GALA: Graph Diffusion-Based Alignment With Jigsaw for Source-Free Domain Adaptation

Junyu Luo, Yiyang Gu, Xiao Luo, Wei Ju, *Member, IEEE*, Zhiping Xiao, Yusheng Zhao, Jingyang Yuan, and Ming Zhang, *Member, IEEE*

***Abstract*—Source-free domain adaptation is a crucial machine learning topic, as it contains numerous applications in the real world, particularly with respect to data privacy. Existing approaches predominantly focus on euclidean data, such as images and videos, while the exploration of non-euclidean graph data remains scarce. Recent graph neural network (GNN) approaches can suffer from serious performance decline due to domain shift and label scarcity in source-free adaptation scenarios. In this study, we propose a novel method named Graph Diffusion-based Alignment with Jigsaw (GALA), tailored for source-free graph domain adaptation. To achieve domain alignment, GALA employs a graph diffusion model to reconstruct source-style graphs from target data. Specifically, a score-based graph diffusion model is trained using source graphs to learn the generative source styles. Then, we introduce perturbations to target graphs via a stochastic differential equation instead of sampling from a prior, followed by the reverse process to reconstruct source-style graphs. We feed the source-style graphs into an off-the-shelf GNN and introduce class-specific thresholds with curriculum learning, which can generate accurate and unbiased pseudo-labels for target graphs. Moreover, we develop a simple yet effective graph-mixing strategy named graph jigsaw to combine confident graphs and unconfident graphs, which can enhance generalization capabilities and robustness via consistency learning. Extensive experiments on benchmark datasets validate the effectiveness of GALA.**

***Index Terms*—Graph diffusion model, graph neural network, source-free domain adaptation.**

I. Introduction

Graph neural networks (GNNs) have recently demonstrated superior performance in graph-level representation learning, which facilitates various graph machine learning problems. Among them, graph classification aims to predict the labels of the whole graphs [1], [2], [3], [4], which has various real-world applications, such as molecule property prediction [5], [6] and social network analysis [7], [8]. Typically, these methods adopt the message-passing mechanism to generate informative node-level representations [9]. These representations are subsequently summarized into graph-level representations via summarization operators for classification.

In spite of the promising performance of GNN approaches, they typically operate under the assumption that training and testing graphs come from the same distribution. However, in real-world applications, this is rarely the case. For example, in the discovery of novel molecules in unexplored environments, particularly in response to urgent events, there is often a substantial domain shift from existing training data, resulting in inferior testing performance. One potential solution is to use graph domain adaptation methods, which combine GNN with domain alignment techniques. These methods usually utilize adversarial learning [10] to implicitly minimize distribution differences, or use cross-domain graph contrastive learning [11] for explicit alignment.

However, data privacy has become an important issue in both daily life and scientific research [12]. This is evident in the practices of AI companies, which generally release only pre-trained models rather than the entire dataset. Recent graph domain adaptation approaches typically require complete access to both the source and target graphs [11], [13], which can be impracticable due to privacy regulations. Towards this end, this paper investigates source-free graph domain adaptation, a practical but under-explored problem, which adapts an off-the-shelf source model to unlabeled target graphs without accessing the source graphs. In addition, our scheme is data-efficient, as we sufficiently leverages the open-source pre-trained models.

In literature, various works have focused on source-free domain adaptation [14], [15], [16], [17], [18]. These strategies usually employ self-supervised techniques for semantic exploration in the target domains and conduct domain alignment based on training statistics and weight matrices. However, these works often concentrate on normal euclidean data, such as images and text, while this problem remains under-explored on non-euclidean graph data. In fact, developing a source-free graph domain adaptation framework meets two significant challenges. (1) *How to sufficiently align two graph domains without having access to the source graphs?* The graph samples contain complicated structures, including various nodes and topologies [19]. The complicated structures and rich semantics lead

Manuscript received 24 October 2023; revised 16 April 2024; accepted 7 June 2024. Date of publication 18 June 2024; date of current version 5 November 2024. This work was supported in part by the National Key Research and Development Program of China under Grant 2023YFC3341203, and in part by the National Natural Science Foundation of China under Grant 62276002 and Grant 62306014. Recommended for acceptance by M. Zhang. *(Corresponding authors: Xiao Luo, Wei Ju, and Zhiping Xiao.)*

Junyu Luo, Yiyang Gu, Wei Ju, Yusheng Zhao, Jingyang Yuan, and Ming Zhang are with the School of Computer Science, National Key Laboratory for Multimedia Information Processing, Anker Embodied AI Lab, Peking University, Beijing 100871, China (e-mail: luojunyu@stu.pku.edu.cn; yiyanggu@pku.edu.cn; juwei@pku.edu.cn; yusheng.zhao@stu.pku.edu.cn; yuanjy@pku.edu.cn; mzhang_cs@pku.edu.cn).

Xiao Luo and Zhiping Xiao are with the Department of Computer Science, University of California, Los Angeles, CA 90095 USA (e-mail: xiaoluo@cs.ucla.edu; patricia.xiao@cs.ucla.edu).

The source code is available at https://github.com/luo-junyu/GALA.

Digital Object Identifier 10.1109/TPAMI.2024.3416372

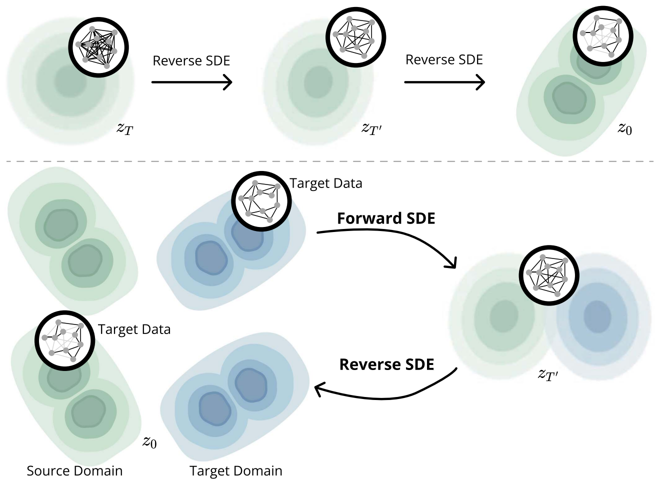


Fig. 1. A brief motivation of GALA. Previous score-based diffusion models usually generate new data by sampling from a prior distribution (upper), while ours GALA transforms the target graphs back to the source domain (lower).

to a hierarchical domain shift, which is difficult to capture [20]. Even worse, we are unable to explicitly align two domains by minimizing the distribution divergence, since source graphs are unavailable. (2) *How to overcome the label scarcity in the target domain?* In real-world applications, we can only access limited unlabeled target graphs, since labeling graph-structured data can be very costly [21]. Previous techniques often adopt pseudo-labeling for additional supervision. Given the complexities inherent in graph structures, these pseudo-labels probably contain errors and are skewed toward the dominant class.

To tackle these two challenges, this paper proposes a novel method named Graph Diffusion-based Alignment with Jigsaw (GALA), tailored for source-free graph domain adaptation. The main idea of our GALA is to convert the target graphs into source-style graphs using a graph diffusion model. Specifically, we first employ source graphs to train a score-based graph diffusion model, which connects intricate source graph structures with a prior distribution using a stochastic differential equation (SDE). As shown in Fig. 1, we use standard diffusion models to generate new data from a prior distribution, while we perturb target graphs using forward SDE, and then reconstruct source-style graphs using reverse SDE with source styles. In this way, we bridge target graphs with the source domain in the protection of data privacy.

In addition, to generate more accurate and impartial pseudo-labels under label scarcity, we introduce adaptive class-specific thresholds, which will progressively increase with the spirit of curriculum learning. Target graphs are regarded as confident if their confidence scores are above the threshold. To exploit unconfident graphs, we provide a simple yet effective graph-mixing method called graph jigsaw, which executes graph clustering based on community detection to select a subgraph from each graph. Subsequently, a confident graph and an unconfident graph exchange subgraphs, much like a jigsaw puzzle. Mixed and original graphs are assumed to have consistent predictions for model generalization and robustness. We conduct comprehensive experiments on benchmark datasets and experiments demonstrate the superiority of the proposed GALA compared with existing baselines.

To summarize, the contribution of this work is as follows:

- We explore a practical but under-explored problem of source-free graph domain adaptation. To the best of our knowledge, we are the first to explore this problem.
- We propose a novel framework named GALA, which transforms target graphs into source-style using a graph diffusion model to mitigate the domain shift in data.
- To overcome the label scarcity, we not only introduce adaptive class-specific thresholds with curriculum learning to learn accurate and unbiased pseudo-labels, but also utilize graph jigsaw with consistency learning for model generalization and robustness.
- Extensive experiments on a range of benchmark datasets demonstrate the superiority of the proposed GALA compared with various competing baselines. Extensive ablation studies and visualization further validate our superiority.

II. Related Work

In this section, we briefly review three related topics, i.e., graph classification, graph domain adaptation, and source-free domain adaptation.

A. Graph Classification

Graph neural networks (GNNs) have remarkable performance in graph classification by mapping graph-structured data into embedding vectors [3], [4], [22], [23]. Among various GNNs, message passing neural networks have been the most prevalent [9], [24], which update node representations via neighborhood aggregation. Recently, graph kernels have been incorporated into GNNs to learn from various substructures such as motifs and paths [25]. To generate graph-level representations, various graph pooling operators are proposed using the attention mechanism [26], reinforcement learning [27], and graph clustering [28]. In addition, various semi-supervised methods [29] are proposed for data-efficient graph-level learning [30]. Despite extensive progress, existing methods mostly assume that training and test graphs come from the same distribution, which is usually not the case in real-world applications. Therefore, we adopt diffusion adaptation to transform target graphs into source-style graphs effectively.

B. Graph Domain Adaptation

Graph domain adaptation has attracted substantial interest in recent research [31], [32], [33], [34], [35], [36], [37], [38], [39]. Early efforts typically focus on node-level adaptation [13], [40], [41], [42], where knowledge is transferred from a labeled source graph to an unlabeled target graph. These strategies usually utilize improved GNN encoders and adversarial learning for domain-invariant representations. For instance, UDA-GCN [40] uses dual graph encoders to explore semantics from multiple perspectives and then combines them with the attention mechanism. In contrast to node-level, graph-level adaptation would encounter numerous graphs with complex distribution shifts [43],

[44]. The recent CoCo [11] combines graph kernel networks with GNNs and employs a coupled graph contrastive learning framework to tackle domain disparities. StruRW [45] utilizes a structural reweighting method to address the domain shift caused by graph structure and node attributes. SA-GDA [21] uses spectral augmentation and dual graph convolutional networks for graph domain adaptation. UDANE [46] leverages the transferable node representation to transfer knowledge across domains. P-Mixup [47] enforces the target domain progressively moving to the source domain for effective domain transfer. However, these works generally require access to both source and target graphs, which could be impractical in real-world scenarios with strict privacy restrictions. To this end, we investigate source-free graph domain adaptation, which adapts an off-the-shelf model to an unlabeled target domain.

