Non-IID Transfer Learning on Graphs

Jun Wu[1], **Jingrui He**[1], **Elizabeth Ainsworth**[1,2]

[1]University of Illinois Urbana-Champaign
[2]USDA ARS Global Change and Photosynthesis Research Unit
{junwu3, jingrui, ainswort}@illinois.edu

Abstract

Transfer learning refers to the transfer of knowledge or information from a relevant source domain to a target domain. However, most existing transfer learning theories and algorithms focus on IID tasks, where the source/target samples are assumed to be independent and identically distributed. Very little effort is devoted to theoretically studying the knowledge transferability on non-IID tasks, e.g., cross-network mining. To bridge the gap, in this paper, we propose rigorous generalization bounds and algorithms for cross-network transfer learning from a source graph to a target graph. The crucial idea is to characterize the cross-network knowledge transferability from the perspective of the Weisfeiler-Lehman graph isomorphism test. To this end, we propose a novel Graph Subtree Discrepancy to measure the graph distribution shift between source and target graphs. Then the generalization error bounds on cross-network transfer learning, including both cross-network node classification and link prediction tasks, can be derived in terms of the source knowledge and the Graph Subtree Discrepancy across domains. This thereby motivates us to propose a generic graph adaptive network (**GRADE**) to minimize the distribution shift between source and target graphs for cross-network transfer learning. Experimental results verify the effectiveness and efficiency of our **GRADE** framework on both cross-network node classification and cross-domain recommendation tasks.

1 Introduction

Transfer learning (Pan and Yang 2009) tackles the knowledge transferability from a source domain to a relevant target domain under a distribution shift. It has been theoretically shown (Ben-David et al. 2010; Zhang et al. 2019a; Acuna et al. 2021) that the generalization performance of a learning algorithm can be improved by leveraging the knowledge from the source domain, when the source and target domains have the same labeling space (also known as domain adaptation (Pan and Yang 2009)). To be more specific, the expected target error could be bounded in terms of the prediction error of the source domain and the distribution discrepancy across domains. It thus motivates a line of practical approaches with domain discrepancy minimization in the latent feature space (Ganin et al. 2016; Acuna et al. 2021). However, it is noteworthy that most of the existing theoretical guarantees and empirical algorithms hold the IID assumption that all the source/target samples are drawn independently from an identical source/target distribution. This hinders their applications on other tasks with non-IID data, e.g., node classification (Kipf and Welling 2017; Wu and He 2019) and recommendation (Zhao, Li, and Fu 2019; Zhou et al. 2021a,b) across domains.

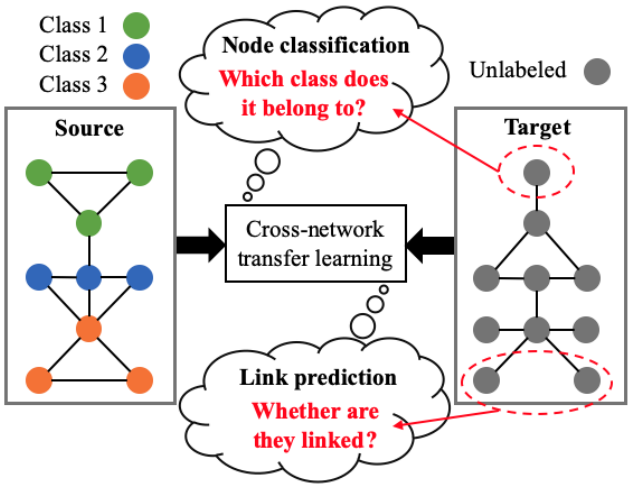


Figure 1: Illustration of the cross-network transfer learning (best viewed in color). Given a labeled source graph (color indicates node label) and an unlabeled target graph, cross-network transfer learning tackles the classification and link prediction tasks in the target graph, by leveraging the auxiliary information from the source graph.

In this paper, we focus on studying the problem of cross-network transfer learning, where the knowledge can be transferred from a source graph to a target graph[1]. To be specific, we consider the following cross-network mining tasks (see Figure 1). (1) Node classification (e.g., cross-network role identification (Zhu et al. 2021)): It aims to predict the class labels of nodes, by leveraging the knowledge from a source graph with fully labeled nodes. Following (Wu et al. 2020), we consider the unsupervised learning scenarios where the target domain has no label information. (2) Link prediction (e.g., cross-domain recommendation (Li and Tuzhilin 2020; Zhao, Li, and Fu 2019)): It predicts the

[1]In this paper, we use "graph" and "network" interchangeably to denote the graph-structured data in every domain.

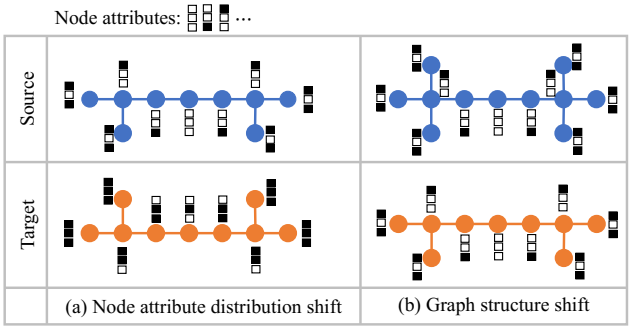


Figure 2: Illustration of distribution shift on graphs. (a) Source and target graphs share the same graph structure, but they have different node attribute distributions (node attribute is assumed to be a 3-dimensional feature vector, where the "black" box denotes 1 and the "white" box denotes 0). (b) Source and target graphs share a similar node attribute distribution but different graph structures.

missing links in the incomplete target graph, by leveraging knowledge from a complete source graph. The unique challenge of cross-network transfer learning lies in the interdependence nature of nodes within the graph. As shown in Figure 2, the distribution shift between source and target graphs can be induced by node attribute and graph structure.

We start by developing a novel distribution discrepancy measure named Graph Subtree Discrepancy between source and target graphs. This is motivated by the connection of existing message-passing graph neural networks (Xu et al. 2019; Hamilton, Ying, and Leskovec 2017) and Weisfeiler-Lehman graph kernel (Shervashidze et al. 2011). On one hand, the Weisfeiler-Lehman graph kernel holds that the non-parametric graph similarity can be decomposed into the similarity of a sequence of subtrees rooted at every node. On the other hand, message-passing graph neural networks tend to iteratively aggregate the messages from nodes' local neighborhoods in a parametric way. Then, Graph Subtree Discrepancy is designed to measure the distribution shift of graphs by estimating the similarity/difference of subtree representations learned from a message-passing graph neural network. As a result, it can inherit the benefits of high expressiveness from message-passing graph neural networks and feasible explanations from the Weisfeiler-Lehman graph kernel. Based on Graph Subtree Discrepancy, the generalization error bounds of cross-network transfer learning can be derived for cross-network mining tasks. By empirically minimizing the error upper bounds, we propose a generic graph adaptive network (**GRADE**) for cross-network transfer learning. The efficacy of the proposed **GRADE** framework is confirmed on various cross-network mining data sets. The major contributions of this paper are summarized as follows.

- We propose a novel Graph Subtree Discrepancy to measure the distribution shift of nodes' data distribution between source and target graphs. The generalization error bounds of cross-network transfer learning can then be derived based on Graph Subtree Discrepancy.
- We propose a generic Graph Adaptive Network (**GRADE**) framework for cross-network transfer learning, followed by the instantiations on cross-network node classification and cross-domain recommendation tasks.
- Extensive experiments demonstrate the effectiveness of our proposed **GRADE** framework on cross-network node classification and cross-domain recommendation tasks.

The rest of this paper is organized as follows. We review the related work in Section 2, and introduce our problem setting of cross-network transfer learning in Section 3. Section 4 provides the theoretical analysis for cross-network transfer learning, followed by the proposed **GRADE** framework in Section 5. We show the experimental results in Section 6, and finally conclude this paper in Section 7.

2 Related Work

2.1 Transfer Learning

Transfer learning (Pan and Yang 2009) refers to the knowledge transferability from a source domain to a relevant target domain. It is theoretically guaranteed (Mansour, Mohri, and Rostamizadeh 2009; Ben-David et al. 2010; Acuna et al. 2021; Wu and He 2021, 2022a,b; Wu et al. 2022a,b) that under mild conditions, the generalization performance of a learning algorithm on the target domain can be improved by leveraging the knowledge from the source domain. One common assumption behind those theoretical guarantees is that all the source/target samples are drawn independently from an independent and identical source/target probability distribution. More recently, (Levie, Isufi, and Kutyniok 2019; Ruiz, Chamon, and Ribeiro 2020; Zhu et al. 2021) proposed to analyze the transferability of graph neural networks using graphons or ego-graphs. Nevertheless, those works explore whether graph neural networks are transferable given two graphs. In contrast, in this paper, by using a hypothesis-dependent Graph Subtree Discrepancy, we show how knowledge can be transferred across graphs. The resulting theoretical analysis provides insight into designing practical cross-network transfer learning algorithms.

2.2 Cross-Network Mining

Cross-network mining aims to exhibit the informative patterns from multiple relevant networks/graphs for a variety of mining tasks, e.g. cross-network node classification (Wu et al. 2020; Zhang et al. 2019b; Zhu et al. 2021), multi-domain graph clustering (Ni et al. 2015), cross-domain recommendation (Zhao, Li, and Fu 2019; Li and Tuzhilin 2020), etc. There are two lines of solutions in exploring the knowledge transferability among different graphs. One is (Fang et al. 2015; Wu et al. 2020; Zhang et al. 2019b; Dai et al. 2022) that it extracts the signature subgraphs or consistent aggregation patterns from source and target graphs without a theoretical explanation. The other one is (Hu et al. 2020a,b; Qiu et al. 2020; Han et al. 2021) that it first pretrains the graph neural networks on a large source graph for encoding the general graph structures, and then fine-tunes on the target graph for extracting the task-specific information. This might lead to a sub-optimal solution for unsupervised cross-network node classification tasks where no labeled target nodes are available for fine-tuning.

3 Preliminaries

3.1 Notation

Suppose that a graph is represented as $G = (V, E)$, where $V = \{v_1, \cdots, v_n\}$ is the set of n nodes and $E \subseteq V \times V$ is the edge set in the graph. In this paper, we consider the attributed graph. That is, each node is associated with a D-dimensional feature vector $x_v \in \mathbb{R}^D$. In the node classification task, each node is associated with a class label $y_v \in \{1, \cdots, C\}$, where C is the total number of classes. The graph G can also be represented by an adjacency matrix $A \in \mathbb{R}^{n \times n}$, where A_{ij} is the similarity between v_i and v_j on the graph. In the context of cross-network network mining, we denote $G^s = (V^s, E^s, X^s)$ and $G^t = (V^t, E^t, X^t)$ to be the source and target graphs, respectively. The associated adjacent matrices of source and target graphs are represented as A^s and A^t, respectively.

3.2 Problem Setting

Following (Wu et al. 2020; Zhu et al. 2021), we formally define the cross-network transfer learning problem as follows.

