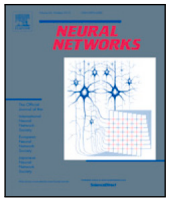
Neural Networks 177 (2024) 106396

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet



Full Length Article

Structure enhanced prototypical alignment for unsupervised cross-domain node classification

Meihan Liu [a,b], Zhen Zhang [d], Ning Ma [a,b], Ming Gu [a,b], Haishuai Wang [a,b], Sheng Zhou [b,c,*], Jiajun Bu [a,b]

[a] College of Computer Science, Zhejiang University, Hangzhou, 310027, China
[b] Zhejiang Provincial Key Laboratory of Service Robot, Zhejiang University, Hangzhou, 310027, China
[c] School of Software Technology, Zhejiang University, Ningbo, 315048, China
[d] School of Computing, National University of Singapore, 117417, Singapore

ARTICLE INFO

Keywords:
Graph domain adaptation
Graph neural networks
Node classification
Graph representation learning
Transfer learning

ABSTRACT

Graph Neural Networks (GNNs) have demonstrated remarkable success in graph node classification task. However, their performance heavily relies on the availability of high-quality labeled data, which can be time-consuming and labor-intensive to acquire for graph-structured data. Therefore, the task of transferring knowledge from a label-rich graph (source domain) to a completely unlabeled graph (target domain) becomes crucial. In this paper, we propose a novel unsupervised graph domain adaptation framework called Structure Enhanced Prototypical Alignment (SEPA), which aims to learn domain-invariant representations on non-IID (non-independent and identically distributed) data. Specifically, SEPA captures class-wise semantics by constructing a prototype-based graph and introduces an explicit domain discrepancy metric to align the source and target domains. The proposed SEPA framework is optimized in an end-to-end manner, which could be incorporated into various GNN architectures. Experimental results on several real-world datasets demonstrate that our proposed framework outperforms recent state-of-the-art baselines with different gains.

1. Introduction

Real-world complex systems are often represented as networks, such as citation networks (Sen et al., 2008; Tang et al., 2008), social networks (Rozemberczki et al., 2021), and so on. Mining valuable information from graphs has gained substantial research attention from both academic and industrial communities (Bhagat et al., 2011; Rao et al., 2022). For example, in citation networks, papers are interconnected with each other through citations, which can be categorized into different topics based on their contents and relationships. In social media platforms, users can be grouped for personalized services according to their interaction characteristics and behaviors. Recently, graph neural networks (GNNs) (Hamilton et al., 2017; Kipf & Welling, 2017; Velickovic et al., 2018) have emerged as a powerful deep learning technique for modeling graph-structured data. By incorporating neighborhood information, GNNs can effectively capture the meaningful representations and achieve impressive performance in node classification task.

Despite the impressive progress of graph neural networks, their success heavily relies on the availability of high-quality labeled data. However, in real-world applications, graphs coming from different periods or sources are ubiquitous and it could be labor-intensive and time-consuming to annotate sufficient labels. For example, in the context of citation networks, graphs acquired from different publishers and times demonstrate distinct node attributes and graph structure properties. As a result, the performance of a GNN model trained on one source, such as papers extracted from ACM between 2000 and 2004, may not be satisfactory when applied to another source, such as papers obtained from DBLP between 2004 and 2008. One natural question is why not transfer knowledge from a label-rich graph to a label-scare graph. Unfortunately, most existing GNN models fail to generalize to new domains due to the domain shift problem (Quionero-Candela et al., 2009; Torralba & Efros, 2011), especially when the distributions of the source domain and the target domain are highly different.

Recently, domain adaptation (Wang & Deng, 2018) has become an appealing solution to tackle the distribution shift problem, which aims to transfer knowledge from source domain to target domain. It motivates a line of researches (Long et al., 2015; Wang & Zheng, 2015) under the IID assumption, where the source and target data are independent and identically distributed in each domain. Although these models (Gretton et al., 2012; Kang et al., 2019; Sun et al., 2016; Tzeng et al., 2017) have obtained satisfied performance in the fields of

* Corresponding author at: Zhejiang Provincial Key Laboratory of Service Robot, Zhejiang University, Hangzhou, 310027, China.
E-mail addresses: lmh_zju@zju.edu.cn (M. Liu), zhousheng_zju@zju.edu.cn (S. Zhou).

https://doi.org/10.1016/j.neunet.2024.106396
Received 17 October 2023; Received in revised form 9 May 2024; Accepted 15 May 2024
Available online 18 May 2024

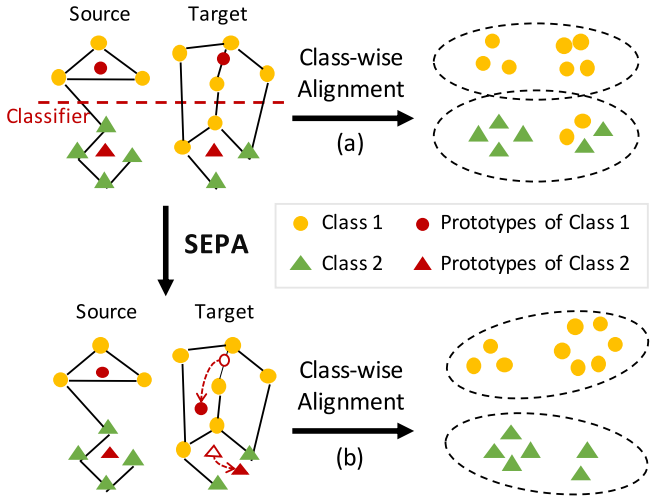


Fig. 1. An example illustrates the class-wise alignment by minimizing distance between prototypes from source and target graph. The red dashed line represents the decision boundary of the classifier. The classifier is supervised by source domain and cannot generalize well to the target graph. (a) The performance of class-wise alignment is limited by the unrepresentative prototypes. (b) SEPA finds more representative prototypes for target domain and conducts class-wise alignment effectively.

computer vision and natural language processing, they are not suitable to other tasks with non-IID data, e.g., node classification in graph-structured data. When it comes to graph-structured data, the situation becomes more complicated. First, each node in the graph could have complex interactions with its neighborhood. Second, the distribution shift may come from either graph structures or node attributes, which are highly related. Therefore, it is necessary to investigate how to transfer knowledge across graph-structured non-IID data.

Several recent efforts have been devoted to unsupervised graph domain adaptation (Dai et al., 2022; Shen et al., 2020, 2021; Wu et al., 2023, 2020; You et al., 2023; Zhang et al., 2021). Existing research can be categorized into two main lines: (1) Learning domain-invariant features by minimizing a divergence that quantifies the distance between distributions (Shen et al., 2021; Wu et al., 2023). Commonly employed divergence measures include maximum mean discrepancy (Shen et al., 2021), graph subtree discrepancy (Wu et al., 2023). These metrics define the difference between two distributions with their mean embeddings. (2) Utilizing adversarial learning techniques (Dai et al., 2022; Shen et al., 2020; Wu et al., 2020; You et al., 2023; Zhang et al., 2021) for domain discrimination by incorporating a domain discriminator to induce domain-level confusion. However, existing methods that rely on pre-defined distance metrics or adversarial learning face a common limitation: they primarily focus on minimizing distribution discrepancy at the domain level while often neglecting the semantic correlations between classes. Consequently, there is a risk of erroneously aligning samples belonging to different classes, resulting in misalignment between target domain samples and source domain samples.

In this paper, we propose a novel framework named **S**tructure **E**nhanced **P**rototypical **A**lignment (SEPA) for unsupervised cross-domain node classification task. SEPA first estimates prototypes to capture semantics information, and then performs domain alignment by minimizing a divergence in measuring the distance between the prototypes. The performance of domain alignment is determined by the representation ability of the prototypes. A simple method for estimating prototypes is to average the representations of the same pseudo labels obtained from the classifiers (Pan et al., 2019; Yue et al., 2021). Nevertheless, utilizing pseudo-labeling under domain shift creates a dilemma. On the one hand, the presence of unreliable pseudo-labels deteriorates the model's performance. On the other hand, only selecting nodes with high confidence provides limited information. In this way, the prototypes induced from pseudo labels are usually not representative enough, thereby limiting the performance of domain alignment.

As shown in Fig. 1, since the classifier is under the supervision of source domain, some target nodes might be wrongly classified based on the decision boundary due to the existence of domain shift. Thus, the features learnt in the embedding space are not representative enough, making the prototypes not suitable to represent the classes. In this scenario, the prototypes we get are biased against the expected prototypes and directly conducting class-wise alignment can alter the decision boundary, further consolidating the dominant position of the source domain over the classifier.

To overcome the above problem, our proposed SEPA avoids using pseudo-labeling methods and instead attempts to calibrate the prototypes to better reflect the commonality within the category of target graph. First, we estimate the transition matrix for each target node to reveal the class possibility that the instance might belong to. The calculation of transition matrix is based on the prediction of the classifier, which provides underlying semantic information for modeling the target graph distribution. Second, we assign a soft prototype to each target node based on the transfer matrix and a prototype-based graph is constructed by the assignment. By constructing the prototype-based graph, SEPA is able to directly map conditionally independent prototypes to the nodes, enabling an effective capture and representation of the characteristics of the target domain. Propagating on the prototype-based graph can intuitively represent the impact of target structure on prototypes. Lastly, we estimate prototypes for target graph and propose an explicit alignment loss function, which enables us to address the domain discrepancy at a class-wise level.

