Open-Set Graph Domain Adaptation via Separate Domain Alignment

Yu Wang[1]*, **Ronghang Zhu**[2]*, **Pengsheng Ji**[3], **Sheng Li**[4]

[1] LinkedIn Corporation
[2] School of Computing, University of Georgia
[3] Department of Statistics, University of Georgia
[4] School of Data Science, University of Virginia
ywang11@linkedin.com, ronghangzhu@uga.edu, psji@uga.edu, shengli@virginia.edu

Abstract

Domain adaptation has become an attractive learning paradigm, as it can leverage source domains with rich labels to deal with classification tasks in an unlabeled target domain. A few recent studies develop domain adaptation approaches for graph-structured data. In the case of node classification task, current domain adaptation methods only focus on the closed-set setting, where source and target domains share the same label space. A more practical assumption is that the target domain may contain new classes that are not included in the source domain. Therefore, in this paper, we introduce a novel and challenging problem for graphs, i.e., open-set domain adaptive node classification, and propose a new approach to solve it. Specifically, we develop an algorithm for efficient knowledge transfer from a labeled source graph to an unlabeled target graph under a separate domain alignment (SDA) strategy, in order to learn discriminative feature representations for the target graph. Our goal is to not only correctly classify target nodes into the known classes, but also classify unseen types of nodes into an unknown class. Experimental results on real-world datasets show that our method outperforms existing methods on graph domain adaptation.

Introduction

Many top-performing machine learning models are trained on large-scale labeled data. However, in practice, labels can be hard to obtain due to the huge cost and/or considerable difficulty of labeling. To handle these challenges, domain adaptation (DA) (Zhu et al. 2023) is proposed to transfer knowledge from a labeled dataset, namely source domain, to an unlabeled dataset, namely target domain, while domain divergence always exists among source and target domains. DA has drawn much attention in recent years for several reasons. First, traditional machine learning methods require a large amount of labeled data for model training, but unlabeled data from new domains constantly emerge. Second, compared to labeling the data from every new domain, it is more efficient to transfer knowledge from a similar domain that already has sufficient labels. Third, studies have shown promising performance on knowledge transfer using domain adaptation techniques in multiple fields, such as computer vision (Zhu and Li 2022; Zhu et al. 2021b), natural language processing (Jiang and Zhai 2007; Rezayi et al. 2023), and facial expression recognition (Zhu, Sang, and Zhao 2016; Zhu et al. 2015).

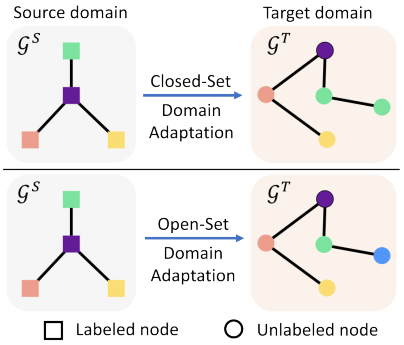


Figure 1: Problem setting difference between closed-set and open-set graph domain adaptation. Different colors denote different categories.

Graph data, which efficiently represents the relationships (edges) between objects (nodes), is ubiquitous and has various applications in the real world. One of the most important learning tasks on graph data is node classification, where the algorithms learn to predict the category of each node. Examples of this task are found in diverse areas, including social networks[1] (Bhagat, Cormode, and Muthukrishnan 2011), citation networks (Ji and Jin 2016), protein–protein association networks (Szklarczyk et al. 2018), and product co-purchasing networks (Bhatia et al. 2016). Although many algorithms have been developed for supervised and semi-supervised graph learning, the topic of cross-network (or cross-graph) domain adaptation has been largely underexplored. In recent years, a few graph domain adaptation methods are brought up (Shen et al. 2021; Dai et al. 2022; Wu et al. 2020). They focus on the closed-set cross-network node classification problem, holding the assumption that the target graph contains nodes of categories only in the source graph, however, which is unrealistic. In many real-world applications, the target graph always contains novel nodes that

*These authors contributed equally to this research.

[1] In this paper, we use the two terms, graph and network, interchangeably to refer to graph-structured data.

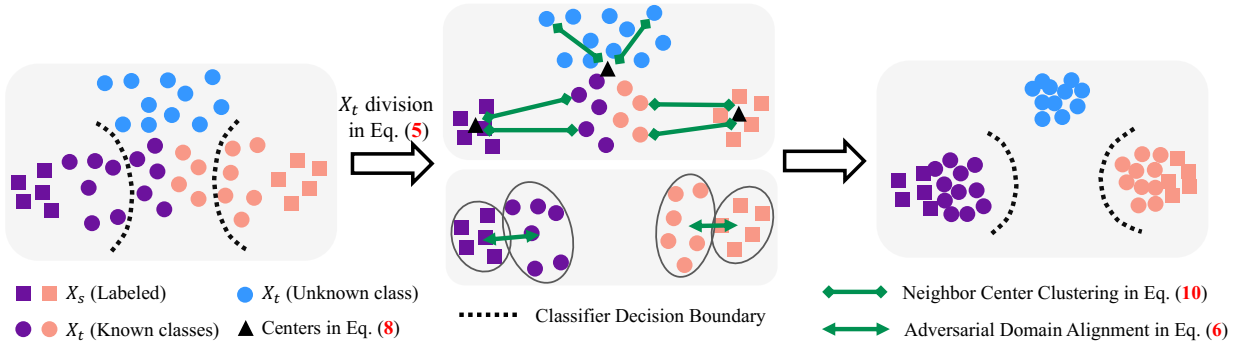


Figure 2: Illustration of the proposed separate domain alignment (SDA), which consists of three parts: (1) dividing target nodes into two groups, i.e., certain group and uncertain group; (2) for certain group, we utilize adversarial domain alignment to align target nodes to source nodes; (3) for uncertain group, we propose a neighbor center clustering loss to cluster target nodes.

