Contextual Structure Knowledge Transfer for Graph Neural Networks

Zhiyuan Yu[1], Wenzhong Li[1*], Zhangyue Yin[2], Xiaobin Hong[1], Shijian Xiao[1], Sanglu Lu[1]

[1]State Key Laboratory for Novel Software Technology, Nanjing University
[2]School of Computer Science, Fudan University
{zhiyuan_yu, xiaobinhong, xiaoshijian}@smail.nju.edu.cn, {yinzy21}@m.fudan.edu.cn, {lwz, sanglu}@nju.edu.cn

Abstract

Graph transfer learning endeavors to develop a Graph Neural Network (GNN) model in a fully-labeled source domain, with the intention of deploying it on a target domain that has limited labeled data for inference. We reveal that prevalent graph transfer learning methods are susceptible to the homophily shift problem. This issue arises from the divergence in homophily structures between the source and target graphs, leading to a notable deterioration in the performance of GNN models. In this paper, we introduce a novel Contextual Structural Graph Neural Network (CS-GNN) method, leveraging a tailored attention mechanism to apprehend a variety of local structural cues, facilitating structural knowledge transfer across domains. It features an ego-network module to distill local structural diversity and a moment-based approach to gauge structural patterns without needing ground-truth labels. CS-GNN crafts a feature smoothness matrix from node attributes, guiding a customized attention mechanism for feature aggregation. A group-wise fairness loss is employed to balance learning across various structural patterns, enhancing the model's ability to transfer knowledge across domains. Comprehensive experiments conducted on six benchmark datasets substantiate the superiority of CS-GNN over the state-of-the-art methods, demonstrating significant improvements in accuracy and robustness against homophily shifts.

Introduction

Graph neural networks (GNNs) (Kipf and Welling 2017; Keriven and Peyré 2019; Wu et al. 2021; Hong et al. 2021a) have become prominent in the field of graph machine learning due to their ability to transform unstructured data into low-dimensional representations by capturing both node features and topological dependencies. However, many GNN methods struggle with sparse label scenarios, limiting their effectiveness in real-world datasets where obtaining labels is challenging or impractical. To address this, graph transfer learning has emerged as a promising solution, which involves transferring knowledge from a relevant source graph to a target graph (Tan et al. 2018; Zhuang et al. 2019). For example, in protein-protein interaction networks, abundant functional information from a source network can be leveraged to predict the functionalities of proteins in a new target network. While standard transfer learning for image or text data often focuses on minimizing distribution discrepancies across domains, graph-structured datasets present a more complex challenge due to the additional distribution shift over their intricate structures.

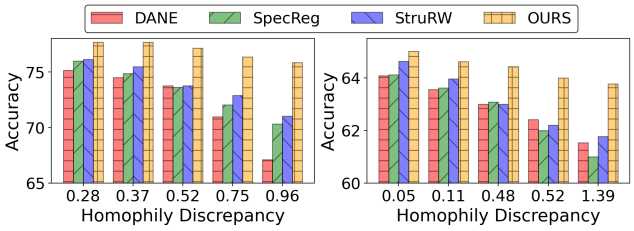

Figure 1: Performance of graph transfer learning methods with varying homophily discrepency on the *Citation* **(left)** and *Social* **(right)** dataset, where different levels of homophily discrepency were generated by adding 0, 5%, 10%, 15%, 20% of heterophilous edges on the source graph accordingly.

	Citation	Social	WebKB	Airport		
Domains	A&D	B1&B2	C&T	U&B	U&E	B&E
Homo Diff	0.2814	0.0533	0.1957	0.3076	0.1994	0.2833

Table 1: The homophily discrepancy of different real-world datasets.

Recent researches have sought to enhance the applicability of GNNs in transfer learning. For example, DANE (Zhang et al. 2019b) applies adversarial training of domain classifiers based on the last-layer node representations. UDAGCN (Wu et al. 2020) further imposes inter-graph attention mechanisms on top of adversarial training. EGI (Zhu et al. 2021) adopts Ego-Graph Information Maximization tactics to align structural information and node embeddings. GRADE (Wu, He, and Ainsworth 2023) proposes optimizing Graph Subtree Discrepancy inspired by the Weisfeiler-Lehman Subtree Kernel. However, these works either neglect the unique characteristics of graph-structured datasets or focus solely on the topological discrepancies. Recent approaches like SpecReg (You et al. 2023) and StruRW (Liu

*corresponding author

et al. 2023) propose leveraging graph spectral regularization or structural re-weighting methods to address structural pattern shift problems, yet their spectrum/spatial-based theories still underestimate the complexity of real-world datasets.

Despite great efforts, the existing methods have failed to effectively address the *homophily shift* problem, resulting in suboptimal performance in certain scenarios. Homophily is a common graph property that measures whether nodes with the same labels or similar features tend to be linked. Homophily shift denotes the discrepancy in this property between the source and target domains. To quantify the impact of homophily shift on graph transfer learning, we conduct experiments using two graph dataset with varying degrees of homophily shift. As depicted in Figure 1, the state-of-the-art graph transfer learning methods (e.g., DANE (Wu et al. 2020), SpecReg (You et al. 2023), StuRW (Liu et al. 2023)), exhibit a notable decline in performance as the level of homophily discrepancy increases, where up to 10% accuracy drop is observed. Unfortunately, homophily shift is prevalent in real-world datasets, as shown in Table 1. This underscores the need for a more nuanced approach that can effectively manage homophily shift to improve graph transfer learning across diverse datasets.

Although homophily varies across different networks, substantial differences in internal structural patterns also exist within individual networks (Lim et al. 2021; Li et al. 2022). This insight prompts us to harness knowledge from subgraphs with structural patterns similar to those of the target network, thereby enhancing transferability. To achieve this, we propose a Contextual Structure knowledge transfer method for GNNs (CS-GNN) that integrates feature smoothness and contextual message-passing for graph transfer learning. As shown in Figure 1, our approach exhibits superior robustness across diverse levels of homophily shift.

The pipeline of the proposed CS-GNN is illustrated in Figure 2. Specifically, an *ego-network construction* module is first employed for each node in the graph to create subgraphs that capture diverse local structural information which is essential for structure knowledge transfer. Next, we introduce a *moment-based feature smoothness* method to provide a quantified measure of contextual structure patterns in both source and target domains, without the need for ground-truth labels. It computes multi-order moments for each node attribute to form a feature smoothness matrix, based on which *contextual message passing* introduced for customized attention-based feature aggregation for GNNs. To avoid over-reliance on dominant structural patterns within the graph, we introduce a *group-wise fairness loss* to ensures the model fairly learns diverse structural patterns, thus enhancing the model's transferability across different domains. The efficacy of our proposed CS-GNN is confirmed on six real-world benchmark datasets. The source code for CS-GNN is publicly available at https://github.com/yyy0959/CS-GNN.

