

# Joint Domain Adaptive Graph Convolutional Network

Niya Yang<sup>1</sup>, Ye Wang<sup>2</sup>, Zhizhi Yu<sup>1</sup>, Dongxiao He<sup>1\*</sup>, Xin Huang<sup>3</sup> and Di Jin<sup>1</sup>

<sup>1</sup>College of Intelligence and Computing, Tianjin University, Tianjin, China

<sup>2</sup>Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, Changchun, China

<sup>3</sup>Department of Computer Science, Hong Kong Baptist University, Hong Kong, China  
 {niya\_yang1, yuzhizhi, hedongxiao, jindi}@tju.edu.cn, wangye1@ciomp.ac.cn, xinhuang@comp.hkbu.edu.hk

## Abstract

In the realm of cross-network tasks, graph domain adaptation is an effective tool due to its ability to transfer abundant labels from nodes in the source domain to those in the target domain. Existing adversarial domain adaptation methods mainly focus on domain-wise alignment. These approaches, while effective in mitigating the marginal distribution shift between the two domains, often ignore the integral aspect of structural alignment, potentially leading to negative transfer. To address this issue, we propose a joint adversarial domain adaptive graph convolutional network (JDA-GCN) that is uniquely augmented with structural graph alignment, so as to enhance the efficacy of knowledge transfer. Specifically, we construct a structural graph to delineate the interconnections among nodes within identical categories across the source and target domains. To further refine node representation, we integrate the local consistency matrix with the global consistency matrix, thereby leveraging the learning of the sub-structure similarity of nodes to enable more robust and effective representation of nodes. Empirical evaluation on diverse real-world datasets substantiates the superiority of our proposed method, marking a significant advancement over existing state-of-the-art graph domain adaptation algorithms.

## 1 Introduction

Domain adaptation (DA) techniques [Zhu *et al.*, 2020; Wei *et al.*, 2021] aim to transfer knowledge from a well-labeled source domain to a less-labeled target domain, addressing the challenge of distribution discrepancies. Recent advances have extended DA’s horizon, adapting its principles to graph-structured data [Liu *et al.*, 2023b; Xie *et al.*, 2020; Wu *et al.*, 2022]. This extension is pivotal, as graph-structured data often encapsulates complex relationships and patterns, making traditional DA techniques insufficient. In cross-network tasks like node classification, DA techniques help in effectively bridging the structural and distributional gaps between

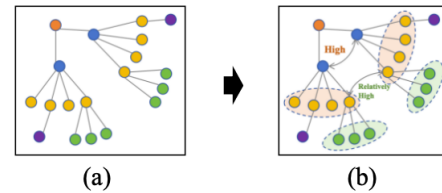


Figure 1: An illustrative example that two ‘blue’ nodes inherit high similarity as they share similar sub-structure.

networks, thereby enhancing the accuracy and reliability of node categorization. Furthermore, these advanced DA methods have fostered significant improvements in diverse real-world scenarios such as recommendation [Li *et al.*, 2020], article publication [Zhao *et al.*, 2021], and social risk identification [Wang *et al.*, 2020a].

Building upon the foundational principles of DA in graph-structured data, graph domain adaptation (GDA) [Liang *et al.*, 2023] represents a more targeted approach, focusing on minimizing disparities between graph domains. Particularly, adversarial domain adaptation [Ganin *et al.*, 2016] methods have been widely adopted in GDA due to their remarkable performance. This strategy ensures that the model captures the fundamental structures common to both source domain and target domain, enhancing transferability and reducing the distributional difference between these two domains through heuristic and adversarial approaches.

While adversarial domain adaptation methods have shown remarkable success [Wu *et al.*, 2020; Dai *et al.*, 2022], there are two key issues that are still insufficient explored. First, traditional graph convolutional networks (GCNs) [Kipf and Welling, 2016; Wang *et al.*, 2019] primarily focus on direct (local consistency) neighbor nodes for node embedding but neglect the indirect (global consistency) connections between nodes, which can be crucial for understanding complex structures. For instance, Figure 1(a) illustrates that two ‘blue’ nodes belonging to the same category, which are not connected to each other directly but have similar sub-structure. Existing GCNs may ignore the similarity relationship of these two nodes in the process of information propagation, leading to incomplete node embedding. Second, most adversarial domain adaptation methods only conduct domain-wise alignment to mitigate marginal distribution shift between the two domains. However, this simple alignment strategy may dam-

\* Corresponding author

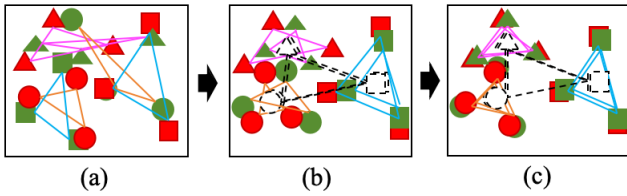


Figure 2: The red color denotes source domain, green color denotes target domain, black color denotes class center, and different shapes are various categories. (a) Domain-wise alignment for marginal distribution shift. (b) Constructing structural graph of class center and minimizing the loss of structural graph alignment between two domains to mitigate the marginal distribution shift. (c) Introducing the loss of structural graph alignment in the same category to prompt more sufficient transfer.

age latent discriminative structural graph in the same category and lead to corresponding negative transfer. As shown in Figure 2(a), we can find that nodes from the source and target domains align in marginal distribution, yet exhibit significant structural differences within the same category. To mitigate the distribution shift, we incorporate structural alignment of class centers between domains, as demonstrated in Figure 2(b). This strategy improves classification but reveals a new challenge: the distribution of nodes within the same category varies significantly between domains. For examples, the ‘circle’ class in the source domain is more decentralized compared to the target domain, whereas the ‘square’ class in the source domain is more compact. Additionally, the nodes of the ‘triangle’ class and the ‘circle’ class are closely situated, indicating the need for a more nuanced transfer strategy.