Definition 1. *(***Cross-Network Transfer Learning***) Given a source graph G^s and a target graph G^t, cross-network transfer learning aims to improve the prediction performance of node classification or link prediction in the target graph by using knowledge from the source graph, with the assumption that source and target graphs are related.*

As illustrated in Figure 2, the distribution shift across graphs can be induced by both node attribute[2] and graph structure. Compared to standard transfer learning (Ben-David et al. 2010), the additional distribution shift over complex graph structure leads to a much more challenging cross-network transfer learning problem setting.

4 Theoretical Results

In this section, we propose a novel Graph Subtree Discrepancy (GSD) to measure the distribution shift across graphs. It is inspired by the association of Weisfeiler-Lehman (WL) graph kernels (Shervashidze et al. 2011) and graph neural networks (GNNs) (Kipf and Welling 2017; Hamilton, Ying, and Leskovec 2017).

4.1 Connection of WL Kernels and GNNs

Weisfeiler-Lehman graph subtree kernel (Shervashidze et al. 2011) aims to measure the semantic similarity of a pair of input graphs. It learns a sequence of Weisfeiler-Lehman subgraphs for an input graph G: $\{G_0, G_1, \cdots, G_m, \cdots\} = \{(V, E, f_0), (V, E, f_1), \cdots, (V, E, f_m), \cdots\}$, where $G_0 = G$ and $f_0(v)$ denotes the raw node attributes of v for any $v \in G$. The "relabeling" function f_j ($j = 1, \cdots, m, \cdots$) aims to represent the subtree structure rooted at $v \in G$ into a novel representation at every iteration (see Definition 2). Then, the structural information around node v can be represented as a sequence of Weisfeiler-Lehman subtrees $\{f_0(v), f_1(v), \cdots, f_m(v), \cdots\}$ with different depths m.

[2]In this paper, we consider that source and target graphs share the same node attribute space, but they might have different node attribute distributions.

Definition 2. *(Weisfeiler-Lehman Subtree (Shervashidze et al. 2011)) Given a graph $G = (V, E)$ associated with initial node attributes $f_0(v)$ for $v \in V$, the Weisfeiler-Lehman subtree of depth m rooted at $v \in V$ can be represented as $f_m(v) := f_m\left(f_{m-1}(v); \cup_{u \in N(v)} f_{m-1}(u)\right)$ where $N(v)$ denotes the nearest neighbors of root node v.*

Note that in the original work (Shervashidze et al. 2011), the "relabeling" function of the WL subtree is simply given by the hashing table due to the discrete node attributes in the graph. Later, it is revealed (Hamilton, Ying, and Leskovec 2017; Wu, He, and Xu 2019; Geerts, Mazowiecki, and Perez 2021) that this WL subtree can actually recover the crucial message-passing modular in many popular message-passing GNNs, where the "relabeling" function $f_m(\cdot)$ is instantiated by deep neural networks for learning continuous node representation. We observe that for single graph mining task, e.g., node classification and link prediction, only the message-passing philosophy of the Weisfeiler-Lehman subtree is studied to design the graph neural networks (Hamilton, Ying, and Leskovec 2017; Kipf and Welling 2017). It maps the structurally equivalent nodes within one graph into the same low-dimensional representation in a latent feature space. However, in the context of cross-network transfer learning, we highlight that the following WL subtree kernel sheds light on measuring the distribution shift of source and target graphs in the feature space learned by GNNs.

Definition 3. *(Weisfeiler-Lehman Subtree Kernel (Shervashidze et al. 2011)) Given any two graphs $G = (V, E)$ with n nodes and $G' = (V', E')$ with n' nodes, the Weisfeiler-Lehman subtree kernel on two graphs G and G' with M iterations is defined as:*

$$k\left(G, G'\right) = \frac{1}{nn'} \sum_{m=0}^{M} \sum_{v \in G} \sum_{v' \in G'} \delta\left(f_m(v), f_m(v')\right)$$

where $\delta(\cdot, \cdot)$ is the Dirac kernel, that is, it is 1 when its arguments are equal and 0 otherwise, and $f_m(v)$ represents the subtree pattern of depth m rooted at v.

The WL subtree kernel on graphs G and G' is rewritten as

$$k\left(G, G'\right) = \sum_{m=0}^{M} s\left(\hat{\mathbb{P}}\left(G_m\right), \hat{\mathbb{Q}}\left(G'_m\right)\right)$$

where $\hat{\mathbb{P}}$ (or $\hat{\mathbb{Q}}$) is the empirical node distribution of graph G (or G'), i.e., $\hat{\mathbb{P}}(\tau | G_m) = \frac{1}{n} \sum_{i=1}^{n} \delta(f_m(v_i), \tau)$ for any subtree pattern τ. Here $s(\cdot, \cdot)$ is an inner product metric to measure the distribution similarity of $\hat{\mathbb{P}}(G_m)$ and $\hat{\mathbb{Q}}(G'_m)$, i.e., $s(\hat{\mathbb{P}}(G_m), \hat{\mathbb{Q}}(G'_m)) = \langle (\hat{\mathbb{P}}(\tau_1 | G_m), \cdots, \hat{\mathbb{P}}(\tau_k | G_m), \cdots), (\hat{\mathbb{P}}(\tau_1 | G'_m), \cdots, \hat{\mathbb{P}}(\tau_k | G'_m), \cdots) \rangle$. Furthermore, we have the following observations. (1) This nonparametric graph kernel (Shervashidze et al. 2011) can be exploited to measure the distribution shift between source and target graphs by counting the occurrence of subtrees when the node attributes are discrete. However, it might suffer when using continuous node attribute to estimate the graph similarity (or distribution discrepancy in the context of cross-network transfer learning). (2) Previous

works (Wu et al. 2020; Zhu et al. 2021; Dai et al. 2022) focus on characterizing the distribution discrepancy over only the m^{th} subtree representation. They can thereby partially reveal the distribution discrepancy between source and target graphs in practice.

4.2 Graph Subtree Discrepancy

Inspired by the connection of the WL subtree kernel and message-passing GNNs, we propose a parametric Graph Subtree Discrepancy (GSD). GSD measures the distribution discrepancy of graphs in the latent feature space induced by the message-passing GNNs.

Following WL subtree kernel (Shervashidze et al. 2011), we assume that given a graph $G = (V, E)$, the WL subtrees (i.e., $\{f_m(v)|v \in V\}$) with a fixed depth m are conditionally independent with respect to WL subgraph G_{m-1} at depth $m-1$, i.e., $f_m(u) \perp f_m(v)|G_{m-1}$. In this case, given the WL subgraph G_{m-1}, the subtree representations $\{f_m(v)|v \in V\}$ can thus be considered as IID samples. This tells us that the subtree (at depth m) distribution shift of source and target graphs can be measured by any existing distribution discrepancy measures, e.g, JS-divergence (Ben-David et al. 2010; Ganin et al. 2016), discrepancy distance (Mansour, Mohri, and Rostamizadeh 2009), Maximum Mean Discrepancy (Gretton et al. 2012) and f-divergence (Acuna et al. 2021). Therefore, our Graph Subtree Discrepancy can be formally defined as follows.

Definition 4. *(Graph Subtree Discrepancy) Given two graphs $G^s = (V^s, E^s)$ and $G^t = (V^t, E^t)$, the graph subtree discrepancy between them can be defined as:*

$$d_{GSD}\left(G^s, G^t\right) = \lim_{M\to\infty} \frac{1}{M+1} \sum_{m=0}^{M} d_b\left(G_m^s, G_m^t\right) \quad (1)$$

where $d_b(\cdot,\cdot)$ is the base domain discrepancy.

For example, we can use the discrepancy distance (Mansour, Mohri, and Rostamizadeh 2009) to instantiate the base domain discrepancy $d_b(\cdot,\cdot)$, which is defined as

$$d_b(G_m^s, G_m^t) = \sup_{h,h'\in\mathcal{H}} \Big|\mathbb{E}_{v\in V^s}\left[L\left(h\left(f_m(v)\right), h'(f_m(v))\right)\right] - \mathbb{E}_{v\in V^t}\left[L\left(h\left(f_m(v)\right), h'(f_m(v))\right)\right]\Big| \quad (2)$$

We see that GSD recursively estimates the subtrees' distribution discrepancy between source and target graphs at different depths. Here the subtree representation can be learned by existing passage-passing GNNs (Kipf and Welling 2017; Hamilton, Ying, and Leskovec 2017; Veličković et al. 2018; Xu et al. 2019). More discussion on the properties of GSD can be found in Appendix A.1.

4.3 Error Bounds

Next, we derive the error bounds for cross-network transfer learning based on GSD. Let $\mathcal{H}$ be the hypothesis space. For any hypothesis $h \in \mathcal{H}$, the node classification error on graph G of a message-passing GNN (with L convolutional layers) can be defined as $\epsilon(h \circ f) = \mathbb{E}_{v\in G}[\mathcal{L}(h(f(v)), y)]$, where $f(\cdot)$ extracts the node representation[3] and $\mathcal{L}(\cdot,\cdot)$ is the loss function. Without loss of generality, we focus on the commonly used GNNs with the feature extraction $f(v) = f_L(v)$ (using only the output of the final graph convolutional layer (Kipf and Welling 2017; Hamilton, Ying, and Leskovec 2017; Veličković et al. 2018)). The following theorem shows that in the cross-network node classification, the expected error in the target graph can be bounded in terms of the classification error in the source graph and the distribution discrepancy across graphs.

Theorem 1. *(Cross-Network Node Classification) Assume that there are a source graph G^s and a target graph G^t and the base domain discrepancy $d_b(\cdot,\cdot)$ of GSD is instantiated by the discrepancy distance (see Eq. (2)). Given a message-passing GNN with the feature extractor f and the hypothesis $h \in \mathcal{H}$, the node classification error in the target graph can be bounded as follows.*

$$\epsilon_t(h \circ f) \le \epsilon_s(h \circ f) + d_{GSD}\left(G^s, G^t\right) + \lambda^* + R^*$$

where $\lambda^* = \mathbb{E}_{v\in V^t}[\mathcal{L}(h_*^s(f(v)), h_*^t(f(v)))]$ measures the prediction difference of optimal source and target hypotheses on the target nodes, and $R^* = \mathbb{E}_{v\in V^s}[\mathcal{L}(y, h_*^s(f(v)))] + \mathbb{E}_{v\in V^t}[\mathcal{L}(h_*^t(f(v)), y)]$ is the Bayes error on the source and target graphs. y is the class label of v. In this case, $h_*^s \in \arg\min_{h\in\mathcal{H}} \mathbb{E}_{v\in V^s}[\mathcal{L}(h(f(v)), y)]$ and $h_*^t \in \arg\min_{h\in\mathcal{H}} \mathbb{E}_{v\in V^t}[\mathcal{L}(h(f(v)), y)]$ are the optimal source and target hypotheses, respectively[4].

Compared to previous work (Zhu et al. 2021), our error bound of Theorem 1 is hypothesis-dependent. That is, the knowledge transferability can be enhanced, if the message-passing GNN learns a latent feature space such that the subtree distribution shift of source and target graphs is minimized. This is in sharp contrast to previous work which focuses on evaluating the transferability of a trained GNN model. Therefore, Theorem 1 provides insights on designing practical cross-network transfer learning algorithms (shown in Section 5) by minimizing the error upper bound.

