributions similar, i.e., $\mathcal{H}^{s,\xi} \approx \mathcal{H}^t$. We define the shift parameters as randomly initialized multi-layer parameter matrices, i.e., $\boldsymbol{\xi} = \{\xi^{(1)}, \xi^{(2)}, ..., \xi^{(L)}\}$, where each $\xi^{(i)}$ is specific to the i-th layer hidden output of $f(\cdot)$, formulated as:

$$H^{s,(l)} = \begin{cases} Conv^{(l)}(P^s, X^s) + \xi^{(l)} & l = 1 \\ Conv^{(l)}(P^s, H^{s,(l-1)}) + \xi^{(l)} & 1 < l \leq L \end{cases} \tag{3}$$

Then, we can obtain the shifted source node embeddings $H^{s,\xi}$ from the final output. We propose an adversarial transformation optimization objective on the source node embeddings and target node embeddings, which is defined as:

$$\begin{gathered} \max_{\theta, \boldsymbol{\xi}} \left\{ \min_{\phi_d} \left\{ \mathcal{L}_{AT}(H^{s,\xi}, H^t; \phi_d) \right\} \right\}, \\ s.t., ||\xi^{(l)}||_F \leq \epsilon, \forall \xi^{(l)} \in \boldsymbol{\xi}. \end{gathered} \tag{4}$$

The loss function $\mathcal{L}_{AT}$ is defined as:

$$\begin{aligned} \mathcal{L}_{AT} = & -\mathbb{E}_{h_i^{s,\xi} \sim \mathcal{H}^{s,\xi}} \left[log \left(\mathcal{D}_d(h_i^{s,\xi}, \phi_d) \right) \right] \\ & - \mathbb{E}_{h_j^t \sim \mathcal{H}^t} \left[log \left(1 - \mathcal{D}_d(h_j^t, \phi_d) \right) \right], \end{aligned} \tag{5}$$

where $h_i^{s,\xi}, h_i^t$ is the i-th row of $H^{s,\xi}, H^t$, respectively. $\mathcal{D}_d(h_i, \phi_d)$ with the parameters ϕ_d is a domain discriminator that learns a logistic regressor: $\mathcal{D}_d : \mathbb{R}^h \rightarrow \mathbb{R}^1$ to model the probability of the given the input node embedding h_i from the source graph or the target graph. The domain discriminator is trained to distinguish which domain the node embeddings are from, while the encoder with shift parameters is forced to generate the source node embeddings as indistinguishable as possible from target ones for the domain discriminator, thus resulting in domain-invariant node embeddings, We constrain the gradient of shift parameters in each training step within a certain radius ϵ to avoid excessive distribution shift, making the adversarial task impossible.

4.3 Pseudo-Labeling with Posterior Scores

In this module, we define the classifier on the node embeddings, e.g., $\mathcal{D}_c(h_i, \phi_c) : \mathbb{R}^h \rightarrow \mathbb{R}^C$ to model the label probability of nodes, which is a multi-layer perception followed with a softmax layer. Then, we can obtain the probability $p_i^s = \mathcal{D}_c(h_i^{s,\xi}, \phi_c)$ of each node in the source graph and the probability $p_j^t = \mathcal{D}_c(h_j^t, \phi_c)$ of each node in the target graph. We define the supervised loss function on the probabilities of nodes in the labeled node set $\mathcal{V}^{s,l}$ of the source graph:

$$\mathcal{L}_{Sup} = -\frac{1}{|\mathcal{V}^{s,l}|} \sum_{n_i \in \mathcal{V}^{s,l}} \sum_{k=1}^{C} y_{i,k} \log(p_{i,k}^s). \tag{6}$$

Since only a few nodes are labeled in the source graph whereas all nodes are unlabeled in the target graph, the model will be easily over-fitting if we only have Eq. 6. In particular, without any supervision, the nodes in the target graph distributed near the border and far away from the centroid of clusters of their corresponding classes are easily misclassified by the hyperplane learned from the label information of the source graph. Thus, we propose a novel pseudo-labeling strategy with posterior scores of nodes to improve the prediction accuracy on unlabeled nodes.

Specifically, in each training iteration, we update the pseudo-labels for the unlabeled nodes in both the source and target graph by mapping their output probabilities into one-hot encoding, denoted as $\widehat{y}_i = \mathcal{M}(p_i)$, where $\mathcal{M}(\cdot)$ is the one-hot map and p_i is the probability of node $n_i \in \mathcal{V}^{s,u} \cup \mathcal{V}^t$. Note that we treat all unlabeled nodes cross-domain in the same level and omit the notations of domain superscript for brevity. We assume that the nodes close to the structural centroid of their pseudo-label cluster on the graph are more likely classified correctly, while the pseudo-labels of those close to the cluster boundary are less reliable. Based on the hypothesis, we treat the pseudo-labels of former nodes as more high-quality self-supervised signals and aim to improve the discriminative ability of these node embeddings. Thus, we introduce a posterior score to define how n_i is close to the structural centroid of its pseudo label cluster on its reconstructed adjacency matrix P computed in Section 4.1:

$$w_i = \sum_{j=1}^{N} (P_{ij} * P_{\mathcal{C}_{\widehat{y}_i}, j} - \frac{1}{C-1} \sum_{k=1, k \neq \widehat{y}_i}^{C} P_{ij} * P_{\mathcal{C}_k, j}), \tag{7}$$

where $P_{\mathcal{C}_{\widehat{y}_i}, j} = \frac{1}{|\mathcal{C}_{\widehat{y}_i}|} \sum_{x \in \mathcal{C}_{\widehat{y}_i}} P_{x,j}$ indicates the overall mutual information from the nodes belonging to the class $\widehat{y}_i$ to the node n_j. The posterior score defines that if a node n_i with the pseudo label $\widehat{y}_i$ encounters high mutual information from other nodes in terms of the class $\widehat{y}_i$ and low ones from other nodes in terms of other classes, we have the conclusion that node n_i close to the centroid of class $\widehat{y}_i$ and w_i has a high value, and vice versa. Then, we apply a cosine annealing function [Chen *et al.*, 2021] to scale w_i into a certain range:

$$\widehat{w}_i = \alpha + \frac{1}{2}(\beta - \alpha)(1 + \cos(\frac{Rank(w_i)}{|\mathcal{V}|}\pi)), \tag{8}$$

where $\mathcal{V} \in \{\mathcal{V}^{s,u}, \mathcal{V}^t\}$ and $n_i \in \mathcal{V}$. $[\alpha, \beta]$ controls the scale range. $Rank(w_i)$ is the ranking order of w_i from the largest to the smallest. Then, we define a pseudo-labeling loss function with posterior scores as follows:

$$\mathcal{L}_{PL} = -\frac{1}{|\mathcal{V}|}\sum_{n_i \in \mathcal{V}} \widehat{w}_i \sum_{k=1}^{C} \widehat{y}_{i,k} \log(p_{i,k}) + \sum_{k=1}^{C} \widehat{p}_k \log \widehat{p}_k, \quad (9)$$

where $\widehat{p}_k = \mathbb{E}_{n_i \in \mathcal{V}}[p_{i,k}]$ and the second term is a diversity regularization to promote the diversity of output probabilities, which can circumvent the problem of some large posterior scores dominating in training to make all unlabeled nodes over-fit into the same pseudo-label. By $\mathcal{L}_{SL}$, the model is more encouraged to focus on the high-confidence nodes close to its corresponding cluster centroid and less influenced by those ambiguous nodes near the boundary, so as to improve the discriminative ability on unlabeled nodes.

4.4 Optimization

By combining the three losses above, the optimization of the proposed method SGDA is as follows:

$$\begin{aligned} &\min_{\theta, \boldsymbol{\xi}, \phi_c} \left\{ \mathcal{L}_{Sup} + \lambda_1 \mathcal{L}_{PL} + \lambda_2 \max_{\phi_d}\{-\mathcal{L}_{AT}\} \right\}, \\ &\qquad s.t., ||\xi^{(l)}||_F \le \epsilon, \forall \xi^{(l)} \in \boldsymbol{\xi}. \end{aligned} \quad (10)$$

where λ_1 and λ_2 is the weights to balance different losses. In practice, we introduce a gradient reversal layer (GRL) [Ganin *et al.*, 2016] between the graph encoder and the domain discriminator so as to conveniently perform min-max optimization under $\mathcal{L}_{AT}$ in one training step. The GRL acts as an identity transformation during the forward propagation and changes the signs of the gradient from the subsequent networks during the backpropagation. Particularly, for each $\xi^{(l)} \in \boldsymbol{\xi}$, the update rule is defined as follow:

$$\begin{aligned} &g(\xi^{(l)}) = \left(\frac{\partial \mathcal{L}_{Sup}}{\partial \xi^{(l)}} + \lambda_1 \frac{\partial \mathcal{L}_{PL}}{\partial \xi^{(l)}} - \lambda_2 \frac{\partial \mathcal{L}_{AT}}{\partial \xi^{(l)}} \right), \\ &\xi^{(l)} \leftarrow \xi^{(l)} + \mu g(\xi^{(l)}) / ||g(\xi^{(l)})||_F. \end{aligned} \quad (11)$$

where μ is the learning rate. The shift parameters $\boldsymbol{\xi}$ are optimized by Projected Gradient Descent (PGD). Following [Yang *et al.*, 2021; Kong *et al.*, 2020], we use the unbounded adversarial transformation as one is not aware of the shifting scale in advance.

5 Experiments

5.1 Dataset

We conduct experiments on three real-world graphs: *ACMv9* (A), *Citationv1* (C), and *DBLPv7* (D) from ArnetMiner [Tang *et al.*, 2008]. These graphs are constructed from three different source datasets in different periods, i.e., Association for Computer Machinery (after the year 2010), Microsoft Academic Graph (before the year 2008), and DBLP Computer Science Bibliography (between years 2004 and 2008), respectively so that they have varied distributions in their domain spaces. Each node in these graphs represents a paper whose attribute is the sparse bag-of-words vector of the paper's title. The edges represent a citation relationship between these papers, where the direction is ignored. As these graphs do not share the same feature set of node attributes, we union their attribute set and reshape the attribute dimension as 6775. Each node is assigned a five-classes label based on its relevant research areas, including *Artificial Intelligence*, *Computer Version*, *Database*, *Information Security*, and *Networking*. Table 1 presents the statistics of graph scale, attributes, average degree, and label proportion, indicating the intrinsic discrepancy between the three graphs. In our paper, we alternately select one of these graphs as the source domain and the rest two as the target domain.