The proposed SEPA framework is optimized in an end-to-end manner, which could be easily incorporated into various GNN architectures. Empirical results on two types of real-world datasets demonstrate that SEPA outperforms recent state-of-the-art baselines on unsupervised cross-domain node classification task. To summarize, our main contributions are as follows:

- We propose a novel framework for unsupervised cross-domain node classification task, which takes high-level semantic information into consideration and performs prototypical alignment for graph data.
- To facilitate the prototype alignment, we propose prototype-based graph and propagate on it to estimate prototypes for target graph, which takes target personalized structure into consideration and avoids the classifier dominated by the source graph.
- We perform extensive experiments to validate its effectiveness and analyze the properties of SEPA via thorough comparisons with state-of-the-art methods on two types of real-world graph datasets.

2. Related work

2.1. Graph representation learning

Graph representation learning aims to transform the complex network structure and associated attributes into low-dimensional vector representations, enabling efficient analysis for downstream tasks such as node classification, link prediction, and graph clustering (Cui et al., 2019; Ji et al., 2022). Graph representation learning methods can be divided into two groups: transductive methods and inductive methods. Transductive embedding methods require that all nodes in the graph are present during training (Grover & Leskovec, 2016; Perozzi et al., 2014; Shi et al., 2018; Xu et al., 2019). DeepWalk (Perozzi et al., 2014) and node2vec (Grover & Leskovec, 2016) employ random walk sampling to generate contextual neighborhood information, which is then fed into a skip-gram model to generate node representations. However, these methods mainly focus on modeling network structure information and cannot infer embeddings for new nodes.

To address these limitations, inductive embedding methods exploit network structures, node attributes, and the available node labels from

known graphs to learn more informative node representations (Hamilton et al., 2017; Huang et al., 2017; Kipf & Welling, 2017; Velickovic et al., 2018; Zhang et al., 2018). The most popular embedding methods are graph neural networks (GNNs). GCN (Kipf & Welling, 2017) employs message passing to update and aggregate node representations from its neighborhood, which achieves great success in various graph related tasks. GraphSAGE (Hamilton et al., 2017) extends GCN by introducing various aggregation strategies. GAT (Velickovic et al., 2018) leverage multi-head-based self-attention mechanisms to selectively attend and weigh the importance of neighbor nodes. Despite their powerful capability to capture complex node relationships, they often experience significant performance degeneration when directly applying them to graphs with different distributions.

2.2. Domain adaptation

Domain adaptation transfers knowledge from labeled source domain to improve the performance of unlabeled target domain in the presence of distribution shift (Pan & Yang, 2010; Wang & Deng, 2018). Many approaches based on deep learning are proposed for domain adaptation in the field of computer vision (Ganin & Lempitsky, 2015; Tzeng et al., 2017), which can be roughly categorized into two groups: adversarial domain discrimination and explicit distribution alignment. Adversarial domain discrimination typically involves a domain classifier that determines whether a feature representation originated from a source or target domain. The underlying assumption is that if the domain classifier is unable to accurately classify the source of the feature representation, features follow the same underlying distribution across the domains (Ganin et al., 2016; Long et al., 2017; Pei et al., 2018; Shen et al., 2018). DANN (Ganin et al., 2016) learns domain-invariant features by training feature extractor to make the domain classifier perform poorly. WDGRL (Shen et al., 2018) is a variant of DANN that learns an approximate Wasserstein distance to replace the domain classifier.

Instead of performing domain discrimination, explicit distribution alignment methods aim to reduce the statistic discrepancy across domains to obtain domain-invariant features (Ganin & Lempitsky, 2015; Long et al., 2015; Pei et al., 2018; Tzeng et al., 2017). TPN (Pan et al., 2019) matches each target example to the nearest source prototype and minimizes the distances between the prototypes. CAN (Kang et al., 2019) employs collaborative learning to generate domain-invariant features and selects pseudo-labeled target samples for re-training the entire framework. SRDC (Tang et al., 2020) enhances target discrimination through a deep clustering framework and minimizes the KL divergence to reduce domain discrepancy. UDA (Yue et al., 2021) encodes and aligns semantic features in the shared embedding space across domains, then a threshold is utilized to determine the robustness of the prototype. Although these models have obtained satisfied performance in the fields of computer vision, they are not applicable to other tasks with non-IID data for the following two reasons: (1) the node's representation is highly independent on their neighboring nodes (2) the distribution shifts between the two graph domains are more complex, which may come from either graph structures or node attributes.

2.3. Graph domain adaptation

Recently, there exists a branch of research in exploring the knowledge transferability among different graphs in an unsupervised setting. There are two lines of solutions: one involves minimizing pre-defined domain discrepancy metrics, while the other employs adversarial learning techniques (Liu et al., 2024; Zhang et al., 2024). CDNE (Shen et al., 2021) introduces the concept of minimizing distribution divergence using the Maximum Mean Discrepancy (MMD) (Gretton et al., 2012) for graph-structured data. MuSDAC (Yang et al., 2020) proposes a two-level selection strategy to address the unexplored problem of transferable classification among heterogeneous information networks. GRADE (Wu et al., 2023) explores graph distribution shift by considering the Weisfeiler–Lehman graph isomorphism test. Rather than minimizing domain discrepancy in an explicit way, DANE (Zhang et al., 2019) achieves embedding space alignment and distribution alignment by employing a shared weight graph convolutional network and adversarial learning regularization. AdaGCN (Dai et al., 2022) models the domain adaptation process as a two-player game similar to GANs (Mirza & Osindero, 2014). ACDNE (Shen et al., 2020) enhances the end-to-end framework by a domain discriminator equipped with Gradient Reversal Layer (GRL) (Ganin et al., 2016). UDAGCN (Wu et al., 2020) and ASN (Zhang et al., 2021) further improve network embedding with attention mechanisms and feature disentanglement. SpecReg (You et al., 2023) derives an optimal transport-based domain adaptation bound closely related to the spectral properties of GNNs and uses it to regularize these properties for improving model's transferability. StruRW (Liu et al., 2023) reduces the conditional shift of neighborhoods by computing the edge probabilities between different classes based on the pseudo node labels estimated on the target graphs. SA-GDA (Pang et al., 2023) proposes a spectral augmentation module to enhance the node representation learning, which combines the target domain spectral information with the source domain. DGDA (Cai et al., 2024) addresses graph domain adaptation in a generative view, which disentangles the generation process into the semantic latent variables, the domain latent variables, and the random latent variables. Nevertheless, the existing approaches suffer from distribution misalignment for neglecting the class from which the samples are drawn. In this paper, we propose a new framework where semantics information is considered.

3. Preliminary

3.1. Notation

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{A}, \mathbf{X}, \mathbf{Y})$ denote a graph, where $\mathcal{V} = \{v_i\}_{i=1,\dots,N}$ is the node set and $\mathcal{E} = \{e_{i,j} = (v_i, v_j)\}$ is the edge set. N represents the number of nodes and the edge $e_{i,j}$ indicates the relationship between node i and j. An adjacency matrix $\mathbf{A} \in \mathbb{R}^{N\times N}$ represents the topological structure of $\mathcal{G}$, where $A_{i,j} = 1$ if $e_{i,j} \in \mathcal{E}$ and vice versa. Let $\mathbf{X} \in \mathbb{R}^{N\times f}$ denote the node attribute matrix, where f is the dimension of node attributes. $\mathbf{Y} \in \mathbb{R}^{N\times C}$ represents the node label matrix, where C is the number of categories of node labels.

3.2. Cross-domain node classification

In this paper, we focus on a challenging task called cross-domain node classification. Following Shen et al. (2020), Zhang et al. (2021), we formally define the problem as follows: Given a fully labeled source graph $\mathcal{G}^s = (\mathcal{V}^s, \mathcal{E}^s, \mathbf{A}^s, \mathbf{X}^s, \mathbf{Y}^s)$ and a completely unlabeled target graph $\mathcal{G}^t = (\mathcal{V}^t, \mathcal{E}^t, \mathbf{A}^t, \mathbf{X}^t)$, our goal is to predict the node labels of the target graph using a co-classifier with the help of fully labeled source graph. Note that, $\mathcal{G}^s$ and $\mathcal{G}^t$ are assumed to have no common nodes, but share the same node categories. The setting of cross-domain node classification is challenging because of graph distribution shift includes both feature distribution and structure distribution, which means relationships between nodes are different across different domains. The covariance shift between source graph and target graph can be denoted as follows: $\mathbb{P}_S(Y|G) = \mathbb{P}_T(Y|G)$ and $\mathbb{P}_S(G) \neq \mathbb{P}_T(G)$.