We have similar results for cross-network link prediction. That is, the label space of link prediction is $\mathcal{Y} = \{0, 1\}$, where $y = 1$ if the link of a pair of nodes exists, $y = 0$ otherwise. In this case, the loss of the intra-graph link prediction is defined as $\epsilon(h \circ f) = \mathbb{E}_{u,v\in V\times V}[\mathcal{L}(h([f_L(u)||f_L(v)]), y)]$, where $[\cdot||\cdot]$ denote the vector concatenation.

Theorem 2. *(Cross-Network Link Prediction) With assumptions in Theorem 1, and let $\mathcal{L}(y, \tilde{y}) = |y - \tilde{y}|$ and the hypothesis class $\mathcal{H}$ is given by the multi-layer perceptrons, if the loss of the link prediction is defined as $\epsilon^{link}(h \circ f) = \mathbb{E}_{u,v\in V\times V}[\mathcal{L}(h([f_L(u)||f_L(v)]), y)]$, then the link prediction error in the target graph can be bounded as follows.*

$$\epsilon_t^{link}(h) \le \epsilon_t^{link}(h) + d_{GSD}\left(G_s, G_t\right) + \lambda_{link}^* + R_{link}^*$$

[3] In this case, a message-passing GNN can be explained as extracting the high-order node representation of v or the subtree representation rooted v, when stacking multiple convolutional layers.

[4] In addition, when the message-passing GNN uses the jumping knowledge from all graph convolutional layers (Xu et al. 2019, 2018), we have similar observation (see Corollary 1 in Appendix).

where $\lambda^*_{link} = \mathbb{E}_{(u,v)\in V^t\times V^t}[\mathcal{L}(h^s_*([f(u)||f(v)]), h^t_*([f(u)||f(v)]))]$ *measures the difference of optimal source and target hypotheses on the target graph, and* $R^*_{link} = \mathbb{E}_{(u,v)\in V^s\times V^s}[\mathcal{L}(y, h^s_*([f(u)||f(v)]))] + \mathbb{E}_{(u,v)\in V^t\times V^t}[\mathcal{L}(h^t_*([f(u)||f(v)]), y)]$ *is the Bayes error. In this case,* $h^s_* \in \arg\min_{h\in\mathcal{H}} \mathbb{E}_{(u,v)\in V^s\times V^s}[\mathcal{L}(h([f(u)||f(v)]), y)]$, *and* $h^t_* \in \arg\min_{h\in\mathcal{H}} \mathbb{E}_{(u,v)\in V^t\times V^t}[\mathcal{L}(h([f(u)||f(v)]), y)]$ *are optimal source and target hypothesises, respectively.*

5 Proposed Framework

In this section, we propose a cross-network transfer learning framework named Graph Adaptive Network (**GRADE**).

5.1 Objective Function

The objective function of a generic cross-network transfer learning framework (**GRADE**) is summarized as follows.

$$\min_{\theta} C(G^s;\theta) + \lambda \cdot d_{GSD}(G^s, G^t;\theta) \tag{3}$$

where θ denotes all the trainable parameters. $C(G^s;\theta)$ is the task-specific loss function on the source graph, and $d_{GSD}(G^s, G^t;\theta)$ is the discrepancy minimization between source and target graphs. $\lambda \geq 0$ is a hyper-parameter to balance these terms. Note that $C(G^s;\theta)$ might also contain the task-specific loss function on the target graph, if label information is partially available in the target domain.

5.2 Algorithms

Following the framework of Eq. (3), we present the instantiated algorithms for two cross-network mining tasks, including cross-network node classification (**GRADE-N**) and cross-domain recommendation (**GRADE-R**).

Cross-Network Node Classification We focus on the cross-network node classification setting from a source graph with labeled nodes to a target graph with only unlabeled nodes. The goal is to identify the class label of every node in the target domain, by leveraging the relevant knowledge from the source domain. The objective function of **GRADE-N** can be instantiated as follows.

$$\begin{aligned}\min_{\theta_f,\theta_h} \quad & \mathcal{L}\left(h\left(f(G^s;\theta_f);\theta_h\right), Y^s\right) \\ & + \lambda \cdot d_{GSD}\left(f(G^s;\theta_f), f(G^t;\theta_f)\right)\end{aligned} \tag{4}$$

where $f(\cdot)$ is the message-passing graph neural network function parameterized by θ_f, and $h(\cdot)$ is the classifier function (MLP is adopted in the experiments) parameterized by θ_h. $\mathcal{L}(\cdot,\cdot)$ is the cross-entropy loss function for cross-network node classification in the experiments (mean square error loss function can be applied for regression task).

Specifically, we adopt Graph Convolutional Network (GCN) (Kipf and Welling 2017) as the base model to extract the subtree representations of a graph. Then, the subtree pattern of depth m rooted at v can be represented as follows.

$$f_m(v) = \sigma\Big(\sum\nolimits_{u\in\{v\}\cup N(v)} \widehat{a}_{vu} W^m f_{m-1}(u)\Big) \tag{5}$$

where $\widehat{A} = (\widehat{a}_{vu}) \in \mathbb{R}^{n\times n}$ (n is the number of nodes) is the re-normalization of the adjacency matrix A with added self-loops, and W^m is the trainable matrix at m^{th} layer. $\sigma(\cdot)$ is the non-linear activation function. After M iterations, we obtain the sequence of subtree representations rooted at v: $f_0(v), f_1(v), \cdots, f_M(v)$, where $f_0(v)$ is the raw node attributes of v. Following (Kipf and Welling 2017), the final representation $f_M(v)$ can be applied to identify the class label of node v. In addition, we consider finite iterations (e.g., M) of the message-passing graph neural network for estimating GSD. That is,

$$d_{GSD}\left(f(G^s;\theta_f), f(G^t;\theta_f)\right) = \frac{1}{M+1}\sum_{m=0}^{M} d_b\left(G^s_m, G^t_m\right) \tag{6}$$

Cross-Domain Recommendation Cross-domain recommendation learns the user preference in the target domain, by leveraging the rich information from a relevant source domain. The objective function of **GRADE-R** can be instantiated as follows.

$$\begin{aligned}\min_{\theta_f,\theta_{h'}} \quad & \mathcal{L}\left(h'\left(f(G^s;\theta_f);\theta_{h'}\right)\right) + \mathcal{L}\left(h'\left(f(G^t;\theta_f);\theta_{h'}\right)\right) \\ & + \lambda \cdot d_{GSD}\left(f(G^s;\theta_f), f(G^t;\theta_f)\right)\end{aligned} \tag{7}$$

where $f(\cdot)$ is the message-passing graph neural network function parameterized by θ_f, and $h'(\cdot)$ is the link prediction function parameterized by $\theta_{h'}$.

More specifically, we adopt GCN (see Eq. (5)) as the base model $f(\cdot)$ to extract the subtree representations of a graph. The graph subtree discrepancy $d_{GSD}(\cdot,\cdot)$ can also be given by Eq. (6) over those subtree representations. In addition, following (He et al. 2017; Zhao, Li, and Fu 2019), we adopt the multi-layer perceptron as the link prediction function $h'(\cdot)$ to infer whether a link of two nodes exists. That is,

$$h'\left((u,v), y_{uv};\theta_{h'}\right) = \text{BCE}\left(\text{MLP}_{\theta_{h'}}\left(f(u)||f(v)\right), y_{uv}\right)$$

where y_{uv} is the link label (i.e., $y_{uv} = 1$ if u and v are linked, $y_{uv} = 0$ otherwise) for any $u, v \in V^s$ or $u, v \in V^t$. Here $\cdot||\cdot$ denotes the vector concatenation, BCE$(\cdot)$ denotes the binary cross-entropy, and $\text{MLP}_{\theta_{h'}}(\cdot)$ is a multi-layer perceptron function.

6 Experiment

6.1 Experimental Setup

Data Sets For cross-network node classification, we use the following data sets: Airport networks (Ribeiro, Saverese, and Figueiredo 2017) (Brazil, USA and Europe); Citation network (Wu et al. 2020) (ACMv9 (A) and DBLPv8 (D)); Social network (Shen et al. 2020) (Blog1 (B1) and Blog2 (B2)); and Agriculture data (Wang et al. 2021) (Maize (M) and Maize_UNL (MU)).

For cross-domain recommendation, we evaluate the models on the Amazon data set (He and McAuley 2016). We adopt two pairs of real-world cross-domain data sets from Amazon-5cores, including CD and Music, Book and Movie. Note that most existing cross-domain recommendation algorithms (Hu, Zhang, and Yang 2018; Zhang et al. 2020) assume that source and target domains have the same group

Methods	USA → Brazil	USA → Europe	Brazil → USA	Brazil → Europe	Europe → USA	Europe → Brazil	Avg.
GCN	$0.366_{\pm 0.011}$	$0.371_{\pm 0.004}$	$0.491_{\pm 0.011}$	$0.452_{\pm 0.012}$	$0.439_{\pm 0.001}$	$0.298_{\pm 0.022}$	0.403
SGC	$0.527_{\pm 0.022}$	$0.430_{\pm 0.009}$	$0.432_{\pm 0.005}$	$0.479_{\pm 0.000}$	$0.447_{\pm 0.002}$	$0.481_{\pm 0.011}$	0.466
GCNII	$0.344_{\pm 0.086}$	$0.393_{\pm 0.025}$	$0.470_{\pm 0.056}$	$0.494_{\pm 0.018}$	$0.460_{\pm 0.012}$	$0.542_{\pm 0.011}$	0.450
DAN	$0.504_{\pm 0.020}$	$0.393_{\pm 0.000}$	$0.436_{\pm 0.006}$	$0.393_{\pm 0.010}$	$0.436_{\pm 0.003}$	$0.542_{\pm 0.000}$	0.451
DANN	$0.500_{\pm 0.005}$	$0.386_{\pm 0.011}$	$0.402_{\pm 0.048}$	$0.350_{\pm 0.062}$	$0.436_{\pm 0.000}$	$0.538_{\pm 0.005}$	0.435
MDD	$0.500_{\pm 0.005}$	$0.378_{\pm 0.000}$	$0.402_{\pm 0.048}$	$0.350_{\pm 0.062}$	$0.402_{\pm 0.048}$	$0.477_{\pm 0.081}$	0.418
AdaGCN	$0.466_{\pm 0.065}$	$0.434_{\pm 0.004}$	$\mathbf{0.501_{\pm 0.003}}$	$0.486_{\pm 0.021}$	$0.456_{\pm 0.034}$	$0.561_{\pm 0.081}$	0.484
UDA-GCN	$\mathbf{0.607_{\pm 0.059}}$	$0.388_{\pm 0.007}$	$0.497_{\pm 0.005}$	$\mathbf{0.510_{\pm 0.019}}$	$0.434_{\pm 0.042}$	$0.477_{\pm 0.024}$	0.486
EGI	$0.523_{\pm 0.013}$	$0.451_{\pm 0.011}$	$0.417_{\pm 0.021}$	$0.454_{\pm 0.046}$	$0.452_{\pm 0.029}$	$\mathbf{0.588_{\pm 0.011}}$	0.481
GRADE-N	$0.550_{\pm 0.062}$	$\mathbf{0.457_{\pm 0.027}}$	$0.497_{\pm 0.010}$	$0.506_{\pm 0.004}$	$\mathbf{0.463_{\pm 0.001}}$	$\mathbf{0.588_{\pm 0.032}}$	**0.510**