are out of the label space of the source graph. Inspired by this, we introduce the concept of open-set (Geng, Huang, and Chen 2020) cross-network node classification. Specifically, we allow the target graph to contain nodes of unknown classes that don't belong to the source label space, and require the learned model to not only correctly classify target node if its label is in the source label space but also to be able to identify it as "unknown". The difference between closed-set domain adaptation and open-set graph domain adaptation is illustrated in Fig. 1.

To address the new challenging problem, i.e., Open-Set Graph Domain Adaptation (OS-GDA), we propose a novel separate domain alignment (SDA) framework. An overview is shown in Fig. 2. Rather than directly aligning source and target domains without considering target unknown class nodes, SDA provides different domain alignment strategies for different target nodes. Specifically, we roughly split the target nodes into two groups, i.e., certain group and uncertain group, based on their entropy values of the classifier's outputs. The data in certain group have smaller entropy values compared with threshold while those in uncertain group have larger entropy values. In principle, entropy estimates the prediction uncertainty. The smaller the entropy, the higher the certainty of the prediction is. For the certain group, we utilize adversarial learning to align them to the source domain. For the uncertain group, we propose a neighbor center clustering method to better separate target data from known classes and those from unknown classes. In this way, the target data coming from known classes would align to the source domain while those from unknown classes would be far from data of known classes.

Our contributions are summarized as follows:

- We introduce a practical and challenging task, namely open-set graph domain adaptation (OS-GDA), that allows target graphs to contain unknown class nodes.
- We propose the Separate Domain Alignment (SDA) framework, which provides suitable domain alignment strategies for different target nodes.
- We conduct extensive experiments and show that our method successfully tackles the novel OS-GDA problem and surpasses all baselines with large margins.

Related Work

Graph Neural Networks

Since graph neural network (GNN) was first brought up to extend general neural network methods to graph domains (Scarselli et al. 2009), numerous GNN algorithms have been developed and shown impressive performance in graph learning tasks, which include node classification, edge classification, graph classification, link prediction, regression, etc. Representative GNNs such as graph convolutional network (GCN) (Kipf and Welling 2017) and GraphSAGE (Hamilton, Ying, and Leskovec 2017) utilize the adjacency matrix and define their own convolution operators on the graphs. GCN makes a connection between spectral convolution and spatial convolution methods, which inspires GraphSAGE, a spatial-based method that becomes one of the most popular GNN methods because of its simplicity and top-tier performance (Hamilton, Ying, and Leskovec 2017; Wu et al. 2021; Zhu et al. 2021c). However, these GNN modes would fail to solve the cross-network problem due to the domain divergence between training graph and testing graph (Wu et al. 2020). GCN is widely used for node classification tasks and is one of the most popular GNN structures used in graph DA (Wu et al. 2020; Zhang et al. 2021; Dai et al. 2022). In those graph DA algorithms, GCN and its variations act as the backbone structure to solve the cross-network node classification problem and demonstrate promising performances.

Closed-Set Domain Adaptation

Many DA methods have been proposed and achieved success in various fields. Most existing methods belong to closed-set domain adaptation (CS-DA), which aims to reduce the domain divergence between source and target domains and extract domain-invariant features (Ganin and Lempitsky 2015; Zhu et al. 2021a; Shi, Zhu, and Li 2022). They can be roughly divided into two categories: (1) moment matching based methods which statistically match the data distributions such as maximum mean discrepancy (MMD) (Gretton et al. 2012) and CORrelation ALignment (CORAL) (Sun, Feng, and Saenko 2016); (2) adversarial learning based methods which play a minimax game between feature extractor and domain discriminator to learn

domain-invariant features including Domain Adversarial Neural Networks (DANN) (Ganin et al. 2016) and Conditional Domain Adversarial Networks (CDAN) (Long et al. 2018a).

Compared to DA in CV and NLP fields, graph domain adaptation is a relatively new topic. There are only a limited number of methods proposed for the closed-set graph domain adaptation (CS-GDA). All of them share the same idea, i.e., utilizing the adversarial domain alignment to mitigate the domain divergence between source and target graphs. Adversarial domain adaptation with graph convolutional networks (AdaGCN) (Dai et al. 2022) utilizes GCN and adversarial domain adaptation to model both graph structures and node attributes. Unsupervised domain adaptive graph convolutional networks (UDAGCN) (Wu et al. 2020) proposes a dual GCN to simultaneously leverage local and global consistency to aggregate features. Adversarial Separation Network (ASN) (Zhang et al. 2021) utilizes an adversarial separation network to explicitly distinguish domain-private and domain-shared information by introducing a shared encoder along with two private encoders. However, all of these methods require the source and target graphs to share the same label space, which is not practical in real-world applications.