C. Source-Free Domain Adaptation

Source-free domain adaptation seeks to transfer a model trained on a source domain to an unlabeled target domain without access to source data. Existing approaches can be essentially categorized into self-training approaches and domain alignment approaches. Self-training approaches typically employ pseudo-labeling and mutual information maximization to discover semantics from unlabeled target data [14], [15], [48]. For instance, SHOT [15] generates prototypes of classes in the hidden space, which can be used to train the nearest centroid classifier for unlabeled target data. In contrast, domain alignment approaches attempt to explore stored batch statistics and weight matrices for reduced distribution shift [16], [49]. However, these methods are designed for visual data, which performs poorly on non-euclidean graph data. Towards this end, we propose a diffusion-based source-free graph domain adaptation method, which transforms target graphs back to the source domain via a graph diffusion model.

III. Preliminaries

A. Problem Definition

We denote a graph as $G = (V, E)$ where V denotes nodes and E denotes edges. Each graph is associated with a node attribute matrix $\boldsymbol{X} \in \mathbb{R}^{|V| \times d^f}$ where d^f is the attribute dimension. We denote a labeled source domain as $\mathcal{D}^{so} = \{(G_i^{so}, y_i^{so})\}_{i=1}^{N_s}$ where G_i^{so} stands for the i-th source graph and y_i^{so} is its label. An unlabeled target domain is $\mathcal{D}^{ta} = \{G_j^{ta}\}_{j=1}^{N_t}$ where G_j^{ta} denote the j-th target graph. While two domains share the same label space, they have different data distributions. In the problem of source-free graph domain adaptation, we initially pre-train neural network models using source data and subsequently learn a model to predict the labels of target graphs. Importantly, to safeguard data privacy, the source data remains inaccessible during the adaptation on the target domain.

B. Graph Neural Networks

Graph neural networks (GNNs) are commonly used to encode graph-structured data based on a message-passing mechanism. In particular, given a graph G, we denote the node representation of $v \in V$ at the layer k as $\boldsymbol{h}_i^{(k)}$ and formulate the updating rule as follows:

$$\boldsymbol{h}_{\mathcal{S}(v)}^{(k)} = \text{AGG}^{(k)}\left(\left\{\boldsymbol{h}_u^{(k-1)} : u \in \mathcal{S}(v)\right\}\right), \quad (1)$$

$$\boldsymbol{h}_v^{(k)} = \text{COM}^{(k)}\left(\boldsymbol{h}_v^{(k-1)}, \boldsymbol{v}_{\mathcal{S}(v)}^{(k)}\right), \quad (2)$$

where $\mathcal{S}(v)$ represents the neighboring nodes of v and $\boldsymbol{h}_{\mathcal{S}(v)}^{(k)}$ denotes the neighborhood representations for v. AGG$^{(k)}$ and COM$^{(k)}$ denote the aggregation and combination operators. Finally, a global pooling function is used to summarize the node representations into a graph-level representation,

$$\boldsymbol{z} = \text{GP}\left(\left\{\boldsymbol{v}_i^{(K)}\right\}_{i=1}^{n}\right), \quad (3)$$

in which GP$(\cdot)$ is for global pooling. Finally, we utilize an MLP classifier with softmax activation, i.e., HEAD$(\cdot)$ to generate label distributions:

$$\boldsymbol{p} = \Phi(G) = \text{HEAD}(\boldsymbol{z}), \quad (4)$$

where Φ represents the whole graph neural network. In our setting, we would first train an off-the-shelf GNN model using source graphs by minimizing the cross-entropy objective as a preliminary:

$$\mathcal{L}_{source} = -\frac{1}{|\mathcal{D}^{so}|} \sum_{G_i^{so} \in \mathcal{D}^{so}} \log \boldsymbol{p}_i^{so}\left[y_i^{so}\right]. \quad (5)$$

where $\boldsymbol{p}_i^{so} = \Phi(G_i^{so})$ denotes the output of the GNN.

IV. The Proposed Approach

A. Framework Overview

This paper introduces GALA for source-free graph domain adaptation. The model, as illustrated in Fig. 2, does not have access to source graphs when learning from target graphs. To solve the problem, we are required to sufficiently align two graph domains and overcome the label scarcity on the target domain. At a high level, GALA transforms target graphs into source-style with a graph diffusion model. Specifically, source graphs are utilized to learn a score-based graph diffusion model, which will be fixed to keep the source styles. On this basis, we perturb target graphs by forwarding a stochastic differential equation (SDE) and then reconstruct source-style graphs using the reverse process. Moreover, we adapt the off-the-shelf GNN model with class-specific thresholds through curriculum learning. This aids in generating accurate and class-balanced pseudo-labels for each target graph. Then, we introduce our proposed GALA in detail as below.

B. Graph Diffusion Model for Domain Alignment

The principal challenge faced in graph domain adaptation is the alignment of source and target domains. This becomes increasingly challenging in our scenarios since we cannot simultaneously access the source and target data. Our approach is to employ a generative model for source-style graphs that can make precise predictions using the pre-trained GNNs. To achieve

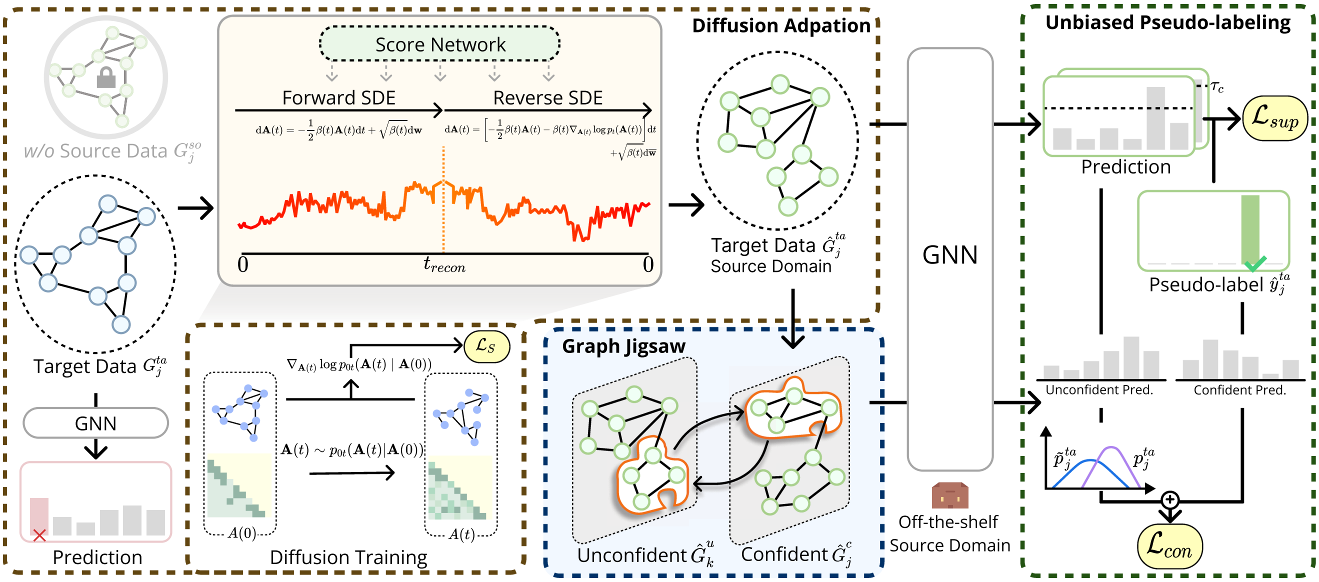

w/o Source Data G_j^{so}
Score Network
Forward SDE
Reverse SDE
$\mathrm{d}\mathbf{A}(t) = -\frac{1}{2}\beta(t)\mathbf{A}(t)\mathrm{d}t + \sqrt{\beta(t)}\mathrm{d}\mathbf{w}$
$\mathrm{d}\mathbf{A}(t) = \left[-\frac{1}{2}\beta(t)\mathbf{A}(t) - \beta(t)\nabla_{\mathbf{A}(t)}\log p_t(\mathbf{A}(t))\right]\mathrm{d}t + \sqrt{\beta(t)}\mathrm{d}\overline{\mathbf{w}}$
0
t_{recon}
0
Diffusion Adpation
Target Data $\hat{G}_j^{ta}$
Source Domain
Target Data G_j^{ta}
GNN
Prediction
$\mathcal{L}_S$
$\nabla_{\mathbf{A}(t)}\log p_{0t}(\mathbf{A}(t) \mid \mathbf{A}(0))$
$\mathbf{A}(t) \sim p_{0t}(\mathbf{A}(t)|\mathbf{A}(0))$
A(0)
A(t)
Diffusion Training
Graph Jigsaw
Unconfident $\hat{G}_k^u$
Confident $\hat{G}_j^c$
GNN
Off-the-shelf
Source Domain
Unbiased Pseudo-labeling
τ_c
$\mathcal{L}_{sup}$
Prediction
Pseudo-label $\hat{y}_j^{ta}$
Unconfident Pred.
Confident Pred.
$\tilde{p}_j^{ta}$
p_j^{ta}
$\mathcal{L}_{con}$

Fig. 2. Overview of GALA. We employ source graphs to train a score-based graph diffusion model and then transform target graphs into source-style graphs. Moreover, we introduce adaptive class-specific thresholds to generate confident graphs with pseudo-labels and then utilize graph jigsaw to exchange subgraphs between graph pairs for consistency learning.

this, we introduce an off-the-shelf graph diffusion model in the source domain, with the ability to generate graphs with source style. Subsequently, the target data can be fed into the diffusion model to generate a source-style graph via the reverse process.

1) Graph Diffusion Model: We first introduce our graph diffusion model, which reconstructs the graph structure using an SDE [50], [51], [52], [53]. As the most crucial component, the graph structure is expressed via the adjacency matrix as a variable from the timestamp 0 to 1 with an interval Δt when adding noise, i.e., $\mathbf{A}(t)$. Then, the SDE for the adjacency matrix is:

$$\mathrm{d}\mathbf{A}(t) = -\frac{1}{2}\beta(t)\mathbf{A}(t)\mathrm{d}t + \sqrt{\beta(t)}\mathrm{d}\mathbf{w}, \tag{6}$$

where $\beta(t)$ is a scalar of adding noise and $\mathbf{w}$ is a Wiener process indicating random noise. The key of an effective diffusion model is to obtain the score function, i.e., the gradient of the likelihood $\nabla_{\mathbf{A}(t)} \log p_{0t}(\mathbf{A}(t) \mid \mathbf{A}(0))$ where $\mathbf{A}(t) \sim p_{0t}(\mathbf{A}(t)|\mathbf{A}(0))$. Here, we utilize a score-based GNN $\rho(\mathbf{A}(t), t)$ to approximate the ground truth by minimizing the following objective:

$$\begin{aligned}\mathcal{L}_S = \mathbb{E}_t\{\lambda(t)\mathbb{E}_{\mathbf{A}(0)}\mathbb{E}_{\mathbf{A}(t)|\mathbf{A}(0)}[\|\rho(\mathbf{A}(t), t) \\ - \nabla_{\mathbf{A}(t)} \log p_{0t}(\mathbf{A}(t) \mid \mathbf{A}(0))\|_2^2]\},\end{aligned} \tag{7}$$

where $\lambda(t)$ is a positive weighting function. By minimizing (7) on the source domain, we can obtain a graph diffusion model with source styles embedded.