3.3. Graph convolutional networks

Graph neural network (Cui et al., 2019) is adopted as node representation learning module to capture both the structural and attribute information of each node, which has gained remarkable progress in various graph mining tasks like node classification. For a given graph $\mathcal{G}$, node representations are generated via recursively propagating and aggregating information from each node's neighborhoods. Specifically,

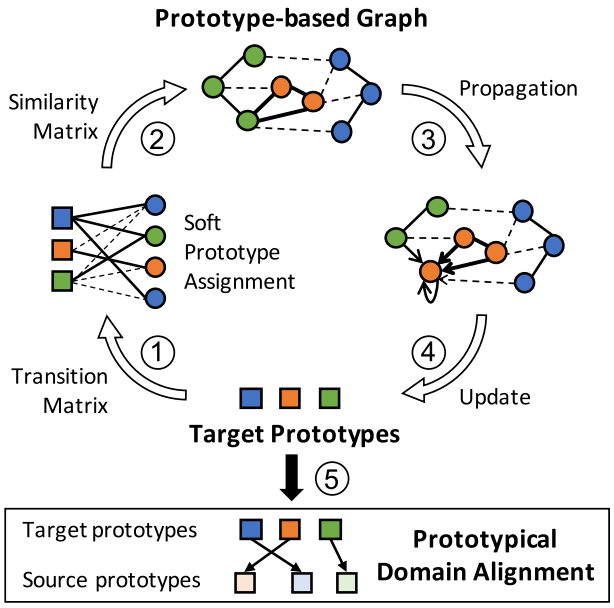


Fig. 2. The overall pipeline of SEPA. The squares denote prototypes and the circles denote target nodes. The solid lines represent connections with a higher probability, while the dashed lines represent the opposite. Our model SEPA on the highest level consists of five steps: ① Calculate base target prototypes by representation predicted by the source domain supervised classifier. ② Assign each target node with a soft prototype according to a transition matrix. ③ Construct a prototype-based graph and propagate to update representations. ④ Update the base prototype and ⑤ conduct prototypical domain alignment. This step allows us to estimate prototypes for the unlabeled target graph by considering the local structure, leading to improved transfer performance through prototypical domain alignment.

at the lth layer of graph convolutional network (Kipf & Welling, 2017), the representation of node v is updated as follows:

$$\boldsymbol{h}_v^l = \textsc{Update}^l\left(\boldsymbol{h}_v^{l-1}, \textsc{Agg}^l\left(\{\boldsymbol{h}_u^{l-1} | u \in \mathcal{N}(v)\}\right)\right), \tag{1}$$

where $\textsc{Agg}(\cdot)$ denotes an aggregation function that projects a set of neighborhood representations to an aggregated representation. $\textsc{Update}(\cdot)$ is another function that unifies the node's current representation and its aggregated neighborhood representation.

4. The proposed method

4.1. Overview

Our proposed framework SEPA addresses the issue of misalignment by incorporating class-wise alignment through the capture of semantic information. To achieve this goal, we first estimate prototypes for unlabeled target graph and then design a method to reduce the domain differences between the source and target graphs. A straightforward method to generate each class's prototype is averaging latent node representations in each class. Unfortunately, the target prototypes calculated in this way is inaccurate due to incorrect pseudo labels with a high probability under domain shift. Moreover, the existing unsupervised techniques (Kang et al., 2019; Pan et al., 2019; Tang et al., 2020; Yue et al., 2021) for obtaining prototypes under domain shift creates a dilemma: Selecting nodes with high confidence provides limited information for alignment, while the presence of unreliable pseudo-labels negatively affects the model's performance, especially when a small threshold is used.

Therefore, we aim at calibrating prototypes for unlabeled target nodes to conduct class-wise alignment and improve the overall performance. Fig. 2 illustrates the overview of our proposed SEPA, which is a non-parametric method to update the unlabeled target domain prototypes. First, we take the average representation of a target class predicted by the source domain supervised classifier as basic target prototypes. Second, we assign each target node a soft prototype by a transition matrix based on the base prototypes. Then, a prototype-based graph is constructed based on the assignment, followed by the prototype-based feature propagation. Finally, we update base target prototypes based on the updated soft-assigned prototypes.

The network architecture of SEPA is simple, which consists of two basic modules: a graph feature encoder $E_\theta(\cdot)$ and a node classifier $F_\phi(\cdot)$. Specifically, graph feature encoder $E_\theta(\cdot)$ is applied to capture informative node representations and the classifier $F_\phi(\cdot)$ is employed to predict the categories of the nodes. These two modules are shared by the source and target domains, which are jointly trained in an end-to-end manner. It is worth noting that various architectures can be incorporated into this node representation learning module, and the impact of different graph neural network architectures will be studied in the experimental section. In the following sections, we will introduce the details of different key components.

4.2. Transition matrix estimation

Pseudo labels directly obtained from predictions of the source-trained model often lead to relatively poor transfer performance. This is primarily due to the large number of incorrectly pseudo-labeled samples that are used during training (Lee, 2013). Therefore, we utilize a transition matrix $\boldsymbol{Q} \in \mathbb{R}^{C\times C}$ to represent the relationships between the true label and the pseudo label. This matrix serves to explicitly capture the label uncertainty. Since the ground truth of $\boldsymbol{Q}$ is unknown, we leverage the predictions obtained from the source-trained model to estimate the transition matrix for target graph $\mathcal{G}^t$, which is defined as:

$$\boldsymbol{Q} = \begin{bmatrix} 0 & q_{12} & \cdots & q_{1C} \\ q_{21} & 0 & \cdots & q_{2C} \\ \vdots & \vdots & \ddots & \vdots \\ q_{C1} & \vdots & q_{C(C-1)} & 0 \end{bmatrix}_{C\times C}, \tag{2}$$

where $\boldsymbol{Q}_i = q_i = \frac{1}{|S_i|}\sum_{v\in S_i}\hat{\boldsymbol{p}}_v$ denotes the average probability distribution of the ith class'. $\hat{\boldsymbol{p}}_v = \mathrm{softmax}(F_\phi(\boldsymbol{h}_v^k))$ is the probability distribution of target node v. S_i is the set of target nodes belonging to class i based on pseudo labels. To eliminate the transition probability from a class to itself, we set the diagonal of $\boldsymbol{Q}$ as zeros and normalize the non-diagonal probabilities within each row.

Next, we integrate $\boldsymbol{Q}$ into each sample's prediction to estimate the transition matrix for target nodes, denoted as $\boldsymbol{T} \in \mathbb{R}^{N^t\times C}$. The elements $\boldsymbol{T}_{ic}$ indicate the probabilities of the sample i belonging to potentially class c, with larger values suggesting a higher likelihood. The ith row of the transition matrix $\boldsymbol{T}$ can be derived as:

$$\boldsymbol{T}_i = \hat{\boldsymbol{p}}_i \odot \boldsymbol{Q}_c, \tag{3}$$

where $\hat{\boldsymbol{p}}_i$ is the predicted probability of target node v_i, and v_i's pseudo label belongs to class c. $\odot$ is element-wise multiplication. Note that $\boldsymbol{T}_{ic} = 0$ because we have eliminated the transition probability from a class to itself.

4.3. Prototype-based graph construction

The structure of the target domain plays a crucial role in encoding node attributes, which further impacts domain alignment. To leverage the personalized structure of the target graph, we construct a prototype-based graph. In this graph, the conditionally independent prototypes are directly mapped to the nodes (Liu et al., 2010; Ruis et al., 2021), allowing us to effectively capture and represent the characteristics of the target domain.

Firstly, we initialize each prototype μ_c for target graph as the mean vector of the corresponding samples predicted to its class. Given the

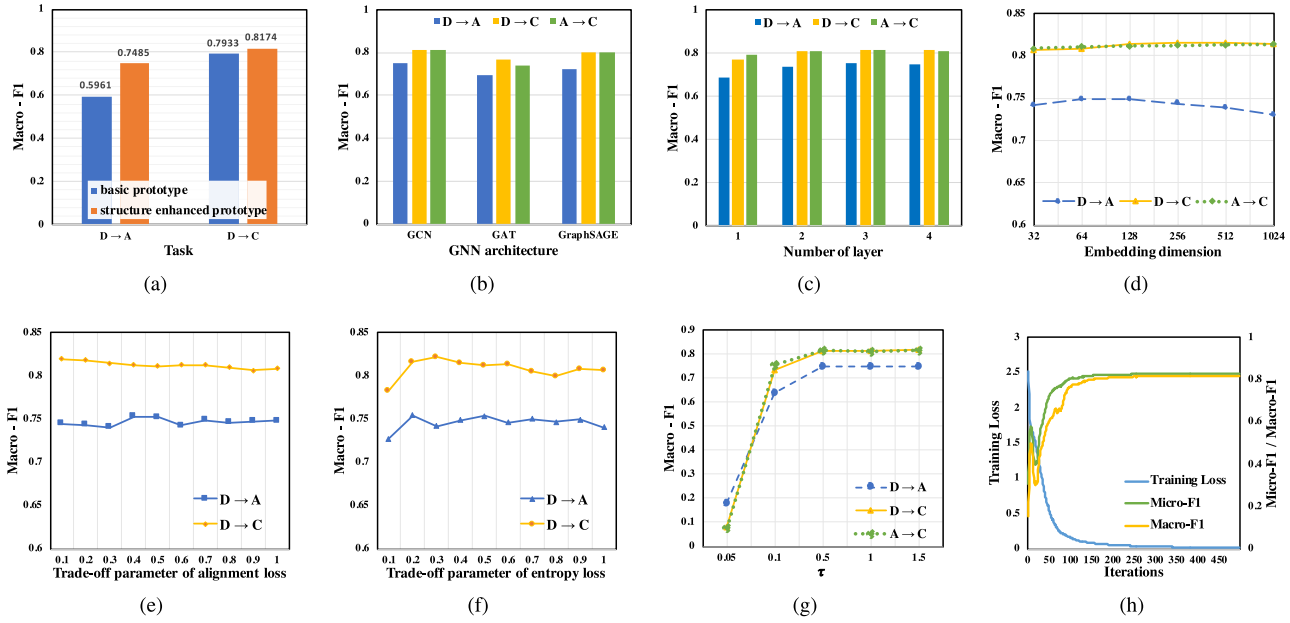


Fig. 3. (a) shows the effectiveness of structure enhanced prototype. (b)–(d) shows the impact of different hyper-parameters, namely GNN architecture, layer dimension and the number of GNN layers. (e) and (g) shows the sensitivity of two trade-off parameters. (g) shows the impact of temperature parameter τ. (h) shows the converge curves.