Table 1: Cross-network node classification on the Airport network

Methods	Citation		Social	
	A → D	D → A	B1 → B2	B2 → B1
GCN	$0.435_{\pm 0.013}$	$0.567_{\pm 0.046}$	$0.408_{\pm 0.025}$	$0.451_{\pm 0.044}$
SGC	$0.430_{\pm 0.001}$	$0.611_{\pm 0.000}$	$0.331_{\pm 0.110}$	$0.268_{\pm 0.059}$
GCNII	$0.465_{\pm 0.002}$	$0.559_{\pm 0.009}$	$0.392_{\pm 0.022}$	$0.473_{\pm 0.061}$
DAN	$0.338_{\pm 0.005}$	$0.421_{\pm 0.060}$	$0.407_{\pm 0.015}$	$0.422_{\pm 0.015}$
DANN	$0.368_{\pm 0.021}$	$0.381_{\pm 0.013}$	$0.409_{\pm 0.019}$	$0.419_{\pm 0.022}$
MDD	$0.349_{\pm 0.029}$	$0.391_{\pm 0.034}$	$0.388_{\pm 0.012}$	$0.421_{\pm 0.015}$
AdaGCN	$0.451_{\pm 0.013}$	$0.566_{\pm 0.042}$	$0.498_{\pm 0.057}$	$0.516_{\pm 0.025}$
UDA-GCN	$\mathbf{0.516_{\pm 0.028}}$	$0.600_{\pm 0.014}$	$0.471_{\pm 0.010}$	$0.468_{\pm 0.009}$
EGI	$0.489_{\pm 0.036}$	$0.404_{\pm 0.006}$	$0.494_{\pm 0.030}$	$0.516_{\pm 0.010}$
GRADE-N	$0.475_{\pm 0.011}$	$\mathbf{0.635_{\pm 0.009}}$	$\mathbf{0.567_{\pm 0.042}}$	$\mathbf{0.541_{\pm 0.008}}$

Table 2: Cross-network node classification on the citation and social networks

Methods	M → MU		MU → M	
	MAE	R^2	MAE	R^2
GCN	$0.489_{\pm 0.005}$	$0.132_{\pm 0.020}$	$0.679_{\pm 0.030}$	$0.295_{\pm 0.051}$
GCNII	$0.467_{\pm 0.022}$	$0.192_{\pm 0.056}$	$0.687_{\pm 0.042}$	$0.254_{\pm 0.023}$
DANN	$0.492_{\pm 0.002}$	$0.104_{\pm 0.010}$	$0.721_{\pm 0.030}$	$0.192_{\pm 0.069}$
RSD	$0.523_{\pm 0.011}$	$0.019_{\pm 0.061}$	$0.729_{\pm 0.009}$	$0.170_{\pm 0.047}$
AdaGCN	$0.454_{\pm 0.069}$	$0.245_{\pm 0.082}$	$\mathbf{0.649_{\pm 0.023}}$	$0.351_{\pm 0.038}$
UDA-GCN	$0.449_{\pm 0.072}$	$0.257_{\pm 0.086}$	$0.684_{\pm 0.012}$	$0.271_{\pm 0.003}$
GRADE-N	$\mathbf{0.353_{\pm 0.038}}$	$\mathbf{0.527_{\pm 0.092}}$	$0.652_{\pm 0.007}$	$\mathbf{0.352_{\pm 0.031}}$

Table 3: Plant phenotyping on the agriculture data set

of users. To validate the effectiveness of our proposed approach, we consider two scenarios: (1) Overlapping users: following (Hu, Zhang, and Yang 2018), source and target domains have the same group of users; (2) Disjoint users: the users of source and target domains are different.

Baselines For cross-network node classification, we use the following baselines: (1) SourceOnly: GCN (Kipf and Welling 2017), SGC (Wu et al. 2019), GCNII (Chen et al. 2020); (2) Feature-only adaptation: DAN (Long et al. 2015), DANN (Ganin et al. 2016), MDD (Zhang et al. 2019a); (3) Cross-network adaptation: AdaGCN (Dai et al. 2022), UDAGCN (Wu et al. 2020), EGI (Zhu et al. 2021), and our proposed **GRADE-N** algorithm.

For cross-domain recommendation, we use the following baselines: (1) Single-domain recommendation: BPR (Rendle et al. 2009), NeuMF (He et al. 2017); (2) Cross-domain recommendation: CoNet (Hu, Zhang, and Yang 2018), PPGN (Zhao, Li, and Fu 2019), CGN (Zhang et al. 2020), EGI (Zhu et al. 2021), and our **GRADE-R** algorithm.

Model Configuration We adopt two hidden layers in the base GCN model when implementing the **GRADE**[5] algorithms. We set $\lambda = 0.02$ for cross-network node classification and $\lambda = 0.1$ for the cross-domain recommendation. We train the model using Adam optimizer with a learning rate of 0.01. The hidden layer size of neural units is set as 8.

6.2 Cross-Network Node Classification

Table 1 and Table 2 provide the cross-network node classification results of **GRADE-N**. Here we report the classification accuracy, i.e., mean and standard deviation over 5 runs. We observe that (1) the cross-network node classification algorithms outperform the vanilla graph neural networks and the feature-only adaptation methods, and (2) in most cases, the proposed **GRADE-N** algorithm improves the node classification performance (up to 13%) over baselines.

In addition, we investigate the effectiveness of **GRADE-N** on the regression task. In this case, we use mean square error (MSE) as the loss function of Eq. (4). Table 3 provides the results of **GRADE-N** on the agriculture data set. Here we use two regression evaluation metrics: mean absolute error (MAE) and R^2. Since MDD (Zhang et al. 2019a) focuses only on the classification setting, we use another state-of-the-art adaptation regression baseline RSD (Chen et al. 2021) in our experiments. It is observed that our proposed **GRADE-N** outperforms the state-of-the-art baselines for both MAE (lower is better) and R^2 (higher is better).

6.3 Cross-Domain Recommendation

Results on overlapping users Table 4 provides the cross-domain recommendation results on the Amazon data set. Here we use three recommendation metrics to evaluate our algorithms: hit ratio (HR@k), mean reciprocal rank (MRR@k), and normalized discounted cumulative gain (NDCG@k) where k is 10. Following (Hu, Zhang, and Yang 2018; Zhang et al. 2020), we only consider the users shared by both source and target domains. We have the following observations. (1) Single-domain recommendation methods study the user preference in the target domain from limited observed user-item interactions. They have inferior performance due to network sparsity. (2) Cross-domain rec-

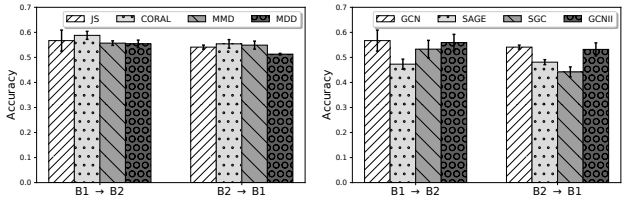
[5] https://github.com/jwu4sml/GRADE

Methods	CD → Music			Music → CD			Book → Movie			Movie → Book		
	HR@10	MRR@10	NDCG@10	HR@10	MRR@10	NDCG@10	HR@10	MRR@10	NDCG@10	HR@10	MRR@10	NDCG@10
BPRMF	$0.182_{\pm 0.003}$	$0.061_{\pm 0.003}$	$0.089_{\pm 0.003}$	$0.259_{\pm 0.008}$	$0.097_{\pm 0.006}$	$0.134_{\pm 0.007}$	$0.198_{\pm 0.003}$	$0.070_{\pm 0.002}$	$0.099_{\pm 0.002}$	$0.128_{\pm 0.007}$	$0.040_{\pm 0.003}$	$0.060_{\pm 0.004}$
NeuMF	$0.286_{\pm 0.012}$	$0.104_{\pm 0.007}$	$0.145_{\pm 0.008}$	$0.328_{\pm 0.014}$	$0.109_{\pm 0.015}$	$0.160_{\pm 0.013}$	$0.294_{\pm 0.020}$	$0.102_{\pm 0.006}$	$0.146_{\pm 0.009}$	$0.142_{\pm 0.009}$	$0.045_{\pm 0.006}$	$0.066_{\pm 0.007}$
CoNet	$0.405_{\pm 0.018}$	$0.161_{\pm 0.002}$	$0.214_{\pm 0.006}$	$0.333_{\pm 0.010}$	$0.119_{\pm 0.018}$	$0.168_{\pm 0.015}$	$0.319_{\pm 0.023}$	$0.116_{\pm 0.020}$	$0.162_{\pm 0.020}$	$0.142_{\pm 0.025}$	$0.041_{\pm 0.011}$	$0.064_{\pm 0.014}$
CGN	$0.357_{\pm 0.018}$	$0.120_{\pm 0.015}$	$0.175_{\pm 0.015}$	$0.476_{\pm 0.042}$	$0.192_{\pm 0.020}$	$0.255_{\pm 0.026}$	$0.359_{\pm 0.032}$	$0.136_{\pm 0.018}$	$0.187_{\pm 0.021}$	$0.205_{\pm 0.014}$	$0.071_{\pm 0.004}$	$0.102_{\pm 0.006}$
PPGN	$0.419_{\pm 0.016}$	$0.179_{\pm 0.008}$	$0.231_{\pm 0.009}$	$0.564_{\pm 0.044}$	$0.278_{\pm 0.032}$	$0.336_{\pm 0.035}$	$0.489_{\pm 0.011}$	$0.239_{\pm 0.007}$	$0.291_{\pm 0.007}$	$0.296_{\pm 0.007}$	$0.154_{\pm 0.010}$	$0.184_{\pm 0.009}$
EGI	$0.446_{\pm 0.011}$	$0.196_{\pm 0.003}$	$0.250_{\pm 0.006}$	$0.599_{\pm 0.007}$	$0.265_{\pm 0.015}$	$0.338_{\pm 0.009}$	$0.461_{\pm 0.021}$	$0.224_{\pm 0.017}$	$0.274_{\pm 0.014}$	$0.274_{\pm 0.024}$	$0.147_{\pm 0.008}$	$0.174_{\pm 0.011}$
GRADE-R	$\mathbf{0.450}_{\pm \mathbf{0.006}}$	$\mathbf{0.197}_{\pm \mathbf{0.002}}$	$\mathbf{0.251}_{\pm \mathbf{0.003}}$	$\mathbf{0.600}_{\pm \mathbf{0.011}}$	$\mathbf{0.313}_{\pm \mathbf{0.007}}$	$\mathbf{0.373}_{\pm \mathbf{0.008}}$	$\mathbf{0.505}_{\pm \mathbf{0.022}}$	$\mathbf{0.249}_{\pm \mathbf{0.004}}$	$\mathbf{0.302}_{\pm \mathbf{0.008}}$	$\mathbf{0.318}_{\pm \mathbf{0.006}}$	$\mathbf{0.170}_{\pm \mathbf{0.009}}$	$\mathbf{0.202}_{\pm \mathbf{0.006}}$