2) Graph-Based Score Function: Different from diffusion models in computer vision, our approach requires incorporating graph-structural information during propagation. Combining a random walk with a message-passing mechanism, we develop an effective graph-based score function $\rho(\mathbf{A}(t), t)$. Specifically, to better learn structural information, we first discretize the adjacency matrix and then utilize a random walk to generate edge representations $\boldsymbol{e}_{mn}$:

$$\dot{\mathbf{A}}(t) = \mathbf{1}_{\mathbf{A}(t) > 1/2}, \tag{8}$$

$$\boldsymbol{e}_{mn}^{(0)} = \phi([\boldsymbol{R}_{mn}, \boldsymbol{R}_{mn}^2, \ldots, \boldsymbol{R}_{mn}^r]), \tag{9}$$

where $\boldsymbol{R}^k = (\dot{\mathbf{A}}(t)\dot{\mathbf{D}}^{-1}(t))^k$ indicates the likelihood of starting and end points for every k-length walk. r denotes the maximal random walk length and $\dot{\mathbf{D}}^{-1}(t)$ is the degree matrix of $\dot{\mathbf{A}}(t)$. $\phi(\cdot)$ is an MLP to map these likelihoods into the embedding space. Besides edge information, we also extract node representations $\boldsymbol{f}_i^{(0)} = [\boldsymbol{e}_{ii}, \boldsymbol{d}_i]$ where $\boldsymbol{d}_i$ is the one-hot degree vector, and then utilize a message passing neural network with to update $\boldsymbol{f}_i^k$ as:

$$\boldsymbol{f}_{\mathcal{S}(v)}^{(k)} = \mathrm{AGG}^{(k)}\left(\left\{\boldsymbol{f}_i^{(k-1)} : j \in \mathcal{S}(i)\right\}\right) \tag{10}$$

$$\boldsymbol{f}_i^{(k)} = \mathrm{COM}^{(k)}\left(\boldsymbol{f}_i^{(k-1)}, \boldsymbol{f}_{\mathcal{S}(v)}^{(k)}\right). \tag{11}$$

Finally, we update edge representations as:

$$\boldsymbol{e}_{ij}^{(k+1)} = \widetilde{\mathrm{COM}}^{(k)}\left(\boldsymbol{e}_{ij}^{(k)}, \boldsymbol{f}_i^{(k)}, \boldsymbol{f}_j^{(k)}\right), \tag{12}$$

where $\widetilde{\mathrm{COM}}^{(k)}$ is a different combination operator to update the edge representation. After stacking K^d layers, we utilize an MLP to generate the estimated adjacency matrix.

3) Target Graph Adaptation: By sampling from the prior distribution, the well-trained graph diffusion model can generate new graphs carrying source styles. Instead of sampling from the prior distribution $p_1(\mathbf{A}(1))$, GALA adds noise to the target graphs and then uses the reverse process to endow the target graphs with source styles. This process only utilizes the trained diffusion model, effectively addressing the domain shift while protecting the privacy of source graphs.

In detail, we perturb the adjacency matrix of the target graph, which generates noise with target semantics,

$$\mathbf{A}(t_{recon}) \sim p_{0t_{recon}}(\mathbf{A}(t_{recon})|\mathbf{A}(0)). \tag{13}$$

where t_{recon} is the starting point of reconstruction. Consequently, we utilize an iterative process to alter the noisy input $\mathbf{A}(t_{recon})$ for reconstruction with the reverse process. The reserve SDE can be written as:

$$\mathrm{d}\mathbf{A}(t) = \left[-\frac{1}{2}\beta(t)\mathbf{A}(t) - \beta(t)\nabla_{\mathbf{A}(t)}\log p_t(\mathbf{A}(t))\right]\mathrm{d}t + \sqrt{\beta(t)}\mathrm{d}\overline{\mathbf{w}}, \tag{14}$$

where $\overline{\mathbf{w}}$ denotes a standard Wiener process for backward. On this basis, the updating process of the generated adjacency matrix $\mathbf{A}^g(t-\Delta t)$ can be:

$$\mathbf{A}^g(t-\Delta t) \sim p(\mathbf{A}^g(t-\Delta t)|\mathbf{A}^g(t)). \tag{15}$$

In our implementation, the Euler-Maruyama algorithm is employed to simply the propagation as:

$$\mathbf{A}^g(t-\Delta t) = \mathbf{A}^g(t) + \left[\frac{1}{2}\beta(t)\mathbf{A}^g(t) + \beta(t)\mathbf{s}_\theta\left(\mathbf{A}^g(t), t\right)\right] + \sqrt{\beta(t)}\sqrt{\Delta t}\mathbf{z}_t, \tag{16}$$

where $\mathbf{z}_t \sim \mathcal{N}(0,1)$. Finally, the adjacency matrix $\mathbf{A}^g(0)$ is for the reconstructed graph, which is denoted as $\hat{G}_j^{ta}$. In this manner, we can adapt the target graphs into source-type graphs from a data-centric perspective, which can facilitate accurate semantics exploration using the out-of-shelf GNN.

C. Unbiased Pseudo-Labeling With Curriculum Learning

To overcome label scarcity in the target domain, we turn to pseudo-labeling techiniques. Note that after transforming target graphs into source-style graphs, we can use the pre-trained GNN with a fixed threshold to generate target graph pseudo-labels [54]. However, these pseudo-labels from the pre-trained GNN could be biased toward the dominant class. Worse still, GNNs could produce overconfident pseudo-labels due to the potential class competition from normalization operators. To avoid this, we propose unbiased pseudo-labeling with class-specific thresholds for accurate and unbiased pseudo-labels. In addition, a curriculum learning approach is employed, which initially concentrates on all graph samples and then progressively focuses on more reliable ones. In other words, we increase the threshold to reject samples from easy to hard.

Primarily, we generate the label distribution $\boldsymbol{p}_j^{ta}$ using the pre-trained GNN for each source-style target graph $\hat{G}_j^{ta}$, i.e.,

$$\boldsymbol{p}_j^{ta} = \Phi(\hat{G}_j^{ta}), \tag{17}$$

where $\hat{G}_j^{ta}$ is the output of the diffusion model. To generate the rigid and class-balanced pseudo-labels, we first calculate the confidence distribution for each class and subsequently establish an adaptive threshold based on the maximum. In formulation, the confidence of each sample is defined as:

$$s_j^{ta} = \max_c \boldsymbol{p}_j^{ta}[c]. \tag{18}$$

Then, the maximum confidence scores for class c are:

$$\mathcal{M}_c = \max\{s_j^{ta}|argmax_{c'}\boldsymbol{p}_j^{ta}[c'] = c\}. \tag{19}$$

The adaptive threshold for class c is defined as:

$$\tau_c = \mathcal{M}_c \cdot \alpha(e), \tag{20}$$

where $\alpha(e)$ is shared across different classes. Following the spirit of curriculum learning, we would increase $\alpha(e)$ linearly according to epoch number e, which gradually selects graphs with higher confidence. Consequently, this would yield a collection of confident target graphs:

$$\mathcal{C} = \{G_j^{ta}|c = argmax_{c'}\boldsymbol{p}_j^{ta}[c'], s_j^{ta} > \tau_c, \}. \tag{21}$$

The standard cross-entropy loss is minimized in $\mathcal{C}$:

$$\mathcal{L}_{sup} = -\frac{1}{|\mathcal{C}|}\sum_{G_j^{ta}\in\mathcal{C}} \log \boldsymbol{p}_j^{ta}[\hat{y}_j^{ta}], \tag{22}$$

where $\hat{y}_j^{ta}$ denotes the pseudo-label of G_j^{ta}. Our pseudo-labeling strategy provides a reliable and unbiased optimization process with reduced error accumulation.

D. Consistency Learning With Graph Jigsaw

Despite the effectiveness of pseudo-labeling, there could be a large number of unconfident target graphs without sufficient exploration. To tackle this, we aim to expand the dataset by combining confident and unconfident graphs. Recently, a number of works have been developed on graph Mixup [55], [56], which combines graph representations in the latent space. However, these methods cannot promise to generate realistic graphs and are dependent on GNN encoders. In contrast, we employ a simple yet effective augmentation strategy, graph jigsaw, that works in the graph space. Graph jigsaw uses a community detection algorithm to derive a small subgraph from each graph and then exchange them in a jigsaw puzzle mechanism. Finally, we propose consistency learning to ensure that post-augmentation predictions remain consistent, thereby enhancing robustness and generalization.

In detail, we adopt a community detection algorithm, i.e., the Louvain algorithm [57] to partition graphs into diverse clusters. The benefit of this algorithm is unnecessary to decide the cluster number. Then, we randomly choose a cluster, from each confident graph $\hat{G}_j^c$ and denote the complementary part and subgraph as, $\hat{G}_{j,1}^c$ and $\hat{G}_{j,2}^c$. In parallel, each unconfident graph $\hat{G}_k^u$ can be segmented into $\hat{G}_{k,1}^u$ and $\hat{G}_{k,2}^u$. We exchange semantics by combining $\hat{G}_{j,1}^c$ and $\hat{G}_{k,2}^u$ to generate augmented graph $\tilde{G}_j^c$, and $\hat{G}_{j,2}^c$ and $\hat{G}_{k,1}^u$ would be connected to generate graph $\tilde{G}_k^u$. A detailed example can be seen in Fig. 2.

Then, consistency learning is leveraged to encourage the mixed graph to yield similar predictions. In formulation, the learning objective is written as:

$$\mathcal{L}_{con} = -\frac{1}{|\mathcal{S}|}\sum_{G_j^{ta}\in\mathcal{S}} \log \tilde{\boldsymbol{p}}_j^{ta}[\hat{y}_j^{ta}] - \frac{1}{|\mathcal{U}|}\sum_{G_j^{ta}\in\mathcal{U}} KL(\tilde{\boldsymbol{p}}_j^{ta}||\boldsymbol{p}_j^{ta}), \tag{23}$$

Algorithm 1: Optimization Algorithm of GALA.

```
Input: Source graphs $\mathcal{D}^{so}$, target graphs $\mathcal{D}^{ta}$,
Output: GNN-based classifier $\Phi(\cdot)$
Pre-train $\Phi(\cdot)$ using source graphs;
Train diffusion model using source graphs by
  minimizing (7);
for epoch $= 1, 2, \ldots$ do
    Obtain class-specific thresholds using (20);
    Generate confident target graphs with pseudo-labels
    using (21);
    for each batch do
        Sample a mini-batch of target graphs;
        Calculate the loss objective using (24);
        Update the parameters of GNN through
        back-propagation;
    end for
end for
```

where $\tilde{\boldsymbol{p}}_j^{ta}$ is the label distribution of the augmented view and $\mathcal{U} = \mathcal{D}^{ta}/\mathcal{S}$ are unconfident graphs. Moreover, pseudo-labels are adopted to supervise confident samples, while the KL divergence of predictions between unconfident graphs and their augmented views are minimized. Our GALA enjoys the regularization from consistency learning to learn from extensive unconfident graph samples, which can further minimize the information loss.