Open-Set Domain Adaptation

In early open-set domain adaptation (OS-DA) definition (Panareda Busto and Gall 2017), both source and target domains have private label spaces, respectively, and the common label space is known. Later, the setting of OS-DA (Saito et al. 2018) is adjusted by claiming no source private label space, which means target label space contains source label space. In other words, the source label space is the subset of the target label space. The goal of OS-DA is to learn a model with source and target domains that can not only correctly classify target data if it belongs to source label space but also successfully identify target data from unknown classes. Recent OS-DA methods (Liu et al. 2019; Bucci, Loghmani, and Tommasi 2020; Zhu and Li 2021) mainly focus on the later challenging setting. Among recent OS-DA methods, OSBP (Saito et al. 2018) and DANCE (Saito et al. 2020) have shown superior performances and inspired many related works. OSBP proposes an adversarial learning method that enables feature generation that can separate unknown target samples from known target samples (Saito et al. 2018). Later, DANCE proposes a neighborhood clustering strategy that performs unsupervised clustering of target samples to its neighbor and gives a threshold to divide target samples into common-class group and private-class group (Saito et al. 2020). However, DANCE aligns every target sample to their neighbor samples could lead to more inaccurate predictions within a local target cluster when the cluster has inaccurate predictions.

Our Approach

Unlike the closed-set graph domain adaptation (CS-GDA) problem, we target at open-set problem in graph domain adaptation where the target domain contains categories that do not belong to source label space. Compared with CS-GDA, open-set graph domain adaptation (OS-GDA) is more challenging and practical, since we cannot always guarantee all target nodes in the source label space. As a result, the task of OS-GDA is two-fold: both to align distributions of source and target domains, and to identify target nodes that are out of source label space. Here, we propose a novel separate domain alignment (SDA) scheme to address the OS-GDA problem. Fig. 3 illustrates the overall framework.

Our framework SDA contains two novel contributions for graph domain adaptation: 1) target domain separation, and 2) neighbor center clustering. The first part helps to dynamically split target nodes into certain and uncertain groups through entropy value, while the second part aims to refine the coarsely divided unknown group by pushing these nodes close to their neighbor centers. Our method provides novelty and overcomes some issues commonly found in current OS-DA methods such as OSBP and DANCE.

In this section, we first introduce the preliminaries. Then, we recap the details of local and global node embedding. Last, we illustrate our newly proposed SDA and summarize the overall pipeline of our framework.

Preliminaries

In the OS-GDA problem, we focus on the graph node classification task. Let $\mathcal{G}$ be an undirected graph with node set V and edge set $E \subseteq V \times V$. $N_v = |V|$ and $N_e = |E|$ denote the number of nodes and edges in the graph. Let $\mathbf{A} \in \mathbb{R}^{N_v \times N_v}$ be the adjacency matrix of $\mathcal{G}$, where each element $\mathbf{A}_{ij} = \mathbf{A}(i,j)$ indicates the connectivity of node v_i and node v_j. $\mathbf{A}_{i,j} = 1$ if edge $(v_i, v_j) \in E$, otherwise $\mathbf{A}_{i,j} = 0$. $X \in \mathbb{R}^{N_v \times d}$ represents the content features of V where d is the feature dimension. Y is the label set for V that comes from label space C.

Source Graph: Let $\mathcal{G}_s = (V_s, E_s, \mathbf{A}_s, \mathbf{X}_s, Y_s)$ indicate the labeled source network with node set V_s, edge set E_s, and label matrix Y_s with label space C_s.

Target Graph: Let $\mathcal{G}_t = (V_t, E_t, \mathbf{A}_t, \mathbf{X}_t)$ be the unlabeled target network with unlabeled node set V_t and edge set E_t. The target label space is denoted as C_t.

Open-Set Graph Domain Adaptation: Different from CS-GDA which requires source and target graphs to share the same label space, i.e., $C_s = C_t$, OS-GDA relaxes this claim by allowing target graph to contain nodes from classes out of source label space, i.e., $C_s \subset C_t$. Specifically, we denote the known label space shared by both domains as $C = C_s \cap C_t = C_s$ and the unknown label space for target graph as $\bar{C}_t = C_t \setminus C_s$. The goal of OS-GDA is to train the model with $\mathcal{G}_s$ and $\mathcal{G}_t$, classify target nodes into $|C|+1$ categories, where $\bar{C}_t$ are gathered as one unknown class, and require the learned model to classify the target node correctly if it is associated with a label in C, or identify it as "unknown" otherwise. In general, the model consists of three modules, i.e., G, F, and O. Here, $G : x \rightarrow g$ represents the graph feature extractor that maps the content feature of node x into an embedding space, $F : g \rightarrow f$ is the classifier using input embedding to predict the category, and domain discriminator O is for adversarial domain alignment.