Table 4: Cross-domain recommendation on Amazon data set with overlapping users

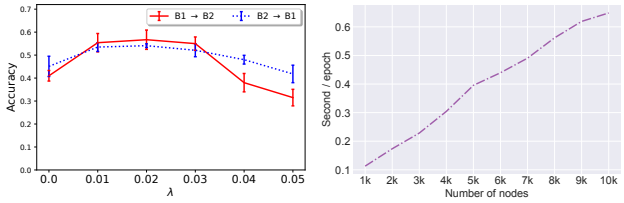
Methods	CD → Music			Music → CD			Book → Movie			Movie → Book		
	HR@10	MRR@10	NDCG@10	HR@10	MRR@10	NDCG@10	HR@10	MRR@10	NDCG@10	HR@10	MRR@10	NDCG@10
BPRMF	$0.167_{\pm 0.006}$	$0.058_{\pm 0.002}$	$0.083_{\pm 0.003}$	$0.122_{\pm 0.001}$	$0.039_{\pm 0.001}$	$0.058_{\pm 0.001}$	$0.165_{\pm 0.004}$	$0.056_{\pm 0.002}$	$0.081_{\pm 0.002}$	$0.116_{\pm 0.001}$	$0.037_{\pm 0.001}$	$0.055_{\pm 0.001}$
NeuMF	$0.283_{\pm 0.020}$	$0.101_{\pm 0.012}$	$0.143_{\pm 0.014}$	$0.133_{\pm 0.006}$	$0.042_{\pm 0.002}$	$0.063_{\pm 0.003}$	$0.233_{\pm 0.006}$	$0.083_{\pm 0.002}$	$0.117_{\pm 0.002}$	$0.132_{\pm 0.006}$	$0.042_{\pm 0.001}$	$0.063_{\pm 0.002}$
NeuMF (S+T)	$0.312_{\pm 0.012}$	$0.123_{\pm 0.010}$	$0.165_{\pm 0.009}$	$0.163_{\pm 0.016}$	$0.056_{\pm 0.009}$	$0.081_{\pm 0.010}$	$0.271_{\pm 0.015}$	$0.110_{\pm 0.012}$	$0.146_{\pm 0.013}$	$0.151_{\pm 0.013}$	$0.054_{\pm 0.010}$	$0.076_{\pm 0.010}$
EGI	$0.415_{\pm 0.008}$	$0.169_{\pm 0.028}$	$0.224_{\pm 0.023}$	$\mathbf{0.231}_{\pm \mathbf{0.035}}$	$0.099_{\pm 0.011}$	$0.128_{\pm 0.015}$	$0.252_{\pm 0.002}$	$0.145_{\pm 0.004}$	$0.167_{\pm 0.004}$	$0.226_{\pm 0.025}$	$0.127_{\pm 0.019}$	$0.148_{\pm 0.020}$
GRADE-R	$\mathbf{0.423}_{\pm \mathbf{0.001}}$	$\mathbf{0.185}_{\pm \mathbf{0.001}}$	$\mathbf{0.238}_{\pm \mathbf{0.000}}$	$0.222_{\pm 0.027}$	$\mathbf{0.110}_{\pm \mathbf{0.004}}$	$\mathbf{0.135}_{\pm \mathbf{0.008}}$	$\mathbf{0.324}_{\pm \mathbf{0.059}}$	$\mathbf{0.187}_{\pm \mathbf{0.022}}$	$\mathbf{0.216}_{\pm \mathbf{0.029}}$	$\mathbf{0.248}_{\pm \mathbf{0.008}}$	$\mathbf{0.127}_{\pm \mathbf{0.025}}$	$\mathbf{0.155}_{\pm \mathbf{0.019}}$

Table 5: Cross-domain recommendation on Amazon data set with disjoint users



(a) Base discrepancy (b) Base GNN

Figure 3: Performance of **GRADE-N** with different base discrepancies and base GNNs on social network



(a) Hyper-parameter sensitivity (b) Efficiency analysis

Figure 4: Model analysis of **GRADE-N**

ommendation methods improve the model performance by leveraging the user preference information from the source domain. (3) The proposed **GRADE-R** outperforms the state-of-the-art cross-domain recommendation baselines.

Results on disjoint users Table 5 provides the results when the users of source and target domains are disjoint. Most existing cross-domain recommendation approaches (Hu, Zhang, and Yang 2018; Zhang et al. 2020) cannot be applied to this scenario, because they require the shared users to explore the common knowledge across domains. Thus, we only consider the single-domain recommendation baselines BPR (Rendle et al. 2009), NeuMF (He et al. 2017), and cross-domain recommendation baseline EGI (Zhu et al. 2021). In addition, we also extend NeuMF to the cross-domain recommendation scenarios by incorporating the recommendation loss in the source domain (denoted as "NeuMF (S+T)"). It is observed that the proposed **GRADE-R** outperforms the baselines by a large margin (up to 18% on HR@10) when the users are disjoint.

6.4 Analysis

Flexibility Figure 3 shows the cross-network node classification performance of **GRADE-N** with different base discrepancies and base graph neural network architectures on social networks. It shows that our **GRADE** framework is flexible to incorporate existing domain discrepancy measures (i.e., JS-divergence (Ganin et al. 2016), CORAL (Sun and Saenko 2016), MMD (Gretton et al. 2012) and MDD (Zhang et al. 2019a)) and message-passing graph neural networks (i.e., GCN (Kipf and Welling 2017), SAGE (Hamilton, Ying, and Leskovec 2017), SGC (Wu et al. 2019) and GCNII (Chen et al. 2020)).

Hyper-parameter Sensitivity We investigate the hyper-parameter sensitivity of **GRADE-N**, i.e., the impact of λ on **GRADE-N**. Figure 4(a) shows the results on the social networks. It shows that the proposed **GRADE-N** can achieve much better performance for $\lambda \in (0.01, 0.03)$. Thus, we use $\lambda = 0.02$ in the experiments.

Computational Efficiency We empirically investigate the computational efficiency of **GRADE** framework. Following (Kipf and Welling 2017), we report the running time (measured in seconds wall-clock time) per epoch on the synthetic source and target graphs. Both graphs have n (i.e., $n_s = n_t = n$) nodes and $2n$ edges. As shown in Figure 4(b), we observe that the wall-clock time of our proposed **GRADE-N** algorithm is linear with respect to the number of nodes within the source and target graphs.

7 Conclusion

In this paper, we study the problem of non-IID transfer learning on graph data. We start by proposing a novel Graph Subtree Discrepancy to measure the distribution shift across graphs. Then we derive the theoretical analysis of non-IID transfer learning on different cross-network mining tasks. It provides insights into designing the practical **GRADE** framework for cross-network transfer learning. Extensive experiments on public data sets demonstrate the effectiveness and efficiency of our **GRADE** framework.

Acknowledgements

This work is supported by National Science Foundation under Award No. IIS-1947203, IIS-2117902, IIS-2137468, and Agriculture and Food Research Initiative (AFRI) grant no. 2020-67021-32799/project accession no.1024178 from the USDA National Institute of Food and Agriculture. The views and conclusions are those of the authors and should not be interpreted as representing the official policies of the funding agencies or the government.

References

Acuna, D.; Zhang, G.; Law, M. T.; and Fidler, S. 2021. f-Domain Adversarial Learning: Theory and Algorithms. In *International Conference on Machine Learning*, 66–75.

Ben-David, S.; Blitzer, J.; Crammer, K.; Kulesza, A.; Pereira, F.; and Vaughan, J. W. 2010. A theory of learning from different domains. *Machine Learning*, 79(1): 151–175.

Chen, M.; Wei, Z.; Huang, Z.; Ding, B.; and Li, Y. 2020. Simple and deep graph convolutional networks. In *International Conference on Machine Learning*, 1725–1735.

Chen, X.; Wang, S.; Wang, J.; and Long, M. 2021. Representation Subspace Distance for Domain Adaptation Regression. In *International Conference on Machine Learning*, 1749–1759. PMLR.

Dai, Q.; Wu, X.-M.; Xiao, J.; Shen, X.; and Wang, D. 2022. Graph Transfer Learning via Adversarial Domain Adaptation with Graph Convolution. *IEEE Transactions on Knowledge and Data Engineering*.

Fang, M.; Yin, J.; Zhu, X.; and Zhang, C. 2015. TrGraph: Cross-network transfer learning via common signature subgraphs. *IEEE Transactions on Knowledge and Data Engineering*, 27(9): 2536–2549.

Ganin, Y.; Ustinova, E.; Ajakan, H.; Germain, P.; Larochelle, H.; Laviolette, F.; Marchand, M.; and Lempitsky, V. 2016. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1): 2096–2030.

Geerts, F.; Mazowiecki, F.; and Perez, G. 2021. Let's agree to degree: Comparing graph convolutional networks in the message-passing framework. In *International Conference on Machine Learning*, 3640–3649.

Gretton, A.; Borgwardt, K. M.; Rasch, M. J.; Schölkopf, B.; and Smola, A. 2012. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1): 723–773.

Hamilton, W.; Ying, Z.; and Leskovec, J. 2017. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30.

Han, X.; Huang, Z.; An, B.; and Bai, J. 2021. Adaptive Transfer Learning on Graph Neural Networks. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 565–574.

He, R.; and McAuley, J. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *proceedings of the 25th international conference on world wide web*, 507–517.

He, X.; Liao, L.; Zhang, H.; Nie, L.; Hu, X.; and Chua, T.-S. 2017. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*, 173–182.

Hu, G.; Zhang, Y.; and Yang, Q. 2018. CoNet: Collaborative cross networks for cross-domain recommendation. In *Proceedings of the 27th ACM international conference on information and knowledge management*, 667–676.

Hu, W.; Liu, B.; Gomes, J.; Zitnik, M.; Liang, P.; Pande, V.; and Leskovec, J. 2020a. Strategies for Pre-training Graph Neural Networks. In *International Conference on Learning Representations*.

Hu, Z.; Dong, Y.; Wang, K.; Chang, K.-W.; and Sun, Y. 2020b. GPT-GNN: Generative pre-training of graph neural networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 1857–1867.

Kipf, T. N.; and Welling, M. 2017. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*.

Levie, R.; Isufi, E.; and Kutyniok, G. 2019. On the transferability of spectral graph filters. In *2019 13th International conference on Sampling Theory and Applications (SampTA)*, 1–5. IEEE.

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*, 1041–1050.

Li, P.; and Tuzhilin, A. 2020. DDTCDR: Deep dual transfer cross domain recommendation. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, 331–339.

Long, M.; Cao, Y.; Wang, J.; and Jordan, M. 2015. Learning transferable features with deep adaptation networks. In *International Conference on Machine Learning*, 97–105.