To address the aforementioned issues, we propose a novel joint domain adaptive graph convolutional networks for graph domain adaption (JDA-GCN) to capture the intricate structures within graph data more effectively. Initially, we construct a global similarity matrix by comparing the graph structures surrounded with each node. This process ensures that both global and local consistencies are considered, leading to the formation of a new propagation matrix applicable to both source and target domains. Furthermore, we address the challenge of classifying unlabeled nodes in target graph. We employ a non-parametric spherical  $k$ -means [Hornik *et al.*, 2012] to generate pseudo labels for these nodes. In addition, we seek to minimize the joint distribution differences between domains. It goes beyond mere domain-wise alignment by facilitating in-category structural graph transfer within the adversarial domain alignment framework. Specifically, in order to mitigate potential negative migration that can be caused by only considering marginal distribution alignment, we utilize a Gram Matrix [de Almeida *et al.*, 2008] to model the structural relationship of class centers and narrow the structural relationships discrepancy between the two domains, and further reduce the shift of node distribution by MMD [Gretton *et al.*, 2012] to jointly explore their latent intra-class structural graph transfer hidden feature space for the two domains to promote more sufficient transfer, as shown in Figure 2(c).

The main contributions are summarized as follows:

- We propose a joint adversarial domain adaptive graph convolutional network (JDA-GCN), augmented with

structural graph alignment in an adversarial graph domain adaptation framework. This augmentation significantly enhances knowledge transfer through exploring intrinsic graph structures, ensuring a more effective domain adaptation.

- We combine local consistency matrix and global consistency matrix to construct new feature propagation matrix, which ensures a more holistic node representation by capturing the comprehensive graph structure.
- Empirical evaluation on Citation and Blog datasets demonstrates the effectiveness of JDA-GCN. Our model not only demonstrates significant improvement in accuracy compared to existing methods but also showcases its robustness across various graph structures.

## 2 Preliminaries

**Notations.** A graph  $G$  is composed of node set  $V$  and edge set  $E$ , where  $V = \{v_1, v_2, \dots, v_N\}$ ,  $E = e_{\{ij\}} \subseteq V \times V$ .  $G$  can be represented by  $G = (A, X, Y)$ , where  $A = [a_{ij}] \in R^{N \times N}$ ,  $X = [x_{ij}] \in R^{N \times F}$  and  $Y = [y_{ij}] \in R^{N \times C}$  denote adjacency matrix, node feature matrix and label matrix, respectively.  $N = |V|$  is the size of node set,  $F$  is the dimension of node feature and  $C$  is the number of class. If  $(v_i, v_j) \in E$ ,  $a_{ij} = 1$ , otherwise  $a_{ij} = 0$ . If a node  $v_i \in V$  is associated with label  $l$ ,  $y_{il} = 1$ , otherwise  $y_{il} = 0$ .

**Problem Definition.** We focus on the downstream task of cross-network node classification to evaluate our proposed JDA-GCN. There are a labeled source domain  $G^s = (A^s, X^s, Y^s)$  and an unlabeled target domain  $G^t = (A^t, X^t)$ , where  $X^s$  and  $X^t$  are in the same feature space but have different joint distribution (i.e.,  $P^s(X^s, Y^s) \neq P^t(X^t, Y^t)$ ), and the classification task keeps unchanged across domains. Our goal is to predict  $Y^t$  by jointly learning on source and target domains with minimizing the difference between  $P^s(X^s, Y^s)$  and  $P^t(X^t, Y^t)$ .

## 3 Methodology

We first give a brief overview of our proposed approach, and then introduce the detailed descriptions of each component.

### 3.1 Overview

In order to leverage cross-domain graphs to learn a effective classifier for node classification, we propose a joint adversarial domain adaptive graph convolutional network (JDA-GCN) to reduce the distribution gap. The whole structure of the proposed approach is shown in Figure 3, which consists of three components: node embedding modeling, pseudo label modeling, as well as joint domain adaptation modeling. For node embedding modeling, we capture local and global consistency relationship to construct new feature propagation matrices to learn effective node embedding with parameter sharing. For pseudo label modeling, we obtain robust pseudo-labels for nodes of target domain by utilizing clustering centers of source domains and spherical  $k$ -means clustering. For joint domain adaptation modeling, we construct intra-class structural graph for two domains and fully consider structural graph transfer except for marginal distribution alignment to reduce the discrepancy of two domains.

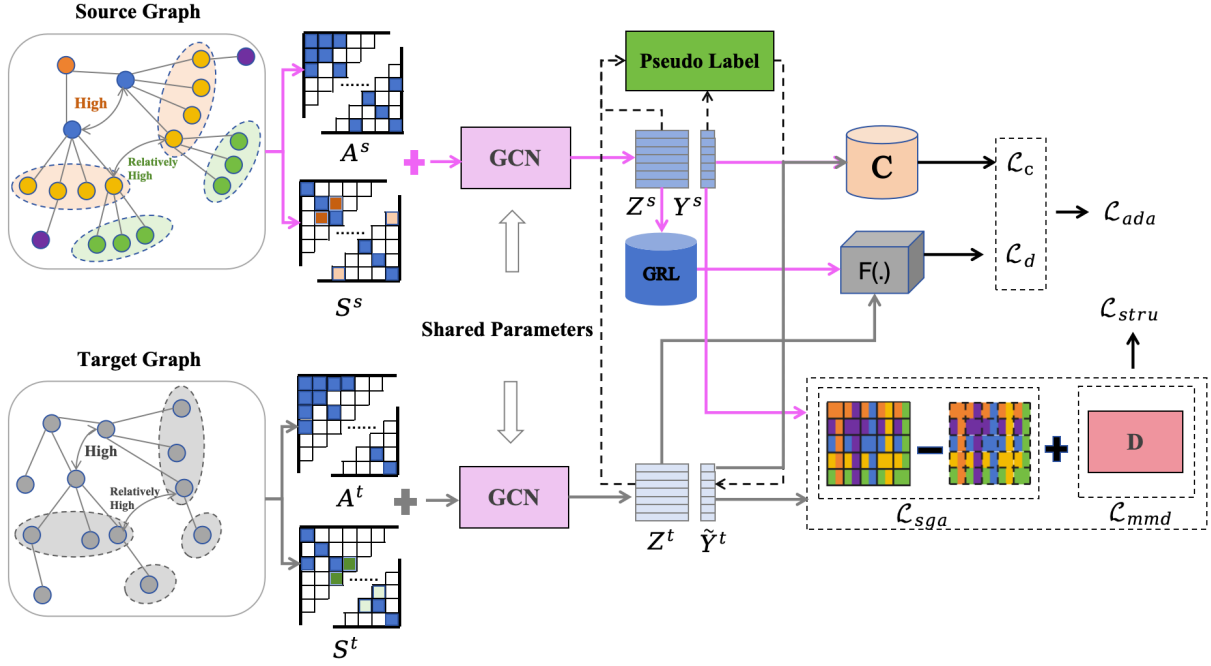


Figure 3: The architecture of JDA-GCN. We first combine global consistency matrix and local consistency matrix to construct a new feature propagation matrix to obtain effective node embedding. Then, we tag pseudo label for nodes in target domain, and utilize nodes and the corresponding labels to construct intra-class structural graph for source and target domains, thereby uniquely considering transferable structures except for domain-wise marginal alignment to prompt sufficient knowledge transfer.