E. Summarization

In a nutshell, the final loss of training our GALA is summarized as:

$$\mathcal{L} = \mathcal{L}_{sup} + \mathcal{L}_{con}. \tag{24}$$

As a preliminary, we adopt an off-the-shelf GNN and graph diffusion model trained on the source domain. Subsequently, the loss objective is minimized in the target domain. The overall algorithm is summarized in Algorithm 1.

Complexity Analysis: We assume that $|D^{ta}|$ is the number of target graphs, d is the feature dimension, $|V|^a$ is the average number of nodes, W is the reconstruction steps. The computational complexity of the diffusion is $\mathcal{O}(K^d|D^{ta}||V|^a d^2 W)$. The GNN model takes $\mathcal{O}(K|D^{ta}||V|^a d^2)$ where K is the layer number. Therefore, the complexity of our GALA is $\mathcal{O}((K + K^d W)|D^{ta}||V|^a d^2)$ which is linear to both $|V|^a$ and $|D^{ta}|$.

V. Experiments

A. Experimental Settings

Datasets: We perform experiments on real-world benchmark datasets with source-free domain adaption settings. In order to demonstrate the performance of our approach in various scenarios, we conducted experiments on both dataset split and cross-dataset source free domain adaptation. In ENZYMES, Mutagenicity, PROTEINS, and FRANKENSTEIN datasets, we split the data, thereby introducing domain discrepancies. Then, we perform source-free domain adaptation across the sub-datasets. As for the COX2 and BZR datasets, we directly conduct source-free domain adaptation on the sub-datasets associated with each of them.

- *ENZYMES* [58] is a bioinformatic data set comprising 600 tertiary protein structures. It constitutes a classic collection of data rooted in the structural information of biological molecules and proteins.
- *Mutagenicity* [59] encompasses a multitude of molecular structures, each intricately paired with its corresponding Ames test data, amounting to a total of 4337 molecular structures.
- *PROTEINS* [60] contain protein data in graph form, where each label indicates whether a protein is a non-enzyme. The amino acids serve as nodes. The edges exist when the distance between two nodes is less than 6 angstroms.
- *FRANKENSTEIN* [61] is a composite dataset that amalgamates the BURSI and MNIST datasets. Each data sample is depicted as a graph, with the chemical atom symbols representing vertices and the bond types representing edges.
- *COX2* [62]. We engage the COX2 and COX2_MD datasets, consisting of 467 and 303 inhibitors specifically targeting cyclooxygenase-2. Within these datasets, individual graphs serve as graphical representations of distinct chemical compounds. These graphs feature edges annotated with distance values, while the vertex labels are indicative of the atom types found in the compounds.
- *BZR* [62]. We use BZR and BZR_MD datasets, which consist of 405 and 306 ligands designed to interact with the benzodiazepine receptor. The process of graph construction employed in these datasets is analogous to that of the COX2 dataset.

Domain Settings. This work follows the setting of source-free domain adaptation [15], [48] in the image classification task. The division of the dataset is executed by using graph density, which is defined as the ratio of the number of existing edges to the number of potential edges. It can be calculated as

$$D = \frac{2|E|}{|V|(|V| - 1)}, \tag{25}$$

where $|E|$ is the number of edges and $|V|$ is the number of nodes in the graph G. Within each sub-dataset, the ratio of the training set to the test set is 8:2. Each dataset is split according to the above instructions, leading to domain discrepancies among these sub-datasets and subsequently yield inferior cross-domain performance.

Diffusion models and pre-trained GNNs are trained on the source domain as off-the-shelf models to simulate real-world scenarios. The off-the-shelf models adapt to the target domain. In the adaptation process, only the target domain data is used, and the source domain data is not available. This is more aligned with real-world application scenarios.

Baselines: For credibility, we compare GALA with extensive competing baselines, which can be categorized into three aspects: (a) Graph Neural Networks, including GCN [63], GIN [64], GAT [65] and GraphSAGE [66]. (b) Graph Semi-Supervised Methods, including Mean-Teacher [67], InfoGraph [68] and TGNN [69]. (c) Source-free Adaptation Methods.

We adopt a recent state-of-the-art source-free domain adoption method PLUE [18] of image classification. Note that the semi-supervised methods are designed to utilize both source and target data. Therefore, the comparison with semi-supervised methods may not be entirely fair.

- *GCN* [63] adapts localized first-order approximation of spectral graph convolution to represent the graph data.
- *GIN* [64] takes the message passing neural networks for capturing the different topological structures of graphs.
- *GraphSAGE* [66] leverages node feature information to efficiently generate node embeddings for unseen data.
- *GAT* [65] introduces the attention mechanism, to facilitate the model focusing on the most informative parts.
- *Mean-Teacher* [67] is an effective semi-supervised algorithm in the graph domain. The method uses a student to make predictions and a teacher model to generate training targets. The teacher model is an exponential moving average of the student model.
- *InfoGraph* [68] is a semi-supervised method for GNNs that uses mutual information maximization to learn representations of graphs.
- *TGNN* (Twin Graph Neural Network) [69] is a semi-supervised approach for graph classification, which contains a message-passing module and a graph kernel module to explore structural information.
- *PLUE* [18] is a recent state-of-the-art source-free domain adaption method for the image classification task, which proposes to evaluate the reliability of refined labels while excluding the noisy label. We implement PLUE in graph classification tasks.

B. Implementation Details

In this passage, we introduce the implementation details of GALA at different stages. The baseline methods are initiated with hyperparameters as the corresponding paper and fine-tuned to achieve the best performance. To mitigate the potential impact of randomness, we employ 5 runs and report the average accuracy and standard deviation.

Model Training on Souce Domain: We adopt GCN as the default GNN encoder, with the embedding dimension 64 and the layer number 3. We use Adam optimizer and a learning rate of 0.001. We train the model in the source data domain with total epochs of 100 and batch size of 64.

Diffusion Training on Source Domain: In our graph score networks, we employ a configuration of 4 message-passing layers equipped with 8 attention heads. For the estimation of the final score, we utilize an MLP with 2 hidden layers. The training of these models is accomplished with the use of an Adam optimizer, maintaining a constant learning rate of $2e-5$ and batch size of 128. To enhance stability in the parameter updating process, we also implement the exponential moving average (EMA) methodology, applying a momentum of 0.9999. We employ the variance preserve SDE in this work. We utilize the Euler-Maruyama method, setting δt at 0.001, with 1000 discretization steps for graph sampling during the reconstruction phase.

Source-free Adaptation on Target Domain: During the target domain adaptation, we initially perform domain alignment on the target data using an off-the-shelf diffusion model. The reconstruction starting point t_{recon} is default set at 0.1. That means we first add noise from $t = 0$ to $t = t_{recon}$, followed by a denoising process back to $t = 0$ for domain transfer. Subsequently, we employ unbiased pseudo-labeling with curriculum learning and consistency learning with graph jigsaw for the domain adaptation. In unbiased pseudo-labeling learning, the initial setting for the pseudo-label threshold $\alpha(0)$ is 0.95. In curriculum learning, the confidence threshold $\alpha(e)$ increases linearly with the epoch e from $\alpha(0)$ to 0.99. In the graph jigsaw, we dynamically partition graphs into communities using the Louvain algorithm. Following this partition, subgraphs are exchanged among confident and unconfident graphs, and the exchanged subgraphs inherit the edges connecting them from the original graph.

C. Performance Comparison

Tables I, II, III, IV, and V present the performance of different methods. We observe the following:

1) Source-free domain adaptation presents a challenging task, as the inferior accuracy in the target domain, highlights the necessity of investigating this problem. Both domain shifts and the unavailability of source data impose constraints on the model's capabilities, rendering prior research inapplicable.

2) Semi-supervised methods (e.g., InfoGraph) perform generally better than source-only methods. Semi-supervised methods can utilize both labeled data (source domain) and unlabeled data (target domain). Note that in practical scenarios, it is often not feasible to access the source data, as only off-the-shelf models are available. Nevertheless, semi-supervised methods exhibit poor stability and performance, as the high standard deviation in experiments. This is attributed to the absence of consideration for domain shift between the source domain and the target domain.

3) Source-free method PLUE performs better than the other approaches, especially in tackling the challenges posed by more complex domain adaptation tasks. Despite PLUE being the state-of-the-art method for source-free domain adaptation in image classification tasks, it is noteworthy that this method is not devised for graph data and substantial domain shifts. Consequently, there still remains a need for further improvements.

4) GALA has yielded significant improvements, both within source-free domain adaptation on the split sub-datasets and cross-datasets. Notably, GALA demonstrates an average improvement of 5.3% in cross-dataset experiments when compared to the best-performing compared baseline methods. Furthermore, GALA exhibits improvements, especially in cases of poorly performing domain adaptation sub-tasks.

The improvements in GALA can be attributed to two primary factors: data transformation and domain transfer. 1) Diffusion-based data transformation allows target data to transform to the source domain while preserving semantics. This mitigates the adverse effects of domain shift and facilitates better predictions in the source domain. 2) The integration of graph jigsaw and

TABLE I
THE CLASSIFICATION RESULTS (IN %) ON ENZYMES (SOURCE → TARGET)

Methods	E0→E1	E1→E0	E0→E2	E2→E0	E0→E3	E3→E0	E1→E2	E2→E1	E1→E3	E3→E1	E2→E3	E3→E2	Avg.
GCN	33.3±4.7	29.3±2.7	22.0±2.2	21.3±2.0	27.3±2.3	16.7±3.7	24.7±2.4	24.7±3.8	20.7±0.3	30.0±3.7	28.7±2.8	30.7±1.5	25.7±2.7
GIN	27.3±2.8	22.0±3.1	24.7±4.0	23.3±3.3	20.7±3.5	18.0±2.2	23.3±3.4	25.3±2.5	27.3±2.3	30.7±2.7	23.3±2.2	28.0±2.5	24.5±2.9
GraphSAGE	32.7±2.9	25.3±0.7	21.3±2.8	22.7±1.9	20.0±3.0	16.0±3.1	24.7±2.8	24.7±3.1	24.0±3.3	25.3±2.5	26.7±2.6	34.7±3.6	24.8±2.7
GAT	29.3±2.1	24.0±1.5	28.0±3.6	23.3±3.3	23.3±2.7	20.7±2.9	24.0±2.7	27.3±1.5	23.3±3.0	26.7±2.6	28.0±2.4	23.3±2.0	25.1±2.5
Mean-Teacher	16.0±2.3	22.0±3.6	21.3±3.3	20.0±2.7	20.0±3.2	20.0±3.7	26.7±2.2	19.3±1.5	18.7±3.8	18.7±2.0	23.3±3.4	22.0±2.5	20.7±2.9
InfoGraph	29.3±3.0	28.0±0.7	26.0±2.3	18.0±3.5	22.7±1.4	26.7±2.3	28.7±3.5	25.3±3.2	24.0±2.8	28.7±2.9	26.0±2.8	28.7±4.2	26.0±2.7
TGNN	27.3±1.9	26.7±2.7	27.3±3.8	20.0±1.1	18.7±2.5	14.7±2.5	27.3±3.6	25.3±1.0	30.0±2.0	28.7±2.7	24.7±0.6	26.7±1.0	24.8±2.1
PLUE	29.8±2.3	24.5±2.2	26.2±3.0	21.5±2.1	20.7±2.4	22.7±2.7	29.0±2.0	24.3±1.3	27.0±3.1	30.3±3.0	25.3±2.3	31.5±2.7	26.1±2.4
GALA	33.3±2.6	26.7±2.3	27.3±2.6	26.7±1.8	21.3±1.5	23.3±1.7	36.7±2.0	24.7±2.9	34.7±2.9	35.3±2.3	34.0±1.3	44.7±2.5	**30.7**±2.2

E0, E1, E2, and E3 are split by the graph density.