Mansour, Y.; Mohri, M.; and Rostamizadeh, A. 2009. Domain adaptation: Learning bounds and algorithms. In *22nd Conference on Learning Theory*.

Ni, J.; Tong, H.; Fan, W.; and Zhang, X. 2015. Flexible and robust multi-network clustering. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 835–844.

Pan, S. J.; and Yang, Q. 2009. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10): 1345–1359.

Qiu, J.; Chen, Q.; Dong, Y.; Zhang, J.; Yang, H.; Ding, M.; Wang, K.; and Tang, J. 2020. GCC: Graph contrastive coding for graph neural network pre-training. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 1150–1160.

Rendle, S.; Freudenthaler, C.; Gantner, Z.; and Schmidt-Thieme, L. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, 452–461.

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*.

Ruiz, L.; Chamon, L.; and Ribeiro, A. 2020. Graphon neural networks and the transferability of graph neural networks. *Advances in Neural Information Processing Systems*, 33: 1702–1712.

Shen, X.; Dai, Q.; Chung, F.-l.; Lu, W.; and Choi, K.-S. 2020. Adversarial deep network embedding for cross-network node classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2991–2999.

Shervashidze, N.; Schweitzer, P.; Van Leeuwen, E. J.; Mehlhorn, K.; and Borgwardt, K. M. 2011. Weisfeiler-Lehman graph kernels. *Journal of Machine Learning Research*, 12(9).

Sun, B.; and Saenko, K. 2016. Deep coral: Correlation alignment for deep domain adaptation. In *European conference on computer vision*, 443–450. Springer.

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*, 990–998.

Tang, X.; Yao, H.; Sun, Y.; Wang, Y.; Tang, J.; Aggarwal, C.; Mitra, P.; and Wang, S. 2020. Investigating and mitigating degree-related biases in graph convoltuional networks. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, 1435–1444.

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

Wang, S.; Guan, K.; Wang, Z.; Ainsworth, E. A.; Zheng, T.; Townsend, P. A.; Li, K.; Moller, C.; Wu, G.; and Jiang, C. 2021. Unique contributions of chlorophyll and nitrogen to predict crop photosynthetic capacity from leaf spectroscopy. *Journal of experimental botany*, 72(2): 341–354.

Weisfeiler, B.; and Leman, A. 1968. The reduction of a graph to canonical form and the algebra which appears therein. *NTI, Series*, 2(9): 12–16.

Wu, F.; Souza, A.; Zhang, T.; Fifty, C.; Yu, T.; and Weinberger, K. 2019. Simplifying graph convolutional networks. In *International Conference on Machine Learning*, 6861–6871.

Wu, J.; and He, J. 2019. Scalable manifold-regularized attributed network embedding via maximum mean discrepancy. In *Proceedings of the 28th ACM international conference on information and knowledge management*, 2101–2104.

Wu, J.; and He, J. 2021. Indirect Invisible Poisoning Attacks on Domain Adaptation. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 1852–1862.

Wu, J.; and He, J. 2022a. Domain Adaptation with Dynamic Open-Set Targets. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2039–2049.

Wu, J.; and He, J. 2022b. A Unified Meta-Learning Framework for Dynamic Transfer Learning. In *The Thirty-First International Joint Conference on Artificial Intelligence*.

Wu, J.; He, J.; Wang, S.; Guan, K.; and Ainsworth, E. 2022a. Distribution-Informed Neural Networks for Domain Adaptation Regression. In *Advances in Neural Information Processing Systems*.

Wu, J.; He, J.; and Xu, J. 2019. DEMO-Net: Degree-specific Graph Neural Networks for Node and Graph Classification. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 406–415.

Wu, J.; Tong, H.; Ainsworth, E.; and He, J. 2022b. Adaptive Knowledge Transfer on Evolving Domains. In *2022 IEEE International Conference on Big Data (Big Data)*. IEEE.

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*, 1457–1467.

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.; Kawarabayashi, K.-i.; and Jegelka, S. 2018. Representation learning on graphs with jumping knowledge networks. In *International Conference on Machine Learning*, 5453–5462.

Zhang, Y.; Liu, T.; Long, M.; and Jordan, M. 2019a. Bridging theory and algorithm for domain adaptation. In *International Conference on Machine Learning*, 7404–7413.

Zhang, Y.; Liu, Y.; Han, P.; Miao, C.; Cui, L.; Li, B.; and Tang, H. 2020. Learning Personalized Itemset Mapping for Cross-Domain Recommendation. In *Proceedings of the 29th International Joint Conference on Artificial Intelligence*, 2561–2567.

Zhang, Y.; Song, G.; Du, L.; Yang, S.; and Jin, Y. 2019b. DANE: Domain Adaptive Network Embedding. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, 4362–4368.

Zhao, C.; Li, C.; and Fu, C. 2019. Cross-domain recommendation via preference propagation graphnet. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 2165–2168.

Zhao, H.; Des Combes, R. T.; Zhang, K.; and Gordon, G. 2019. On learning invariant representations for domain adaptation. In *International Conference on Machine Learning*, 7523–7532.

Zhou, Y.; Wang, H.; He, J.; and Wang, H. 2021a. From Intrinsic to Counterfactual: On the Explainability of Contextualized Recommender Systems. *arXiv preprint arXiv:2110.14844*.

Zhou, Y.; Xu, J.; Wu, J.; Taghavi, Z.; Korpeoglu, E.; Achan, K.; and He, J. 2021b. PURE: Positive-unlabeled recommendation with generative adversarial network. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 2409–2419.

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. *Advances in Neural Information Processing Systems*, 34.

A Appendix

A.1 More Discussion on Graph Subtree Discrepancy

Properties of GSD we show that Graph Subtree Discrepancy (GSD) satisfies the following properties

Lemma 1. *If the "relabeling" function f_m ($m = 0, 1, \cdots$) is injective, then for any graphs G^s, G^t, it holds*

(a) $d_{GSD}\left(G^s, G^t\right)$ *exists when M goes to infinity.*
(b) $d_b(G^s_{m-1}, G^t_{m-1}) \leq d_b(G^s_m, G^t_m)$ *for any depth m.*
(c) $d_{GSD}(G^s, G^t) = d_b(G^s, G^t)$*, when G^s, G^t are null graphs containing only isolated nodes.*
(d) $d_{GSD}\left(G^s, G^t\right)$ *is equivalent to the WL subtree kernel (see Definition 3), when* $d_b(G^s_h, G^t_h) = \langle \phi(G^s_m), \phi(G^t_m) \rangle$ *where $\phi(\cdot)$ counts the number of occurrences of subtree patterns.*

Proof. We first show the second **property (b) of GSD**. The function $f_m(v) = f_m\left(f_{m-1}(v); \cup_{u \in N(v)} f_{m-1}(u)\right)$ learns the WL subtree of depth m rooted at $v \in V$ by compressing the representation of v and its neighbors from previous depth $m-1$. In this case, for any node $v^s \in V^s$ and $v^t \in V^t$, there are two scenarios for studying $f_m(v^s)$ and $f_m(v^t)$. (i) If $f_{m-1}(v^s) = f_{m-1}(v^t)$ at depth $m-1$, then $f_m(v^s) = f_m(v^t)$ only when v^s and v^t has the same node degree and their neighbors have the same representation; $f_m(v^s) \neq f_m(v^t)$, otherwise. (ii) If $f_{m-1}(v^s) \neq f_{m-1}(v^t)$ at depth $m-1$, then $f_m(v^s) \neq f_m(v^t)$. As a result, it can be seen that the distribution discrepancy of G^s_m and G^t_m would become larger than that of G^s_{m-1} and G^t_{m-1}, as the WL subtrees become more diverse with the increase of depth m. Therefore, we conclude that $d_b(G^s_m, G^t_m)$ is monotonically increasing with respect to the depth m, i.e., $d_b(G^s_{m-1}, G^t_{m-1}) \leq d_b(G^s_m, G^t_m)$ for any integer m.

For **property (a) of GSD**, we need to show that $d_{GSD}\left(G^s, G^t\right)$ is bounded and monotonically increasing with respect to the depth M.

For **property (c) of GSD**, when one graph only contains the isolated nodes, it can be viewed as a set of independent nodes (variables). Thus, it would degenerate into a standard adaptation setting, where the distribution shift across domains can be measured by $d_b(G^s, G^t)$ on IID nodes. In other words, when there is no edge in the graph, each subtree at any depth m rooted at v is the node v itself. Therefore, it holds that $d_b(G^s_0, G^t_0) = \cdots = d_b(G^s_m, G^t_m)$ for any m. Then we have $d_{GSD}(G^s, G^t) = d_b(G^s_0, G^t_0) = d_b(G^s, G^t)$.

For **property (d) of GSD**, it can be derived using the definition of WL subtree kernel. □

Benefits of Graph Subtree Discrepancy The benefits of Graph Subtree Discrepancy could be summarized as follows. (1) *Interpretability:* It measures the difference of subtree distributions in the source and target graphs. When the subtree representation is countable, it would be equivalent to the standard Weisfeiler-Leman graph subtree kernel by counting the number of occurrences of subtree patterns. (2) *Computational Efficiency:* When using GCN (Kipf and Welling 2017) as the base model for subtree representation learning and multi-layer perceptrons as the hypothesis space $\mathcal{H}$ for learning the subtree representation, the time complexity of GSD is $\mathcal{O}(M(n_s + n_t)d^2 + M(m_s + m_t)d)$ where n_s, n_t denote the number of nodes within the source and target graphs, m_s, m_t denote the number of edges within the source and target graphs, d is the dimensionality of hidden layers, and M is the number of neural layers. (3) *Flexibility:* It enables the estimate of distribution shift across graphs using existing expressive graph neural networks (Kipf and Welling 2017; Hamilton, Ying, and Leskovec 2017; Veličković et al. 2018) and standard domain discrepancy measures (Gretton et al. 2012; Ganin et al. 2016; Zhang et al. 2019a) (see Subsection 6.4 for empirical analysis).

Improved Graph Subtree Discrepancies We provide two improved variants of Graph Subtree Discrepancy by incorporating specific structure and label information.