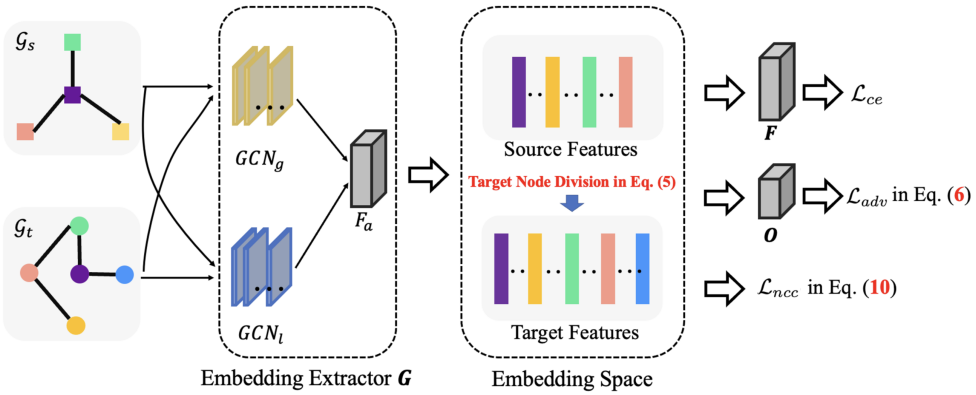


Figure 3: Illustration of our novel Separate Domain Alignment (SDA) scheme for open-set graph domain adaptation, which includes three losses: cross-entropy loss $\mathcal{L}_{CE}$, domain alignment loss $\mathcal{L}_{DA}$, and neighbor center clustering loss $\mathcal{L}_{NCC}$, and three modules: feature extractor $G = \{GCN_g, GCN_l, F_a\}$, classifier F, and domain discriminator O.

Recap of Local and Global Node Embedding

Many methods (Xu et al. 2021; Zhang et al. 2020, 2021; Wu et al. 2020) have proven the advantage of combining local and global graph information to learn the semantic node embedding. The core idea is to both recover the local 1-hop neighbor information and extract the global topological features from the given graph. Furthermore, the attention mechanism is adopted to aggregate the local and global embeddings to capture the semantic information from both aspects.

Local GCN (GCN_l) To capture the local information in a graph, We directly utilize the GCN model proposed by (Kipf and Welling 2017) and formulate the local GCN GCN_l as a type of feed-forward neural networks. Give the graph $\mathcal{G} = (V, E, \mathbf{A}, \mathbf{X}, Y)$, the output of the i-th hidden layer $\mathbf{Z}_l^{(i)}$ of the GCN_l is defined as:

$$\mathbf{Z}_l^{(i)}(\mathbf{X}) = \sigma(\tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-\frac{1}{2}}\mathbf{Z}_l^{(i-1)}\mathbf{W}_l^{(i)}), \tag{1}$$

where $\sigma(\cdot)$ is the activation function, $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}_N$ denotes the adjacency matrix with self-loops ($\mathbf{I}_N$ is an identity matrix), $\tilde{\mathbf{D}}_{ii} = \sum_j \tilde{\mathbf{A}}_{ij}$ and $\tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-\frac{1}{2}}$ is the symmetric normalized adjacency matrix, $\mathbf{Z}_l^{(i-1)}$ indicates the output of the $(i-1)$-th layer and $\mathbf{Z}_l^0 = \mathbf{X}$, and $\mathbf{W}_l^{(i)}$ represents the learnable parameters of the i-th layer.

Global GCN (GCN_g) To excavate the global topological features, we introduce the PPMI-based GCN (Zhuang and Ma 2018), leveraging the PPMI matrix $\mathbf{P}$ to assess the topological proximity between nodes within a given graph $\mathcal{G}$ over k steps. Please refer to (Zhuang and Ma 2018) for more details of $\mathbf{P}$.

Utilizing the calculated matrix $\mathbf{P}$, we formulate the global GCN GCN_g as a type of feed-forward neural networks, which is defined as follows:

$$\mathbf{Z}_g^{(i)}(\mathbf{X}) = \sigma(\mathbf{D}^{-\frac{1}{2}}\mathbf{P}\mathbf{D}^{-\frac{1}{2}}\mathbf{Z}_g^{(i-1)}\mathbf{W}_g^{(i)}), \tag{2}$$

where $\sigma(\cdot)$ is the activation function, $\mathbf{P}$ denotes the PPMI matrix, $\mathbf{D}_{ii} = \sum_j \mathbf{P}_{ij}$ is the normalized matrix, $\mathbf{Z}_g^{(i-1)}$ represents the output of the $(i-1)$-th layer and $\mathbf{Z}_g^{(0)} = \mathbf{X}$, and $\mathbf{W}_g^{(i)}$ is the trainable parameters of the i-th layer.

Embedding Attention (F_a) To further excavate the contribution of both embeddings, i.e., local embedding $\mathbf{Z}_l$ and global embedding $\mathbf{Z}_g$, and generate a unified node embedding space, an attention layer F_a is introduced. α_1 and α_2 are from the self-attention layer F_a with the input of concatenated Z_l and Z_g. F_a takes $\mathbf{Z}_l$ and $\mathbf{Z}_g$ as input and produces weight coefficients α_1 and α_2 for $\mathbf{Z}_l$ and $\mathbf{Z}_g$, respectively:

$$[\alpha_1, \alpha_2] = F_a([\mathbf{Z}_l, \mathbf{Z}_g]). \tag{3}$$

The unified node embedding is the combination of local and global embeddings with their corresponding weight coefficients:

$$\mathbf{Z} = \frac{\exp(\alpha_1)}{\exp(\alpha_1) + \exp(\alpha_2)}\mathbf{Z}_l + \frac{\exp(\alpha_2)}{\exp(\alpha_1) + \exp(\alpha_2)}\mathbf{Z}_g. \tag{4}$$

Fig. 3 illustrates the details of how to extract the node embedding. For simplicity, we use $G = \{GCN_l, GCN_g, F_a\}$ to denote the node embedding extractor.