### 3.2 Node Embedding Modeling

In order to encode the semantic information of each node, we propose a feature propagation matrix to capture both local and global information of the graph. Specifically, it mainly consists of two parts: local consistency relationship and global consistency relationship. For local consistency, we denote relationships of directly connected nodes by graph adjacency matrix  $A$ . For global consistency, inspired by studies on similarity [Jeh and Widom, 2002], that is, nodes surrounded with similar graph structures are likely to share the same label, we construct the global similarity matrix (i.e., the nodes surrounded with similar graph structures have high similarity).

**Global Consistency Matrix.** Random walk [Codling *et al.*, 2008] allows efficient exploration of graph structure, which visits the neighbors of the ego node with equal probability. Thus, the random walk probability can measure the probability of random walks from node  $v_i$  to node  $v_j$  for all paths with length  $l$ , which can be formulated as:

$$p(v_j|v_i, t_{v_i:v_j}^{(l)}) = \prod_{v_t \in t_{v_i:v_j}^{(l)}} \frac{1}{|N(v_t)|} \quad (1)$$

where  $v_t$  is the  $t$ -th node in path  $t_{v_i:v_j}^{(l)} : \{v_i, \dots, v_j\}$ ,  $t_{v_i:v_j}^{(l)}$  is one possible path of length  $l$ ,  $N(v_t)$  is the number of neighbor nodes of node  $v_t$ . In this way, we can obtain the random walk probability between all nodes in graph.

Then, inspired by SIMGA [Liu *et al.*, 2023a], which suggests the random walk probability of the node to all nodes can be seen as the node embedding. Thus, for nodes  $v_i$  and  $v_j$  and  $l \geq 0$ , the length- $l$  random walk probability from  $v_i$

to  $v_j$  under all paths  $t^{(l)}$  equals to the  $l$ -th layer embedding value  $z_{i,j}^{(l)}$  of node  $v_i$ :

$$z_{i,j}^{(l)} = p(v_j|v_i, t_{v_i:v_j}^{(l)}) \quad (2)$$

After obtaining node embedding, we can compute the global consistency matrix  $S$  by calculating the substructure similarity around the nodes. Formally, the similarity value  $S(v_i, v_j)$  of node  $v_i, v_j$  can be equivalent to the layer-wise summation of inner-product of node embedding  $z_i^{(l)}$  and  $z_j^{(l)}$  with a decay factor  $c$ :

$$S(v_i, v_j) = \sum_{l=1}^{\infty} c^l \cdot \langle z_i^{(l)}, z_j^{(l)} \rangle \quad (3)$$

Finally, we integrate the local and global matrices as the new propagation matrix  $H$  in GCN [Kipf and Welling, 2016]:

$$H = A + S \quad (4)$$

Thus, the  $l$ -th layer propagation rule can be defined as:

$$Z^{(l)} = \sigma \left( \tilde{D}^{-\frac{1}{2}} \tilde{H} \tilde{D}^{-\frac{1}{2}} Z^{(l-1)} W^{(l)} \right) \quad (5)$$

where  $\tilde{H} = H + I_N$ ,  $I_N \in R^{N \times N}$  is the identity matrix,  $\tilde{D}_{i,i} = \sum_j \tilde{H}_{i,j}$ . Accordingly,  $\tilde{D}^{-\frac{1}{2}} \tilde{H} \tilde{D}^{-\frac{1}{2}}$  is the normalized probability matrix,  $Z^{(l-1)}$  is the output of the  $(l-1)$  layer, and  $Z^{(0)} = X$ .  $W^{(l)}$  is the trainable parameters, and  $\sigma(\cdot)$  denotes the activation function.

### 3.3 Pseudo Label Modeling

We introduce the pseudo labeling strategy for unlabeled target graph. Classical MLP [Pinkus, 1999] is mostly used to generate pseudo labels, however, this method generally introduces noise to the domain transfer. Motivated by the principle that target nodes should align closely with source nodes of the same class in the embedding space, we propose a non-parametric strategy for generating pseudo labels.

First, we set the cluster number  $m^t$  of target domain equal to the class number  $m^s$  of source domain, and set class center  $O_{s,k}$  ( $k = 1, \dots, m^s$ ) of source domain as initial clusters center  $O_{t,k}^{(0)}$  ( $k = 1, \dots, m^t$ ) of target domain.

Then, we perform spherical  $k$ -means iteratively: 1) assigning pseudo-labels via minimum-distance classifier. 2) computing new cluster centers. Iteration will stop when all class centers converge.

$$\hat{y}_i^t = \operatorname{argmin}_k \operatorname{Dist} \left( O_{t,k}^{(m)}, z_i^t \right) \quad (6)$$

$$\operatorname{Dist} \left( O_{t,k}^{(m)}, z_i^t \right) = \frac{1}{2} \left( 1 - \frac{O_{t,k}^{(m)T} z_i^t}{\left| O_{t,k}^{(m)} \right| \cdot \left| z_i^t \right|} \right) \quad (7)$$

$$O_{t,k}^{(m+1)} = \frac{\sum_{i=1}^{N_t} B(\hat{y}_{ik}^t = 1) \cdot z_i^t}{\sum_{i=1}^{N_t} B(\hat{y}_{ik}^t = 1)} \quad (8)$$

where  $m$  denotes the number of current iterations,  $O_{t,k}^{(m)}$  is clustering center of class  $k$  for  $m$  iterations in target domain, and  $B(\cdot)$  is a boolean function.