basic prototype set $\{\mu_c\}_{c=1}^{C}$, a target node v_i is soft-assigned (Zhu & Koniusz, 2023) to each cluster μ_c, yielding z_{ic}. Specifically, soft-assigned prototypes of target graph are obtained as follows:

$$\boldsymbol{\mu}_c = \frac{\sum_{i=1}^{N^t} \mathbb{I}(\hat{y}_i = c) \cdot \boldsymbol{h}_i}{\sum_{i=1}^{N^t} \mathbb{I}(\hat{y}_i = c)}, \tag{4}$$

$$z_{ic} = \sum_{j=1}^{C} \boldsymbol{T}_{ij} \cdot \boldsymbol{\mu}_j, \tag{5}$$

where N^t indicates the number of target nodes. $\mathbb{I}(\cdot)$ is an indicator function. $\boldsymbol{h}_i$ denotes the target node v_i's representation, which is extracted from the output of the node representation learning module $E_\theta(\cdot)$. The pseudo label denoted as $\hat{y}_i = \arg\max \hat{\boldsymbol{p}}_i$ are generated based on the predictions of $F_\phi(\cdot)$. For different instances, the soft-assigned prototypes of the same class are different.

Then, we design a adjacency matrix A' with pair-wise similarity of the soft-assigned prototypes, where A'_{ij} captures relation between the ith and jth samples. This adjacency matrix is often dense, representing a fully connected graph structure. Thus, we integrate it with given target graph structure A. We form the adjacency matrix A' as:

$$A' = \left(\boldsymbol{Z}^\top \boldsymbol{Z}\right) \odot A, \tag{6}$$

where $\odot$ is element-wise multiplication. Thus, we form a prototype-based graph $\mathcal{G}' = (\mathcal{V}, \mathcal{E}, \mathbf{A}', \mathbf{Z})$, where vertices $\mathcal{V}$ are the same as the target graph $\mathcal{G}^t$ and edges $\mathcal{E}$ are weighted by a similarity between soft-assigned prototypes.

4.4. Prototype-based graph propagation

With the prototype-based graph, we further perform propagation to update soft-assigned prototypes. This propagation step allows us to refine and improve the representations of the prototypes, taking into account both the structural information within the target graph and the semantic similarities between nodes (Klicpera et al., 2018). Here, we choose a simple and effective way to perform a node-dependent local smoothing of the feature over the constructed prototype-based graph $\mathcal{G}'$:

$$\tilde{\mathbf{Z}}^{(r+1)} = (1-\alpha)\mathbf{L}'\tilde{\mathbf{Z}}^{(r)} + \alpha\mathbf{Z}, \tag{7}$$

where $\mathbf{L}' = \mathbf{D}'^{-\frac{1}{2}}\mathbf{A}'\mathbf{D}'^{-\frac{1}{2}}$ is the corresponding Laplacian matrix of $\mathbf{A}'$. α is a hyper-parameter to control the probability of propagating information to its neighbors. We use the power iteration method (Huang et al., 2020; Klicpera et al., 2018) to accelerate the computation procedure, which usually converges within several iterations r. The impact of different propagation schemes has been shown in Section 5.3.

Then, we integrate the local structure into the prototype-based graph to obtain $\tilde{z}_{ik}$, which provides potential candidate prototype for each target node. By combining the first choice and the candidate choice, we find finds more representative prototypes for target domain and conducts class-wise alignment effectively. The corresponding experiments is shown in Fig. 3(a).

Finally, the prototypes of each class in target graph can be updated as follows:

$$\tilde{\boldsymbol{\mu}}_c = \boldsymbol{\mu}_c + \frac{1}{N_c}\sum_{i=1}^{N^t} \mathbb{I}(y_i = c) \cdot (\mathcal{H}(\boldsymbol{p}_i) \cdot \tilde{z}_i) \tag{8}$$

where $\mathcal{H}(\cdot)$ denotes the entropy function used to represent the confidence level for each node and N_c means the number of target nodes of class c. A smaller entropy means there is less need to calibrate the prototype, and vice versa. We call the updated prototypes as *structure enhanced prototypes*, which are more representative by combining the basic prototype and the candidate prototype. The ablation study is shown in Section 5.3.

4.5. Prototypical domain alignment

To explicitly minimize domain discrepancy, we incorporate class-level alignment to capture the semantic similarities between classes. The main objective is to maximize the similarities between the prototypes of each class from the source and target domains, while simultaneously minimizing the similarities among the prototypes of different classes. This ensures that clusters belonging to the same class in both domains are brought closer together in the latent space, while clusters from different classes and domains are pushed further apart. By optimizing this objective, we can effectively reduce the domain discrepancy at the semantic level.

We propose class-wise domain alignment in a prototype self-supervised form: if $\{\hat{\boldsymbol{\mu}}_i^t, \boldsymbol{\mu}_i^s\}$ forms the positive pair, then the prototypes $\{\hat{\boldsymbol{\mu}}_i^t, \boldsymbol{\mu}_{j\neq i}^s\}$ can form $C-1$ negative pairs, where $j \neq i$ denotes the classes except i. To encourage the prototype of same class to be more compact and those of different classes to be more separated, we apply a separation costs proposed by Chen et al. (2019). The formulation for class-wise domain alignment is as follows:

$$\begin{aligned}\mathcal{L}_{\text{align}}^{t,s} = &-\sum_{i=1}^{C}\left(\log\exp\left(\hat{\boldsymbol{\mu}}_i^t\cdot\boldsymbol{\mu}_i^s/\tau\right)-\log\sum_{j\neq i}^{C}\exp\left(\hat{\boldsymbol{\mu}}_i^t\cdot\boldsymbol{\mu}_j^s/\tau\right)\right)\\ &+\sum_{i=1}^{C}\left(\log\frac{1}{C-1}\sum_{j=1}^{C}\mathbb{I}(j\neq i)\cdot\exp\left(\boldsymbol{\mu}_i^s\cdot\boldsymbol{\mu}_j^s/\tau\right)\right),\end{aligned} \tag{9}$$

where τ is the temperature hyper-parameter, which controls the degree of distinguishing semantically similar prototypes. We discussed the impact of different values of τ on the effect in Section 5.4 and compared the difference between different implementation forms in Section 5.3.

To better learn knowledge transferred from source domain to target domain, we incorporate a node classification loss for the label-rich source graph, and an entropy loss for the unlabeled target graph. Ideally, a perfect prediction for target domain should be highly similar to the one-hot encoding, meanwhile covering diverse classes. To achieve this goal, we minimize the entropy for each individual sample, and maximize the entropy for each class (Liang et al., 2020). To summarize, our proposed SEPA framework has the following overall objectives:

$$\mathcal{L} = \mathcal{L}_{\text{cls}}^s + \lambda_{\text{align}}\cdot\mathcal{L}_{\text{align}}^{t,s} + \lambda_{\text{ent}}\cdot\mathcal{L}_{\text{ent}}^t, \tag{10}$$

where $\mathcal{L}_{\text{cls}}^s$ is the cross-entropy loss for source domain. $\mathcal{L}_{\text{ent}}^t = \mathbb{E}_{i\in N^t}[\mathcal{H}(\boldsymbol{p}_i^t)] - \mathcal{H}\left(\mathbb{E}_{c\in C}[\hat{\boldsymbol{p}}_c^t]\right)$ is the entropy loss, where $\mathcal{H}(\boldsymbol{p}_i) = -\sum_{c=1}^{C}\boldsymbol{p}_{i,c}\log(\boldsymbol{p}_{i,c})$ is the entropy function. $\hat{\boldsymbol{p}}_c$ is mean predictions of class c. λ_{align} and λ_{ent} represent the trade-off parameters to balance different loss terms. We discussed the effectiveness of each module in Section 5.3. The whole training procedure is summarized in Algorithm 1.

4.6. Time complexity analysis

We analyze the time complexity of our proposed framework in this subsection. Suppose the graph has n nodes and m edges, the node representation dimension is d and the number of graph neural network layers is K, the time of graph encoder module $E_\theta(\cdot)$ is $\mathcal{O}(Knd^2 + Kmd)$. The time complexity of class prototype calculation is $\mathcal{O}(n)$. Although the calibration of target prototype involves a propagation procedure, the computation cost is actually very small and it has a time complexity of $O(m)$ due to the utilization of power iteration. At last, the prototype alignment has the time complexity of $\mathcal{O}(C)$. Thus, the overall time complexity is $\mathcal{O}(Knd^2+Kmd+C)$, which is in the same time complexity scope of vanilla graph neural networks.