Separate Domain Alignment

Previous CS-GDA methods (Wu et al. 2020; Zhang et al. 2021) utilize adversarial learning to align source and target graphs and minimize the entropy value of target nodes to traverse low-density regions within the target embedding space. As a result, these methods cannot be applied to our proposed OS-GDA problem. Without considering the target nodes from unknown label space $\bar{C}_t$, directly aligning source and target domains would lead to a negative knowledge transfer.

To solve the problem, we propose a novel separate domain alignment (SDA) scheme that enables the model to align known-class target nodes to source nodes while separating unknown-class target nodes from known-class target nodes. SDA consists of two alignment operations, i.e., adversarial domain alignment and neighbor center clustering, for different target nodes.

Inspired by (Grandvalet and Bengio 2004), we employ entropy as a measure to assess the uncertainty of classifier predictions. A lower entropy value indicates a higher level of prediction certainty. Target nodes that have lower entropy values are more inclined to belong to the known label space C, whereas those with higher entropy are more likely to come from the unknown label space $\bar{C}_t$. We employ the threshold γ to adaptively divide the target nodes into two groups, i.e, certain group and uncertain group, by utilizing the entropy $e_t = H(f_t)$ where $H(f_t) = -\sum_{i=1}^{|C_s|} f_t^i \log(f_t^i)$, as the follows:

$$x_t \in \begin{cases} Group_c, & e_t < \gamma \\ Group_u, & e_t \geq \gamma. \end{cases} \tag{5}$$

Instead of tuning the hyper-parameter γ to divide target nodes, we set γ to $\frac{\log(|C_s|)}{2}$ where $|C_s|$ denotes the number of source classes and $\log(|C_s|)$ is the maximum entropy value of the classifier.

Adversarial Domain Alignment for Certain Group For target nodes from certain group $Group_c$, there is a high probability that these nodes belong to known label space C. We utilize adversarial domain alignment to learn domain invariant embedding for nodes from C. Specifically, we employ a domain discriminator $O(\cdot)$ to play a minimax game with embedding extractor $G(\cdot)$. The objective function is:

$$\mathcal{L}_{adv} = \mathbb{E}_{x_s \in X_s} \log(O(G(x_s)) + \mathbb{E}_{x_t \in Group_c} \log(1 - O(G(x_t)). \tag{6}$$

The domain discriminator tries to identify the source and target nodes while the embedding extractor aims to fool the domain discriminator. The overall process is:

$$\min_O \max_G \mathcal{L}_{adv}. \tag{7}$$

Neighbor Center Clustering for Uncertain Group During model training, for the target nodes from uncertain group $Group_u$, directly utilizing adversarial learning to align them to source domain could cause negative transfer, because $Group_u$ may contain target nodes from both known and unknown classes. Violently enforcing target unknown class nodes to align with source nodes will deteriorate the learned domain-invariant embedding. To address this challenge, we exploit a novel neighbor center clustering (NCC) to better identify target nodes from known class and those from unknown class while softly aligning target nodes to source nodes. The main idea of our NCC is to move each target node in $Group_u$ either to source class centers or to cluster centers in $Group_u$. The target unknown class nodes are more likely to share similar semantic information with the centers which are close to ground truth unknown class centers. Likewise, those from known class would possibly have similar characteristics with the known class centers.

Given the uncertain group $Group_u$, we utilize K-means to group them into K clusters and obtain corresponding embedding centers $\{\mu_t^1, ..., \mu_t^K\}$. Meanwhile, we utilize the weight vectors $\mathbf{W}_f = [w_f^1, ..., w_f^{|C_s|}]$ in the classifier F as source class centers. Let $\mathbf{M}$ represents the center matrix which consists of cluster centers from $Group_u$ and source class centers:

$$\mathbf{M} = [\mu_t^1, ..., \mu_t^K, w_f^1, ..., w_f^{|C_s|}], \tag{8}$$

where both μ_t^i and w_f^j are L2-normalized.

Although our NCC module is inspired by DANCE, we would like to point out two essential differences: (1) unlike DANCE which aligns every target sample to their neighbor samples, we require target data to align to their neighbor cluster centers. (2) DANCE does not consider different target data samples to have different alignments, while we proposed two different alignment strategies for target data from different groups. DANCE can be restricted due to its sole neighborhood alignment and fail to identify unknown class, while our method will not have this disadvantage.

Another thing to note is that although $Group_u$ contains both known and unknown classes, the majority of data in $Group_u$ belong to unknown class due to threshold γ. Thus, $\mathbf{M}$ is mainly used to cluster unknown class data; meanwhile, to avoid negative clustering for known classes data (a minority of data in $Group_u$), we add the source class centers in $\mathbf{M}$. To better discriminate between known and unknown data, we propose Eq. (10). Hopefully, after the model converges, using threshold γ could clearly identify known and unknown data.