Further, to reduce the noise in target pseudo labels, we remove the ambiguous samples far from its assigned clustered centers. Concretely, node  $v_i^t$  ( $i = 1, \dots, N_t$ ) will be removed from class  $k$  in the target domain when the cosine similarity between its feature and its assigned cluster center is below a manually set threshold  $d$ . Thus, we set pseudo-labels for all nodes in the target domain in an optimal way.

### 3.4 Joint Domain Adaptation Modeling

Joint domain adaptation consists of two components: adversarial domain adaptation and structural graph alignment.

**Marginal Adversarial Domain Adaptation.** Inspired by UDA-GCN [Wu *et al.*, 2020], we employ marginal adversarial domain adaptation to capture domain-invariant of node embedding between the source and target domains. Specifically, the marginal adversarial domain adaptation consists of three components: a domain discriminator  $d$ , a source domain classifier  $c_s$ , as well as a target domain classifier  $c_t$ .

The source classifier loss  $\zeta_{c_s}(Z^s, Y^s)$  and the target classifier loss  $\zeta_{c_t}(Z^t)$  are minimized by using cross-entropy loss:

$$\zeta_{c_s}(Z^s, Y^s) = -\frac{1}{N_s} \sum_{i=1}^{N_s} y_i^s \log(\hat{y}_i^s) \quad (9)$$

$$\zeta_{c_t}(Z^t) = -\frac{1}{N_t} \sum_{i=1}^{N_t} \hat{y}_i^t \log(\hat{y}_i^t) \quad (10)$$

where  $y_i^s$  denotes the label of node  $v_i$  in the source domain,  $\hat{y}_i^s$  is the classification prediction for node  $v_i$  in the source

domain, and  $\hat{y}_i^t$  is the classification prediction for node  $v_i$  in the target domain.

Assuming that the source domain label is 1 and the target domain label is 0. The domain discriminator loss  $\zeta_d(Z^s, Z^t)$  is to increase discrimination error of discriminator  $d$  by gradient reversal propagation, meanwhile, to reduce discrimination error discriminator  $d$  by minimizing the cross-entropy loss for domain classifier:

$$\zeta_d(Z^s, Z^t) = -\frac{1}{N_s + N_t} \sum_{i=1}^{N_s + N_t} d_i \log(\hat{d}_i) + (1 - d_i) \log(1 - \hat{d}_i) \quad (11)$$

where  $d_i \in \{0, 1\}$  and  $\hat{d}_i$  denotes the ground truth and domain prediction of node  $v_i$  in the source domain and target domain, respectively.

Finally, by integrating the above loss, the loss of marginal adversarial domain adaptation can be defined as:

$$\zeta_{ada} = \zeta_{c_s}(Z^s, Y^s) + \gamma_1 \zeta_d(Z^s, Z^t) + \gamma_2 \zeta_{c_t}(Z^t) \quad (12)$$

where  $\gamma_1$  and  $\gamma_2$  are the balance parameters.

In addition, to further promote sufficient knowledge transfer from source domain to target domain, we fully explore intra-class structural graph alignment except for marginal adversarial domain adaptation to promote more sufficient transfer for the two domains.

**Structural Graph Alignment.** Inspired by establishing instance relationship graphs of teacher's and student's networks, which enable the student's network to mimic the one of teacher's by aligning these two graphs [Liu *et al.*, 2019]. After pseudo-labeling, we model the instance relationship graph (i.e., structural graph) among class centers with the Gram Matrix and implement structural graph alignment of class centers for source and target domains. However, the phenomenon of compactness and decentralization for intra-class nodes between source and target domains are still existing (Figure 2(b)). Thus, we further minimize the instance distances of source and target domains by MMD (Figure 2(c)) to promote latent structural graph of intra-class nodes. In this way, we can further narrow structural graph discrepancy in the same category and promote efficient knowledge transfer.

First, gram matrix of class centers for source and target domains  $R^{s/t}$  can be computed as below:

$$R^{s/t} = O_{s/t} (O_{s/t})^\top \quad (13)$$

where  $R_{i,j}^{s/t}$  is the inner products between  $O_{s/t,i}$  and  $O_{s/t,j}$ .

Then, the final structural graph of class centers  $Q^{s/t,l}$  can be obtained by forming  $L_2$  normalization for each row of  $R_{i \cdot}^{s/t}$ , defined as:

$$Q^{s/t,l} = \left[ \frac{R_{1 \cdot}^{s/t,l}}{\|R_{1 \cdot}^{s/t,l}\|_2}, \dots, \frac{R_{m^{s/t} \cdot}^{s/t,l}}{\|R_{m^{s/t} \cdot}^{s/t,l}\|_2} \right] \quad (14)$$

Once the structural graph of class centers for source and target domains are established, we can propose our structural

graph alignment of class centers shift loss, which requires the bias between two graphs is minimized:

$$\zeta_{sga} = \sum \frac{1}{m^{s/t}} \|Q^s - Q^t\|_2^2 \quad (15)$$

To further consider intra-class structural graph transfer, we construct intra-class structural graph alignment along the basis of class-centred structural graph alignment by narrowing the difference of node distribution. We define the difference between two distributions with their mean node embedding in the reproducing kernel Hilbert space (RKHS) [Gretton *et al.*, 2012]. Given kernel mapping  $\varphi(\cdot)$  in the corresponding RKHS, MMD is empirical estimated by comparing the square distance between the empirical kernel mean node embedding:

$$\begin{aligned} \zeta_{mmd} &= \text{MMD}(Z^s, Z^t) \\ &= \left\| \frac{1}{N^s} \sum \varphi(Z^s) - \frac{1}{N^t} \sum \varphi(Z^t) \right\|_2^2 \end{aligned} \quad (16)$$

By minimizing  $\zeta_{sga}$  and  $\zeta_{mmd}$ , the JDA-GCN we proposed will enable latent intra-class structural information hidden in feature space aligned as possible between two domains, and benefit the cross-domain node classification task.

$$\zeta_{stru} = \alpha \zeta_{mmd} + (1 - \alpha) \zeta_{sga} \quad (17)$$

Finally, by integrating Eq. (12) and Eq. (17), the overall loss of our JDA-GCN is defined as:

$$\zeta = \beta \zeta_{ada} + (1 - \beta) \zeta_{stru} \quad (18)$$

where  $\beta$  is the hyper-parameter for balancing the relative importance of proposed structural graph alignment loss.