Dataset	#Nodes	#Edges	#Attr.	Avg. Degree	Label Proportion (%)
ACMv9	9,360	15,602	5,571	1.667	20.5/29.6/22.5/8.6/18.8
Citationv1	8,935	15,113	5,379	1.691	25.3/26.0/22.5/7.7/18.5
DBLPv7	5,484	8,130	4,412	1.482	21.7/33.0/23.8/6.0/15.5

Table 1: The statistics of three graphs. '#' means 'the number of'. 'Attr.' means 'Attributes'. 'Avg.' means 'Average'.

5.2 Baselines

We select two groups of baseline methods. The first group is traditional solutions for graph semi-supervised learning, which learns a node classification model on the source graph and directly uses the model to perform inductive prediction on the target graph without explicit transfer learning. We first use Multi-Layer Perceptron (*MLP*), which is directly trained on the attributes of nodes in the source graph. We choose four GNN variants, including GCN [Welling and Kipf, 2016], GraphSAGE (*GSAGE*) [Hamilton *et al.*, 2017], GAT [Veličković *et al.*, 2018], and GIN [Xu *et al.*, 2018], which are acknowledged as state-of-the-art models for graph semi-supervised learning. The second group is specific to domain adaptation. We first choose two general approaches DANN [Ganin *et al.*, 2016] and CDAN [Ganin *et al.*, 2016], which are initially designed for transfer learning on images or text. We train them on node attributes. For graph semi-supervised learning, we create their variants DANN$_{GCN}$ and CDAN$_{GCN}$ by replacing the encoder from MLP to GCN and training them on graphs. Finally, we select two methods most similar to us, UDA-GCN [Wu *et al.*, 2020a] and AdaGCN [Dai *et al.*, 2022], which are also designed for semi-supervised domain adaptation on graphs.

5.3 Experimental Setting

We choose a two-layer GCN as the backbone of SGDA. We set the loss wrights λ_1 always as 1 and $\lambda_2 \in [0, 1]$ as a dynamic value that is linearly increased with the training epoch, i.e., $\lambda_2 = m/M$ where m is the current epoch and M is the maximum epoch. We consider that at the early training steps, the classifier is not completely converged, making the pseudo labels produced by the classifier inferior for self-supervised learning. We randomly initialize the $\boldsymbol{\xi}$ under the uniform distribution $U(-\epsilon, \epsilon)$ and set the ϵ always as 0.5. We set the scale range α and β always as 0.8 and 1.2. We train SGDA for 200 epochs with the learning rate as 0.001, the weight decay as