- `Structure-aware GSD`: As one of the most important properties in a graph, node degree encodes the intrinsic graph structure (Wu, He, and Xu 2019; Tang et al. 2020; Geerts, Mazowiecki, and Perez 2021) despite its simplicity. Following the Weisfeiler-Lehman graph isomorphism test (Weisfeiler and Leman 1968; Shervashidze et al. 2011), two graphs can be easily distinguished from the node degree distribution. Therefore, we can design a degree-specific GSD by taking the node degree into consideration.

$$d^{deg}_{GSD}\left(G^s, G^t\right) = \lim_{M \to \infty} \frac{1}{M+1} \sum_{m=0}^{M} d_b\left(G^s_m \circ D^s, G^t_m \circ D^t\right) \tag{8}$$

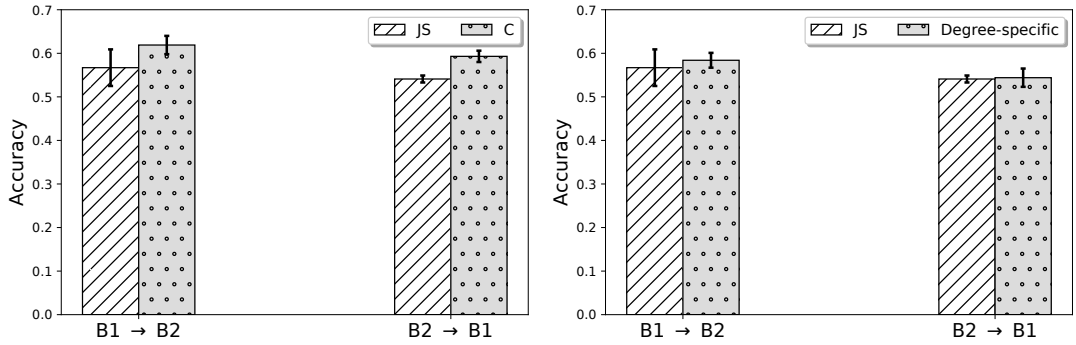
  where $\circ$ denotes the vector concatenation of subtree representation around $v \in V$ and one-hot degree representation of v. This improves the quality of GSD, when the distribution shift of graphs is largely induced by the node attributes. In this case, source and target nodes are more likely to have the identical class label, when they have the same node degree. Thus, the degree-specific distribution discrepancy measure can better characterize the distribution shift across graphs.

- `Label-informed GSD`: In the context of cross-network node classification, the class labels of nodes can help refine the definition of distribution shift across graphs. As demonstrated in previous works (Zhao et al. 2019), minimizing the feature-based marginal distribution discrepancy cannot guarantee the success of the knowledge transfer across domains.

Therefore, inspired by (Wu et al. 2022b), we proposed a label-informed GSD by considering the distribution shifts across graphs induced by both input features and output labels.

$$d_{GSD}^{cls}\left(G^s, G^t\right) = \lim_{M\to\infty} \frac{1}{M+1} \sum_{m=0}^{M} d_b\left(G_m^s \circ Y^s, G_m^t \circ Y^t\right) \tag{9}$$

Here, $\circ$ denotes the vector concatenation of subtree representation around $v \in V$ and the one-hot class label representation of v. In practice, when the class labels of target nodes are unknown, we can use their pseudo-labels to estimate this discrepancy. Similarly, we can derive the generalization error bounds based on the improved GSD, because it holds that $d_{GSD}\left(G^s, G^t\right) \leq d_{GSD}^{deg}\left(G^s, G^t\right)$ and $d_{GSD}\left(G^s, G^t\right) \leq d_{GSD}^{cls}\left(G^s, G^t\right)$.



(a) Label-informed discrepancy (b) Degree-specific discrepancy

Figure 5: Performance of **GRADE-N** with improved discrepancies on social network

Figure 5 provides the results of **GRADE-N** using the improved discrepancy above. Since the target domain is unlabeled, we use the pseudo-labels of target nodes to estimate the label-informed GSD of Eq. (9). It is observed that both label and degree information can help improve the quality of GSD, thus leading to better model performance.

A.2 Illustration of GRADE Framework

Figure 6 shows the proposed **GRADE** framework on cross-network node classification, where Graph Subtree Discrepancy (GSD) is defined over the subtrees involving both the structure and node attributes in the center node's local neighborhood.

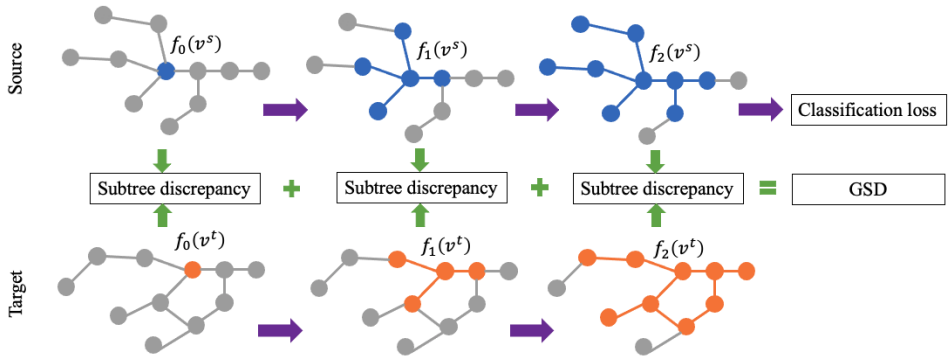


Figure 6: Illustration of **GRADE** on cross-network node classification

A.3 Proof of Theorem 1

Theorem 1 states that assume there are a source graph G^s and a target graph G^t and the base domain discrepancy $d_b(\cdot,\cdot)$ of GSD is instantiated by the discrepancy distance (see Eq. (2)). Given a message-passing GNN with the feature extractor f and the hypothesis $h \in \mathcal{H}$, the node classification error in the target graph can be bounded as follows.

$$\epsilon_t(h \circ f) \leq \epsilon_s(h \circ f) + d_{GSD}\left(G^s, G^t\right) + \lambda^* + R^*$$

where $\lambda^* = \mathbb{E}_{v\in V^t}[\mathcal{L}(h_*^s(f(v)), h_*^t(f(v)))]$ measures the prediction difference of optimal source and target hypotheses on the target nodes, and $R^* = \mathbb{E}_{v\in V^s}[\mathcal{L}(y, h_*^s(f(v)))] + \mathbb{E}_{v\in V^t}[\mathcal{L}(h_*^t(f(v)), y)]$ is the Bayes error on the source and target graphs. y

is the class label of v. In this case, $h_*^s \in \arg\min_{h\in\mathcal{H}} \mathbb{E}_{v\in V^s}[\mathcal{L}(h(f(v)),y)]$ and $h_*^t \in \arg\min_{h\in\mathcal{H}} \mathbb{E}_{v\in V^t}[\mathcal{L}(h(f(v)),y)]$ are optimal source and target hypotheses, respectively.

Proof. Using Lemma 1, it holds that for any $M > 0$

$$\begin{aligned}
&\frac{1}{M+1}\sum_{m=0}^{M} d_b\left(G_m^s, G_m^t\right) - \frac{M-L+1}{M+1} d_b\left(G_L^s, G_L^t\right) \\
&= \frac{1}{M+1}\sum_{m=0}^{L-1} d_b\left(G_m^s, G_m^t\right) + \frac{1}{M+1}\sum_{m=L}^{M}\left(d_b\left(G_m^s, G_m^t\right) - d_b\left(G_L^s, G_L^t\right)\right) \geq 0
\end{aligned}$$

Then it holds that $d_{GSD}\left(G^s, G^t\right) = \lim_{M\to\infty}\frac{1}{M+1}\sum_{m=0}^{M} d_b\left(G_m^s, G_m^t\right) \geq \lim_{M\to\infty}\frac{M-L+1}{M+1} d_b\left(G_L^s, G_L^t\right) = d_b\left(G_L^s, G_L^t\right)$. That means that the following holds.

$$\begin{aligned}
d_{GSD}\left(G^s, G^t\right) &= \lim_{M\to\infty}\frac{1}{M+1}\sum_{m=0}^{M}\sup_{h,h'\in\mathcal{H}}\left|\mathbb{E}_{v\in V^s}\left[\mathcal{L}\left(h\left(f_m(v)\right), h'\left(f_m(v)\right)\right)\right] - \mathbb{E}_{v\in V^t}\left[\mathcal{L}\left(h\left(f_m(v)\right), h'\left(f_m(v)\right)\right)\right]\right| \\
&\geq \sup_{h,h'\in\tilde{\mathcal{H}}}\left|\mathbb{E}_{v\in V^s}\left[\mathcal{L}\left(h\left(f_L(v)\right), h'\left(f_L(v)\right)\right)\right] - \mathbb{E}_{v\in V^t}\left[\mathcal{L}\left(h\left(f_L(v)\right), h'\left(f_L(v)\right)\right)\right]\right| \\
&\geq \mathbb{E}_{v\in V^t}[\mathcal{L}(h(f_L(v)), h_*^s(f_L(v)))] - \mathbb{E}_{v\in V^s}[\mathcal{L}(h(f_L(v)), h_*^s(f_L(v)))] \\
&= \mathbb{E}_{v\in V^t}[\mathcal{L}(h(f(v)), h_*^s(f(v)))] - \mathbb{E}_{v\in V^s}[\mathcal{L}(h(f(v)), h_*^s(f(v)))]
\end{aligned}$$

If the classification error is defined using the output of the last graph convolutional layer, i.e., $\epsilon_t(h\circ f) = \mathbb{E}_{v\in G}[L(h(f_L(v)),y)]$, we have the following results.

$$\begin{aligned}
&\epsilon_t(h\circ f) = \mathbb{E}_{v\in V^t}[\mathcal{L}(h(f(v)),y)] \\
&\leq \mathbb{E}_{v\in V^t}[\mathcal{L}(h(f(v)), h_*^t(f(v)))] + \mathbb{E}_{v\in V^t}[\mathcal{L}(h_*^t(f(v)),y)] \\
&\leq \mathbb{E}_{v\in V^t}[\mathcal{L}(h(f(v)), h_*^s(f(v)))] + \mathbb{E}_{v\in V^t}[\mathcal{L}(h_*^s(f(v)), h_*^t(f(v)))] + \mathbb{E}_{v\in V^t}[\mathcal{L}(h_*^t(f(v)),y)] \\
&\leq \mathbb{E}_{v\in V^s}[\mathcal{L}(h(f(v)), h_*^s(f(v)))] + d_{GSD}\left(G^s, G^t\right) + \mathbb{E}_{v\in V^t}[\mathcal{L}(h_*^s(f(v)), h_*^t(f(v)))] + \mathbb{E}_{v\in V^t}[\mathcal{L}(h_*^t(f(v)),y)] \\
&\leq \epsilon_s(h\circ f) + \mathbb{E}_{v\in V^s}[\mathcal{L}(y, h_*^s(f(v)))] + d_{GSD}\left(G^s, G^t\right) + \mathbb{E}_{v\in V^t}[\mathcal{L}(h_*^s(f(v)), h_*^t(f(v)))] + \mathbb{E}_{v\in V^t}[\mathcal{L}(h_*^t(f(v)),y)]
\end{aligned}$$

which completes the proof. □

A.4 Corollary 1

Corollary 1. *With assumptions in Theorem 1, and let $\mathcal{L}(y,\tilde{y}) = |y-\tilde{y}|$ and the hypothesis class $\mathcal{H}$ is given by the multi-layer perceptrons, if the classification error is defined using the jumping knowledge, i.e., $\epsilon_t(h\circ f) = \mathbb{E}_{v\in G}[\mathcal{L}(h(f(v)),y)]$ where $f(v) = [f_0(v),\cdots,f_L(v)]$, the node classification error in the target graph can also be bounded as follows.*

$$\epsilon_t(h\circ f) \leq \epsilon_s(h\circ f) + 2d_{GSD}\left(G^s, G^t\right) + \lambda^* + R^*$$

where $\lambda^* = \mathbb{E}_{v\in V^t}[\mathcal{L}(h_*^s(f(v)), h_*^t(f(v)))]$, and $R^* = \mathbb{E}_{v\in V^s}[\mathcal{L}(y, h_*^s(f(v)))] + \mathbb{E}_{v\in V^t}[\mathcal{L}(h_*^t(f(v)),y)]$.