## 4 Experiment

We first give the experimental setup, and then compare the performance of our method with baselines on cross-network node classification. After that, we analyse the performance of different components of JDA-GCN and give the parameter analysis. Finally, we provide a visualization task with t-SNE.

### 4.1 Experimental Setup

**Datasets.** We adopt two-category graphs from Citation [Li *et al.*, 2015] and Blog [Li *et al.*, 2015] to evaluate the performance of our proposed JDA-GCN, as shown in Table 1.

- Citation datasets: Citationv1 (C), ACMv9 (A) and DBLPv7 (D), where nodes represent papers and edges indicate the citation relationship.
- Blog datasets: Blog1 (B1) and Blog2 (B2), where nodes represent bloggers and edges indicates the friendship between two bloggers.

**Baselines.** We compare our proposed JDA-GCN with two category of baselines:

- Classical graph convolutional networks: GCN [Kipf and Welling, 2016], GAT [Veličković *et al.*, 2017] and BM-GCN [He *et al.*, 2022]. It is worth noting that these methods are trained on source domain and tested directly on target domain.

Datasets	#Nodes	#Edges	#Attributes	#Labels
Citationv1	8,935	15,113	6,775	5
ACMv9	9,360	15,602	2,089	5
DBLPv7	5,484	8,130	6,775	5
Blog1	2,300	33,471	8,189	6
Blog2	2,896	53,836	8,189	6

Table 1: Statistics of datasets.

- Graph domain adaptation methods: UDA-GCN [Wu *et al.*, 2020], AdaGCN [Dai *et al.*, 2022], CDNE [Shen *et al.*, 2020], and JHGDA [Shi *et al.*, 2023]. Specifically, UDA-GCN and AdaGCN only use marginal adversarial domain adaptation to narrow the discrepancy between source and target domains, while CDNE and JHGDA only use maximum mean discrepancy to narrow the discrepancy between two domains.

**Implementation Details.** All deep learning algorithms are implemented in Pytorch [Paszke *et al.*, 2017] and are trained with Adam optimizer. We use all labeled source samples and all unlabeled target samples. For all baselines, we follow the default parameter settings in their original papers. For our proposed JDA-GCN, we set the hidden layers in both source and target networks from 128 to 16, the dropout for each GCN layer to 0.3, and a fixed learning rate to  $1e^{-4}$ . In addition, we set balance parameters  $\gamma_1 = 1$  and  $\gamma_2 = 0.8$  on Citation and Blog, respectively, and set balance parameter  $\beta = 0.9$  on Citation and set  $\beta = 0.95$  on Blog.

### 4.2 Cross-network Node Classification

Table 2 lists the accuracy of different methods on cross-domain node classification task. Across all datasets, JDA-GCN achieves the highest average performance, surpassing the best baselines JHGDA by 1.65% on Citation, and UDA-GCN by 3.13% on Blog, respectively. In particular, JDA-GCN surpasses the best baselines UDA-GCN by 5.00% on dataset of B1→B2. The detailed analysis is as follows:

Compared to classical graph convolutional networks, that is, GCN, GAT and BM-GCN, JDA-GCN has an average improvement by 13.27%, 8.63% and 7.03% on Citation, and 6.84%, 6.24% and 5.66% on Blog, respectively. This mainly because classical GCNs often suffer from an inability to transfer abundant knowledge from source domain to target domain, which further indicates the effectiveness and superiority of our proposed strategy of domain alignment in addressing cross-network node classification task.

Compared to UDA-GCN and AdaGCN, JDA-GCN has an average improvement by 4.99% and 6.36% on Citation, and 3.13% and 3.31% on Blog, respectively. This confirms that considering structural graph alignment except for marginal distribution alignment can narrow domain discrepancy. In addition, JDA-GCN also performs better than CDNE and JHGDA on average by 6.12% and 1.65% on Citation, and 5.37% and 3.74% on Blog, respectively. This further validates that modeling both local and global consistency relationships of each graph as a new feature propagation matrix to obtain effective node embedding and joint graph domain

Methods	A→D	D→A	A→C	C→A	D→C	C→D	Avg.	B1→B2	B2→B1	Avg.
GCN	0.6178	0.5397	0.6607	0.6322	0.6111	0.6574	0.6198	0.5041	0.4852	0.4946
GAT	0.6508	0.5718	0.7168	0.6616	0.6963	0.7000	0.6662	0.5082	0.4903	0.5006
BM-GCN	0.6791	0.6123	0.7400	0.6650	0.6458	0.7509	0.6822	0.5126	0.5001	0.5064
CDNE	0.6723	0.6721	0.7082	0.6963	0.7143	0.6847	0.6913	0.5414	0.5032	0.5223
JHGDA	0.7207	0.6910	0.7703	0.7312	0.7602	0.7424	0.7360	0.5374	0.5138	0.5256
AdaGCN	0.7020	0.6731	0.7164	0.6642	0.6861	0.6914	0.6889	0.5335	0.5264	0.5299
UDA-GCN	0.7158	0.6681	0.7352	0.6822	0.7024	0.7120	0.7026	0.5153	0.5221	0.5187
<b>JDA-GCN</b>	<b>0.7338</b>	<b>0.7120</b>	<b>0.7851</b>	<b>0.7442</b>	<b>0.7838</b>	<b>0.7562</b>	<b>0.7525</b>	<b>0.5919</b>	<b>0.5340</b>	<b>0.5630</b>

Table 2: Classification accuracy on eight cross-domain tasks.