Methods	A⇒C		A⇒D		C⇒A		C⇒D		D⇒A		D⇒C	
	Micro	Macro	Micro	Macro	Micro	Macro	Micro	Macro	Micro	Macro	Micro	Macro
MLP	$41.3_{\pm 1.15}$	$35.8_{\pm 0.72}$	$42.8_{\pm 0.88}$	$36.3_{\pm 0.77}$	$39.4_{\pm 0.57}$	$33.7_{\pm 0.58}$	$43.7_{\pm 0.69}$	$36.7_{\pm 0.55}$	$37.3_{\pm 0.32}$	$30.8_{\pm 0.37}$	$39.4_{\pm 0.99}$	$32.8_{\pm 0.99}$
GCN	$54.4_{\pm 1.52}$	$52.0_{\pm 1.62}$	$56.9_{\pm 2.33}$	$53.4_{\pm 2.81}$	$54.1_{\pm 1.40}$	$52.3_{\pm 1.98}$	$58.9_{\pm 0.99}$	$54.5_{\pm 1.55}$	$50.1_{\pm 2.14}$	$48.0_{\pm 3.28}$	$56.0_{\pm 1.24}$	$51.9_{\pm 1.49}$
GSAGE	$49.3_{\pm 2.18}$	$46.4_{\pm 2.06}$	$51.8_{\pm 1.35}$	$47.4_{\pm 1.62}$	$46.8_{\pm 2.56}$	$45.0_{\pm 2.78}$	$51.7_{\pm 1.95}$	$48.1_{\pm 1.97}$	$41.7_{\pm 2.17}$	$37.4_{\pm 4.59}$	$45.4_{\pm 2.11}$	$39.3_{\pm 3.45}$
GAT	$55.1_{\pm 3.22}$	$50.8_{\pm 1.45}$	$55.3_{\pm 2.52}$	$51.8_{\pm 2.60}$	$50.0_{\pm 1.20}$	$45.6_{\pm 2.36}$	$55.4_{\pm 2.73}$	$49.2_{\pm 2.59}$	$44.8_{\pm 2.74}$	$38.3_{\pm 4.84}$	$50.4_{\pm 3.35}$	$42.0_{\pm 4.46}$
GIN	$64.6_{\pm 2.47}$	$56.0_{\pm 2.73}$	$60.0_{\pm 2.09}$	$51.3_{\pm 3.99}$	$57.1_{\pm 1.19}$	$54.4_{\pm 2.57}$	$62.0_{\pm 1.05}$	$56.8_{\pm 1.40}$	$51.9_{\pm 2.00}$	$45.4_{\pm 2.16}$	$60.2_{\pm 3.05}$	$53.0_{\pm 2.10}$
DANN	$44.3_{\pm 2.03}$	$39.3_{\pm 1.86}$	$44.0_{\pm 1.42}$	$38.7_{\pm 1.47}$	$41.8_{\pm 1.95}$	$37.6_{\pm 1.24}$	$45.5_{\pm 0.71}$	$39.6_{\pm 1.55}$	$37.8_{\pm 3.66}$	$33.2_{\pm 2.23}$	$41.7_{\pm 2.32}$	$35.6_{\pm 2.55}$
CDAN	$44.6_{\pm 1.30}$	$38.6_{\pm 1.07}$	$45.5_{\pm 0.85}$	$38.0_{\pm 0.86}$	$42.4_{\pm 0.64}$	$36.2_{\pm 1.17}$	$46.7_{\pm 1.17}$	$39.2_{\pm 0.96}$	$39.0_{\pm 1.08}$	$32.3_{\pm 1.09}$	$41.7_{\pm 1.55}$	$34.8_{\pm 1.56}$
DANN$_{GCN}$	$63.0_{\pm 6.75}$	$59.6_{\pm 6.02}$	$62.2_{\pm 1.90}$	$57.7_{\pm 3.16}$	$56.7_{\pm 0.38}$	$55.2_{\pm 1.03}$	$65.3_{\pm 2.04}$	$59.0_{\pm 2.39}$	$52.3_{\pm 2.59}$	$48.6_{\pm 4.52}$	$58.1_{\pm 2.78}$	$52.4_{\pm 3.81}$
CDAN$_{GCN}$	$70.3_{\pm 0.84}$	$66.5_{\pm 0.66}$	$65.0_{\pm 1.00}$	$61.3_{\pm 0.96}$	$56.3_{\pm 1.78}$	$53.6_{\pm 2.70}$	$65.2_{\pm 2.19}$	$58.8_{\pm 2.38}$	$53.0_{\pm 1.34}$	$48.7_{\pm 3.51}$	$59.0_{\pm 1.52}$	$53.3_{\pm 1.99}$
UDA-GCN	$72.4_{\pm 2.75}$	$65.2_{\pm 6.51}$	$68.0_{\pm 6.38}$	$64.3_{\pm 7.12}$	$62.9_{\pm 0.33}$	$62.2_{\pm 1.44}$	$71.4_{\pm 2.56}$	$67.5_{\pm 2.25}$	$55.8_{\pm 3.50}$	$52.4_{\pm 2.68}$	$65.2_{\pm 4.41}$	$60.7_{\pm 6.84}$
AdaGCN	$70.8_{\pm 0.95}$	$68.5_{\pm 0.73}$	$68.2_{\pm 3.84}$	$64.2_{\pm 3.91}$	$61.5_{\pm 2.20}$	$60.4_{\pm 3.15}$	$69.1_{\pm 1.96}$	$65.8_{\pm 2.87}$	$56.1_{\pm 1.75}$	$53.8_{\pm 2.95}$	$64.1_{\pm 0.91}$	$62.8_{\pm 1.56}$
SGDA	$\mathbf{75.6_{\pm 0.57}}$	$\mathbf{71.4_{\pm 0.82}}$	$\mathbf{69.2_{\pm 0.73}}$	$\mathbf{64.7_{\pm 2.36}}$	$\mathbf{66.3_{\pm 0.68}}$	$\mathbf{62.3_{\pm 0.96}}$	$\mathbf{72.9_{\pm 1.26}}$	$\mathbf{68.9_{\pm 1.83}}$	$\mathbf{60.6_{\pm 0.86}}$	$\mathbf{56.0_{\pm 0.90}}$	$\mathbf{73.2_{\pm 0.59}}$	$\mathbf{69.3_{\pm 1.01}}$

Table 2: The model performance comparison on six domain adaptation tasks with source label rate as 5%. A: ACMv9; C:Citationv1; D: DBLPv7. A⇒C represents that A is the source graph and C is the target graph. The same applies to other tasks.

0.001, and the dropout rate as 0.1 on all datasets. For baselines, we implement DANN, CDAN, AdaGCN, and UDA-GCN by their own codes and report the best results. The dimension of node embeddings is as 512 for all approaches.

5.4 Performance Comparison

This experiment aims to answer: *How is SGDA performance on the semi-supervised domain adaptation task on graphs?* We randomly select 5% of nodes in the source graph as labeled nodes and others as unlabeled nodes while the target graph is completely unlabeled. We use Micro-F1 and Macro-F1 as the metric and report the classification results of different approaches on the target graph in Table 2. Notably, we ran each experiment 5 times, and each time we sampled different label sets to alleviate the randomness. The average results with standard deviation are reported.