The contributions of this paper are as follows.

- We tackle the homophily shift issue in graph transfer learning effectively by emphasizing the internal diversity of structural patterns within the network. By harnessing contextual structural knowledge, we enhance the model's ability to transfer knowledge across different domains.
- We propose a novel Contextual Structural Graph Neural Network (CS-GNN) method, which adopts a customized attention-based design to capture diverse local structural information, enabling contextual structural transfer across domains.
- We introduce a moment-based feature smoothness for measuring diverse structure patterns without the need for ground-truth labels, together with a contextual message passing to enhance GNNs' transferability across different domains. Comprehensive experiments conducted on six benchmark datasets substantiate the superiority of CS-GNN over the state-of-the-art methods, demonstrating significant improvements in accuracy as well as robustness against homophily shifts.

Related Works

Graph Neural Networks

Graph neural networks (GNNs) have recently become the defacto tool to learn the representations of graph-structured data, demonstrating strong performance across various real-world tasks. GNN models typically follow a neighborhood aggregation framework, where node representations are updated by aggregating information from their neighboring nodes. The pioneering work, Graph Convolutional Networks (GCN) (Kipf and Welling 2017), introduced spectral convolution to graph data. This was followed by efforts to develop deeper (Xu et al. 2018; Li et al. 2019; Liu, Gao, and Ji 2020) and more scalable (Wu et al. 2019; Lin et al. 2022; Hong et al. 2024) architectures. Building on GCN, attention-based methods (Veličković et al. 2018; Shi et al. 2021; Hong et al. 2021b) dynamically learn weights (attention scores) on edges during message passing, incorporating attention mechanisms into neighborhood aggregation.

Homophily and Heterophily

Most GNNs rely on the homophily assumption, which posits that nodes of the same class are more likely to be connected. However, as a ubiquitous graph property in numerous real-world scenarios, heterophily, i.e., nodes with different labels tend to be linked, significantly limits the performance of tailor-made homophilic GNNs. Recent researches has sought to make GNNs better handle heterophilic graphs. Some of them extend the neighborhood of central nodes by adding high-order neighbors (Abu-El-Haija et al. 2019; Jin et al. 2021; Zhu et al. 2020). Others challenge predefined edges and seek to discover potential neighbors based on representational similarity (bao Yang et al. 2021; Liu, Wang, and Ji 2020; Pei et al. 2020). Additionally, new aggregation functions have also been designed to discriminate node representations more effectively for heterophilic graphs (Bo et al. 2021; Chen et al. 2020; Yang et al. 2021).

Graph Transfer learning

Transfer learning has emerged as a framework to leverage knowledge from relevant tasks to improve performance on a target task, particularly useful when training data is insufficient. Previous works (Ben-David et al. 2010; Zhang et al.

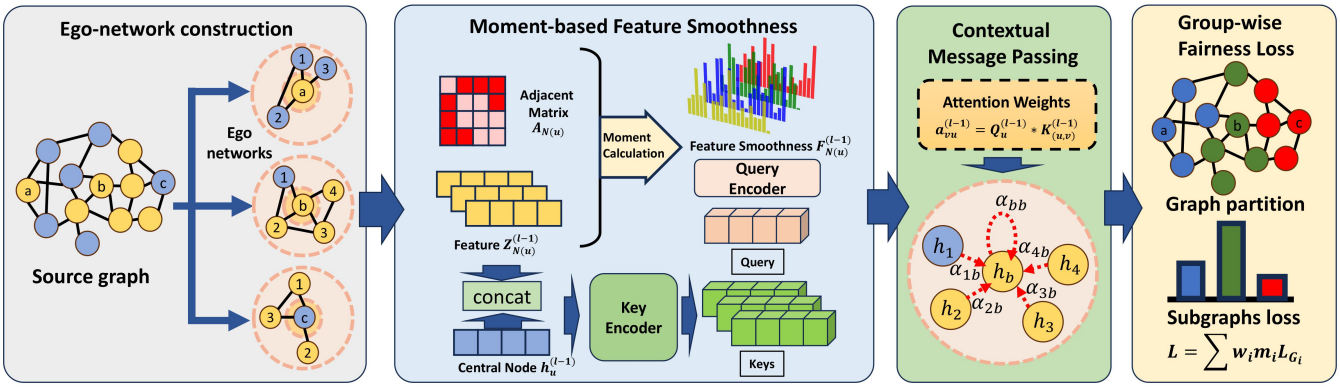

Figure 2: The framework of CS-GNN.

2019a; Acuna et al. 2021) have established the upper bound of transfer learning performance and suggest that generalization can be improved by reducing attribute distribution discrepancies across domains. Methods to reduce distribution shift between the source and target include domain regularizers (Sun and Saenko 2016; Panaretos and Zemel 2019), adversarial learning (Wu et al. 2020; Dai et al. 2023; Lin et al. 2023), selective instance or task training in the source domain(Dai et al. 2007; Yao and Doretto 2010; Mancini et al. 2018; Segu, Tonioni, and Tombari 2023).

Compared with Euclidean data, graph-structured datasets present a more complex challenge in transfer learning scenarios due to their intricate structures. Recent researches have sought to enhance the applicability of GNNs in transfer learning. For example, (Zhang et al. 2019b; Dai et al. 2023) adopt adversarial learning to train domain-invariant encoder, following by (Wu et al. 2020) which incorporates attention to further enhance expressiveness. Alternatively, graph-specific regularizers have be employed to constrain the discrepancy between the source and target, including tree mover distance (Chuang and Jegelka 2024), subtree discrepancy (Wu, He, and Ainsworth 2023), and spectral regularizer (You et al. 2023). Additionally, (Liu et al. 2023) reweights the graphs in the source domain during GNN encoding to alleviate structure shift problem. While these methods can reduce the distribution shift between source and target, they underestimate the complexity of real-world datasets and fail to effectively address the homophily shift problem. In addition, most of their methods require the target graph unavailable during pre-training, which is sometimes impracticable in real-world scenarios.

Methodology

Overall Framework

The proposed CS-GNN aims to achieve better transfer performance by leveraging contextual structural knowledge. To this end, it introduces an implicit local structure matching mechanism across the source and target networks, and achieves knowledge transferable feature aggregation for GNNs by adopting a contextual message passing. The overall framework is depicted in Figure 2.