Methods	A→D	D→A	A→C	C→A	D→C	C→D	Avg.	B1→B2	B2→B1	Avg.
JDA-GCN-p	0.6441	0.5433	0.7201	0.6061	0.6631	0.6601	0.6395	0.5804	0.5014	0.5409
JDA-GCN-c	0.6836	0.6547	0.7376	0.7233	0.7608	0.6661	0.7044	0.5863	0.5283	0.5573
JDA-GCN-s	0.7128	0.7009	0.7250	0.7003	0.5504	0.7192	0.6848	0.5894	0.5297	0.5596
<b>JDA-GCN</b>	<b>0.7338</b>	<b>0.7120</b>	<b>0.7851</b>	<b>0.7442</b>	<b>0.7838</b>	<b>0.7562</b>	<b>0.7525</b>	<b>0.5919</b>	<b>0.5340</b>	<b>0.5630</b>

Table 3: Classification accuracy between JDA-GCN variants on eight cross-domain tasks.

adaptation can promote sufficient transfer between source domain and target domain.

### 4.3 Ablation Study

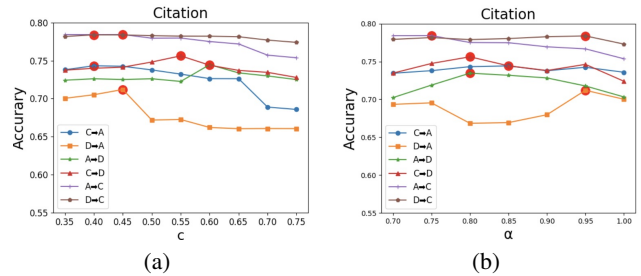
Since the proposed JDA-GCN contains multiple key components, we compare it with the following variants:

- JDA-GCN-p: A variant of JDA-GCN with the global consistency matrix being removed, and only using local consistency matrix to guide information propagation.
- JDA-GCN-c: A variant of JDA-GCN with the class center alignment loss being removed.
- JDA-GCN-s: A variant of JDA-GCN with the structure graph alignment loss being removed.

As shown in Table 3, JDA-GCN outperforms its three variants, which indicating the effectiveness of our proposed JDA-GCN. Specifically, 1) Compared to JDA-GCN-p, JDA-GCN improves by an average of 11.30% on Citation, and 2.21% on Blog, respectively, illustrating the superiority of fully exploring the global consistency matrix to obtain a comprehensive node embedding. 2) Compared to JDA-GCN-c, JDA-GCN improves by an average of 4.81% on Citation, and 0.57% on Blog, respectively, which demonstrates that incorporating structural graph alignment of class centers can mitigate the corresponding negative migration caused by only considering marginal distribution alignment. 3) Compared to JDA-GCN-s, JDA-GCN improves by an average of 6.77% on Citation, and 0.34% on Blog, respectively. This implies that constructing intra-class structural graph and uniquely augmenting the structural graph alignment can promote more sufficient knowledge transfer between the source and target domain.

### 4.4 Parameter Analysis

We estimate the sensitivity of two important hyperparameters, that is, the feature propagation matrix decay factor  $c$  and the structural graph alignment balance factor  $\alpha$ .


 Figure 4: Analysis results of (a) the decay factor  $c$ , and (b) Analysis results of the structural alignment factor  $\alpha$ .

**Analysis of Decay Factor  $c$ .** We obtain the global similarity matrix by comparing the sub-structure surrounded with each node. Considering that each layer has a different effect for ego node, and meanwhile to explore more layers, we denote the effect of different layers for ego node as a decay factor  $c$ .

Figure 4(a) shows the results of decay factor  $c$  on cross-network node classification, where the value of decay factor  $c$  is relatively stable between 0.40 and 0.65, indicating that the impact of decay factor  $c$  on the ego node is halved with the increase of the number of layers. Thus, the similarity between two nodes should consider not only the distance, but also the similarity of sub-structures. This also verifies the rationality and importance of the global similarity matrix, that is, when the decay factor  $c$  is too small or too large, for each node, the global information it received may be limited or with noise, leading to neglect of sub-structure similarity or excessive emphasis on sub-structure similarity.

**Analysis of Balance Factor  $\alpha$ .** We analyze the crucial coefficients  $\alpha$  in loss terms, where  $\alpha$  balances structural graph loss of class centers and intra-class structural graph loss.

Figure 4(b) shows that results of structural graph alignment

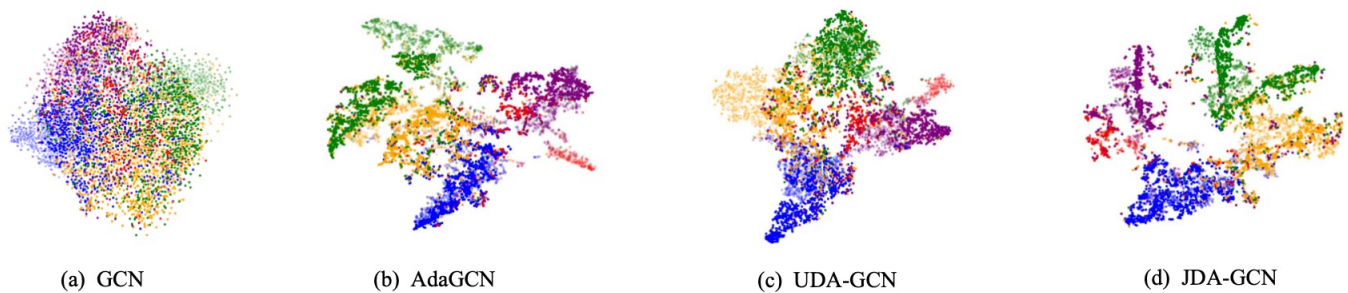


Figure 5: Visualizing results of source and target node embedding via t-SNE. Each color indicates a class, while high and light shades of the same color represent nodes belonging to the same class from source and target domains.

balance factor  $\alpha$  on cross-network node classification, where the balance factor  $\alpha$  varies between 0.75 and 0.95 for the best classification results. This underlines that modeling structural graph alignment of class centers is essential but is not sufficient, and incorporating intra-class structural graph alignment is more beneficial to the cross-network node classification. At the same time, we can find that the larger the balance factor  $\alpha$ , the better the model performance is, which further verifies the importance of intra-class structural graph alignment in our proposed JDA-GCN.

#### 4.5 Visualization

To provide a more intuitive comparison, we take the  $D \rightarrow C$  task as an illustrative example, and visualize our proposed JDA-GCN and three SOTA baselines (GCN, AdaGCN, and UGA-GCN) by t-SNE [Van der Maaten and Hinton, 2008] in a two-dimensional space.