Algorithm 1 SEPA's Training Strategy

Require: Labelled source graph $\mathcal{G}^s = (\mathcal{V}^s, \mathcal{E}^s, \mathbf{X}^s, \mathbf{Y}^s)$, unlabelled target graph $\mathcal{G}^t = (\mathcal{V}^t, \mathcal{E}^t, \mathbf{X}^t)$, the number of graph neural network layer K, the node representation dimension d

Ensure: Predicted node labels for $\mathcal{G}^t$: $\{\hat{y}_1^t, \hat{y}_2^t, \cdots, \hat{y}_{N^t}^t\}$

1: Randomly initialize weights for $E_\theta(\cdot)$ and $F_\phi(\cdot)$
2: **while** not reaching the maximum epochs **do**
3: Learn node representations $\boldsymbol{H}^s$ and $\boldsymbol{H}^t$ for source and target domain by $E_\theta(\cdot)$ via Eq. (1)
4: Calculate base prototypes $\boldsymbol{\mu}^t$ for unlabelled target graph and ground-truth prototypes $\boldsymbol{\mu}^s$ for labelled source graph via Eq. (4)
5: Construct prototype-based graph via Eq. (5)–(6)
6: Update target prototype $\boldsymbol{\mu}^t$ to $\hat{\boldsymbol{\mu}}^t$ via Eq. (7)–(8)
7: Prototypical domain alignment via Eq. (9)
8: Back-propagate loss gradient using Eq. (10)
9: **end while**

5. Experiments

In this section, we conduct extensive experiments on various real-world graph datasets to demonstrate the effectiveness of the proposed model SEPA. We aim to answer three research questions as follows:

- **RQ1:** Is the proposed model effective in the graph domain adaptation task?
- **RQ2:** Are the modules designed in the proposed model effective and necessary?
- **RQ3:** Is the proposed model sensitive to the key hyper-parameters?

5.1. Datasets and setups

We conduct our experiments on two types of real-world graph datasets: *citation networks*[1] and *social networks*.[2] The citation networks consist of three datasets, namely ACMv9, Citationv1, and DBLPv7. In these datasets, each node represents a paper, and the edges denote the citations between them, which are obtained from diverse sources and periods. Among them, **ACMv9 (A)** comprises papers extracted from ACM between 2000 and 2010, **Citationv1 (C)** includes papers obtained from Microsoft Academic Graph prior to 2008, and **DBLPv7 (D)** encompasses papers collected from DBLP during the period from 2004 to 2008. We aim to categorize all the papers into five distinct research topics: "Databases", "Artificial Intelligence", "Computer Vision", "Information Security" and "Networking".

Regarding social networks, we choose Twitch gamer networks, which are collected from different regions. Each node within these networks represent a user, and the connections between nodes indicate their friendships. We extract node features that encompass details about the games users play and favor, their geographical location, and streaming habits, etc. Specifically, we focus on two largest graphs: **Germany (DE)** and **England (EN)**. In this scenario, users are classified into two groups based on whether they employ explicit language. For a comprehensive overview of these datasets, please refer to Table 1 and 2.

Baselines We compare our proposed SEPA with the following three categories of baselines: (1) *Hypothesis transfer with unsupervised graph representation learning*: DeepWalk (Perozzi et al., 2014), node2vec (Grover & Leskovec, 2016) and ANRL (Zhang et al., 2018). The node representations are first learned in an unsupervised manner, and then target graph node representations are evaluated using a classifier trained on source graph. (2) *Source only graph neural networks*: GCN (Kipf & Welling, 2017), GAT (Velickovic et al., 2018), GraphSAGE (Hamilton et al., 2017) and GIN (Xu et al., 2018). This group of methods train graph neural networks on the source graph in a supervised manner, and then they are directly applied to the target graph for evaluation. (3) *Graph domain adaptation methods*: CDNE (Shen et al., 2021), DANE (Zhang et al., 2019), AdaGCN (Dai et al., 2022), ACDNE (Shen et al., 2020), UDAGCN (Wu et al., 2020), ASN (Zhang et al., 2021), TPN (Pan et al., 2019), GRADE (Wu et al., 2023), StruRW (Liu et al., 2023), SA-GDA (Pang et al., 2023), SpecReg (You et al., 2023) and DGDA (Cai et al., 2024). Approaches within this group are strong baselines that are specifically designed to tackle the distribution shift problem in graph domain adaptation.

Parameter Settings To ensure a fair comparison, we utilize the source codes released by the authors and their hyper-parameters are fine-tuned to the optimal values. The node representation dimension is uniformly set to 128 across all methods, and identical GCN layers are employed throughout all methods. Our proposed SEPA is implemented with Pytorch (Paszke et al., 2019) and the model is trained using the Adam optimizer (Kingma & Ba, 2015). The temperature parameter τ is

[1] https://github.com/yuntaodu/ASN/tree/main/data
[2] http://snap.stanford.edu/data/twitch-social-networks.html

Table 1
Statistics of Citation dataset.

Dataset	Node	Edge	Attributes	Labels	Label distribution (#)				
					1	2	3	4	5
ACMv9 (A)	9360	15,556	6775	5	25.32	26.02	23.92	7.88	18.72
Citationv1 (C)	8935	15,098			21.66	32.97	23.83	6.05	15.75
DBLPv7 (D)	5484	8117			20.47	30.68	27.24	8.74	19.07

Table 2
Statistics of Social dataset.

Dataset	Node	Edge	Attributes	Labels	Label distribution (#)	
					1	2
Germany (DE)	9498	153,138	3170	2	39.55	60.45
England (EN)	7126	35,324			45.44	54.56

Table 3
Unsupervised Cross-Domain Node Classification on the Citation Network. The best result is bold and the second best is underlined.

Models	D → A		C → A		A → D		C → D		A → C		D → C		Average	
	Ma-F1	Mi-F1	Ma-F1	Mi-F1	Ma-F1	Mi-F1	Ma-F1	Mi-F1	Ma-F1	Mi-F1	Ma-F1	Mi-F1	Ma-F1	Mi-F1
DeepWalk	19.83	26.23	19.33	21.94	19.87	25.94	17.51	22.57	17.72	21.05	22.76	29.46	19.50	24.53
node2vec	22.05	28.61	17.99	21.76	19.50	24.54	24.98	28.95	25.84	29.89	16.22	21.16	21.09	25.82
ANRL	19.12	29.56	22.04	31.84	23.33	29.54	22.71	25.90	20.93	30.31	18.25	25.99	21.06	28.86
GAT	43.95	52.93	42.14	50.37	41.36	53.80	45.25	55.85	43.64	57.13	50.04	55.52	44.39	54.27
GraphSAGE	57.31	59.22	64.69	65.22	61.80	64.82	66.86	69.96	69.14	71.40	64.90	67.85	64.12	66.41
GIN	56.50	58.98	59.48	60.46	50.49	59.10	63.48	66.27	62.49	68.61	63.21	69.25	59.27	63.78
GCN	59.42	63.35	70.39	70.58	65.29	69.05	71.37	74.53	74.78	77.38	69.79	74.17	68.49	71.51
TPN	62.10	62.99	67.94	67.93	66.73	69.78	72.07	74.65	72.41	74.56	70.09	72.54	68.56	70.41
DANE	53.15	59.88	55.97	63.15	59.50	66.88	66.22	70.42	64.04	72.94	63.69	73.08	60.43	67.72
CDNE	70.45	69.62	75.06	**74.22**	69.24	71.58	71.34	74.36	76.83	78.76	77.36	78.88	73.38	74.57
UDAGCN	55.89	58.16	67.22	66.80	64.83	66.95	69.46	71.77	60.33	72.15	61.12	73.28	63.14	68.18
ACDNE	72.64	71.29	74.79	73.59	73.59	**76.24**	75.74	77.21	80.09	81.75	78.83	80.14	75.95	76.70
ASN	71.49	70.15	73.17	72.74	71.40	73.80	73.98	76.36	77.81	80.64	75.17	78.23	73.84	75.32
AdaGCN	69.47	69.67	70.77	71.67	71.39	75.04	72.34	75.59	76.51	79.32	74.22	78.20	72.45	74.92
StruRW	48.32	57.01	51.96	61.04	45.79	58.64	54.90	66.25	48.09	58.50	53.79	64.90	50.47	61.06
SA-GDA	60.59	63.81	58.38	61.74	61.47	68.28	66.69	71.52	62.62	67.74	55.49	63.11	60.87	66.03
GRADE	59.35	63.72	69.34	69.55	63.03	68.22	70.02	73.95	72.52	76.04	69.32	74.32	67.46	70.97
SpecReg	72.34	71.01	73.15	72.04	73.98	75.93	73.64	75.74	78.83	80.55	77.78	79.04	74.95	75.72
DGDA	51.82	53.19	54.11	55.97	52.65	56.71	56.48	60.55	60.87	64.80	61.24	66.15	56.19	59.56
SEPA	**74.85**	**73.83**	**75.29**	73.88	**74.78**	76.05	**76.97**	**78.08**	**81.11**	**82.46**	**81.74**	**82.82**	**77.39**	**77.79**

set to 1.0 by default and the dropout rate for each layer is set to 0.5. We search the optimal learning rate and weight decay in the range of $\{0.1, 0.01, 1e^{-3}, 1e^{-4}, 5e^{-4}\}$. The trade-off parameters λ_{align} and λ_{ent} are set to 0.1 and 0.2, respectively. In accordance with previous works (Dai et al., 2022; Shen et al., 2020, 2021), we adopt Micro-F1 (Mi-F1) and Macro-F1 (Ma-F1) scores as the evaluation metrics. The experiments are repeated five times and we report their mean performance.