Preliminary

Graph Neural Network We represent the graph-structured data as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{v_1, ..., v_n\}$ is the node set and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the edge set. Each node is associated with node attributes $\mathbf{x}_i \in \mathbb{R}^d$ and a class $y_i \in \{1, ..., C\}$, with C being the total number of classes. Additionally, each graph has an adjacency matrix $\mathbf{A} \in \{0, 1\}^{n \times n}$, where $\mathbf{A}_{ij} = 1$ iff $(i, j) \in \mathcal{E}$, otherwise $\mathbf{A}_{ij} = 0$. Generally, GNN models follow a neighborhood aggregation mechanism called message passing, where node representations are updated by aggregating information from their neighboring nodes. Let $z_i^{(l)}$ denote the output vector of node v_i at the l-th hidden layer and $z_i^{(0)}$ the original feature. The update function for the l-th hidden layer is:

$$z_i^{(l)} = \mathbf{UDT}\left(z_i^{(l-1)}, \mathbf{AGG}\left(\{z_j^{(l-1)} | v_j \in \mathbb{N}_{v_i}\}\right)\right), \quad (1)$$

where $\mathbb{N}_{v_i}$ is the set of neighbors of v_i. The **AGG** function gathers information from neighbors, while the **UDT** is a pooling function (Xu et al. 2019) further fuses the information from neighbors and the central node.

Homophily Ratio and Discrepancy Homophily ratio measures the proportion of edges connecting nodes of the same class. Typically, for node v_i, its homophily ratio can be define as follows:

$$\mathcal{H}_{v_i} = \frac{\sum_{v_j \in \mathbb{N}_{v_i}} \mathbb{I}(y_i = y_j)}{|\mathbb{N}_{v_i}|}, \quad (2)$$

where $\mathbb{N}_{v_i}$ is the neighboring set for node v_i, and $\mathbb{I}(\cdot)$ is the indicator function that equals to 1 if $(\cdot)$ holds true and otherwise 0.

To measure the the inner diversity of structure pattern within networks, we define homophily discrepancy as

$$\Delta\mathcal{H} = \mathbf{DTW}(\{\mathcal{H}_{s_1}, \mathcal{H}_{s_2}, ...\}, \{\mathcal{H}_{t_1}, \mathcal{H}_{t_2}, ...\}), \quad (3)$$

where $\mathcal{H}_{s_i}$ and $\mathcal{H}_{t_j}$ are the node-level homophily ratio for node i and node j in the source and target domains respectively, and DTW is the Dynamic Time Warping distance (Geler et al. 2019) that measures the difference between

these two lists even when they are unequal in length. Such a distance metric can discern the distribution of homophily ratios across networks, offering a nuanced assessment of homophily shifts.

Problem Formulation In cross-network classification tasks, we have the source graph $\mathcal{G}^s = (\mathcal{V}^s, \mathcal{E}^s)$ and target graph $\mathcal{G}^t = (\mathcal{V}^t, \mathcal{E}^t)$. For simplicity, we assume these graphs share the same feature space, otherwise we could project feature vectors that lie in different feature spaces into the same latent factor space, as did in (Zhao et al. 2024). We utilize GNN backbone $f(\cdot)$ to encode nodes into embeddings $\mathbf{Z}^s$ and $\mathbf{Z}^t$, Subsequently, a classifier $g(\cdot)$ is applied for predictions. The goal is to train a classification model $f \circ g : G \to Y$ using the available data from the source domain and then assess its performance on the target domain. The objective in graph transfer learning is to minimize the empirical risk on the target domain, defined as $R_T(h) = \mathbf{Pr}_{(x,y)\sim\mathcal{D}_T}(g(f(x, A)) \neq y)$.

Ego-Network Construction

To formulate the contextual structural information that is valuable for graph transfer learning, we define the k-hop ego-graph for every central node as follows.

Definition 1 (K-hop ego-graph) *Supposing that the complete network to be $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. We define a graph $g_i = \{V(g_i), E(g_i)\}$ with $V(g_i) \subseteq V$ and $E(g_i) \subseteq E$ as a k-hop ego-graph centered at node v_i, if the greatest distance between v_i and any other nodes in the ego-graph is k, i.e., $\forall v_j \in V(g_i), |d(v_i, v_j)| \leq k$, where $d(v_i, v_j)$ is the graph distance between v_i and v_j. And g_i is the induced subgraph of node set $V(g_i)$, which means for each node pair $a, b \in V(g_i)$, $(a, b) \in E(g_i)$ if and only if $(a, b) \in E$.*

By default, in the following subsections, we restrict the message passing process for node i within its k-hop ego-network. This approach allows us to construct networks with different structural patterns while retaining local contextuality within each of them.

Moment-based Feature Smoothness

After obtaining the K-hop ego-graphs, we aim to integrate quantified contextual structure pattern measurements, such as homophily ratio, into the message passing process to allow for diverse aggregation mechanisms within the same network. However, cross-network node classification tasks are usually semi-supervised, meaning only part of nodes in source domain have accessible labels, and few or no labels on the target domain. To address this, we leverage feature smoothness as a metric to reflect structural patterns in the absence of ground-truth labels.

Definition 2 (*Feature Smoothness*) *Given graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with its normalized feature $\mathbf{X} \in \mathbb{R}^{n\times d}$, its feature smoothness can be calculated by:*

$$\boldsymbol{FS}_{\mathcal{G}} = \frac{1}{|\mathcal{E}| \cdot d} \Big|\Big| \sum\nolimits_{u\in\mathcal{V}} \Big(\sum\nolimits_{v\in\mathbb{N}_u} (x_u - x_v) \Big)^2 \Big|\Big|_1, \quad (4)$$

where x_u, x_v are the feature vectors of node u, v acccordingly, and $||\cdot||_1$ is the Manhattan norm.

Theoretically, feature smoothness assesses how nodes acquire information from their neighbors (Hou et al. 2020), which can indicate how the message passing process in GNNs alters node embeddings. However, the feature smoothness score evaluates the overall magnitude and does not capture the variability within different attributes. Additionally, it does not account for the distribution of feature differences between individual pairs of connected nodes. To address these limitations, we propose a novel moment-based feature smoothness measurement to offer a more nuanced and comprehensive assessment of the contextual structure.