As shown in Figure 5, we have the following two observations. First, compared to the other baselines, the boundary of JDA-GCN is more clear. This shows that JDA-GCN can effectively distinguish different classes in the embedding space. Second, nodes belonging to the same class from source domain and target domain are clustered together and clusters are overlapped, which indicates that JDA-GCN can significantly narrow domain discrepancy. It worth noting that the first observation indicates well performance on node classification, and the latter presents well performance on graph domain adaptation.

## 5 Related Work

**Classical Graph Convolutional Networks.** Researchers have proposed many works to obtain effective node embedding [Yu *et al.*, 2021; Cui *et al.*, 2022], including methods that focus on improving the propagation matrix. For example, HOG-GCN [Wang *et al.*, 2022] automatically learns the propagation process according to the homophily degree between node pairs. BM-GCN [He *et al.*, 2022] realizes “block-guided classified aggregation”, and automatically learns the corresponding aggregation rules for neighbors of different classes to aggregate information from homophilic and heterophilic neighbors discriminately with their homophily degree. AS-GAT [Li *et al.*, 2021a] handles graphs without being restricted by their underlying homophily via adaptive propagation on graphs. Inspired by the above approaches that using an effective

propagation matrix to benefit the node embedding, in this paper, we compute the global similarity matrix based on the sub-structure similarity around nodes, and combine it with the adjacency matrix to obtain the new propagation matrix to guide the information propagation.

**Graph Domain Adaptation Methods.** The powerful representation of graph make domain adaptation technique achieve desirable performance on cross-network tasks [Beigi and Moattar, 2021; Wang *et al.*, 2020b]. For instance, GDA-SpecReg [You *et al.*, 2023] improves the transfer ability of graph convolutional network by constructing a model-based GDA bound closely related to two GNN spectral properties. PACS [Yang *et al.*, 2020] solves the problem of the semantic alignment from a source domain to multiple target domains by constructing a multiple-domain feature transfer network with semantic propagation. Furthermore, DASGA [Pillancı and Vural, 2020] learns the spectrum of the label function in a source graph with many labeled nodes, and transfers the information of the spectrum to the target graph with fewer labeled nodes for domain adaptation. DAGCN [Li *et al.*, 2021b] models the three types of information in a unified deep network and achieving Unsupervised domain adaptation. However, the above models consider neither class distribution shift nor structural graph distribution shift, which is of great importance for graph domain adaption. For this purpose, we propose joint domain adaptive graph convolutional network with intra-class structural graph transfer to prompt more sufficient transfer.

## 6 Conclusion

In this paper, we propose a joint adversarial domain adaptive graph convolutional network (JDA-GCN). We learn effective node embedding on both source and target graphs by combining global and local consistency matrices to guide information propagation, where the global consistency matrix fully capture sub-structure information in graphs. To prompt more sufficient knowledge transfer between these two domains, we further construct structural graphs and align them except for considering domain-wise marginal alignment, so as to enable the two domains as close as possible. Experimental results on diverse real-world datasets show that JDA-GCN outperforms existing state-of-the-art methods for cross-domain network node classification and indicates the superiority of our proposed method.

## Acknowledgments

This work is supported by National Key Research and Development Program of China (2023YFC3304503), National Natural Science Foundation of China (62276187).

## Contribution Statement

Niya Yang and Ye Wang are both authors.

## References

- [Beigi and Moattar, 2021] Omid Mohamad Beigi and Mohammad H Moattar. Automatic construction of domain-specific sentiment lexicon for unsupervised domain adaptation and sentiment classification. *Knowledge-Based Systems*, 213:106423, 2021.
- [Codling *et al.*, 2008] Edward A Codling, Michael J Plank, and Simon Benhamou. Random walk models in biology. *Journal of the Royal Society Interface*, 5(25):813–834, 2008.
- [Cui *et al.*, 2022] Zeyu Cui, Zekun Li, Shu Wu, Xiaoyu Zhang, Qiang Liu, Liang Wang, and Mengmeng Ai. Dygn: Efficient dynamic graph embedding with graph convolutional network. *IEEE Transactions on Neural Networks and Learning Systems*, 2022.
- [Dai *et al.*, 2022] Quanyu Dai, Xiao-Ming Wu, Jiaren Xiao, Xiao Shen, and Dan Wang. Graph transfer learning via adversarial domain adaptation with graph convolution. *IEEE Transactions on Knowledge and Data Engineering*, 35(5):4908–4922, 2022.
- [de Almeida *et al.*, 2008] Madson C de Almeida, Eduardo N Asada, and Ariovaldo V Garcia. On the use of gram matrix in observability analysis. *IEEE Transactions on Power Systems*, 23(1):249–251, 2008.
- [Ganin *et al.*, 2016] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario March, and Victor Lempitsky. Domain-adversarial training of neural networks. *Journal of Machine Learning Research*, 17(59):1–35, 2016.
- [Gretton *et al.*, 2012] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773, 2012.
- [He *et al.*, 2022] Dongxiao He, Chundong Liang, Huixin Liu, Mingxiang Wen, Pengfei Jiao, and Zhiyong Feng. Block modeling-guided graph convolutional neural networks. In *Proceedings of the AAAI conference on Artificial Intelligence*, volume 36, pages 4022–4029, 2022.
- [Hornik *et al.*, 2012] Kurt Hornik, Ingo Feinerer, Martin Kober, and Christian Buchta. Spherical k-means clustering. *Journal of Statistical Software*, 50:1–22, 2012.
- [Jeh and Widom, 2002] Glen Jeh and Jennifer Widom. Simrank: a measure of structural-context similarity. In *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 538–543, 2002.
- [Kipf and Welling, 2016] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.
- [Li *et al.*, 2015] Jundong Li, Xia Hu, Jiliang Tang, and Huan Liu. Unsupervised streaming feature selection in social media. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 1041–1050, 2015.
- [Li *et al.*, 2020] Zhao Li, Xin Shen, Yuhang Jiao, Xuming Pan, Pengcheng Zou, Xianling Meng, Chengwei Yao, and Jiajun Bu. Hierarchical bipartite graph neural networks: Towards large-scale e-commerce applications. In *Proceedings of the 36th IEEE International Conference on Data Engineering*, pages 1677–1688, 2020.
- [Li *et al.*, 2021a] Shouheng Li, Dongwoo Kim, and Qing Wang. Beyond low-pass filters: Adaptive feature propagation on graphs. In *Machine Learning and Knowledge Discovery in Databases*, pages 450–465, 2021.
- [Li *et al.*, 2021b] Tianfu Li, Zhibin Zhao, Chuang Sun, Ruqiang Yan, and Xuefeng Chen. Domain adversarial graph convolutional network for fault diagnosis under variable working conditions. *IEEE Transactions on Instrumentation and Measurement*, 70:1–10, 2021.
- [Liang *et al.*, 2023] Pengfei Liang, Leitao Xu, Hanqin Shuai, Xiaoming Yuan, Bin Wang, and Lijie Zhang. Semisupervised subdomain adaptation graph convolutional network for fault transfer diagnosis of rotating machinery under time-varying speeds. *IEEE/ASME Transactions on Mechatronics*, 2023.
- [Liu *et al.*, 2019] Yufan Liu, Jiajiong Cao, Bing Li, Chunfeng Yuan, Weiming Hu, Yangxi Li, and Yunqiang Duan. Knowledge distillation via instance relationship graph. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7096–7104, 2019.
- [Liu *et al.*, 2023a] Haoyu Liu, Ningyi Liao, and Siqiang Luo. Simga: A simple and effective heterophilous graph neural network with efficient global aggregation. *arXiv preprint arXiv:2305.09958*, 2023.
- [Liu *et al.*, 2023b] Shikun Liu, Tianchun Li, Yongbin Feng, Nhan Tran, Han Zhao, Qiang Qiu, and Pan Li. Structural re-weighting improves graph domain adaptation. In *International Conference on Machine Learning*, pages 21778–21793, 2023.
- [Paszke *et al.*, 2017] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.
- [Pilancı and Vural, 2020] Mehmet Pilancı and Elif Vural. Domain adaptation on graphs by learning aligned graph bases. *IEEE Transactions on Knowledge and Data Engineering*, 34(2):587–600, 2020.
- [Pinkus, 1999] Allan Pinkus. Approximation theory of the mlp model in neural networks. *Acta Numerica*, 8:143–195, 1999.