5.2. Main results

Table 3 illustrates the node classification performance on citation networks. The results demonstrate that our proposed SEPA surpasses all baselines, exhibiting different gains in terms of Macro-F1. Moreover, it achieves the second-best performance in 4 out of 6 settings with respect to Micro-F1. The performance improvement can be attributed to the integration of semantic information into domain alignment through the utilization of structure enhanced prototype. This incorporation enables a more effective alignment strategy, leading to improved results.

We have the following key observations. Firstly, the performance of source-only graph neural networks outperforms hypothesis transfer approaches. This is not surprising, since the encoder and classifier are trained separately in hypothesis transfer methods. Thus, it implies that it is of great importance to establish a shared representation space for the source and target domains. Secondly, graph domain adaptation methods generally achieve superior performance compared to source-only graph neural networks, with the exception of UDAGCN. This highlights the benefits of addressing domain distribution shift, which improves the model's generalization ability. For our proposed SEPA, we have the following analyses:

- Compared with adversarial-based domain adaptation methods (e.g., UDAGCN (Wu et al., 2020), AdaGCN (Dai et al., 2022), ASN (Zhang et al., 2021)), our proposed SEPA can consistently achieve better performance. The reason is that adversarial-based methods often suffer an unstable training procedure, and it cannot guarantee class-wise alignment even the discriminator reaches optimal.
- Our model demonstrates significant improvements compared to the prototype-based domain adaptation method TPN (Pan et al., 2019) developed for computer vision. This highlights the limitations of directly applying existing prototype-based domain adaptation algorithms designed for computer vision to cross-domain node classification task. It emphasizes the importance of developing methods that specifically consider the unique characteristics of graphs in domain adaptation, due to the inherent complexity of graph data and its non-IID nature.
- Among all the baselines and SEPA, it can be observed that the Micro-F1 score tends to be higher than the Macro-F1 score in most cases. This phenomenon can be attributed to the fact that the Micro-F1 score may be influenced by larger classes if a method incorrectly classifies a majority of the data into a few classes. On the other hand, the Macro-F1 score considers the F1 score within each class and is less influenced by class imbalances. Interestingly, the SEPA approach has narrowed the gap between Micro-F1 and Macro-F1, indicating its ability to improve classification results for individual classes rather than blindly grouping data into a few predominant classes. This highlights the

Table 4
Unsupervised Cross-Domain Node Classification on the Social Network. The best result is bold and the second best is underlined.

Models	DE→EN		EN→DE		Average	
	Ma-F1	Mi-F1	Ma-F1	Mi-F1	Ma-F1	Mi-F1
Node2vec	46.96	52.64	50.10	54.61	48.53	53.63
DeepWalk	46.54	52.18	49.97	55.08	48.26	53.63
GAT	49.50	54.84	40.08	43.65	44.79	49.25
GCN	54.55	54.77	51.04	62.03	52.80	58.40
GIN	49.91	52.39	44.26	55.26	47.09	53.83
TPN	42.81	54.42	53.58	53.82	48.20	54.12
DANE	53.13	57.00	57.83	60.65	55.48	58.82
UDAGCN	**58.19**	**59.74**	56.35	58.69	57.27	59.22
SpecReg	50.30	56.43	46.13	61.45	48.22	58.94
ACDNE	56.31	58.08	57.92	58.79	57.12	58.44
ASN	51.21	55.45	45.90	60.45	48.56	57.95
AdaGCN	35.30	54.56	31.18	40.22	33.24	47.39
StruRW	50.13	56.06	**59.68**	60.81	54.91	58.44
SA-GDA	37.59	51.62	40.62	60.09	38.91	55.85
GRADE	56.38	56.40	56.83	61.18	56.61	58.79
SpecReg	50.30	56.43	46.13	61.45	48.22	58.94
DGDA	51.60	57.24	51.85	62.70	51.73	59.97
SEPA	57.44	57.90	58.34	**63.58**	**57.89**	**60.74**

Table 5
Performance contribution of each part in SEPA.

Loss	D → A		A → C		D → C	
	Ma-F1	Mi-F1	Ma-F1	Mi-F1	Ma-F1	Mi-F1
$\mathcal{L}^s_{cls}$	59.52	63.52	70.74	70.63	70.01	74.32
+ $\mathcal{L}^{t,s}_{align}$	72.27	70.72	74.67	73.40	78.66	79.74
+ $\mathcal{L}^t_{ent}$ (SEPA)	**74.85**	**73.83**	**81.11**	**82.46**	**81.74**	**82.82**

Table 6
Comparisons among different propagation schemes.

Propagation schemes	Ma-F1	Mi-F1
Simple Propagate	73.57	72.08
Adaptive Propagate	73.53	71.98
PageRank	74.85	73.83

capability of SEPA to better preserve the semantic information during domain adaptation.

Similar to citation networks, we can draw similar conclusions on social networks. The results are shown in Table 4. In general, our proposed SEPA outperforms or shows comparable performance among all the baselines. More specifically, SEPA obtains the highest scores on and Micro-F1 for the $EN \rightarrow DE$ task, and it ranks second in terms of Macro-F1 for the $DE \rightarrow EN$ task. Additionally, we observe that some domain adaptation methods are outperformed by source-only graph neural networks, which is caused by negative transfer. This is attributed to the heterophilic nature of these social graphs, which poses significant challenges for domain adaptation tasks. It is worth noting that UDAGCN performs well in social networks while showing poor performance in citation networks. One possible reason for this discrepancy is that they incorporate the PPMI matrix to ensure global consistency, which mitigates the negative affect of heterogeneity in the graph. Finally, our proposed SEPA demonstrates effective performance across different types of graphs, which validates its capability in domain adaptation and answers **RQ1**.

5.3. Ablation study

To investigate the contributions of different components and losses in our proposed model, we conducted ablation studies on $D \rightarrow A$, $D \rightarrow C$ and $A \rightarrow C$.

Table 7
Comparisons among different alignments implementation forms. E-Dist and p-MMD are two distance-based alignment methods. $SEPA_{ip}$ and $SEPA$ are two self-supervision based methods.

Tasks	Metric	E-Dist	p-MMD	SEPA$_{ip}$	SEPA
D → A	Ma-F1	55.24	52.49	73.03	**74.85**
	Mi-F1	65.15	62.79	72.14	**73.83**
D → C	Ma-F1	77.03	63.44	81.70	**81.74**
	Mi-F1	62.69	76.30	82.64	**82.82**

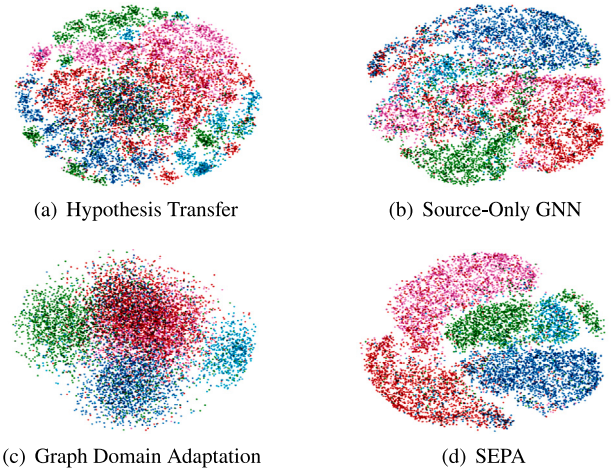
The effectiveness of structure enhanced prototype. To verify the effectiveness of structure enhanced prototypes, we conduct a comparison with a variant of SEPA, where the prototypes are estimated based on the source graph supervised classifier (Eq. (4)), referred to as the *basic prototypes*. As depicted in Fig. 3-(a), without using structure enhanced prototypes causes 15.24% and 2% decrease in Macro-F1. This indicates that the prototypes estimated by the source graph supervised classifier are biased due to the existence of domain distribution shift. Using biased prototypes could lead to misalignment between source and target domains, resulting in sub-optimal performance.

Impact of different propagation schemes. To explore the influence of various propagation schemes on SEPA, we conduct additional experiments on the $D \rightarrow A$ task by replacing the original PageRank with alternative widely used propagation schemes, namely Simple Propagate (Wu et al., 2019) and Adaptive Propagate (Du et al., 2017) which propagate information with power and accumulated adjacency matrix respectively. The results, presented in the Table 6, demonstrate that the performance of SEPA remains consistent regardless of the chosen propagation scheme. This robustness indicates that our proposed SEPA is not significantly affected by different propagation schemes. Furthermore, it confirms the effectiveness of SEPA in capturing class-specific information from the graph structure.

Discussion on the implementation form of class-wise alignment. Given that class-wise domain alignment can be implemented in various forms, we conduct comparisons with three variants of SEPA: (*i*) E-Dist: This variant minimizes the Euclidean distance between prototypes with the same semantics. (*ii*) p-MMD: This approach measures the class-level domain discrepancy by computing pairwise Reproducing Kernel Hilbert Space (RKHS) distance between the prototypes of the same class from different domains (Pan et al., 2019). (*iii*) SEPA$_{ip}$: In contrast to SEPA, which emphasizes prototype–prototype alignment, SEPA$_{ip}$ focuses on modeling positive and negative matching between instances and prototypes across different classes. Specifically, SEPA$_{ip}$ aims to maximize the similarities between each node and the corresponding target prototype, while minimizing their similarities among the remaining class prototypes.