Definition 3 (*Method of Moments*) *For a random variable X, its k-th order origin moment is denoted as $\mathbb{E}(X^k)$, where $\mathbb{E}(X^k) = \frac{1}{Z(\theta)} \int_{-\infty}^{+\infty} x^k f(x, \theta) dx$. The k-th order central moment of X is $\mathbb{E}((X - \mu_x)^k) = \frac{1}{Z(\theta)} \int_{-\infty}^{+\infty} (x - \mu_x)^k f(x, \theta) dx$, where μ_x is the 1-th order origin moment of X. The k-th order standardized moment of X is $\mathbb{E}((X - \mu_x)^k) = \frac{1}{Z(\theta)} \int_{-\infty}^{+\infty} \frac{(x-\mu_x)^k}{\sigma_x^k} f(x, \theta) dx$, where σ_x is the standard deviation, and $Z(\theta) = \int_{-\infty}^{+\infty} f(x, \theta) dx$ is the nonmalized functions so that $\frac{f(x,\theta)}{Z(\theta)}$ can obey the formula of probability density function.*

Based on the formulation of moments, we propose the computation of moment-based feature smoothness.

Definition 4 (*Moment-based Feature Smoothness*) *Given graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with its normalized feature $\mathbf{X} \in \mathbb{R}^{n\times d}$. Let x_i denote the feature vector of node i, and $\delta_{uv} = (x_u - x_v)^2$. The k-th order of moment-based feature smoothness can be then calculated by:*

$$\boldsymbol{FS}_{\mathcal{G}}^{(k)} = \begin{cases} \frac{1}{|\mathcal{E}|} \sum\limits_{(u,v)\in\mathcal{E}} \delta_{uv} \triangleq \mu_i, & k = 1, \\ \frac{1}{|\mathcal{E}|} \Big(\sum\limits_{(u,v)\in\mathcal{E}} (\delta_{uv} - \mu_i)^2 \Big)^{\frac{1}{2}} \triangleq \sigma_i, & k = 2, \\ \frac{1}{|\mathcal{E}|} \Big(\sum\limits_{(u,v)\in\mathcal{E}} \frac{(\delta_{uv} - \mu_i)^k}{\sigma_i^k} \Big)^{\frac{1}{k}}, & k \geq 3. \end{cases}$$

Based on the definition, we formulate the moment-based feature smoothness matrix $\mathbf{F}_{\mathcal{G}}$ to measure the structure pattern of graph $\mathcal{G}$, where the i-th column of it being the i-th order of moment-based feature smoothness, i.e., $\mathbf{F}_{\mathcal{G}}[i, :] = \mathbf{FS}_{\mathcal{G}}^{(i)}$.

Contextual Message Passing

We propose contextual message passing for GNNs to enhance its transferability. As illustrated in Figure 3, conventional GNNs perform message passing by treating neighborhood nodes equally, which can be formulated as

$$z_u^{(l)} = \sum_{v\in N(u)} \mathbf{MLP}(\alpha(z_u^{(l-1)}, z_v^{(l-1)}) * z_v^{(l-1)}), \quad (5)$$

where $\alpha(\cdot) : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ is the attention mechanism that computes the importance of node u to node v.

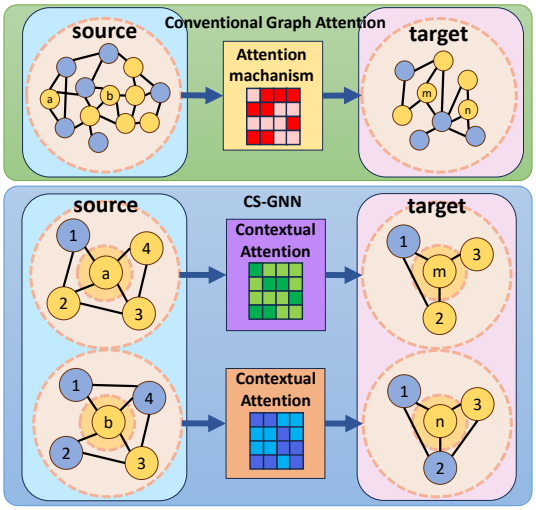

Figure 3: Illustration of conventionl graph attention and the proposed contextual attention mechanism.

However, conventional graph attention mechanisms uses the same attention machanism on both source and target domains without taking into account the homophily characteristics. This approach makes them particularly sensitive to the homophily shift during graph transfer learning. To address this issue, we propose contextual message passing that incorporates feature smoothness for knowledge transfer across domains, which is particularly designed to provide personalized attention weights for diverse structural patterns.

Specifically, for an ego-network centered at node v_u, we compute the feature smoothness matrix $\mathbf{F}_{g_u}$ and encode it into the query vector as:

$$Q_u = [\boldsymbol{q} \cdot \sigma(\mathbf{W}_{src}\mathbf{F}_{g_u}) || \boldsymbol{q} \cdot \sigma(\mathbf{W}_{dst}\mathbf{F}_{g_u})], \tag{6}$$

where $\mathbf{W}_{src}^{(l-1)}, \mathbf{W}_{dst}^{(l-1)} \in \mathbb{R}^{h\times k}$ are learnable parameters that fuse information from different moments of feature smoothness; $\sigma(\cdot)$ is the activation function; $\boldsymbol{q} \in \mathbb{R}^h$ is learnable vector that further project $\sigma(\mathbf{W}_{src}\mathbf{F}_{g_u})$ into a d-dimensional embedding space. We derive the key vectors as

$$K_{(u,v)} = [z_u^{(l-1)}\mathbf{W}_K || z_v^{(l-1)}\mathbf{W}_K], \tag{7}$$

where $\mathbf{W}_K \in \mathbb{R}^{d\times h'}$ is a learnable transformation matrix and $||$ is the concatenation operation.

Then we calculate the importance of node v's features to node u by the inner product of the key and query vectors, i.e., $e_{uv} = Q_u^T K_{(u,v)}$. Once $\{e_{uv} \mid v \in \mathbb{N}_u\}$ is obtained, we use the normalized attention coefficients to compute the linear combination of the neighbor features, serving as the final output features for the central node:

$$\alpha_{uv} = \frac{\exp(\mathbf{LeakyRelu}(e_{uv}))}{\sum_{v\in\mathbb{N}_u}\exp(\mathbf{LeakyRelu}(e_{uv}))}, \tag{8}$$

$$z_u^{(l)} = \sigma(\sum_{v\in\mathbb{N}_u}\alpha_{mk} z_v^{(l-1)}\mathbf{W}^{(l-1)}). \tag{9}$$

Different from conventional graph attention, our CS-GNN provides a customized attention mechanism for each node based on their contextual structure. Intuitively, our method serves as an implicit structure matching machanism, which assigns similar attention mechanisms to nodes with similar structure patterns across domains.

Group-Wise Fairness Loss

Previous works apply cross-entropy loss as the overall loss function, aiming for optimal accuracy performance on a global scale. However, this approach can be problematic since it will leads to model's over-reliance on the majority structural pattern on the graph. In graph transfer learning settings, this can lead to over-emphasize on particular structure patterns in the source domain, thus impairing transfer performance in the target domain. Our objective is to maintain an unbiased performance across nodes that exhibit a variety of structural patterns. To this end, we propose a group-wise fairness loss based on a graph partition approach.