From the results, we can observe that: Firstly, SGDA consistently achieved the best results on six transfer learning tasks. Particularly, SGDA achieved a significant improvement compared with two methods of semi-supervised domain adaption across graphs—AdaGCN and UDA-GCN, indicating the effectiveness of SGDA in solving this problem. Another observation is that the performance of MLP is worse than GNNs; two general domain adaptation methods DANN and CDAN are worse than their two variants DANN$_{GCN}$ and CDAN$_{GCN}$, indicating incorporating graph topology information is critical. Thirdly, the domain adaptation methods generally performed better than the inductive learning methods. That proves that it is necessary to perform domain transformation to eliminate the cross-domain distribution discrepancy in the graph transfer learning task. Lastly, as the domain adaptation methods with the GNN-based encoders, AdaGCN and UDA-GCN achieved better results than DANN$_{GCN}$ and CDAN$_{GCN}$. That is because they use more complex graph encoders and improved optimization objectives. Still, they are worse than SGDA due to the shortages of domain transformation and incompetence in handling the label scarcity.

5.5 Ablation Study

This experiment aims to answer: *Are all the proposed technologies of SGDA have the claimed contribution to the semi-supervised domain adaptation on graphs?* For that, we design four variant methods for SGDA to verify the effectiveness of node embedding generalization, adversarial transformation via shifting, and pseudo-labeling: ***w/o NEG***: we directly use the original graph adjacency matrix rather than reconstruct the adjacency. ***w/o Shift***: we remove the shift parameters added on the source graph and only use the graph encoder to learn domain-invariant node embeddings. ***w/o AT***: we remove the loss $\mathcal{L}_{AT}$ so the transformation of the source graph to the target graph is deactivated. ***w/o PL***: we remove the loss $\mathcal{L}_{PL}$ so pseudo-labeling on unlabeled nodes is deactivated. Figure 3 reported the performance of these variants.

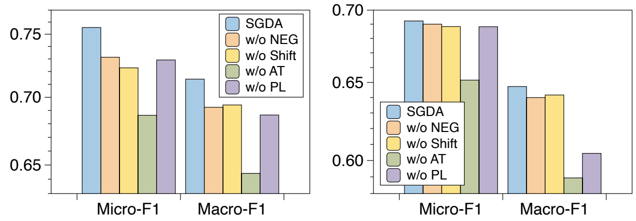
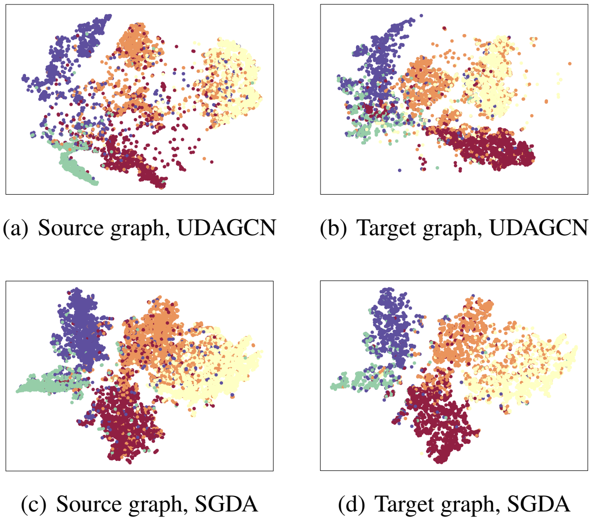


Figure 3: The results of ablation study on the A ⇒ C task (left) and the A ⇒ D task (right).

Effect of node embedding generalization.*w/o NEG* performs worse than SGDA. The reason is that without exploiting high-order graph topology information, the node embeddings only incorporated local neighborhood information are not generalized enough to perform the transformation.

Effect of shift parameters. *w/o Shift* performs worse than SGDA. The reason is that the shift parameters in $\mathcal{L}_{AT}$ can facilitate the transfer ability of the model. Only utilizing the graph encoder is inefficient to shift distributions finely. Apart from this, we can observe that *w/o Shift* is still superior to other SOTA domain adaptation methods. This phenomenon shows the limitation of those baselines on an incomplete labeled source graph and proves the advantage of $\mathcal{L}_{PL}$.

Effect of adversarial transformation. *w/o AT* without any cross-domain distribution alignment performs worse but still achieves considerable performance compared with baselines in Table 2. That is because our proposed pseudo-labeling on the target graph can adaptively pay more attention to those nodes close to the class centroid, who are more convinced to be correctly classified, and thus can constantly optimize the



(a) Source graph, UDAGCN (b) Target graph, UDAGCN

(c) Source graph, SGDA (d) Target graph, SGDA

Figure 4: Visualization of node embeddings learned by UDAGCN and SGDA on the A ⇒ C task.

classifier to have better discrimination on the target graph.

Effect of pseudo-labeling. *w/o PL* performs worse because, without $\mathcal{L}_{PL}$, the model is easier to get over-fitting on the source graph under limited labels. Additionally, the model may learn similar distributions between the source and target graph, but their class-level distributions may be inconsistent. The proposed pseudo-labeling can utilize the unlabeled graph topology to find better label space for nodes adaptively.