TABLE II
THE CLASSIFICATION RESULTS (IN %) ON MUTAGENICITY (SOURCE → TARGET)

Methods	M0→M1	M1→M0	M0→M2	M2→M0	M0→M3	M3→M0	M1→M2	M2→M1	M1→M3	M3→M1	M2→M3	M3→M2	Avg.
GCN	68.0±2.0	68.8±1.5	60.5±2.8	64.4±1.5	53.7±1.3	58.1±1.4	75.2±0.8	76.2±1.5	67.5±1.2	55.4±1.5	62.5±1.0	68.5±1.5	64.9±1.5
GIN	70.6±0.4	64.2±0.9	63.5±1.2	62.5±0.7	57.0±0.1	56.4±0.3	73.3±0.5	76.5±0.4	65.2±0.6	53.3±1.8	64.4±0.8	66.8±0.5	64.5±0.7
GraphSAGE	71.2±0.6	65.6±0.3	64.3±0.3	65.5±0.2	57.3±0.5	56.5±0.3	74.7±0.4	77.6±0.7	62.3±0.4	51.7±0.3	62.4±0.5	62.3±0.7	64.3±0.4
GAT	69.7±1.0	67.0±1.6	62.7±2.3	67.0±1.5	56.1±2.1	57.8±1.7	76.6±1.2	77.2±0.4	63.4±1.3	53.0±3.9	60.7±0.6	61.8±3.2	64.6±1.7
Mean-Teacher	65.3±4.7	52.1±3.4	70.6±2.5	52.2±1.9	49.9±0.5	49.0±0.4	66.2±1.4	62.7±4.1	50.1±1.3	72.2±1.6	48.9±1.7	65.8±3.1	58.7±2.2
InfoGraph	69.1±1.8	68.9±0.3	66.6±2.5	64.9±1.2	55.9±1.0	57.8±2.1	74.7±0.3	76.8±1.5	65.6±0.6	57.1±3.2	64.7±1.9	64.2±2.9	65.5±1.6
TGNN	73.3±4.9	61.9±2.4	65.3±4.0	58.1±2.4	55.5±3.5	58.1±2.4	65.9±1.1	66.7±3.9	66.5±2.1	70.1±1.0	55.5±3.5	65.3±3.0	63.5±2.9
PLUE	75.2±1.4	68.5±0.5	66.3±1.1	67.9±1.6	54.0±1.3	56.4±1.4	68.4±1.0	76.9±3.5	62.9±0.4	57.6±3.0	62.0±0.6	67.4±2.6	65.3±1.5
GALA	76.4±0.8	69.6±1.3	70.0±2.2	63.2±1.2	58.4±1.2	60.6±1.3	76.9±1.8	80.1±1.9	65.7±1.1	66.5±3.1	65.6±1.5	70.6±1.3	**68.6**±1.6

M0, M1, M2, and M3 are split by the graph density.

TABLE III
THE CLASSIFICATION RESULTS (IN %) ON PROTEINS (SOURCE → TARGET)

Methods	P0→P1	P1→P0	P0→P2	P2→P0	P0→P3	P3→P0	P1→P2	P2→P1	P1→P3	P3→P1	P2→P3	P3→P2	Avg.
GCN	71.9±0.9	74.7±2.9	62.6±1.2	68.3±3.8	51.1±3.3	45.8±3.1	57.6±2.1	70.4±2.2	39.7±3.6	49.7±2.6	58.3±1.3	52.9±3.0	58.6±2.5
GIN	70.0±2.1	60.7±3.6	61.8±2.6	72.9±2.7	47.1±3.3	44.3±4.2	62.5±2.1	68.9±2.0	41.1±3.2	47.9±3.3	48.6±4.0	56.1±2.6	57.4±3.0
GraphSAGE	72.2±1.0	78.3±3.0	64.7±2.3	67.2±1.1	46.9±1.1	42.2±4.0	62.6±1.8	69.7±0.8	32.9±2.0	50.8±2.3	56.1±3.5	56.9±3.4	58.9±2.2
GAT	70.0±3.7	71.4±3.7	66.8±2.0	73.9±2.6	49.3±1.3	40.4±2.8	61.4±4.6	68.9±2.2	44.3±3.8	49.3±3.6	50.7±2.4	52.1±2.9	58.3±3.0
Mean-Teacher	64.3±4.1	71.4±5.2	60.4±3.6	72.1±4.4	25.0±5.6	55.4±4.0	61.1±2.3	60.7±5.3	29.6±4.8	49.3±3.1	31.8±4.0	55.4±4.9	51.0±4.3
InfoGraph	74.0±2.7	77.6±2.9	68.3±3.6	71.1±1.1	46.9±3.2	46.5±2.1	64.4±1.5	72.2±1.9	41.9±1.1	35.4±1.9	54.7±3.1	62.6±4.2	59.4±2.4
TGNN	60.4±3.4	41.8±2.8	45.7±4.6	46.6±4.2	64.8±5.4	42.0±3.9	58.6±3.2	58.4±3.7	75.7±4.0	68.4±3.2	65.5±4.8	53.8±2.0	55.9±3.8
PLUE	64.3±2.3	71.5±2.1	65.0±3.0	78.4±2.1	45.9±3.7	63.7±3.0	58.3±3.3	68.9±2.7	41.9±1.3	45.7±4.1	47.5±3.2	58.3±2.7	59.1±2.8
GALA	72.3±1.0	73.9±2.0	66.8±3.4	78.5±1.9	47.1±2.2	65.6±3.7	61.1±1.1	71.8±3.1	43.9±2.0	50.4±2.6	43.2±2.0	60.0±3.4	**61.2**±2.4

P0, P1, P2, and P3 are split by the graph density.

TABLE IV
THE CLASSIFICATION RESULTS (IN %) ON FRANKENSTEIN (SOURCE → TARGET)

Methods	F0→F1	F1→F0	F0→F2	F2→F0	F0→F3	F3→F0	F1→F2	F2→F1	F1→F3	F3→F1	F2→F3	F3→F2	Avg.
GCN	55.3±0.8	56.4±1.6	60.4±1.9	54.6±1.4	46.7±1.8	51.6±1.7	60.7±0.8	58.3±1.3	47.9±1.8	47.5±0.8	52.1±2.6	54.6±1.8	53.8±1.5
GIN	56.5±0.5	53.8±1.4	57.2±0.4	57.9±1.8	50.7±3.5	51.0±0.6	58.7±1.3	58.3±0.8	46.7±1.4	47.9±1.3	50.5±0.7	52.0±2.1	53.4±1.3
GraphSAGE	56.6±1.3	53.5±0.4	55.4±1.0	57.8±1.3	49.9±0.3	55.5±1.6	59.4±1.4	59.6±0.2	47.1±1.0	49.3±0.8	49.2±1.7	52.7±1.3	53.8±1.0
GAT	57.1±0.9	56.0±0.8	58.5±1.8	56.4±2.2	48.7±1.1	51.6±2.8	62.5±1.8	57.1±0.7	46.2±2.2	47.2±1.4	51.7±2.6	53.6±0.9	53.9±1.6
Mean-Teacher	57.0±4.6	53.8±3.3	55.6±3.5	54.2±2.5	47.6±3.8	49.7±1.5	56.2±3.4	59.1±4.8	48.5±3.4	52.9±5.2	51.4±3.3	53.1±4.7	53.2±3.7
InfoGraph	57.0±2.7	55.7±2.2	60.1±2.6	60.0±3.0	48.9±2.0	51.2±1.7	60.6±1.1	61.8±1.9	45.4±2.3	46.3±1.2	53.2±2.0	53.5±0.8	54.5±1.9
TGNN	53.2±5.6	48.8±1.7	54.0±5.2	54.2±1.7	46.2±4.0	47.9±0.5	55.1±5.1	50.8±3.6	56.8±4.0	53.2±5.6	56.8±3.0	48.1±4.5	53.1±3.7
PLUE	57.9±1.2	56.4±1.3	60.0±1.9	59.1±1.6	49.1±0.6	53.2±1.8	60.8±1.5	52.3±3.7	48.1±3.7	52.1±4.1	52.7±1.5	53.9±2.2	54.6±2.1
GALA	59.7±0.6	56.9±0.9	58.8±1.1	53.8±0.4	51.7±1.2	55.9±1.7	61.0±0.5	62.7±0.7	53.0±0.6	54.5±1.0	53.5±0.5	56.4±0.8	**56.5**±0.8

F0, F1, F2, and F3 are split by the graph density.

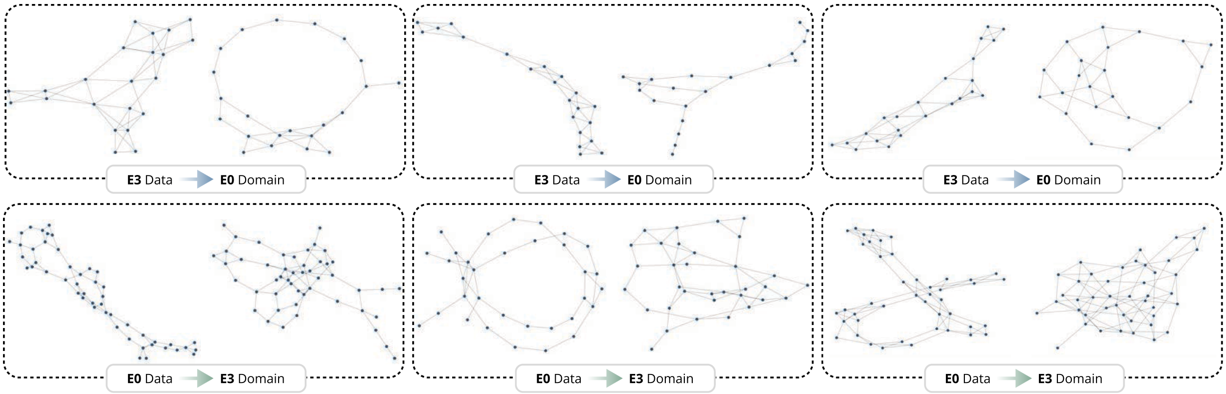


Fig. 3. Visualization of diffusion adaptation on ENZYMES. E0 and E3 are subsets of ENZYMES, with E0 being sparser and E3 being denser. The diffusion process can reconstruct the graph while preserving semantics. As $\mathbf{E3} \rightarrow \mathbf{E0}$, the graphs become sparser. As $\mathbf{E0} \rightarrow \mathbf{E3}$, the graphs become denser.