In real-world scenarios, nodes with similar structure patterns tend to get clustered together. Therefore we utilize the scalable graph partition module METIS (Karypis and Kumar 1998) to split nodes into multiple non-overlapping subgraphs: $\{\mathcal{G}_1, \mathcal{G}_2, ..., \mathcal{G}_K\}$. For each subgraph $\mathcal{G}_i$, we calculate the cross-entropy loss over it, and apply weighted-sum to form the group-wise fairness loss:

$$\mathcal{L} = \sum_{i=1}^{K} w_i m_i \mathcal{L}_{\mathcal{G}_i} \tag{10}$$

where $m_i = \frac{|V_i|}{|V|}$ represents the percentage of node in subgraph i; w_i represents a dynamic weight for the subgraph.

The dynamic weight w_i is initialized as $1/K$, and then determined by the averaged cross-entropy loss in the last training epoch (denoted by $\hat{\mathcal{L}}_{\mathcal{G}_i}$) with a softmax modulation

$$z_i = \frac{1/\hat{\mathcal{L}}_{\mathcal{G}_i}}{\sum_{i=1}^{K} 1/\hat{\mathcal{L}}_{\mathcal{G}_i}}, \; w_i = \frac{\exp((z_i - \max(z))/\tau)}{\sum_{i=1}^{K}\exp((z_i-\max(z))/\tau)}, \tag{11}$$

where τ is a temperature parameter for controlling the sharpness of the distribution.

According to (10)–(11), the weights of underperforming subgraphs (with larger loss) will be dynamically increased during the training process, which avoid the model's over-reliance on particular structure thus achieving fairness across diverse structural patterns.

Experiment

Datasets We conduct experiments on six real-world datasets: Citation (Tang et al. 2008), Social (Li et al. 2015), WebKB (Pei et al. 2020), Airport (Ribeiro, Saverese, and Figueiredo 2017), ARXIV (Hu et al. 2020), and CORA (Bojchevski and Günnemann 2017).

Baselines We compare our methods with graph neural network models and the state-of-the-art graph transfer learning methods. For basic graph neural network models, we include classic GCN (Kipf and Welling 2017) and GAT (Veličković et al. 2018), as well as GCNII (Chen et al. 2020) and JKNet (Xu et al. 2018), which can handle heterophilic graphs. The compared graph transfer learning methods include EERM

	CITATION		SOCIAL		WebKB		
	A→D	D→A	B1→B2	B2→B1	T→C	C→T	Rank
GCN	65.31±1.13	64.51±0.73	60.95±1.13	57.35±1.29	71.91±1.26	72.46±1.32	6.8
GAT	67.50±1.79	65.05±0.94	56.35±2.96	50.73±4.75	72.90±2.98	71.15±1.75	7.3
GCNII	68.27±0.63	62.87±3.75	57.00±0.39	55.96±0.29	71.58±1.25	69.73±1.26	7.6
JKNet	66.42±1.11	63.97±1.57	58.47±1.81	58.30±1.05	70.38±1.52	73.88±0.98	7.3
EERM	55.63±4.39	52.21±4.57	49.78±3.02	48.22±2.60	55.74±0.39	57.70±1.32	10.7
MIXUP	65.10±2.50	65.05±0.94	48.30±1.47	47.22±2.78	71.48±0.46	72.35±0.49	9.0
DANE	70.37±1.96	67.45±1.03	51.05±7.15	53.48±2.13	73.33±1.91	74.86±1.02	4.8
UDAGCN	75.44±3.62	65.49±0.69	59.78±1.66	60.94±1.65	70.82±3.34	72.79±1.99	5.0
GRADE	71.80±1.09	66.07±0.24	62.13±0.99	62.03±0.94	76.94±1.18	74.32±0.95	3.0
STRURW	68.39±4.09	66.38±0.48	63.12±1.21	61.91±1.38	75.96±1.05	74.86±0.77	3.0
CS-GNN	**78.39±2.15**	**69.28±1.35**	**65.09±0.58**	**63.59±0.40**	**77.49±1.75**	**75.41±1.20**	1.0

Table 2: Node classification performance for directly transfer on `Citation`, `Social` and `WebKB` datasets.

	AIRPORT	ARXIV			CORA		
	Avg.	TIME1	TIME2	DEGREE	WORD	DEGREE	Rank
GCN	40.82	30.40±1.26	47.84±0.55	54.76±0.91	64.02±0.43	54.13±0.18	8.0
GAT	45.97	31.26±1.13	48.84±0.53	55.91±0.22	66.15±0.23	56.66±0.48	4.5
GCNII	45.05	24.39±0.92	45.16±1.02	52.00±0.78	64.94±0.56	53.27±0.29	9.0
JKNet	46.40	28.35±1.01	46.43±0.70	54.74±0.21	63.16±1.91	55.78±0.20	7.7
EERM	47.03	OOM	OOM	OOM	60.59±0.39	54.00±0.10	9.8
MIXUP	45.71	21.91±4.29	48.61±0.38	53.90±0.18	65.86±0.13	57.21±0.44	6.2
DANE	43.53	33.58±1.76	46.63±1.82	54.24±1.08	64.68±0.27	54.45±0.35	6.8
UDAGCN	48.55	32.52±0.35	48.08±1.56	55.75±0.38	64.26±0.37	54.34±0.43	5.3
GRADE	51.01	33.00±0.64	50.96±1.33	55.47±0.18	63.90±0.18	57.50±0.43	3.8
STRURW	49.96	32.61±1.46	49.84±0.88	55.28±0.41	65.23±0.42	57.18±0.22	3.8
CS-GNN	**52.81**	**35.32±1.08**	**52.50±0.39**	**57.60±0.83**	**66.34±0.26**	**58.55±0.26**	1.0

Table 3: Node classification performance for directly transfer on `Airport`, `ARXIV` and `CORA` datasets.