5.6 Visualization of Distributions

This experiment aims to answer that: *How is the distribution of SGDA-generated node embeddings compared to the SOTA domain adaptation method?* To illustrate the difference, we obtained the source and target node embeddings learned from SGDA and UDA-GCN with the same 5% label rate setting. Then we separately projected them in 2-D by t-SNE and visualized them in Figure 4. We colored each node by its class. The first observation is that the distribution learned by SGDA could generate more generalized node embeddings, showing that nodes are dispersedly distributed in the space, which is contributed by preserving high-order topology information via random walks. Also, the distributions of source space and target space learned by SGDA are clearly more consistent with each other, proving it can well eliminate the cross-domain discrepancy and learn domain-invariant node embeddings. Lastly, SGDA can significantly separate each class of nodes in both the source and the target graph. On the contrary, UDA-GCN can hardly differentiate each group of nodes, which is more apparent in the target space. That is because SGDA can well handle label scarcity and train a more discriminative classifier on the target nodes.

5.7 Hyper-Paramter Experiment

Effect of Label Rate. This experiment aims to answer: *Is SGDA robust with different ratios of labeled data on the source graph?* We evaluate the performance of different

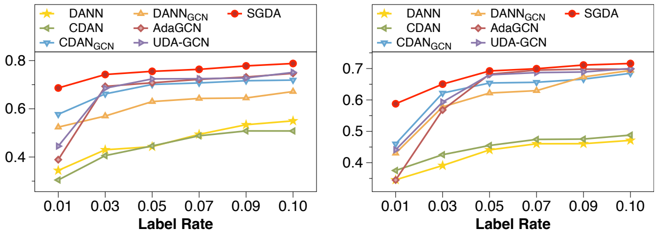


Figure 5: The model performance with different label rates on the A ⇒ C task (left) and the A ⇒ D task (right).

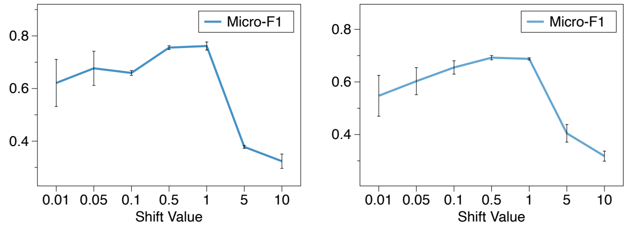


Figure 6: The model performance with different shift values on the A ⇒ C task (left) and the A ⇒ D task (right).

methods with the label rate of the source graph as 1%, 5%, 7%, 9%, and 10%, respectively. The results are reported in Figure 5. The first observation is that SGDA has a remarkable margin compared with other selected baselines, even with only 1% labeled data in the source graph. This shows the proposed pseudo-labeling can significantly handle the label scarcity problem. Also, the GNN-based methods have a great leap compared with NN-based approaches, proving the great potential of utilizing unlabeled graph topology information in improving the model's robustness under limited labels.

Effect of Shift Value. This experiment aims to answer: *How do different shift values affect the performance of SGDA?* The constraint shift value ϵ of the shift parameters is significant to control the scale of distribution shifting. We evaluated SGDA with ϵ as 0.01, 0.05, 0.1, 0.5, 1. 5, and 10, respectively and report the results in Figure 6. We can observe that with low shift values, the model's performance is less robust, showing high standard deviations. Within $\epsilon \in [0.1, 1]$, shift parameters have more impact in training, so SGDA can achieve relatively high and stable results. However, when ϵ is large, the adversarial learning becomes difficult, thus dampening the results.

6 Conclusion

This work presents a novel research problem of semi-supervised domain adaptation on graphs. We propose a method called SGDA that uses shift parameters and adversarial learning to achieve model transferring. Also, SGDA uses pseudo labels with adaptive posterior scores to alleviate the label scarity. Extensive experiments on a variety of publicly available datasets demonstrate the efficacy of SGDA. In future work, we will expand our SGDA to a variety of graph transfer learning tasks including source-free domain adaptation and out-of-domain generalization on graphs.

Acknowledgements

This work is supported in part by Foshan HKUST Projects (FSUST21-FYTRI01A, FSUST21-FYTRI02A) and the Natural Science Foundation of China under Grant No. 61836013.

Contribution Statement

In this work, Ziyue Qiao, Xiao Luo, and, Meng Xiao contributed equally. Specifically, Ziyue Qiao and Xiao Luo contributed to the problem formulation, methodology, model implementation, and paper writing. Meng Xiao contributed to the conduction, visualization, and writing of experiments. Hao Dong assisted in the paper writing. Yuanchun Zhou and Hui Xiong provided valuable feedback on the paper drafts. All authors reviewed and approved the final manuscript.

References

[Chang *et al.*, 2021] Ji Chang, Jing Li, Yu Kang, Wenjun Lv, Ting Xu, Zerui Li, Wei Xing Zheng, Hongwei Han, and Haining Liu. Unsupervised domain adaptation using maximum mean discrepancy optimization for lithology identification. *Geophysics*, 86(2):ID19–ID30, 2021.