Given a target node embedding g_t^i from $Group_u$ and the center matrix $\mathbf{M}$, the probability that the j-th center m_j in M is the neighbor center of g_t^i is,

$$p_{i,j} = \frac{\exp(\langle g_t^i, m_j \rangle / \tau)}{\sum_{k=1}^{K+|C_s|} \exp(\langle g_t^i, m_k \rangle / \tau)}, \tag{9}$$

where $\langle \cdot, \cdot \rangle$ denotes the inner product between two vectors to measure their similarity, τ denotes the temperature parameter which is empirically set as 0.05. Eventually, the neighbor center clustering loss is formulated as:

$$\mathcal{L}_{ncc} = -\sum_{i=1}^{N_{group_u}} \sum_{i=j}^{K+|C_s|} p_{i,j} \log(p_{i,j}), \tag{10}$$

where N_{group_u} is the number of target nodes in uncertain group $Group_u$. By minimizing the above loss, the learned target embedding space will be more discriminative and benefit the target unknown class identification.

Overall Objective

Our total loss objective comprises three components. Cross-entropy loss $\mathcal{L}_{ce}$ is applied for the source graph. Adversarial domain alignment loss $\mathcal{L}_{adv}$ and neighbor center clustering loss $\mathcal{L}_{ncc}$ in Eq. (6) and Eq. (10) are used for source and target graphs. Therefore, the overall loss function is:

$$\mathcal{L}_{SDA} = \mathcal{L}_{ce} + \mathcal{L}_{adv} + \beta \mathcal{L}_{ncc}, \tag{11}$$

where β is hyper-parameters for $\mathcal{L}_{ncc}$. Our method procedure is summarized in Algorithm 1.

Algorithm 1: Seperate Domain Alignment (SDA) Algorithm

Input: $\mathcal{G}_s$, $\mathcal{G}_t$, γ, β, K, initialized G, F, O.
Output: Learned G and F

1: **for** $epoch = 1$ to $epochs$ **do**
2: Extract node embedding by utilizing G.
3: Apply Eq. (5) for target node division.
4: Apply Eq. (7) for target certain group alignment.
5: Apply Eq. (10) for target uncertain group alignment.
6: **end for**
7: **return** G and F

Inference

In the testing phase, given each target node x_t with its classifier output f_t. If its entropy value is larger than the threshold γ, which is the same as the one in Eq. (5), x_t will be marked as unknown class. Otherwise, it will be assigned to a class in the source label space C_s depending on f_t.

Experiments

In this section, we evaluate the effectiveness of our method as the following: Firstly, we introduce the experimental settings. Next, we compare our method to other methods. Then, we provide an extensive ablative study investigating each of our proposed modules. Last, we present the hyper-parameters sensitivity study.

Experimental Settings

Datasets: We leverage three commonly used paper citation networks, i.e., **A**CMv9 (between years 2000 and 2010), **D**BLPv7 (between years 2004 and 2008), and **C**itationv1(before the year 2008), provided by Arnet-Miner (Tang et al. 2008) and construct graphs based on these citation networks. These networks come from three different original sources (ACM, DBLP, and Microsoft Academic Graph, respectively, suggested by the dataset names) with 5 categories. we utilize the same dataset processing techniques as (Zhang et al. 2021). We unify the dimension of attributes by following (Zhang et al. 2021) and consider them as undirected networks with each node indicating a paper and each edge denoting a citation relation between two nodes. The experiment is conducted under six domain adaptation tasks: A→D, D→A, A→C, C→A, C→D, D→C.

Protocols: We introduce an open set protocol for the experiment by setting the size of source label space C_s to 3 and keeping the size of target label space C_t as 5. For each task, we evaluate the method under every possible target unknown class label space, which includes 10 situations. The reported results are calculated by averaging over 10 runs with different target unknown classes label space.

Baselines: We compare our method with four main streams of the state-of-the-art methods: (1) *Graph Node Classification* methods, namely **GCN** and **GraphSAGE**. (2) *Unsupervised Domain Adaptation* methods, namely DANN (Ganin et al. 2016) and **CDAN** (Long et al. 2018b). (3) *Open-set Domain Adaptation* methods, namely **OSBP** (Saito et al. 2018) and **DANCE** (Saito et al. 2020). (4) *Closed-set Graph Domain Adaptation* methods, **UDAGCN** (Wu et al. 2020) and **ASN** (Zhang et al. 2021).

Evaluation Metrics: We use four metrics, i.e., average class accuracy over all classes (Acc), average class accuracy on known classes (Acc_k), average class accuracy on unknown class (Acc_u), and h-score (HS) (Fu et al. 2020), to evaluate the performance of all methods. The Acc is the mean of per-class accuracy over known and unknown classes, which would fail to truly discover the ability of unknown class identification for the methods. Due to its equal weighting of each known class accuracy and unknown class accuracy, it results in the dominance of known class accuracy in the overall accuracy Acc. Thus, we introduce the HS to address the importance of both Acc_k and Acc_u by computing their harmonic mean:

$$HS = \frac{2 \times Acc_k \times Acc_u}{Acc_k + Acc_u}. \tag{12}$$

HS value is high only when both Acc_k and Acc_u are high. In our experiment, we report the averaged results of 10 runs by enumerating every possible source label space C_s.