	ACM → DBLP			DBLP → ACM			
	$p = 0.1\%$	$p = 1\%$	$p = 10\%$	$p = 0.1\%$	$p = 1\%$	$p = 10\%$	Rank
GCN	73.44±1.78	91.58±0.61	97.47±0.22	59.59±0.53	72.21±0.68	79.43±0.37	6.1
GAT	74.35±1.44	89.01±0.51	98.19±0.17	61.57±2.25	71.36±0.42	79.65±0.38	6.0
GCNII	71.90±3.26	81.90±2.35	95.94±0.31	61.17±1.05	70.73±1.54	78.00±0.89	9.2
JKNet	67.80±0.79	86.78±2.75	97.18±0.16	61.77±1.62	70.99±1.59	77.83±0.62	8.5
EERM	64.49±4.08	86.01±2.81	97.43±0.21	63.76±1.39	69.71±1.96	75.93±3.17	8.5
MIXUP	65.42±1.30	90.31±0.49	94.14±0.99	53.42±2.93	69.67±1.00	73.14±2.61	10.2
DANE	73.31±3.76	95.70±1.13	98.26±0.26	63.56±1.05	72.44±1.14	78.85±0.43	4.2
UDAGCN	73.21±1.61	91.33±3.01	97.58±0.20	63.11±1.04	73.51±1.00	79.71±0.22	5.0
GRADE	75.49±2.34	91.89±0.48	97.61±0.25	62.76±0.59	73.30±1.21	79.80±0.74	4.0
STRURW	74.93±2.44	93.21±0.25	98.30±0.03	63.55±1.34	71.92±0.78	**80.24±0.26**	3.2
CS-GNN	**83.54±1.05**	**97.39±0.09**	**98.37±0.16**	**67.94±2.26**	**76.40±1.09**	79.95±0.15	1.2

Table 4: Node classification performance with fine-tuning on `Citation` dataset, where p denotes the percentage of labeled nodes in used for fine-tuning the target graph and rank indicates the average rank over all settings.

(Wu et al. 2022), MIXUP (Wang et al. 2021), DANE (Zhang et al. 2019b), UDAGCN (Wu et al. 2020), GRADE (Wu, He, and Ainsworth 2023), and STRURW (Liu et al. 2023). We use standard GCN as backbone for these methods.

Settings For all compared methods, we set their depth to 2 layers and use the implementations from the PyTorch Geometric Library in all experiments. The representation dimension is set to 64. Other hyperparameters of baseline methods are set to the suggested values in their respective papers or are carefully tuned for fairness. For methods combined with different GNN training pipelines (e.g. STRURW has three options, including STRURW-ERM, STRURW-MIX, and STRURW-ADV), we select the one that performs best. Accuracy scores are used as the final evaluation metric, and we report the mean and standard deviation over five runs.

Performance Comparison The results of direct transfer on six real-world datasets are presented in Tables 2 and 3. For the airport datasets, we report the average accuracy score for all its six pairs of transfers. It can be observed that in all cases, our proposed algorithm achieves superior node classification performance compared to basic graph neural network models and other transfer learning methods. Specifically, CS-GNN outperforms the GNN backbones by 5.8% in average on absolute accuracy, and achieves up to 4.0%

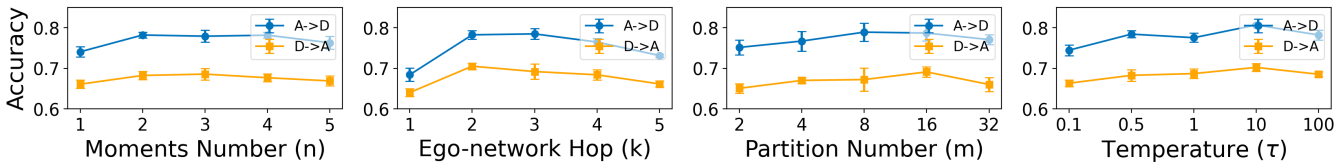

Figure 4: Performance with varying hyperparameter settings (Citation Network).

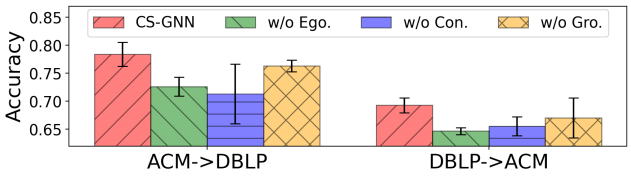

Figure 5: Ablation results for CS-GNN (Citation Network).

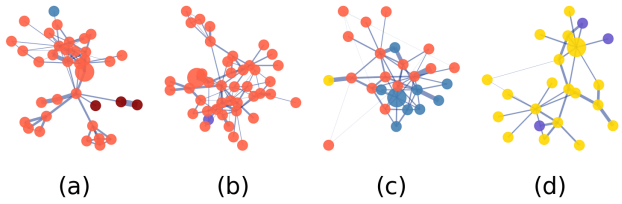

Figure 6: Visualization of ego structure patterns on the Citation dataset, where color represents node class, and the thickness of edge represents the contextual attention weight. (a)-(b) from source domain, (c)-(d) from target domain.

improvement compared with state-of-the-art transfer learning methods on the Citation dataset.

In addition to direct transfer, we also evaluated the performance of transfer learning methods with labeled samples in target graph to fine-tune the GNN, as presented in Table 4. The results show that when the number of labeled nodes in the target domain is sparse (e.g., $p = 0.1\%$), most existing methods struggle to perform well, resulting in lower accuracy. However, the proposed method, CS-GNN, demonstrates a significant improvement, achieving approximately 10% accuracy gain compared to the baseline models. As the percentage of labeled nodes increases, the performance of all methods increases accordingly. When the target domain has a sufficient number of labeled nodes (e.g., $p = 10\%$), the knowledge from the source domain is overshadowed by fine-tuning on the target domain, leading to high classification performance across all methods. Still, the proposed CS-GNN achieves the best performance with different percentage of labeled nodes in the target domain.

Hyperparameter Analysis We conducted a comprehensive hyperparameter analysis to understand the sensitivity and impact of key parameters in CS-GNN. Specifically, we examined the effects of four hyperparameters: the number of moments of feature smoothness considered (denoted by n) and the hop of the ego-networks (denoted by k), the number of network partitions (denoted by m) and temperature τ, as shown in Figure 4. The number of moments was varied from 1 to 5, and results indicate that increasing the moments number improves accuracy, particularly when moving from 1 to 2 moments. Beyond 2 moments, the accuracy stabilizes, suggesting diminishing returns with higher moments. Similarly, the ego-network hop was varied from 1 to 5, and the optimal performance was observed at 2 hops. For the partition number, we tested values from 2 to 32. The accuracy showed a moderate increase with the number of partitions, with the best performance around 8 partitions. Finally, we explored the temperature parameter, varying it from 0.1 to 100. A temperature of 10 provided the highest accuracy, with lower and higher values resulting in a performance drop.

Ablation Study To evaluate the contributions of different components in CS-GNN, we conducted ablation study on *Citation* networks. We compare the full CS-GNN model against three variants: CS-GNN that omits the construction of ego-networks (w/o Ego.); CS-GNN that replaces our contextual message passing with the standard message passing process (w/o Con.); and CS-GNN that eliminates the group-wise fairness loss using only the standard cross-entropy loss instead (w/o Gro.). As shown in Figure 5, all three variants lead to a noticeable drop in accuracy, which confirms that each component of CS-GNN contributes to its superior performance, with the ego-network construction and contextual message-passing playing particularly crucial roles.