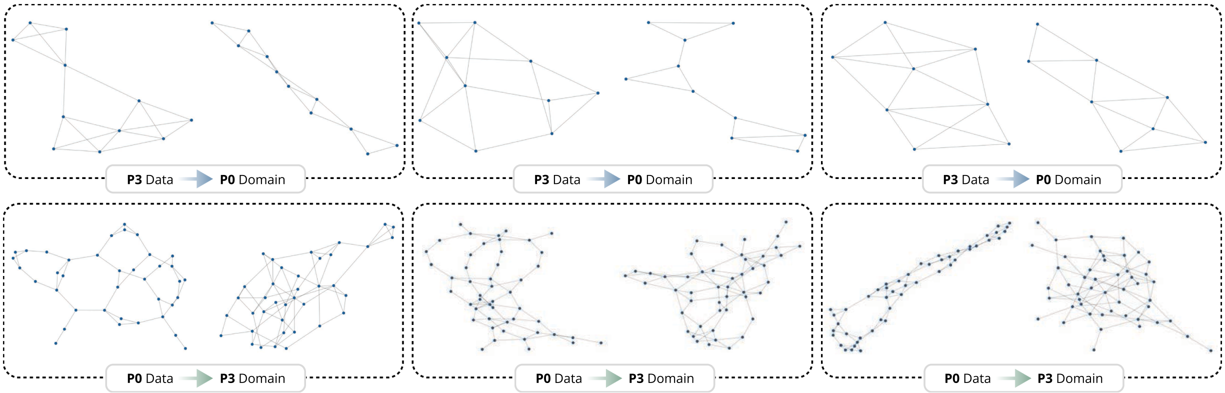


Fig. 4. Visualization of diffusion adaptation on PROTEINS. P0 and P3 are subsets of PROTEINS, with P0 being sparser and P3 being denser. The diffusion process can reconstruct the graph while preserving semantics. As $\mathbf{P3} \rightarrow \mathbf{P0}$, the graphs become sparser. As $\mathbf{P0} \rightarrow \mathbf{P3}$, the graphs become denser.

TABLE V
THE CLASSIFICATION RESULTS (IN %) ON COX2 AND BZR (SOURCE → TARGET)

Methods	C→CM	CM→C	B→BM	BM→B	Avg.
GCN	54.1±2.6	46.6 ±4.1	51.3 ±2.3	62.8 ±3.7	53.7±3.2
GIN	51.1±2.2	46.4±4.6	48.1±3.6	65.6±2.8	52.8±3.3
GraphSAGE	49.2±3.4	42.9±3.9	47.3±1.5	67.5±3.1	51.7±3.0
GAT	52.0±1.8	48.9±3.7	48.4±2.2	61.3±4.2	52.6±3.0
Mean-Teacher	53.0±2.3	42.6±4.9	50.6±2.1	57.6±4.3	50.9±3.4
InfoGraph	45.9±3.4	48.9±3.3	51.9±3.2	65.2±4.7	53.4±3.6
TGNN	48.3±4.2	52.1±5.6	46.4±2.7	68.5±4.3	53.8±4.2
PLUE	54.4±1.4	41.7±3.2	49.8±2.8	74.8±1.5	55.2±2.2
GALA	56.6±0.7	59.1±2.6	53.2±2.0	73.2±3.1	**60.5**±2.1

C, CM, B, and BM are for COX2, COX2_MD, BZR, and BZR_MD datasets.

proposed pseudo-label methods significantly helps the model learn more robust representations during the adaptation process, thereby enhancing the adaptation of the model.

D. Visualization

In order to further analyze the effectiveness of GALA, we investigate the two newly proposed modules, i.e. the graph diffusion and the graph jigsaw.

Graph Diffusion: To validate the effects of diffusion-based data adaptation, we performed visual verification on the target domain data in ENZYMES and PROTEINS. As shown in Figs. 3 and 4, diffusion-based data adaptation makes the target graph more compatible with the source model. Graph diffusion enables domain transfer for graph data and the generation of source-style graphs.

Graph Diffusion modifies the graph structure of a target domain by adding or removing edges between nodes while preserving the original semantic information of the graph. This allows the target graph to become more similar to the source domain, facilitating more accurate predictions by models trained on the source domain.

Graph Jigsaw: We visualize the graph jigsaw process. As in Fig. 5, it can be observed that our proposed graph jigsaw is able to adaptive select subgraphs and perform exchanges between confident and unconfident graphs. In graph jigsaw, we randomly chose a subgraph $\hat{G}^{c}_{j,2}$ from each confident graph $\hat{G}^{c}_{j}$, and exchanged with an unconfident graph $\hat{G}^{u}_{k}$ to generate an augmented graph $\hat{G}^{c}_{j}$. The visualization elucidates the effective execution of graph jigsaw, like a jigsaw puzzle game, by employing the copy-and-paste mechanism with subgraphs.

The exchange of subgraphs and establishment of connections generate new graph data, which helps explore data samples.

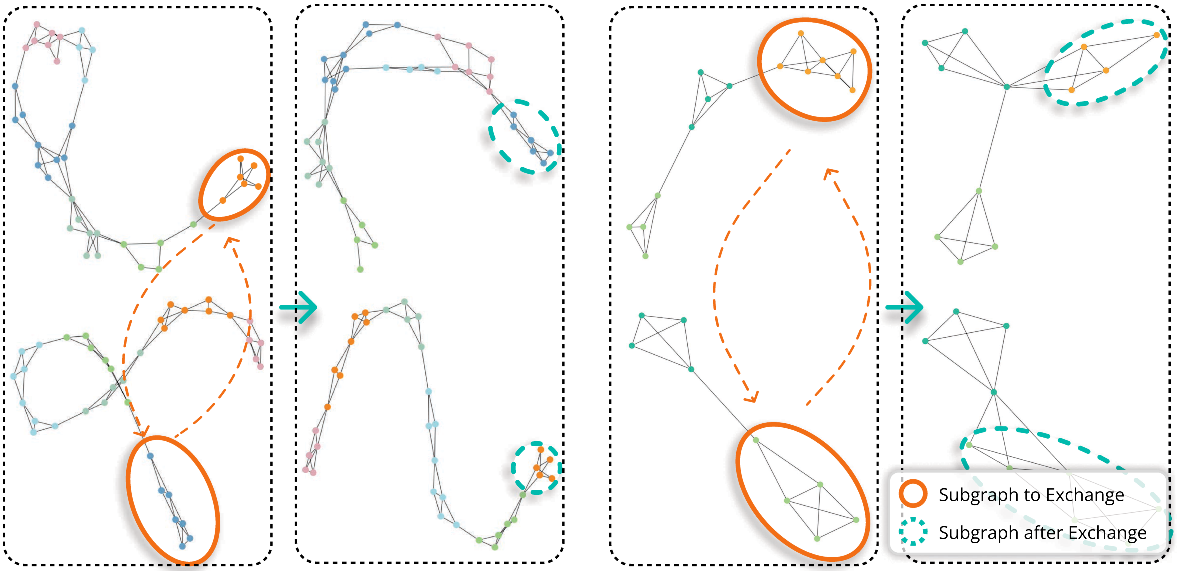


Fig. 5. Visualization of graph jigsaw on ENZYMES. Graph jigsaw, combined with consistency learning, leverages unlabeled target data to enhance the generalization capacity.

TABLE VI
ABLATION STUDIES OF DIFFERENT SUBMODULES

	Methods	ENZYMES	Mutagenicity	PROTEINS	FRANKENSTEIN	COX2	BZR	Avg.
M_1	GALA *w/o* GDM	$27.7_{\pm 3.1}$	$67.2_{\pm 1.8}$	$59.7_{\pm 3.7}$	$55.6_{\pm 0.7}$	$56.1_{\pm 1.9}$	$62.0_{\pm 3.4}$	$54.7_{\pm 2.4}$
M_2	GALA *w/o* UPCL	$28.3_{\pm 1.5}$	$67.5_{\pm 1.5}$	$60.2_{\pm 2.1}$	$55.4_{\pm 0.5}$	$56.5_{\pm 1.6}$	$62.3_{\pm 2.9}$	$55.0_{\pm 1.7}$
M_3	GALA *w/o* CLGJ	$28.7_{\pm 2.5}$	$67.9_{\pm 2.0}$	$60.0_{\pm 2.4}$	$56.1_{\pm 0.9}$	$57.2_{\pm 2.2}$	$63.0_{\pm 2.8}$	$55.5_{\pm 2.1}$
M_4	GALA (ours)	$30.7_{\pm 2.2}$	$68.6_{\pm 1.6}$	$61.2_{\pm 2.4}$	$56.5_{\pm 0.8}$	$57.9_{\pm 1.7}$	$63.2_{\pm 2.6}$	$56.3_{\pm 1.9}$

With graph jigsaw, the model learns more robust representations by consistency learning between the original graphs and post-exchange graphs, therefore enhancing the generalization ability on the target domain.

E. Ablation Study

We investigate the effectiveness of GALA in three aspects: the graph diffusion model for adaptation (GDM), unbiased pseudo-labeling with curriculum learning (UPCL), and consistency learning with graph jigsaw (CLGJ). The average classification results on two datasets are summarized in Table VI and we have four results. *First*, it can be observed that the whole model M_4 achieves the best performance. As such, adaptation is required from both data and models. The collaborative interaction of these three modules can enhance the effectiveness. *Second*, different components can play distinct roles. GDM appears to be a notably effective component (M_1), suggesting that data adaptation serves a crucial role in handling distribution shifts, the significant challenge in source-free domain adaptation. *Third*, the role of UPCL varies in magnitude, depending on the bias present in pseudo-labels. plays a more important role in ENZYMES (M_2), indicating a more serious biased of pseudo-labels. Nonetheless, consistently, they all play a significant role in the overall result of source-free domain adaptation. *Fourth*, we can observe that CLGJ shows relatively consistent efficacy across both datasets (M_3), which represents the gain from consistency learning with graph jigsaw.

In addition, to empirically analyze the effectiveness of UPCL, we conducted studies on the ENZYMES (E0→E3), focusing on the confidence set selection in the target domain. As shown in Fig. 8, the traditional pseudo-labeling methods can generate biased confidence sets that are skewed toward easier categories, which negatively affects adaptation in the target domain. In contrast, our approach to unbiased pseudo-labeling shows significant effectiveness in preventing such biases.

F. Scalability Analysis

Following to the experimental setup in Section V-B, we analyze the scalability of our proposed GALA using the PROTEINS dataset, as shown in Fig. 9. The experiments are conducted on a single RTX 3090 GPU, where we sample data from different scales of graph data and report the adaptation latency per batch. The results show that our method exhibits high efficiency as graph size increases, outperforming other methods. This highlights the ability of our method to achieve efficiency and high performance when handling large-scale graph data.

