Semi-supervised Domain Adaptation in Graph Transfer Learning

Ziyue Qiao\textsuperscript{1,2,3}, Xiaoluopku, Meng Xiao\textsuperscript{4,5}, Hao Dong\textsuperscript{4,5}, Yuanchun Zhou\textsuperscript{4,5} and Hui Xiong\textsuperscript{2,3,†}
\textsuperscript{1}Jiangmen Laboratory of Carbon Science and Technology, Jiangmen
\textsuperscript{2}The Hong Kong University of Science and Technology (Guangzhou), Guangzhou
\textsuperscript{3}Guangzhou HKUST Fok Ying Tung Research Institute, Guangzhou
\textsuperscript{4}Computer Network Information Center, Chinese Academy of Sciences, Beijing
\textsuperscript{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 Semi-supervised Graph Domain Adaptation (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.

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 Semi-supervised Graph Domain Adaptation 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 adaptation 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 problems. 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.
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:

$$
P_{ij} = \frac{F_{ij}}{\sum_{i,j} F_{ij}}, \quad P_i = \sum_j F_{ij}, \quad P_j = \sum_i F_{ij}, \quad P_{ij} = \max\{\log(P_{ij}/P_i \times P_j), 0\},$$

where $P_{ij}$ is the probability of the node $n_j$ occurring in the context of the node $n_i$. $P_i$ and $P_j$ 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., $P_{ij} > P_i \times P_j$. 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:

$$H^{(l)} = Conv_l(P, H^{(l-1)}),$$

$$= \sigma(D^{-\frac{1}{2}} \tilde{P} D^{-\frac{1}{2}} H^{(l-1)} W^{(l)}),$$

where $\sigma(\cdot)$ denotes an activation function. $\tilde{P} = P + I$ where $I$ is the identity matrix and $D$ is the diagonal degree matrix of $P$ (i.e., $D_{ii} = \sum_j \tilde{P}_{ij}$). $W^{(l)}$ is the $l$-th layer weight matrix. $H^{(0)}$ 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 $\theta$ 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 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.

![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 $\xi$ to promote distribution alignment. Three losses $L_{Sup}$, $L_{AT}$, and $L_{PL}$ perform supervised learning, domain adversarial transformation via shifting, and pseudo-labeling with posterior scores, respectively.](image-url)
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 $\xi$ onto the source graph and obtain the shifted source node embeddings $H^{s, \xi}$ 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, $\xi = \{\xi^{(1)}, \xi^{(2)}, ..., \xi^{(L)}\}$, where each $\xi^{(i)}$ is specific to the $i$-th layer hidden output of $f(\cdot)$, formulated as:

$$H^{s, (i)} = \begin{cases} 
    \text{Conv}^{(i)}(P^s, X^s) + \xi^{(i)} & l = 1 \\
    \text{Conv}^{(i)}(P^s, H^{s, (l-1)}) + \xi^{(i)} & 1 < l \leq L
\end{cases}$$  \hspace{1cm} (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:

$$\max_{\theta, \xi} \left\{ \min_{\phi_\delta} \left\{ \mathcal{L}_{AT}(H^{s, \xi}, H^t; \phi_\delta) \right\} \right\}, \hspace{1cm} \text{s.t.,} \|\xi^{(l)}\|_F \leq \epsilon, \forall \xi^{(l)} \in \xi$$  \hspace{1cm} (4)

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

$$\mathcal{L}_{AT} = -\mathbb{E}_{h^{s, \xi} \sim \mathcal{H}^{s, \xi}} \left[ \log \left( \mathcal{D}_d(h^{s, \xi}, \phi_\delta) \right) \right] - \mathbb{E}_{h^t \sim \mathcal{H}^t} \left[ \log (1 - \mathcal{D}_d(h^t, \phi_\delta)) \right], \hspace{1cm} (5)$$

where $h^{s, \xi}, h^t$ is the $i$-th row of $H^{s, \xi}, H^t$, respectively. $\mathcal{D}_d(h_i, \phi_\delta)$ with the parameters $\phi_\delta$ is a domain discriminator that learns a logistic regressor: $\mathcal{D}_d: \mathbb{R}^h \rightarrow \mathbb{R}$ 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 $\epsilon$ 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 $p^s_i = \mathcal{D}_c(h^{s, \xi}_i, \phi_c)$ of each node in the source graph and the probability $p^t_i = \mathcal{D}_c(h^t_i, \phi_c)$ 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:

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

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 $\tilde{y}_i = M(p_i)$, where $M(\cdot)$ 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 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:

$$w_i = \sum_{j=1}^N (P_{ij} * P_{\tilde{y}_j}, \tilde{y}_j) - \frac{1}{C-1} \sum_{k=1}^C \sum_{k \neq \tilde{y}_i} P_{ij} * P_{\tilde{v}_k,j}, \hspace{1cm} (7)$$

where $P_{\tilde{y}_j, \tilde{y}_j} = \frac{1}{|\mathcal{V}^t|} \sum_{x \in \tilde{y}_j} P_{x,j}$ indicates the overall mutual information from the nodes belonging to the class $\tilde{y}_j$ to the node $n_j$. The posterior score defines that if a node $n_i$ with the pseudo label $\tilde{y}_i$ encounters high mutual information from other nodes in terms of the class $\tilde{y}_i$ 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 $\tilde{y}_i$ 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:

$$\bar{w}_i = \alpha + \frac{1}{2}(\beta - \alpha)(1 + \cos(\frac{\text{Rank}(w_i)}{|V| - \pi})), \hspace{1cm} (8)$$
where \( V \in \{ V^{s,u}, V^{t} \} \) and \( n_{t} \in V \). \([\alpha, \beta]\) controls the scale range. \( \text{Rank}(w_{t}) \) is the ranking order of \( w_{t} \) from the largest to the smallest. Then, we define a pseudo-labeling loss function with posterior scores as follows:

\[
\mathcal{L}_{PL} = -\frac{1}{|V|} \sum_{i \in V} \tilde{w}_{i} \sum_{k=1}^{C} \tilde{y}_{i,k} \log(p_{i,k}) + \sum_{k=1}^{C} \tilde{p}_{k} \log \tilde{p}_{k},
\]

where \( \tilde{p}_{k} = \mathbb{E}_{n_{t} \in V}[p_{i,k}] \) 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:

\[
\min_{\theta, \phi, \xi} \left\{ \mathcal{L}_{Sup} + \lambda_{1} \mathcal{L}_{PL} + \lambda_{2} \max_{\phi_{x}} \{-\mathcal{L}_{AT}\} \right\},
\]

\[
\text{s.t.}, \quad ||\xi^{(i)}||_{F} \leq \epsilon, \forall \xi^{(i)} \in \xi,
\]

where \( \lambda_{1} \) and \( \lambda_{2} \) are 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 \( \xi^{(i)} \in \xi \), the update rule is defined as follow:

\[
g(\xi^{(i)}) = \left( \frac{\partial \mathcal{L}_{Sup}}{\partial \xi^{(i)}} + \lambda_{1} \frac{\partial \mathcal{L}_{PL}}{\partial \xi^{(i)}} - \lambda_{2} \frac{\partial \mathcal{L}_{AT}}{\partial \xi^{(i)}} \right),
\]

\[
\xi^{(i)} \leftarrow \xi^{(i)} + \mu g(\xi^{(i)})/||g(\xi^{(i)})||_{F}.
\]

where \( \mu \) is the learning rate. The shift parameters \( \xi \) 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.

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 DANNGCN and CDANGCN 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 weights \( \lambda_{1} \) 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 \( \xi \) under the uniform distribution \( U(\epsilon, \epsilon) \) and set the \( \epsilon \) always as 0.5. We set the scale range \( \alpha \) and \( \beta \) always as 0.8 and 1.2. We train SGDA for 200 epochs with the learning rate as 0.001, the weight decay as
Are all the proposed techniques 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 that 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.

**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 performance.
classifier to have better discrimination on the target graph. 

**Effect of pseudo-labeling.** *w/o PL* performs worse because, without $L_{PL}$, 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: *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 control the scale of distribution shifting. We evaluated SGDA with $\epsilon$ 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 $\epsilon$ 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 scarcity. 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 construction, 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


[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.