We conduct experiments on the $D \rightarrow A$ and $D \rightarrow C$ tasks, and the results are presented in Table 7. As we can see, both SEPA$_{ip}$ and SEPA outperform the two distance-based methods, namely E-Dist and p-MMD. This is because the pre-defined distance metrics might not be appropriate in these scenarios. While the self-supervision based methods enforce close proximity between prototypes of the same class in the embedding space, it could result in consistent representation distribution across different domains and provides greater flexibility. The performance of SEPA$_{ip}$ is comparable to that of SEPA, demonstrating the effectiveness of various self-supervision based alignment methods. Overall, the prototypical domain alignment in SEPA proves to be highly effective.

Effectiveness of different losses. Table 5 demonstrates that the addition of each loss contributes to the final results without any performance degradation. The first line, $\mathcal{L}_{cls^s}$, only considers the source domain information and performs poorly, indicating that the model cannot be effectively applied to the target graph in this case. The second line incorporates the reduction of domain discrepancy by adding



(a) Hypothesis Transfer (b) Source-Only GNN

(c) Graph Domain Adaptation (d) SEPA

Fig. 4. Visualization results on target domain embeddings.

$\mathcal{L}_{align^{t,s}}$ to $\mathcal{L}_{cls^s}$, highlighting the importance of reducing domain discrepancy. Moreover, the increase in Macro-F1 further improves its performance across different categories. The third line demonstrates that the utilization of target predictions through minimizing $\mathcal{L}_{ent}$ can further boosts its performance, resulting in the complete version of our proposed SEPA. Therefore, the overall loss function achieves the best performance, which validates the effectiveness of combining all the aforementioned modules and answers **RQ2**.

5.4. Parameter analysis

We perform a sensitivity analysis of the key hyper-parameters in our proposed SEPA by varying them at different scales. We focus on providing a detailed description of the sensitivity analysis results based on Macro-F1 for Macro-F1 and Micro-F1 have similar trends.

Impact of different GNN architectures. We evaluate the performance of three commonly used GNN architectures, namely GCN (Kipf & Welling, 2017), GAT (Velickovic et al., 2018), and GraphSAGE (Hamilton et al., 2017), with the same network layers. As shown in Fig. 3-(b), the results demonstrate variations among the architectures within a reasonable range. Among them, GCN achieves the highest performance, while GAT performs the poorest. This discrepancy could be attributed to the sub-optimal calculation of attention weights in the target domain caused by distribution shift. These findings further motivate our investigation into graph domain adaptation.

Impact of dimension d. We examine the impact of different dimension values by considering a range of scales, including $\{64, 128, 256, 512, 1024\}$. In the case of $D \rightarrow A$, as depicted in Fig. 3-(c), we observe that setting $d = 64$ yield superior performance compared to other values. Conversely, in the scenarios of $D \rightarrow C$ and $A \rightarrow C$, varying dimension values does not exhibit significant performance differences. Our findings indicate that achieving good performance in node classification tasks does not necessarily require a higher representation dimension, which is beneficial for efficient learning on target domains. Additionally, employing a large representation dimension could potentially lead to overfitting, thus it is important to select an appropriate representation dimension.

Impact of the number of network layers l. We further explore the impact of the number of layers in graph neural networks. Fig. 3-(d) illustrates the results. Using a shallow graph neural network may result in underfitting and lead to a decrease in model performance. On the other hand, employing a deeper network can improve the model's performance to some extent. However, we notice a decline in performance when the number of network layers became excessively large (e.g., $l = 4$). This finding aligns with the well-known over-smoothing problem encountered in graph neural networks.

Sensitivity of trade-off parameters. To assess the sensitivity of the trade-off parameters, we conduct experiments on tasks $D \rightarrow A$ and $D \rightarrow C$, as depicted in Fig. 3-(e) and (f). In each case, we fix all the remaining hyper-parameters to their default values. The results indicate that our proposed SEPA performs well across a range of λ_{align} and λ_{ent} values, suggesting its robustness in the optimization process.

Sensitivity of τ. The temperature parameter τ plays a important role in the prototypical domain alignment, as demonstrated by the sensitivity analysis in Fig. 3-(g). We observe that a small value of $\tau = 0.05$ can break the underlying semantic structure and hinder the formation of alignment features. In contrast, when τ is larger than 0.5, the model's performance is not significantly affected by this parameter. Therefore, we conclude that the proposed method SEPA is not highly sensitive to the parameter τ when it is larger than 0.5. The above analysis all answered **RQ3**.

5.5. Convergence verification.

To verify the convergence of our proposed model SEPA, we perform an experiment on task $D \rightarrow C$ and present the convergence curves in Fig. 3-(h). From the figure, we can observe that the training loss consistently decreases over iterations, while both the Macro-F1 and Micro-F1 metrics steadily increase. This indicates that our proposed model can converge within a reasonable number of iterations, and according to the convergence curve, SEPA reaches convergence within just 100 iterations.

5.6. Visualization

In order to gain deeper insights into the discriminative power of the learned embeddings in the unsupervised cross-domain node classification task, we conduct a visualization of the node embeddings in the target domain for the $D \rightarrow A$ scenario. We compare our proposed SEPA with that of hypothesis transfer (DeepWalk), the source-only method (GCN), and the strongest graph domain adaptation method (ACDNE). The node embeddings are projected onto a 2-dimensional space using t-SNE (Van der Maaten & Hinton, 2008). In this projected space, nodes with similar semantic tend to be close to each other, while nodes with different semantic appear more distant. To facilitate visual analysis, we color the nodes based on their labels. As depicted in Fig. 4, the visualization results reveal that the embeddings learned by our proposed SEPA exhibit better separations between different clusters. Clear boundaries are observed between distinct clusters, indicating the effectiveness of SEPA in learning discriminative embeddings. In contrast, the embeddings generated by the other methods do not exhibit compact clusters and show significant overlapping across clusters.

6. Conclusion

In this paper, we study the problem of unsupervised graph domain adaptation by overcoming the limitation of existing works in capturing the class-wise semantic relation under domain shift. We propose a novel framework named SEPA, which is capable of dynamically adjusting structure enhanced prototype for target graph to implement class-wise alignment. To learn structure enhanced prototype, our method constructs a prototype-based graph to strike a better balance between the target structure and supervised signals. Furthermore, we also propose an explicit domain alignment metric to capture their semantic similarity in a self-supervised manner. Extensive experiments demonstrate the superiority of SEPA in narrowing domain discrepancy.

CRediT authorship contribution statement

Meihan Liu: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Formal analysis. **Zhen Zhang:** Writing – review & editing, Supervision, Formal analysis. **Ning Ma:** Writing – review & editing, Visualization, Formal analysis. **Ming Gu:** Writing – review & editing, Visualization. **Haishuai Wang:** Writing – review & editing, Supervision. **Sheng Zhou:** Writing – review & editing, Supervision, Resources, Funding acquisition. **Jiajun Bu:** Writing – review & editing, Supervision, Funding acquisition.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Sheng Zhou reports financial support was provided by National Natural Science Foundation of China. Sheng Zhou reports financial support was provided by Zhejiang Provincial Natural Science Foundation of China. Sheng Zhou reports financial support was provided by Ningbo Natural Science Foundation. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The data that has been used is publicly available.

Acknowledgments

This work is supported in part by the National Natural Science Foundation of China (Grant No. 62106221, 61972349), Zhejiang Provincial Natural Science Foundation of China (Grant No. LTGG23F030005), and Ningbo Natural Science Foundation (Grant No. 2022J183).

References

Bhagat, S., Cormode, G., & Muthukrishnan, S. (2011). Node classification in social networks. In *Social network data analytics*.

Cai, R., Wu, F., Li, Z., Wei, P., Yi, L., & Zhang, K. (2024). Graph domain adaptation: A generative view. *ACM Transactions on Knowledge Discovery from Data*, *18*(3), 1–24.

Chen, C., Li, O., Tao, D., Barnett, A., Rudin, C., & Su, J. K. (2019). This looks like that: deep learning for interpretable image recognition. *Advances in neural information processing systems*, *32*.

Cui, P., Wang, X., Pei, J., & Zhu, W. (2019). A survey on network embedding. *IEEE Transactions on Knowledge Data Engineering*, *31*, 833–852.

Dai, Q., Wu, X.-M., Xiao, J., Shen, X., & Wang, D. (2022). Graph transfer learning via adversarial domain adaptation with graph convolution. *IEEE Transactions on Knowledge Data Engineering*.

Du, J., Zhang, S., Wu, G., Moura, J. M., & Kar, S. (2017). Topology adaptive graph convolutional networks. arXiv preprint arXiv:1710.10370.

Ganin, Y., & Lempitsky, V. (2015). Unsupervised domain adaptation by backpropagation. In *International conference on machine learning* (pp. 1180–1189). PMLR.

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., & Lempitsky, V. S. (2016). Domain-adversarial training of neural networks. *Journal of Machine Language Reserch*.

Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B., & Smola, A. (2012). A kernel two-sample test. *Journal of Machine Language Reserch*, *13*, 723–773.

Grover, A., & Leskovec, J. (2016). Node2vec: Scalable feature learning for networks. In *KDD*.