G. Sensitivity Analysis

Analysis of pseudo-label threshold: We first vary the initial pseudo-label threshold $\alpha(0)$. Combined with class-wise

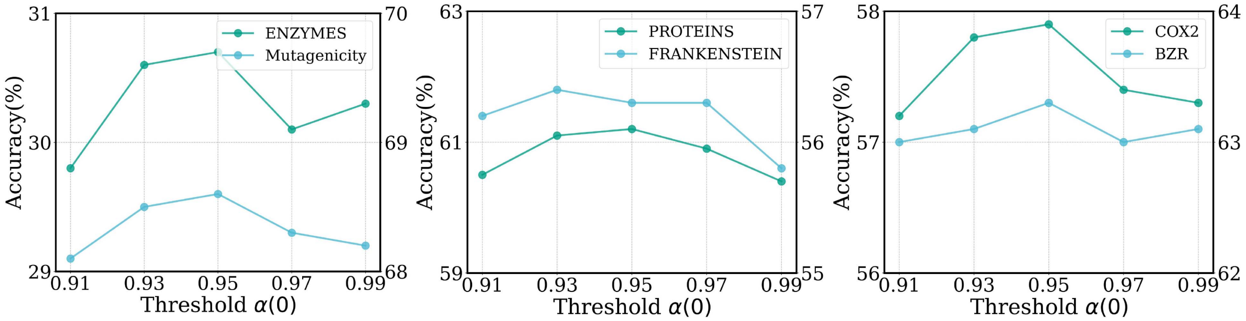


Fig. 6. Sensitivity analysis of the pseudo-label threshold.

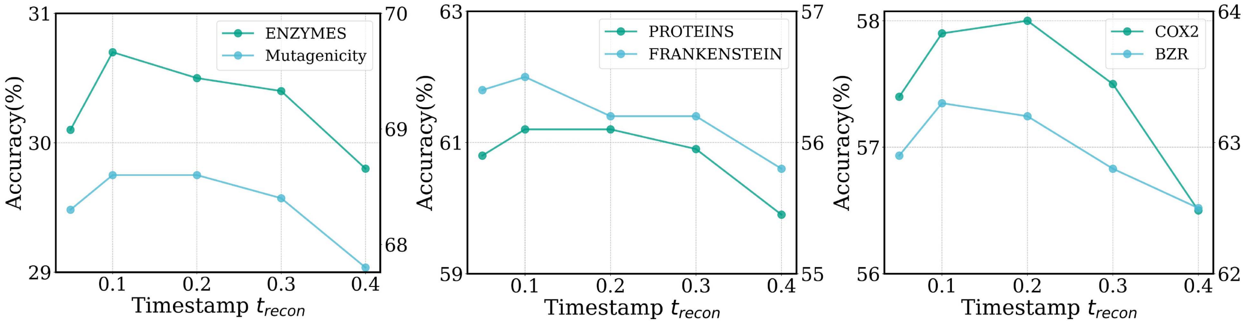
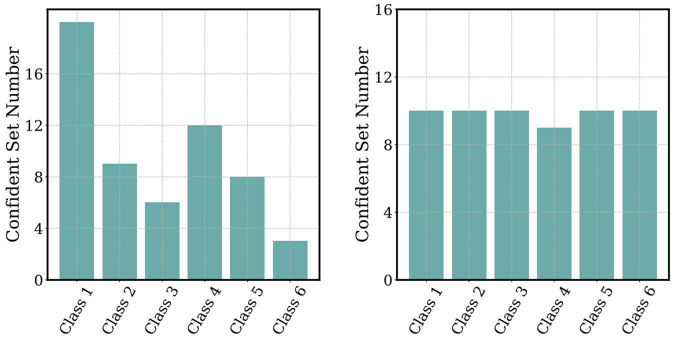


Fig. 7. Sensitivity analysis of the reconstruction timestamp.



(a) Pseudo-labeling (b) Unbiased Pseudo-labeling

Fig. 8. Validation on ENZYMES. Pseudo-labeling with a fixed threshold can be biased towards easy classes (left). Our unbiased pseudo-labeling can provide more unbiased pseudo-labeling (right).

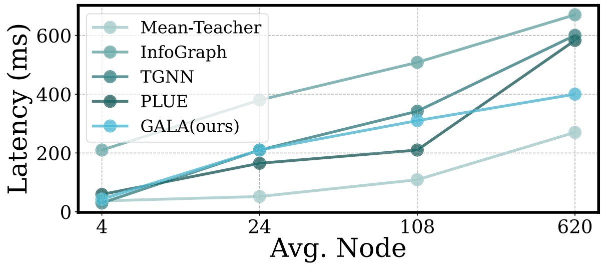


Fig. 9. Scalability analysis. We evaluate the adaptation latency for a single batch across different graph sizes and methods.

accuracy and training epochs, $\alpha(0)$ determines the confidence threshold. Its impact on final average accuracy is investigated in Fig. 6. We test $\alpha(0)$ from 0.91 to 0.99 with uniform sampling intervals. As $\alpha(0)$ from 0.91 to 0.95, the final accuracy improves on all datasets. This indicates that relatively clean pseudo-labels benefit model adaptation. However, when $\alpha(0)$ is further increased to 0.99, there is a decrease in accuracy. This suggests that an excessively strict pseudo-label threshold can limit the model's ability to learn from the target domain. Therefore, in order to make a balance between these two considerations, we take $\alpha(0) = 0.95$ as the default value, enabling us to maximize the utilization of the information within the target domain.

Analysis of reconstruction timestamp: We investigated the impact of reconstruction timestamp t_{recon}, which is the end point of forward SDE and the start point of reverse SDE. t_{recon} influence semantic information preservation during reconstruction. We vary t_{recon} in $[0.05, 0.1, 0.2, 0.3, 0.4]$ to investigate the potential effects arising from diffusion reconstruction. As shown in Fig. 7, the optimal average accuracy is achieved when the t_{recon} is set to 0.1. As t_{recon} increases, the overall accuracy rises, which shows the necessity of employing data domain transfer based on graph diffusion for the target graph data. However, as t_{recon} continues to increase, we observe varying degrees of decline in overall accuracy. This suggests that an excessive alteration of the information in the target graph data can generate side effects, impacting the original semantic information within the graph. Insufficient diffusion leads to inadequate data adaptation, while excessive diffusion hinders the preservation of graph semantics. They both result in an adverse impact on accuracy. This analysis reveals that transforming target graphs into source-style graphs carries a potential risk of information loss due to over-reconstruction. GALA can transfer graph semantics with limited information loss, which can improve the graph domain adaptation performance.

VI. Conclusion and Future Works

This paper studies source-free domain adaptation, a practical scenario in graph domain adaptation, where the source data is inaccessible during adaptation, resulting in the domain shift and label scarcity of the target domain. We propose GALA to address these problems. GALA utilizes graph score-based diffusion for domain alignment, unbiased pseudo-labeling with curriculum learning, and consistency learning with graph jigsaw. Extensive experiments demonstrate the effectiveness of GALA.

Future work: Our methods could be applied to heterogeneous graphs for complex real-world environments in future work. In heterogeneous graphs, capturing intricate structural and semantic information remains a challenge. In addition, the diversity of node and edge types in heterogeneous graphs, each potentially carrying different types of information, increases the risk of information loss during the reconstruction process. These challenges deserve further investigation.

References

[1] M. Zhang, Z. Cui, M. Neumann, and Y. Chen, "An end-to-end deep learning architecture for graph classification," in *Proc. Assoc. Advance. Artif. Intell.*, 2018, pp. 4438–4445.

[2] Z. Ying, J. You, C. Morris, X. Ren, W. Hamilton, and J. Leskovec, "Hierarchical graph representation learning with differentiable pooling," in *Proc. Neural Inf. Process. Syst.*, 2018, pp. 4805–4815.

[3] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip, "A comprehensive survey on graph neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 1, pp. 4–24, Jan. 2021.

[4] H. Gao, Y. Liu, and S. Ji, "Topology-aware graph pooling networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 12, pp. 4512–4518, Dec. 2021.

[5] C. Lu, Q. Liu, C. Wang, Z. Huang, P. Lin, and L. He, "Molecular property prediction: A multilevel quantum interactions modeling perspective," in *Proc. Assoc. Advance. Artif. Intell.*, 2019, pp. 1052–1060.

[6] S. Kim, D. Lee, S. Kang, S. Lee, and H. Yu, "Learning topology-specific experts for molecular property prediction," in *Proc. Assoc. Advance. Artif. Intell.*, 2023, pp. 8291–8299.

[7] J. Yoo, S. Shim, and U. Kang, "Model-agnostic augmentation for accurate graph classification," in *Proc. ACM Web Conf.*, 2022, pp. 1281–1291.

[8] W. Ju, J. Yang, M. Qu, W. Song, J. Shen, and M. Zhang, "KGNN: Harnessing kernel-based networks for semi-supervised graph classification," in *Proc. ACM Int. Conf. Web Search Data Mining*, 2022, pp. 421–429.

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

[10] N. Yin et al., "Deal: An unsupervised domain adaptive framework for graph-level classification," in *Proc. ACM Int. Conf. Multimedia*, 2022.

[11] N. Yin et al., "CoCo: A coupled contrastive framework for unsupervised domain adaptive graph classification," 2023, *arXiv:2306.04979*.

[12] Y. Tan, Y. Liu, G. Long, J. Jiang, Q. Lu, and C. Zhang, "Federated learning on non-IID graphs via structural knowledge sharing," in *Proc. Assoc. Advance. Artif. Intell.*, 2023, pp. 9953–9961.

[13] M. Lin, W. Li, D. Li, Y. Chen, G. Li, and S. Lu, "Multi-domain generalized graph meta learning," in *Proc. Assoc. Advance. Artif. Intell.*, 2023, pp. 4479–4487.

[14] Y. Sun, X. Wang, Z. Liu, J. Miller, A. Efros, and M. Hardt, "Test-time training with self-supervision for generalization under distribution shifts," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 9229–9248.

[15] J. Liang, D. Hu, and J. Feng, "Do we really need to access the source data? Source hypothesis transfer for unsupervised domain adaptation," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 6028–6039.

[16] Z. Nado, S. Padhy, D. Sculley, A. D'Amour, B. Lakshminarayanan, and J. Snoek, "Evaluating prediction-time batch normalization for robustness under covariate shift," 2020, *arXiv: 2006.10963*.

[17] J. Gao, J. Zhang, X. Liu, T. Darrell, E. Shelhamer, and D. Wang, "Back to the source: Diffusion-driven adaptation to test-time corruption," in *Proc. Comput. Vis. Pattern Recognit.*, 2023, pp. 11786–11796.

[18] M. Litrico, A. Del Bue, and P. Morerio, "Guiding pseudo-labels with uncertainty estimation for source-free unsupervised domain adaptation," in *Proc. Comput. Vis. Pattern Recognit.*, 2023, pp. 7640–7650.

[19] 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.

[20] Y. You, T. Chen, Z. Wang, and Y. Shen, "Graph domain adaptation via theory-grounded spectral regularization," in *Proc. Int. Conf. Learn. Representations*, 2022.

[21] J. Pang, Z. Wang, J. Tang, M. Xiao, and N. Yin, "SA-GDA: Spectral augmentation for graph domain adaptation," in *Proc. ACM Int. Conf. Multimedia*, 2023, pp. 309–318.

[22] X. Luo et al., "Dualgraph: Improving semi-supervised graph classification via dual contrastive learning," in *Proc. IEEE Int. Conf. Data Eng.*, 2022, pp. 699–712.

[23] W. Ju et al., "A comprehensive survey on deep graph representation learning," *Neural Netw.*, vol. 173, 2024, Art. no. 106207.

[24] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How powerful are graph neural networks?," in *Proc. Int. Conf. Learn. Representations*, 2019.