Proof. Using Lemma 1, it holds that for any $M > 0$

$$\begin{aligned}
&\frac{1}{M+1}\sum_{m=0}^{M} d_b\left(G_m^s, G_m^t\right) - \sum_{l=0}^{L}\frac{M-l+1}{(L+1)(M+1)} d_b\left(G_l^s, G_l^t\right) \\
&= \frac{1}{L+1}\sum_{l=0}^{L}\left(\frac{1}{M+1}\sum_{m=0}^{M} d_b\left(G_m^s, G_m^t\right) - \frac{M-l+1}{M+1} d_b\left(G_l^s, G_l^t\right)\right) \geq 0
\end{aligned}$$

Then it holds $d_{GSD}\left(G^s, G^t\right) = \lim_{M\to\infty}\frac{1}{M+1}\sum_{m=0}^{M} d_b\left(G_m^s, G_m^t\right) \geq \lim_{M\to\infty}\sum_{l=0}^{L}\frac{M-l+1}{(L+1)(M+1)} d_b\left(G_l^s, G_l^t\right) = \frac{1}{L+1}\sum_{l=0}^{L} d_b\left(G_l^s, G_l^t\right)$. If the classification error is defined using the jumping knowledge, i.e., $\epsilon_t(h\circ f) = \mathbb{E}_{v\in V}[\mathcal{L}(h(f(v)),y)]$ where $f(v) = [f_0(v),\cdots,f_M(v)]$, the hypothesis class $\mathcal{H}$ used in the classification and GSD would be different. So we denote the hypothesis class of GSD as $\tilde{\mathcal{H}}$ in this case. If the hypothesis class $\mathcal{H}$ is instantiated by the multi-layer perceptrons (MLPs), then there exists $h_0, h_1, \cdots, h_L \in \tilde{\mathcal{H}}$ ($\tilde{\mathcal{H}}$ can also be multi-layer perceptrons) such that for any hypothesis $h \in \mathcal{H}$, it could be represented as $h(f(v)) = h([f_0(v),\cdots,f_L(v)]) = \frac{1}{L+1}\sum_{l=0}^{L} h_l(f_l(v))$. That can be derived by decomposing the first-layer parameters of h as follows.

$$W_1[f_0(v),\cdots,f_M(v)] + b_1 = (W_{10}f_0(v) + b_{10}) + \cdots + (W_{1M}f_M(v) + b_{1M})$$

where the weight W_1 can be divided into M parts according to its columns, and $b_1 = b_{10} + \cdots + b_{1M}$, and the constant scaling factor $\frac{1}{L+1}$ can be added, as $\frac{1}{L+1} h_l(\cdot) \in \tilde{\mathcal{H}}$. Therefore, we have

$$
\begin{aligned}
& 2d_{GSD}\left(G^s, G^t\right) \\
&= \lim_{M\to\infty} \frac{2}{M+1} \sum_{m=0}^{M} \sup_{\tilde{h},\tilde{h'}\in\tilde{\mathcal{H}}} \left| \mathbb{E}_{v\in V^s}\left[\mathcal{L}\left(\tilde{h}\left(f_m(v)\right), \tilde{h'}\left(f_m(v)\right)\right)\right] - \mathbb{E}_{v\in V^t}\left[\mathcal{L}\left(\tilde{h}\left(f_m(v)\right), \tilde{h'}\left(f_m(v)\right)\right)\right] \right| \\
&\geq \frac{2}{L+1} \sum_{l=0}^{L} \sup_{\tilde{h},\tilde{h'}\in\tilde{\mathcal{H}}} \left| \mathbb{E}_{v\in V^s}\left[\mathcal{L}\left(\tilde{h}\left(f_l(v)\right), \tilde{h'}\left(f_l(v)\right)\right)\right] - \mathbb{E}_{v\in V^t}\left[\mathcal{L}\left(\tilde{h}\left(f_l(v)\right), \tilde{h'}\left(f_l(v)\right)\right)\right] \right| \\
&= \frac{2}{L+1} \sum_{l=0}^{L} \sup_{\tilde{h},\tilde{h'}\in\tilde{\mathcal{H}}} \left| \int_V \left(p^s(v) - p^t(v)\right) \mathcal{L}\left(\tilde{h}\left(f_l(v)\right), \tilde{h'}\left(f_l(v)\right)\right) dv \right| \\
&\geq \frac{1}{L+1} \sum_{l=0}^{L} \sup_{\tilde{h},\tilde{h'}\in\tilde{\mathcal{H}}} \int_V \left|p^s(v) - p^t(v)\right| \left|\tilde{h}(f_l(v)) - \tilde{h'}(f_l(v))\right| dv \\
&\geq \frac{1}{L+1} \sum_{l=0}^{L} \int_V \left|p^s(v) - p^t(v)\right| \left|h_l(f_l(v)) - h^s_{*l}(f_l(v))\right| dv \\
&\geq \int_V \left|p^s(v) - p^t(v)\right| \left|h(f(v)) - h^s_*(f(v))\right| dv \\
&\geq \mathbb{E}_{v\in V^t}[\mathcal{L}(h(f(v)), h^s_*(f(v)))] - \mathbb{E}_{v\in V^s}[\mathcal{L}(h(f(v)), h^s_*(f(v)))]
\end{aligned}
$$

which completes the proof. □

A.5 Proof of Theorem 2

Theorem 2 states that with assumptions in Theorem 1, and let $\mathcal{L}(y, \tilde{y}) = |y - \tilde{y}|$ and the hypothesis class $\mathcal{H}$ is given by the multi-layer perceptrons, if the loss of the link prediction is defined as $\epsilon^{link}(h \circ f) = \mathbb{E}_{u,v\in V\times V}[\mathcal{L}(h([f_L(u)||f_L(v)]), y)]$, then the link prediction error in the target graph can be bounded as follows.

$$\epsilon^{link}_t(h) \leq \epsilon^{link}_t(h) + d_{GSD}\left(G_s, G_t\right) + \lambda^*_{link} + R^*_{link}$$

where $\lambda^*_{link} = \mathbb{E}_{(u,v)\in V^t\times V^t}[\mathcal{L}(h^s_*([f(u)||f(v)]), h^t_*([f(u)||f(v)]))]$ measures the difference of optimal source and target hypotheses on the target graph, and $R^*_{link} = \mathbb{E}_{(u,v)\in V^s\times V^s}[\mathcal{L}(y, h^s_*([f(u)||f(v)]))] + \mathbb{E}_{(u,v)\in V^t\times V^t}[\mathcal{L}(h^t_*([f(u)||f(v)]), y)]$ is the Bayes error. In this case, $h^s_* \in \arg\min_{h\in\mathcal{H}} \mathbb{E}_{(u,v)\in V^s\times V^s}[\mathcal{L}(h([f(u)||f(v)]), y)]$, and $h^t_* \in \arg\min_{h\in\mathcal{H}} \mathbb{E}_{(u,v)\in V^t\times V^t}[\mathcal{L}(h([f(u)||f(v)]), y)]$ are optimal source and target hypothesises, respectively.

Proof. It can be shown using a similar method in Corollary 1. Here the output feature function $f(\cdot)$ would stack the representations from a pair of nodes, i.e., $f(u, v) = [f_L(u)||f_L(v)]$. □

A.6 Data Description

For cross-network node classification, we use the following benchmark data sets:

- Airport network (Ribeiro, Saverese, and Figueiredo 2017): It contains three airport networks from Brazil, USA and Europe. Each node corresponds to an airport and the edge indicates the existence of commercial flights between two airports. The class labels of nodes are assigned based on the level of activity measured by flights or people that passed the airports. Following (Zhu et al. 2021), we use node degree one-hot encoding as the node feature.
- Citation network (Wu et al. 2020): It has two networks ACMv9 (A) and DBLPv8 (D) from ArnetMine (Tang et al. 2008). Each node is a paper and each edge indicates the citation between two papers. Each paper is associated with a 7537-dimensional feature vector extracted from paper content. Its class label indicates the research topics.
- Social network (Shen et al. 2020): Blog1 (B1) and Blog2 (B2) are two disjoint social networks extracted from BlogCatalog (Li et al. 2015). Each node is a blogger and each edge indicates the friendship between two bloggers. The blogger is associated with an 8189-dimensional feature vector extracted from the blogger's self-description, and its class label indicates the joining group.
- Agriculture data (Wang et al. 2021): Plant Phenotyping predicts diverse traits (e.g., Nitrogen) of plants related to the plants' growth using leaf hyperspectral reflectance. Here we use the agriculture data from two domains: Maize (M) and Maize_UNL (MU) (i.e., maize data are measured from different locations). In our case, the task is to predict the Nitrogen content of maize using the leaf hyperspectral reflectance. Specifically, each example can be represented as a 1901-dimensional feature vector,

Data		#nodes	#edges	#classes
Airport	USA	1,190	13,599	4
	Brazil	131	1,038	4
	Europe	399	5,995	4
Citation	ACMv9	7,410	11,135	6
	DBLPv8	5,578	7,341	6
Social	Blog1	2,300	33,471	6
	Blog2	2,896	53,836	6
Agriculture	Maize	349	1,745	-
	Maize_UNL	1,210	6,050	-
Data		#users	#items	#ratings
CD vs. Music (overlapping)	CD	5,000	55,312	353,942
	Music	5,000	90,248	155,192
Book vs. Movie (overlapping)	Book	5,000	90,248	155,192
	Movie	5,000	28,796	100,302
CD vs. Music (disjoint)	CD	5,000	27,838	51,190
	Music	5,000	3,568	58,408
Book vs. Movie (disjoint)	Book	5,000	51,968	72,804
	Movie	5,000	21,070	52,376

Table 6: Data statistics

which characterizes the spectral wavelengths 500-2400 nm. Then we can adopt k-NN to build the graph for each domain. We use $k = 5$ in our experiments.

For cross-domain recommendation, we evaluate the models on the Amazon data set (He and McAuley 2016). We adopt two pairs of real-world cross-domain data sets from Amazon-5cores, including CD (i.e., CDs and Vinyl) and Music (i.e., Digital Music), Book (i.e., Books) and Movie (i.e., Movies and TV). Note that most of the existing cross-domain recommendation algorithms (Hu, Zhang, and Yang 2018; Zhang et al. 2020) assume that source and target domains have the same group of users. To validate the effectiveness of our proposed approach, we would like to consider the following two scenarios: (1) Overlapping users: following (Hu, Zhang, and Yang 2018), source and target domains have the same group of users; (2) Disjoint users: the users of source and target domains are not overlapping.

Table 6 summarizes all the data sets used in the experiments. All the experiments are performed on a Windows machine with four 3.80GHz Intel Cores, 64GB RAM, and one NVIDIA Quadro RTX 5000 GPU.