[Chen *et al.*, 2021] Deli Chen, Yankai Lin, Guangxiang Zhao, Xuancheng Ren, Peng Li, Jie Zhou, and Xu Sun. Topology-imbalance learning for semi-supervised node classification. *Advances in Neural Information Processing Systems*, 34:29885–29897, 2021.

[Dai *et al.*, 2022] Quanyu Dai, Xiao-Ming Wu, Jiaren Xiao, Xiao Shen, and Dan Wang. Graph transfer learning via adversarial domain adaptation with graph convolution. *IEEE Transactions on Knowledge and Data Engineering*, 2022.

[Fan *et al.*, 2019] Wenqi Fan, Yao Ma, Qing Li, Yuan He, Eric Zhao, Jiliang Tang, and Dawei Yin. Graph neural networks for social recommendation. In *The world wide web conference*, pages 417–426, 2019.

[Ganin *et al.*, 2016] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The journal of machine learning research*, 17(1):2096–2030, 2016.

[Guo *et al.*, 2022] Gaoyang Guo, Chaokun Wang, Bencheng Yan, Yunkai Lou, Hao Feng, Junchao Zhu, Jun Chen, Fei He, and Philip Yu. Learning adaptive node embeddings across graphs. *IEEE Transactions on Knowledge and Data Engineering*, 2022.

[Hamilton *et al.*, 2017] William L Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 1025–1035, 2017.

[Hu *et al.*, 2020] Ziniu Hu, Yuxiao Dong, Kuansan Wang, Kai-Wei Chang, and Yizhou Sun. Gpt-gnn: Generative pre-training of graph neural networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1857–1867, 2020.

[Ingraham *et al.*, 2019] John Ingraham, Vikas Garg, Regina Barzilay, and Tommi Jaakkola. Generative models for graph-based protein design. *Advances in neural information processing systems*, 32, 2019.

[Jiang *et al.*, 2020] Pin Jiang, Aming Wu, Yahong Han, Yunfeng Shao, Meiyu Qi, and Bingshuai Li. Bidirectional adversarial training for semi-supervised domain adaptation. In *IJCAI*, pages 934–940, 2020.

[Jin *et al.*, 2021] Wei Jin, Yaxing Li, Han Xu, Yiqi Wang, Shuiwang Ji, Charu Aggarwal, and Jiliang Tang. Adversarial attacks and defenses on graphs. *ACM SIGKDD Explorations Newsletter*, 22(2):19–34, 2021.

[Ju *et al.*, 2023] Wei Ju, Zheng Fang, Yiyang Gu, Zequn Liu, Qingqing Long, Ziyue Qiao, Yifang Qin, Jianhao Shen, Fang Sun, Zhiping Xiao, et al. A comprehensive survey on deep graph representation learning. *arXiv preprint arXiv:2304.05055*, 2023.

[Kong *et al.*, 2019] Xiangjie Kong, Yajie Shi, Shuo Yu, Jiaying Liu, and Feng Xia. Academic social networks: Modeling, analysis, mining and applications. *Journal of Network and Computer Applications*, 132:86–103, 2019.

[Kong *et al.*, 2020] Kezhi Kong, Guohao Li, Mucong Ding, Zuxuan Wu, Chen Zhu, Bernard Ghanem, Gavin Taylor, and Tom Goldstein. Flag: Adversarial data augmentation for graph neural networks. *arXiv preprint arXiv:2010.09891*, 2020.

[Li *et al.*, 2020] Mengxue Li, Yi-Ming Zhai, You-Wei Luo, Peng-Fei Ge, and Chuan-Xian Ren. Enhanced transport distance for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13936–13944, 2020.

[Qiao *et al.*, 2022] Ziyue Qiao, Yanjie Fu, Pengyang Wang, Meng Xiao, Zhiyuan Ning, Denghui Zhang, Yi Du, and Yuanchun Zhou. Rpt: toward transferable model on heterogeneous researcher data via pre-training. *IEEE Transactions on Big Data*, 9(1):186–199, 2022.

[Qiao *et al.*, 2023] Ziyue Qiao, Pengyang Wang, Pengfei Wang, Zhiyuan Ning, Yanjie Fu, Yi Du, Yuanchun Zhou, Jianqiang Huang, Xian-Sheng Hua, and Hui Xiong. A dual-channel semi-supervised learning framework on graphs via knowledge transfer and meta-learning. *ACM Transactions on the Web*, 2023.

[Qiu *et al.*, 2020] Jiezhong Qiu, Qibin Chen, Yuxiao Dong, Jing Zhang, Hongxia Yang, Ming Ding, Kuansan Wang, and Jie Tang. Gcc: Graph contrastive coding for graph neural network pre-training. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1150–1160, 2020.

[Shen *et al.*, 2020a] Xiao Shen, Quanyu Dai, Fu-lai Chung, Wei Lu, and Kup-Sze Choi. Adversarial deep network embedding for cross-network node classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 2991–2999, 2020.

[Shen *et al.*, 2020b] Xiao Shen, Quanyu Dai, Sitong Mao, Fu-lai Chung, and Kup-Sze Choi. Network together: Node classification via cross-network deep network embedding. *IEEE Transactions on Neural Networks and Learning Systems*, 32(5):1935–1948, 2020.