Implementation Details: Our implementation is based on Pytorch (Paszke et al. 2019). We utilize the same graph embedding extractor structure as ASN (Zhang et al. 2021) which includes local GCN GCN_l, global GCN GCN_g, and attention layer F_a. Both GCN_l and GCN_g are two-layer structures, the hidden dimensions for two layers in GCN_l and GCN_g are set as 128 and 16. The dropout rate is defined as 0.5. To ensure a fair comparison, identical dimensions are set for other baseline models. We optimize the model for 100 epochs by using Adam optimizer with learning rate of 0.005, momentum of 0.9, and weight decay of 5×10^{-4}. The hyper-parameters γ, τ and β are set as $\frac{\log(3)}{2}$, 0.05, and 0.05, respectively. The number of steps k in PPMI matrix for GCN_g is defined as 3, which is the same as ASN.

Results and Analysis

Quantitative comparisons are shown in Table 1 from the aspects of Acc and HS. The results of Acc_k and Acc_u are in supplementary material. We group the methods on top four rows compared with ours on the last row. On the first row, there are two graph node classification GNN methods, i.e., GCN and GraphSAGE. On the second row, there are two unsupervised domain adaptation methods: DANN and CDAN. On the third row, we present two state-of-the-art open-set domain adaptation methods: OSBP and DANCE. On the fourth row, we present two cutting-edge closed-set graph domain adaptation methods: UDAGCN and ASN.

In Table 1, we observe that our method consistently outperforms all other compared methods with a significant margin for all six OS-GDA tasks. One exception is for the task of D → A, where the accuracy for DANCE is slightly better than our SDA. The HS of our method SDA is the highest among all methods throughout the six tasks. For example, we get 4.92% better than ASN and 24.30% better than UDAGCN in terms of Acc. Checking HS, we see 8.39% and 27.68% performance gains compared with ASN and UDAGCN. Further, surprisingly the performances of GCN and GraphSAGE are approaching or surpassing UDAGCN

Methods	A → D		D → A		A → C		C → A		C → D		D → C		Average	
	Acc	*HS*	*Acc*	*HS*	*Acc*	*HS*	*Acc*	*HS*	*Acc*	*HS*	*Acc*	*HS*	*Acc*	*HS*
GCN	45.10	41.80	38.95	39.52	46.36	43.91	44.14	43.66	48.45	44.61	42.26	41.25	44.21	42.46
GraphSAGE	48.26	46.22	43.14	42.84	50.60	49.04	48.13	46.36	51.72	48.66	47.20	46.70	48.17	46.64
DANN	33.30	28.07	34.58	36.53	39.64	41.22	34.47	34.42	36.92	41.88	35.20	35.46	35.68	34.16
CDAN	31.13	21.65	29.03	27.76	30.99	26.00	31.72	30.91	35.69	30.47	28.62	21.82	31.20	26.44
OSBP	28.56	11.27	26.20	12.91	29.32	11.15	27.80	7.34	33.81	18.89	28.63	14.16	29.05	12.62
DANCE	60.54	25.99	**53.27**	39.53	63.23	39.15	60.44	35.88	64.29	28.98	57.62	39.50	59.90	34.84
UDAGCN	36.20	26.59	31.90	12.31	37.44	32.01	35.64	22.76	41.88	36.48	35.50	25.09	36.43	25.87
ASN	56.40	37.55	47.49	43.93	59.88	49.82	57.51	47.87	56.65	45.62	56.97	46.19	55.81	45.16
Ours (SDA)	**61.60**	**49.22**	51.36	**50.86**	**64.47**	**55.27**	**61.67**	**55.89**	**67.51**	**55.35**	**57.74**	**54.72**	**60.73**	**53.55**

Table 1: Results (%) on six open-set graph domain adaptation tasks in terms of Acc and HS.

Loss Objectives	Acc_k	Acc_u	Acc	HS
$\mathcal{L}_{ce}$	44.03	42.42	43.69	38.70
$\mathcal{L}_{ce} + \mathcal{L}_{ncc}$	49.45	48.74	49.27	45.67
$\mathcal{L}_{ce} + \mathcal{L}_{adv}$	48.32	54.04	49.69	48.34
$\mathcal{L}_{ce} + \mathcal{L}_{ncc} + \mathcal{L}_{adv}$ (Ours)	49.68	56.40	51.36	50.86

Table 2: Ablation study for SDA on D→A domain adaptation task. $\mathcal{L}_{ce}$ is the cross-entropy loss objective. $\mathcal{L}_{adv}$ represents the adversarial domain alignment loss objective. $\mathcal{L}_{ncc}$ denotes the neighbor center clustering loss objective.