[25] L. Cosmo, G. Minello, M. Bronstein, E. Rodolà, L. Rossi, and A. Torsello, "Graph kernel neural networks," 2021, *arXiv:2112.07436*.

[26] J. Lee, I. Lee, and J. Kang, "Self-attention graph pooling," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 3734–3743.

[27] Q. Sun et al., "SUGAR: Subgraph neural network with reinforcement pooling and self-supervised mutual information mechanism," in *Proc. ACM Web Conf.*, 2021, pp. 2081–2091.

[28] F. M. Bianchi, D. Grattarola, and C. Alippi, "Spectral clustering with graph neural networks for graph pooling," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 874–883.

[29] J. Li, Y. Rong, H. Cheng, H. Meng, W. Huang, and J. Huang, "Semi-supervised graph classification: A hierarchical graph perspective," in *Proc. ACM Web Conf.*, 2019, pp. 972–982.

[30] W. Ju et al., "GLCC: A general framework for graph-level clustering," in *Proc. Assoc. Advance. Artif. Intell.*, 2023, pp. 4391–4399.

[31] M. Long, Y. Cao, J. Wang, and M. Jordan, "Learning transferable features with deep adaptation networks," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 97–105.

[32] D.-H. Lee et al., "Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2013, p. 896.

[33] M. Assran et al., "Semi-supervised learning of visual features by non-parametrically predicting view assignments with support samples," in *Proc. Int. Conf. Comput. Vis.*, 2021, pp. 8423–8432.

[34] B. Zhang et al., "Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling," in *Proc. Neural Inf. Process. Syst.*, 2021, pp. 18408–18419.

[35] J. Li, Y. Huang, H. Chang, and Y. Rong, "Semi-supervised hierarchical graph classification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 5, pp. 6265–6276, May 2023.

[36] X. Xu, C. Deng, Y. Xie, and S. Ji, "Group contrastive self-supervised learning on graphs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 3, pp. 3169–3180, Mar. 2023.

[37] Y. Xie, Z. Xu, J. Zhang, Z. Wang, and S. Ji, "Self-supervised learning of graph neural networks: A unified review," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 2, pp. 2412–2429, Feb. 2023.

[38] R. Zhang, Y. Zhang, C. Lu, and X. Li, "Unsupervised graph embedding via adaptive graph learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 4, pp. 5329–5336, Apr. 2023.

[39] R. Wang, J. Yan, and X. Yang, "Unsupervised learning of graph matching with mixture of modes via discrepancy minimization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 8, pp. 10500–10518, Aug. 2023.

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

[41] Q. Wu, H. Zhang, J. Yan, and D. Wipf, "Handling distribution shifts on graphs: An invariance perspective," in *Proc. Int. Conf. Learn. Representations*, 2021.

[42] H. Liu, B. Hu, X. Wang, C. Shi, Z. Zhang, and J. Zhou, “Confidence may cheat: Self-training on graph neural networks under distribution shift,” in *Proc. ACM Web Conf.*, 2022, pp. 1248–1258.
[43] Y. Wu, X. Wang, A. Zhang, X. He, and T.-S. Chua, “Discovering invariant rationales for graph neural networks,” in *Proc. Int. Conf. Learn. Representations*, 2021.
[44] D. Buffelli, P. Liò, and F. Vandin, “Sizeshiftreg: A regularization method for improving size-generalization in graph neural networks,” in *Proc. Neural Inf. Process. Syst.*, 2022, pp. 31871–31885.
[45] S. Liu et al., “Structural re-weighting improves graph domain adaptation,” in *Proc. Int. Conf. Mach. Learn.*, 2023, pp. 21778–21793.
[46] J. Chen, F. Dai, X. Gu, J. Zhou, B. Li, and W. Wang, “Universal domain adaptive network embedding for node classification,” in *Proc. ACM Int. Conf. Multimedia*, 2023, pp. 4022–4030.
[47] R. Zhu, X. Yu, and S. Li, “Progressive mix-up for few-shot supervised multi-source domain transfer,” in *Proc. Int. Conf. Learn. Representations*, 2022.
[48] K. Saito, Y. Ushiku, and T. Harada, “Asymmetric tri-training for unsupervised domain adaptation,” in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 2988–2997.
[49] S. Schneider, E. Rusak, L. Eck, O. Bringmann, W. Brendel, and M. Bethge, “Improving robustness against common corruptions by covariate shift adaptation,” in *Proc. Neural Inf. Process. Syst.*, 2020, pp. 11539–11551.
[50] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, “Score-based generative modeling through stochastic differential equations,” 2020, *arXiv: 2011.13456*.
[51] C. Meng et al., “SDEdit: Guided image synthesis and editing with stochastic differential equations,” in *Proc. Int. Conf. Learn. Representations*, 2022.
[52] H. Huang, L. Sun, B. Du, Y. Fu, and W. Lv, “GraphGDP: Generative diffusion processes for permutation invariant graph generation,” in *Proc. IEEE Int. Conf. Data Mining*, 2022, pp. 201–210.
[53] S. Li, X. He, W. Song, A. Hao, and H. Qin, “Graph diffusion convolutional network for skeleton based semantic recognition of two-person actions,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 7, pp. 8477–8493, Jul. 2023.
[54] K. Sohn et al., “FixMatch: Simplifying semi-supervised learning with consistency and confidence,” in *Proc. Neural Inf. Process. Syst.*, 2020, pp. 596–608.
[55] M. Navarro and S. Segarra, “GraphMAD: Graph mixup for data augmentation using data-driven convex clustering,” in *Proc. Int. Conf. Acoustics, Speech, Signal Process.*, 2023, pp. 1–5.
[56] X. Han, Z. Jiang, N. Liu, and X. Hu, “G-Mixup: Graph data augmentation for graph classification,” in *Proc. Int. Conf. Mach. Learn.*, 2022, pp. 8230–8248.
[57] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, “Fast unfolding of communities in large networks,” *J. Statist. Mechanics: Theory Experiment*, vol. 2008, 2008, Art. no. P10008.
[58] I. Schomburg et al., “Brenda, the enzyme database: Updates and major new developments,” *Nucleic Acids Res.*, 2004.
[59] J. Kazius, R. McGuire, and R. Bursi, “Derivation and validation of toxicophores for mutagenicity prediction,” *J. Med. Chem.*, vol. 32, pp. D431–D433, 2005.
[60] K. M. Borgwardt, C. S. Ong, S. Schönauer, S. Vishwanathan, A. J. Smola, and H.-P. Kriegel, “Protein function prediction via graph kernels,” *Bioinformatics*, vol. 21, pp. i1–i10, 2005.
[61] F. Orsini, P. Frasconi, and L. De Raedt, “Graph invariant kernels,” in *Proc. Int. Joint Conf. Artif. Intell.*, 2015, pp. 3756–3762.
[62] J. J. Sutherland, L. A. O’brien, and D. F. Weaver, “Spline-fitting with a genetic algorithm: A method for developing classification structure-activity relationships,” *J. Chem. Inf. Comput. Sci.*, vol. 43, pp. 1906–1915, 2003.
[63] M. Welling and T. N. Kipf, “Semi-supervised classification with graph convolutional networks,” in *Proc. Int. Conf. Learn. Representations*, 2016.
[64] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, “How powerful are graph neural networks?,” in *Proc. Int. Conf. Learn. Representations*, 2018.
[65] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, “Graph attention networks,” in *Proc. Int. Conf. Learn. Representations*, 2018.
[66] W. Hamilton, Z. Ying, and J. Leskovec, “Inductive representation learning on large graphs,” in *Proc. Neural Inf. Process. Syst.*, 2017, pp. 1024–1034.
[67] A. Tarvainen and H. Valpola, “Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results,” in *Proc. Neural Inf. Process. Syst.*, 2017, pp. 1195–1204.
[68] F.-Y. Sun, J. Hoffmann, V. Verma, and J. Tang, “Infograph: Unsupervised and semi-supervised graph-level representation learning via mutual information maximization,” in *Proc. Int. Conf. Learn. Representations*, 2020.
[69] W. Ju et al., “TGNN: A joint semi-supervised framework for graph-level classification,” in *Proc. Int. Joint Conf. Artif. Intell.*, 2022, pp. 2122–2128.

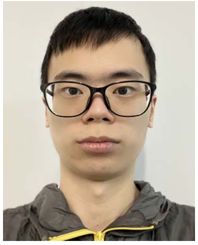
Junyu Luo is working toward the graduate degree with the School of Computer Science, Peking University, Beijing, China. His research focuses on machine learning on graphs, computer vision, and efficient AI.

Yiyang Gu received the BS degree in computer science from Peking University, Beijing, China, in 2021. He is currently working toward the PhD degree in computer science with Peking University. His research interests include data mining, graph machine learning, self-supervised learning, and bioinformatics.

Xiao Luo received the BS degree in mathematics from Nanjing University, Nanjing, China, in 2017, and the PhD degree from the School of Mathematical Sciences, Peking University, Beijing, China. He is a postdoctoral researcher with the Department of Computer Science, University of California, Los Angeles, USA. His research interests include machine learning on graphs, image retrieval, statistical models, and bioinformatics.

Wei Ju (Member, IEEE) received the BS degree in mathematics from Sichuan University, Sichuan, China, in 2017, and the PhD degree in computer science from Peking University, Beijing, China, in 2022. He is currently a postdoc research fellow in computer science with Peking University. His current research interests lie primarily in the area of machine learning on graphs including graph representation learning and graph neural networks, and interdisciplinary applications, such as knowledge graphs, bioinformatics, drug discovery, and recommender systems. He has published more than 30 papers in top-tier venues and has won the best paper finalist in IEEE ICDM 2022.

Zhiping Xiao received the BS degree from Peking University, Beijing, China, in 2016, the MS degree in computer science from the University of California, Berkeley, in 2018, and the PhD degree in computer science from the University of California, Los Angeles, in 2024. She is currently a research scientist with the Computer Science Department, University of California at Los Angeles. Her research interests include graph representation learning, graph neural networks, and applications, such as social network analysis and biomedicine.



Yusheng Zhao is working toward the graduate degree with the School of Computer Science, Peking University, Beijing, China. His research interests include machine learning with graphs, vision and language, and adversarial learning.

Jingyang Yuan received the BS degree in computer science from Peking University, in 2022. He is currently working toward the graduate degree with the School of Computer Science, Peking University. His research interests include graph representation learning, neural physical simulation, large language models, etc.

Ming Zhang (Member, IEEE) received the BS, MS, and PhD degrees in computer science from Peking University, Beijing, China, in 1988, 1991, and 2005, respectively. She is currently a full professor with the School of Computer Science, Peking University. She has authored or coauthored more than 200 research papers on text mining and machine learning in the top journals and conferences. She is the leading author of several textbooks on data structures and algorithms in Chinese and the corresponding course is awarded as the National Elaborate Course, National Boutique Resource Sharing Course, National Fine-designed Online Course, and National First Class Undergraduate Course by Ministry of Education (MOE) China. She is a member of the Advisory Committee of the Ministry of Education in China. She is one of the 15 members of the ACM/IEEE CC2020 Steering Committee. She received the Best Paper Award of the International Conference on Machine Learning (ICML) 2014 and the Best Paper Nominee of World Wide Web (WWW) 2016.