Hamilton, W. L., Ying, Z., & Leskovec, J. (2017). Inductive representation learning on large graphs. In *NIPS*.

Huang, Q., He, H., Singh, A., Lim, S.-N., & Benson, A. R. (2020). Combining label propagation and simple models out-performs graph neural networks. arXiv preprint arXiv:2010.13993.

Huang, X., Li, J., & Hu, X. (2017). Label informed attributed network embedding. In *WSDM*.

Ji, S., Pan, S., Cambria, E., Marttinen, P., & Yu, P. S. (2022). A survey on knowledge graphs: Representation, acquisition, and applications. *IEEE Transactions on Neural Networks Learning Systems*, *33*, 494–514.

Kang, G., Jiang, L., Yang, Y., & Hauptmann, A. (2019). Contrastive adaptation network for unsupervised domain adaptation. In *2019 IEEE/CVF conference on computer vision and pattern recognition* (pp. 4888–4897).

Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization, CoRR abs/1412.6980. arXiv:1412.6980.

Kipf, T., & Welling, M. (2017). Semi-supervised classification with graph convolutional networks, arxiv abs/1609.02907.

Klicpera, J., Bojchevski, A., & Günnemann, S. (2018). Predict then propagate: Graph neural networks meet personalized pagerank. arXiv preprint arXiv:1810.05997.

Lee, D.-H. (2013). Pseudo-label : The simple and efficient semi-supervised learning method for deep neural networks.

Liang, J., Hu, D., & Feng, J. (2020). Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In *International conference on machine learning* (pp. 6028–6039). PMLR.

Liu, M., Fang, Z., Zhang, Z., Gu, M., Zhou, S., Wang, X., & Bu, J. (2024). Rethinking propagation for unsupervised graph domain adaptation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 13963–13971.

Liu, W., He, J., & Chang, S.-F. (2010). Large graph construction for scalable semi-supervised learning. In *Proceedings of the 27th international conference on machine learning (ICML-10)* (pp. 679–686). Citeseer.

Liu, S., Li, T., Feng, Y., Tran, N., Zhao, H., Qiu, Q., & Li, P. (2023). Structural reweighting improves graph domain adaptation. In *International conference on machine learning* (pp. 21778–21793). PMLR.

Long, M., Cao, Y., Wang, J., & Jordan, M. I. (2015). Learning transferable features with deep adaptation networks. In *Proceedings of the 32nd international conference on international conference on machine learning - volume 37* (pp. 97–105). JMLR.org.

Long, M., Cao, Z., Wang, J., & Jordan, M. I. (2017). Conditional adversarial domain adaptation. In *Neural information processing systems*.

Mirza, M., & Osindero, S. (2014). Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784.

Pan, S. J., & Yang, Q. (2010). A survey on transfer learning. *IEEE Transactions on Knowledge Data Engineering*, *22*, 1345–1359.

Pan, Y., Yao, T., Li, Y., Wang, Y., Ngo, C.-W., & Mei, T. (2019). Transferrable prototypical networks for unsupervised domain adaptation. In *CVPR* (pp. 2234–2242).

Pang, J., Wang, Z., Tang, J., Xiao, M., & Yin, N. (2023). Sa-gda: Spectral augmentation for graph domain adaptation. In *Proceedings of the 31st ACM international conference on multimedia* (pp. 309–318).

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., ... Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*.

Pei, Z., Cao, Z., Long, M., & Wang, J. (2018). Multi-adversarial domain adaptation. In *Proceedings of the AAAI conference on artificial intelligence, vol. 32, no. 1*.

Perozzi, B., Al-Rfou, R., & Skiena, S. (2014). DeepWalk: online learning of social representations. In *KDD*.

Quionero-Candela, J., Sugiyama, M., Schwaighofer, A., & Lawrence, N. D. (2009). Dataset shift in machine learning.

Rao, X., Chen, L., Liu, Y., Shang, S., Yao, B., & Han, P. (2022). Graph-flashback network for next location recommendation. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining* (pp. 1463–1471).

Rozemberczki, B., Allen, C., & Sarkar, R. (2021). Multi-scale attributed node embedding. *Journal of Complex Networks*, *9*(2), cnab014.

Ruis, F., Burghouts, G., & Bucur, D. (2021). Independent prototype propagation for zero-shot compositionality. *Advances in Neural Information Processing Systems*, *34*, 10641–10653.

Sen, P., Namata, G., Bilgic, M., Getoor, L., Galligher, B., & Eliassi-Rad, T. (2008). Collective classification in network data. *AI Magazine*, *29*(3), 93.

Shen, X., Dai, Q., Chung, K. F.-L., Lu, W., & Choi, K.-S. T. (2020). Adversarial deep network embedding for cross-network node classification. In *AAAI*.

Shen, X., Dai, Q., Mao, S., lai Chung, F., & Choi, K.-S. T. (2021). Network together: Node classification via cross-network deep network embedding. *IEEE Transactions on Neural Networks and Learning Systems*, *32*, 1935–1948.

Shen, J., Qu, Y., Zhang, W., & Yu, Y. (2018). Wasserstein distance guided representation learning for domain adaptation. In *Proceedings of the AAAI conference on artificial intelligence, vol. 32, no. 1*.

Shi, C., Hu, B., Zhao, W. X., & Philip, S. Y. (2018). Heterogeneous information network embedding for recommendation. *IEEE Transactions on Knowledge and Data Engineering*, *31*(2), 357–370.

Sun, B., Feng, J., & Saenko, K. (2016). Return of frustratingly easy domain adaptation. In *Proceedings of the AAAI conference on artificial intelligence, vol. 30, no. 1*.

Tang, H., Chen, K., & Jia, K. (2020). Unsupervised domain adaptation via structurally regularized deep clustering. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 8722–8732).

Tang, J., Zhang, J., Yao, L., Li, J.-Z., Zhang, L., & Su, Z. (2008). ArnetMiner: extraction and mining of academic social networks. In *KDD*.

Torralba, A., & Efros, A. A. (2011). Unbiased look at dataset bias. In *CVPR 2011* (pp. 1521–1528).

Tzeng, E., Hoffman, J., Saenko, K., & Darrell, T. (2017). Adversarial discriminative domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 7167–7176).

Van der Maaten, L., & Hinton, G. (2008). Visualizing data using t-sne.. *Journal of Machine Learning Research*, *9*(11).

Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Lio', P., & Bengio, Y. (2018). Graph attention networks, arxiv abs/1710.10903.

Wang, M., & Deng, W. (2018). Deep visual domain adaptation: A survey. *Neurocomputing*, *312*, 135–153.

Wang, D., & Zheng, T. F. (2015). Transfer learning for speech and language processing. In *2015 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)* (pp. 1225–1237).

Wu, J., He, J., & Ainsworth, E. (2023). Non-iid transfer learning on graphs. In *Proceedings of the AAAI conference on artificial intelligence, vol. 37, no. 9* (pp. 10342–10350).

Wu, M., Pan, S., Zhou, C., Chang, X., & Zhu, X. (2020). Unsupervised domain adaptive graph convolutional networks. In *Proceedings of the Web Conference 2020*.

Wu, F., Souza, A., Zhang, T., Fifty, C., Yu, T., & Weinberger, K. (2019). Simplifying graph convolutional networks. In *International conference on machine learning* (pp. 6861–6871). PMLR.

Xu, L., Cao, J., Wei, X., & Philip, S. Y. (2019). Network embedding via coupled kernelized multi-dimensional array factorization. *IEEE Transactions on Knowledge and Data Engineering*, *32*(12), 2414–2425.

Xu, K., Hu, W., Leskovec, J., & Jegelka, S. (2018). How powerful are graph neural networks? arXiv preprint arXiv:1810.00826.

Yang, S., Song, G., Jin, Y., & Du, L. (2020). Domain adaptive classification on heterogeneous information networks. In *International joint conference on artificial intelligence*. URL https://api.semanticscholar.org/CorpusID:220480908.

You, Y., Chen, T., Wang, Z., & Shen, Y. (2023). Graph domain adaptation via theory-grounded spectral regularization. In *The eleventh international conference on learning representations*.

Yue, X., Zheng, Z., Zhang, S., Gao, Y., Darrell, T., Keutzer, K., & Sangiovanni-Vincentelli, A. (2021). Prototypical cross-domain self-supervised learning for few-shot unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Zhang, X., Du, Y., Xie, R., & Wang, C. (2021). Adversarial separation network for cross-network node classification. In *CIKM*.

Zhang, Z., Liu, M., Wang, A., Chen, H., Li, Z., Bu, J., & He, B. (2024). Collaborate to adapt: source-free graph domain adaptation via bi-directional adaptation. In *Proceedings of the ACM on Web Conference 2024* (pp. 664–675).

Zhang, Y., Song, G., Du, L., Yang, S., & Jin, Y. (2019). DANE: Domain adaptive network embedding. In *International joint conference on artificial intelligence*. URL https://api.semanticscholar.org/CorpusID:173991053.

Zhang, Z., Yang, H., Bu, J., Zhou, S., Yu, P., Zhang, J., Ester, M., & Wang, C. (2018). ANRL: Attributed network representation learning via deep neural networks. In *IJCAI*.

Zhu, H., & Koniusz, P. (2023). Transductive few-shot learning with prototype-based label propagation by iterative graph refinement. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 23996–24006).