Visualization Figure 6 visualizes the contextual attention mechanism for a subset of nodes on the Citation dataset. It is evident that distinct ego-networks allocate customized attention weights to their neighbors, forming contextual structure knowledge that is beneficial for graph transfer learning.

Conclusion

This paper proposed a novel Contextual Structural Graph Neural Network (CS-GNN) to address the homophily discrepancy challenge in graph transfer learning. It employed a customized attention mechanism to capture diverse local structural cues, facilitating effective structural knowledge transfer between different domains. It included an ego-network module for extracting local structural diversity and a moment-based approach to identify structural patterns without requiring ground-truth labels. A feature smoothness matrix, derived from node attributes, guided the attention mechanism during feature aggregation. Furthermore, a group-wise fairness loss ensured balanced learning across various structural patterns, enhancing the model's capability to transfer knowledge effectively. Comprehensive experiments proved the effectiveness of the proposed method.

Acknowledgments

This work was partially supported by the Natural Science Foundation of Jiangsu Province (Grant No. BK20222003), the Collaborative Innovation Center of Novel Software Technology and Industrialization, and the Sino-German Institutes of Social Computing. The corresponding author is Wenzhong Li (lwz@nju.edu.cn).

References

Abu-El-Haija, S.; Perozzi, B.; Kapoor, A.; Harutyunyan, H.; Alipourfard, N.; Lerman, K.; Steeg, G. V.; and Galstyan, A. G. 2019. MixHop: Higher-Order Graph Convolutional Architectures via Sparsified Neighborhood Mixing. In *International Conference on Machine Learning*.

Acuna, D.; Zhang, G.; Law, M. T.; and Fidler, S. 2021. f-Domain Adversarial Learning: Theory and Algorithms. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, 66–75. PMLR.

bao Yang, T.; Wang, Y.; Yue, Z.; Yang, Y.; Tong, Y.; and Bai, J. 2021. Graph Pointer Neural Networks. In *AAAI Conference on Artificial Intelligence*.

Ben-David, S.; Blitzer, J.; Crammer, K.; Kulesza, A.; Pereira, F. C.; and Vaughan, J. W. 2010. A theory of learning from different domains. *Machine Learning*, 79: 151–175.

Bo, D.; Wang, X.; Shi, C.; and Shen, H. 2021. Beyond Low-frequency Information in Graph Convolutional Networks. In *AAAI Conference on Artificial Intelligence*.

Bojchevski, A.; and Günnemann, S. 2017. Deep Gaussian Embedding of Graphs: Unsupervised Inductive Learning via Ranking. *arXiv: Machine Learning*.

Chen, M.; Wei, Z.; Huang, Z.; Ding, B.; and Li, Y. 2020. Simple and deep graph convolutional networks. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org.

Chuang, C.-Y.; and Jegelka, S. 2024. Tree mover's distance: bridging graph metrics and stability of graph neural networks. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22.

Dai, Q.; Wu, X.-M.; Xiao, J.; Shen, X.; and Wang, D. 2023. Graph Transfer Learning via Adversarial Domain Adaptation With Graph Convolution. *IEEE Trans. on Knowl. and Data Eng.*, 35(5): 4908–4922.

Dai, W.; Yang, Q.; Xue, G.-R.; and Yu, Y. 2007. Boosting for transfer learning. In *Proceedings of the 24th International Conference on Machine Learning*, ICML '07. New York, NY, USA.

Geler, Z.; Kurbalija, V.; Ivanović, M.; Radovanović, M.; and Dai, W. 2019. Dynamic Time Warping: Itakura vs Sakoe-Chiba. In *2019 IEEE International Symposium on INnovations in Intelligent SysTems and Applications (INISTA)*, 1–6.

Hong, X.; Li, W.; Wang, C.; Lin, M.; and Lu, S. 2024. Label Attentive Distillation for GNN-Based Graph Classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 8499–8507.

Hong, X.; Zhang, T.; Cui, Z.; Huang, Y.; Shen, P.; Li, S.; and Yang, J. 2021a. Graph game embedding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 7711–7720.

Hong, X.; Zhang, T.; Cui, Z.; and Yang, J. 2021b. Variational gridded graph convolution network for node classification. *IEEE/CAA Journal of Automatica Sinica*, 8(10): 1697–1708.

Hou, Y.; Zhang, J.; Cheng, J.; Ma, K.; Ma, R. T. B.; Chen, H.; and Yang, M.-C. 2020. Measuring and Improving the Use of Graph Information in Graph Neural Networks. In *International Conference on Learning Representations*.

Hu, W.; Fey, M.; Zitnik, M.; Dong, Y.; Ren, H.; Liu, B.; Catasta, M.; and Leskovec, J. 2020. Open graph benchmark: datasets for machine learning on graphs. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20.

Jin, D.; Yu, Z.; Huo, C.; Wang, R.; Wang, X.; He, D.; and Han, J. 2021. Universal Graph Convolutional Networks. In *Neural Information Processing Systems*.

Karypis, G.; and Kumar, V. 1998. A Fast and High Quality Multilevel Scheme for Partitioning Irregular Graphs. *SIAM J. Sci. Comput.*, 20(1): 359–392.

Keriven, N.; and Peyré, G. 2019. Universal Invariant and Equivariant Graph Neural Networks. In *Neural Information Processing Systems*.

Kipf, T. N.; and Welling, M. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *International Conference on Learning Representations*.

Li, G.; Müller, M.; Thabet, A.; and Ghanem, B. 2019. DeepGCNs: Can GCNs Go As Deep As CNNs? In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 9266–9275.

Li, J.; Hu, X.; Tang, J.; and Liu, H. 2015. Unsupervised Streaming Feature Selection in Social Media. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, CIKM '15, 1041–1050.

Li, X.; Zhu, R.; Cheng, Y.; Shan, C.; Luo, S.; Li, D.; and Qian, W. 2022. Finding Global Homophily in Graph Neural Networks When Meeting Heterophily. In *International Conference on Machine Learning*.

Lim, D.; Hohne, F. M.; Li, X.; Huang, S. L.; Gupta, V.; Bhalerao, O. P.; and Lim, S.-N. 2021. Large Scale Learning on Non-Homophilous Graphs: New Benchmarks and Strong Simple Methods. In *Advances in Neural Information Processing Systems*.