[Tang *et al.*, 2008] Jie Tang, Jing Zhang, Limin Yao, Juanzi Li, Li Zhang, and Zhong Su. Arnetminer: extraction and mining of academic social networks. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 990–998, 2008.

[Tang *et al.*, 2021] Zhengzheng Tang, Ziyue Qiao, Xuehai Hong, Yang Wang, Fayaz Ali Dharejo, Yuanchun Zhou, and Yi Du. Data augmentation for graph convolutional network on semi-supervised classification. In *Web and Big Data: 5th International Joint Conference, APWeb-WAIM 2021, Guangzhou, China, August 23–25, 2021, Proceedings, Part II 5*, pages 33–48. Springer, 2021.

[Tzeng *et al.*, 2017] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7167–7176, 2017.

[Veličković *et al.*, 2018] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018.

[Volpi *et al.*, 2018] Riccardo Volpi, Pietro Morerio, Silvio Savarese, and Vittorio Murino. Adversarial feature augmentation for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5495–5504, 2018.

[Wang *et al.*, 2018] Bo Wang, Armin Pourshafeie, Marinka Zitnik, Junjie Zhu, Carlos D Bustamante, Serafim Batzoglou, and Jure Leskovec. Network enhancement as a general method to denoise weighted biological networks. *Nature communications*, 9(1):1–8, 2018.

[Wang *et al.*, 2020] Yiwei Wang, Wei Wang, Yuxuan Liang, Yujun Cai, Juncheng Liu, and Bryan Hooi. Nodeaug: Semi-supervised node classification with data augmentation. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 207–217, 2020.

[Welling and Kipf, 2016] Max Welling and Thomas N Kipf. Semi-supervised classification with graph convolutional networks. In *J. International Conference on Learning Representations (ICLR 2017)*, 2016.

[Wen *et al.*, 2022] Qianlong Wen, Zhongyu Ouyang, Chunhui Zhang, Yiyue Qian, Yanfang Ye, and Chuxu Zhang. Adversarial cross-view disentangled graph contrastive learning. *arXiv preprint arXiv:2209.07699*, 2022.

[Wu *et al.*, 2020a] Man Wu, Shirui Pan, Chuan Zhou, Xiaojun Chang, and Xingquan Zhu. Unsupervised domain adaptive graph convolutional networks. In *Proceedings of The Web Conference 2020*, pages 1457–1467, 2020.

[Wu *et al.*, 2020b] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, 2020.

[Xhonneux *et al.*, 2020] Louis-Pascal Xhonneux, Meng Qu, and Jian Tang. Continuous graph neural networks. In *International Conference on Machine Learning*, pages 10432–10441. PMLR, 2020.

[Xiao *et al.*, 2022] Jiaren Xiao, Quanyu Dai, Xiaochen Xie, Qi Dou, Ka-Wai Kwok, and James Lam. Domain adaptive graph infomax via conditional adversarial networks. *IEEE Transactions on Network Science and Engineering*, 2022.

[Xu *et al.*, 2018] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *International Conference on Learning Representations*, 2018.

[Xu *et al.*, 2022] Jiarong Xu, Yang Yang, Junru Chen, Xin Jiang, Chunping Wang, Jiangang Lu, and Yizhou Sun. Unsupervised adversarially robust representation learning on graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 4290–4298, 2022.

[Yan *et al.*, 2022] Zizheng Yan, Yushuang Wu, Guanbin Li, Yipeng Qin, Xiaoguang Han, and Shuguang Cui. Multi-level consistency learning for semi-supervised domain adaptation. In Lud De Raedt, editor, *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pages 1530–1536. International Joint Conferences on Artificial Intelligence Organization, 7 2022. Main Track.

[Yang *et al.*, 2021] Longqi Yang, Liangliang Zhang, and Wenjing Yang. Graph adversarial self-supervised learning. *Advances in Neural Information Processing Systems*, 34:14887–14899, 2021.

[Zhang and Davison, 2021] Youshan Zhang and Brian D Davison. Deep spherical manifold gaussian kernel for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4443–4452, 2021.

[Zhang *et al.*, 2018] Weichen Zhang, Wanli Ouyang, Wen Li, and Dong Xu. Collaborative and adversarial network for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3801–3809, 2018.

[Zhang *et al.*, 2021] Xiaowen Zhang, Yuntao Du, Rongbiao Xie, and Chongjun Wang. Adversarial separation network for cross-network node classification. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 2618–2626, 2021.

[Zhang *et al.*, 2022] Wenyu Zhang, Li Shen, Wanyue Zhang, and Chuan-Sheng Foo. Few-shot adaptation of pre-trained networks for domain shift. In Lud De Raedt, editor, *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pages 1665–1671. International Joint Conferences on Artificial Intelligence Organization, 7 2022. Main Track.

[Zhuang and Ma, 2018] Chenyi Zhuang and Qiang Ma. Dual graph convolutional networks for graph-based semi-supervised classification. In *Proceedings of the 2018 World Wide Web Conference*, pages 499–508, 2018.