- [Shen *et al.*, 2020] Xiao Shen, Quanyu Dai, Sitong Mao, Fulai Chung, and Kup-Sze Choi. Network together: Node classification via cross-network deep network embedding. *IEEE Transactions on Neural Networks and Learning Systems*, 32(5):1935–1948, 2020.
- [Shi *et al.*, 2023] Boshen Shi, Yongqing Wang, Fangda Guo, Jiangli Shao, Huawei Shen, and Xueqi Cheng. Improving graph domain adaptation with network hierarchy. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 2249–2258, 2023.
- [Van der Maaten and Hinton, 2008] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.
- [Veličković *et al.*, 2017] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.
- [Wang *et al.*, 2019] Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S Yu. Heterogeneous graph attention network. In *The World Wide Web Conference*, pages 2022–2032, 2019.
- [Wang *et al.*, 2020a] Zhen Wang, Marko Jusup, Hao Guo, Lei Shi, Sunčana Geček, Madhur Anand, Matjaž Perc, Chris T Bauch, Jürgen Kurths, Stefano Boccaletti, et al. Communicating sentiment and outlook reverses inaction against collective risks. *Proceedings of the National Academy of Sciences*, 117(30):17650–17655, 2020.
- [Wang *et al.*, 2020b] Zijian Wang, Yadan Luo, Zi Huang, and Mahsa Baktashmotlagh. Prototype-matching graph network for heterogeneous domain adaptation. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 2104–2112, 2020.
- [Wang *et al.*, 2022] Tao Wang, Di Jin, Rui Wang, Dongxiao He, and Yuxiao Huang. Powerful graph convolutional networks with adaptive propagation mechanism for homophily and heterophily. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 4210–4218, 2022.
- [Wei *et al.*, 2021] Guanqun Wei, Zhiqiang Wei, Lei Huang, Jie Nie, and Xiaojing Li. Center-aligned domain adaptation network for image classification. *Expert Systems with Applications*, 168:114381, 2021.
- [Wu *et al.*, 2020] Man Wu, Shirui Pan, Chuan Zhou, Xiaojun Chang, and Xingquan Zhu. Unsupervised domain adaptive graph convolutional networks. In *Proceedings of The Web Conference 2020*, pages 1457–1467, 2020.
- [Wu *et al.*, 2022] Fei Wu, Pengfei Wei, Guangwei Gao, Chang-Hui Hu, Qi Ge, and Xiao-Yuan Jing. Dual-aligned unsupervised domain adaptation with graph convolutional networks. *Multimedia Tools and Applications*, 81(11):14979–14997, 2022.
- [Xie *et al.*, 2020] Yuan Xie, Tianshui Chen, Tao Pu, Hefeng Wu, and Liang Lin. Adversarial graph representation adaptation for cross-domain facial expression recognition. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 1255–1264, 2020.
- [Yang *et al.*, 2020] Xu Yang, Cheng Deng, Tongliang Liu, and Dacheng Tao. Heterogeneous graph attention network for unsupervised multiple-target domain adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(4):1992–2003, 2020.
- [You *et al.*, 2023] Yuning You, Tianlong Chen, Zhangyang Wang, and Yang Shen. Graph domain adaptation via theory-grounded spectral regularization. In *The 11th International Conference on Learning Representations*, 2023.
- [Yu *et al.*, 2021] Donghan Yu, Yiming Yang, Ruohong Zhang, and Yuexin Wu. Knowledge embedding based graph convolutional network. In *Proceedings of the Web Conference 2021*, pages 1619–1628, 2021.
- [Zhao *et al.*, 2021] Tianxiang Zhao, Xiang Zhang, and Suhang Wang. Graphsmote: Imbalanced node classification on graphs with graph neural networks. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, pages 833–841, 2021.
- [Zhu *et al.*, 2020] Yongchun Zhu, Fuzhen Zhuang, Jindong Wang, Guolin Ke, Jingwu Chen, Jiang Bian, Hui Xiong, and Qing He. Deep subdomain adaptation network for image classification. *IEEE transactions on neural networks and learning systems*, 32(4):1713–1722, 2020.