Lin, M.; Li, W.; Li, D.; Chen, Y.; Li, G.; and Lu, S. 2023. Multi-Domain Generalized Graph Meta Learning. In *AAAI Conference on Artificial Intelligence*.

Lin, M.; Li, W.; Li, D.; Chen, Y.; and Lu, S. 2022. Resource-Efficient Training for Large Graph Convolutional Networks with Label-Centric Cumulative Sampling. *Proceedings of the ACM Web Conference 2022*.

Liu, M.; Gao, H.; and Ji, S. 2020. Towards Deeper Graph Neural Networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery amp; Data Mining*, KDD '20. ACM.

Liu, M.; Wang, Z.; and Ji, S. 2020. Non-Local Graph Neural Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44: 10270–10276.

Liu, S.; Li, T.; Feng, Y.; Tran, N.; Zhao, H.; Qiang, Q.; and Li, P. 2023. Structural re-weighting improves graph domain adaptation. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org.

Mancini, M.; Bulò, S. R.; Caputo, B.; and Ricci, E. 2018. Best Sources Forward: Domain Generalization through Source-Specific Nets. *2018 25th IEEE International Conference on Image Processing (ICIP)*, 1353–1357.

Panaretos, V. M.; and Zemel, Y. 2019. Statistical Aspects of Wasserstein Distances. *Annual Review of Statistics and Its Application*, 6(1): 405–431.

Pei, H.; Wei, B.; Chang, K. C.-C.; Lei, Y.; and Yang, B. 2020. Geom-GCN: Geometric Graph Convolutional Networks. In *International Conference on Learning Representations*.

Ribeiro, L. F.; Saverese, P. H.; and Figueiredo, D. R. 2017. struc2vec: Learning Node Representations from Structural Identity. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '17. ACM.

Segu, M.; Tonioni, A.; and Tombari, F. 2023. Batch normalization embeddings for deep domain generalization. *Pattern Recogn.*, 135(C).

Shi, Y.; Huang, Z.; Feng, S.; Zhong, H.; Wang, W.; and Sun, Y. 2021. Masked Label Prediction: Unified Message Passing Model for Semi-Supervised Classification. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*.

Sun, B.; and Saenko, K. 2016. Deep CORAL: Correlation Alignment for Deep Domain Adaptation. In *Springer International Publishing*.

Tan, C.; Sun, F.; Kong, T.; Zhang, W.; Yang, C.; and Liu, C. 2018. A Survey on Deep Transfer Learning. In *International Conference on Artificial Neural Networks*.

Tang, J.; Zhang, J.; Yao, L.; Li, J.; Zhang, L.; and Su, Z. 2008. ArnetMiner: extraction and mining of academic social networks. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '08, 990–998.

Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; and Bengio, Y. 2018. Graph Attention Networks. In *International Conference on Learning Representations*.

Wang, Y.; Wang, W.; Liang, Y.; Cai, Y.; and Hooi, B. 2021. Mixup for Node and Graph Classification. In *Proceedings of the Web Conference 2021*, WWW '21, 3663–3674.

Wu, F.; Jr., A. H. S.; Zhang, T.; Fifty, C.; Yu, T.; and Weinberger, K. Q. 2019. Simplifying Graph Convolutional Networks. In Chaudhuri, K.; and Salakhutdinov, R., eds., *Proceedings of the 36th International Conference on Machine Learning, ICML*, volume 97 of *Proceedings of Machine Learning Research*, 6861–6871. PMLR.

Wu, J.; He, J.; and Ainsworth, E. A. 2023. Non-IID Transfer Learning on Graphs. In Williams, B.; Chen, Y.; and Neville, J., eds., *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI*, 10342–10350. AAAI Press.

Wu, M.; Pan, S.; Zhou, C.; Chang, X.; and Zhu, X. 2020. Unsupervised Domain Adaptive Graph Convolutional Networks. In *Proceedings of The Web Conference 2020*, WWW '20, 1457–1467.

Wu, Q.; Zhang, H.; Yan, J.; and Wipf, D. 2022. Handling Distribution Shifts on Graphs: An Invariance Perspective. In *International Conference on Learning Representations*.

Wu, Z.; Pan, S.; Chen, F.; Long, G.; Zhang, C.; and Yu, P. S. 2021. A Comprehensive Survey on Graph Neural Networks. *IEEE Transactions on Neural Networks and Learning Systems*, 32(1): 4–24.

Xu, K.; Hu, W.; Leskovec, J.; and Jegelka, S. 2019. How Powerful are Graph Neural Networks? In *International Conference on Learning Representations*.

Xu, K.; Li, C.; Tian, Y.; Sonobe, T.; and Jegelka, S. 2018. Representation Learning on Graphs with Jumping Knowledge Networks. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018*.

Yang, L.; Li, M.; Liu, L.; Niu, B.; Wang, C.; Cao, X.; and Guo, Y. 2021. Diverse Message Passing for Attribute with Heterophily. In *Neural Information Processing Systems*.

Yao, Y.; and Doretto, G. 2010. Boosting for transfer learning with multiple sources. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1855–1862.

You, Y.; Chen, T.; Wang, Z.; and Shen, Y. 2023. Graph Domain Adaptation via Theory-Grounded Spectral Regularization. In *International Conference on Learning Representations*.

Zhang, Y.; Liu, T.; Long, M.; and Jordan, M. I. 2019a. Bridging Theory and Algorithm for Domain Adaptation. In *International Conference on Machine Learning*.

Zhang, Y.; Song, G.; Du, L.; Yang, S.; and Jin, Y. 2019b. DANE: Domain Adaptive Network Embedding. In *International Joint Conference on Artificial Intelligence*.

Zhao, H.; Chen, A.; Sun, X.; Cheng, H.; and Li, J. 2024. All in One and One for All: A Simple yet Effective Method towards Cross-domain Graph Pretraining. *ArXiv*, abs/2402.09834.

Zhu, J.; Yan, Y.; Zhao, L.; Heimann, M.; Akoglu, L.; and Koutra, D. 2020. Beyond homophily in graph neural networks: current limitations and effective designs. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20.

Zhu, Q.; Yang, C.; Xu, Y.; Wang, H.; Zhang, C.; and Han, J. 2021. Transfer Learning of Graph Neural Networks with Ego-graph Information Maximization. In Beygelzimer, A.; Dauphin, Y.; Liang, P.; and Vaughan, J. W., eds., *Advances in Neural Information Processing Systems*.

Zhuang, F.; Qi, Z.; Duan, K.; Xi, D.; Zhu, Y.; Zhu, H.; Xiong, H.; and He, Q. 2019. A Comprehensive Survey on Transfer Learning. *Proceedings of the IEEE*, 109: 43–76.