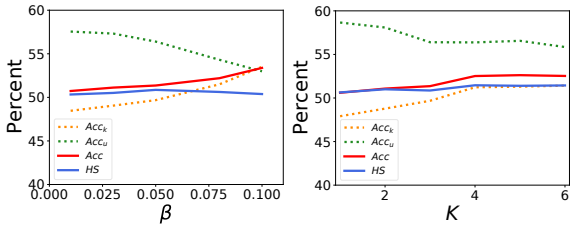
and ASN in terms of HS, which reveals the negative transfer problem among CS-GDA methods when target domain contains unknown classes. This phenomenon is caused by violently enforcing source and target data to align without considering the difference between their label spaces.

Ablation Study

We conduct ablation studies to examine the effectiveness of the proposed $\mathcal{L}_{SDA}$ in Eq. (11) and show the results in Table 2. Firstly, compared with $\mathcal{L}_{ce}$, both $\mathcal{L}_{ce} + \mathcal{L}_{ncc}$ and $\mathcal{L}_{ce} + \mathcal{L}_{adv}$ gain significant improvement over four evaluation metrics. Especially, We can see 6.97% and 9.64% performance gains in terms of HS, which proves the effectiveness of proposed domain alignment strategies for target nodes from different groups. Additionally, comparing "Ours" to $\mathcal{L}_{ce} + \mathcal{L}_{ncc}$ and $\mathcal{L}_{ce} + \mathcal{L}_{adv}$, we achieve another significant improvement over four evaluation metrics, which further verifies the power of our SDA. Overall, each of the incremental combination demonstrates the effectiveness of the components, i.e., the adversarial domain alignment loss $\mathcal{L}_{adv}$ for target nodes from certain group and neighbor center clustering loss $\mathcal{L}_{ncc}$ for target nodes from uncertain group, indicating that SDA possesses a significant advantage in addressing open-set graph domain adaptation challenges.

Hyper-parameter Sensitivity Study

We evaluate the sensitivity of hyper-parameters β in Eq. (11) and number of clusters K in target uncertain group, showing the performance of our method on domain adaptation task D→A in Fig. 4. β is selected from $\{0.01, 0.03, 0.05, 0.08, 0.10\}$, and K is picked from $\{1, 2, 3, 4, 5, 6\}$. As shown in Fig. 4(a), HS is relatively stable in the range $[0.01,\ 0.05]$ while slightly degraded in the range $[0.05, 0.10]$. Acc and Acc_k are continually increased by increasing the value of β while Acc_u is gradually decreased. Fig. 4(b) demonstrates the performance of our method under different number of clusters in target uncertain group, i.e., K. We also observe that the HS is slightly increased when increasing the value of K, Acc and Acc_k are increased, and Acc_u is gradually decreased. Overall, our method is less sensitive to these two hyper-parameters, i.e., β and K, from the aspects of Acc and HS.



(a) Varying β in Eq. (11) (b) Varying #clusters

Figure 4: (a) the impact of the value β, i.e., the hyper-parameter in Eq. (11), on domain adaptation task D→A. (b) the impact of the value K, i.e., the number of clusters in target uncertain group, on domain adaptation task D→A.

Conclusion

In this work, we propose to address a new and challenging problem, namely open-set graph domain adaptation (OS-GDA), where target graph is allowed to contain nodes that are out of source label space. A novel separate domain alignment (SDA) scheme is newly introduced to effectively resolve the open-set cross-network node classification problem. We jointly consider two different domain alignment strategies for different target nodes to sufficiently learn the well-aligned discriminative embedding space, which further improves the capability of the model on OS-GDA. Extensive experiments show that our method achieves significant performance gain over the state-of-the-art methods.

Acknowledgments

This work is supported in part by the U.S. Army Research Office Award under Grant Number W911NF-21-1-0109 and the National Science Foundation under Grant IIS-2316306.

References

Bhagat, S.; Cormode, G.; and Muthukrishnan, S. 2011. *Node Classification in Social Networks*, 115–148. Boston, MA: Springer US. ISBN 978-1-4419-8462-3.

Bhatia, K.; Dahiya, K.; Jain, H.; Kar, P.; Mittal, A.; Prabhu, Y.; and Varma, M. 2016. The extreme classification repository: Multi-label datasets and code.

Bucci, S.; Loghmani, M. R.; and Tommasi, T. 2020. On the Effectiveness of Image Rotation for Open Set Domain Adaptation. In *ECCV*, 422–438.

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*.

Fu, B.; Cao, Z.; Long, M.; and Wang, J. 2020. Learning to Detect Open Classes for Universal Domain Adaptation. In *ECCV*, 567–583.

Ganin, Y.; and Lempitsky, V. 2015. Unsupervised domain adaptation by backpropagation. In *ICML*, 1180–1189.

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.

Geng, C.; Huang, S.-j.; and Chen, S. 2020. Recent advances in open set recognition: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 43(10): 3614–3631.

Grandvalet, Y.; and Bengio, Y. 2004. Semi-supervised learning by entropy minimization. In *NeurIPS*, 529–536.

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. *NeurIPS*, 30.

Ji, P.; and Jin, J. 2016. Coauthorship and citation networks for statisticians. *The Annals of Applied Statistics*.

Jiang, J.; and Zhai, C. 2007. Instance weighting for domain adaptation in NLP. ACL.

Kipf, T. N.; and Welling, M. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *ICLR*.

Liu, H.; Cao, Z.; Long, M.; Wang, J.; and Yang, Q. 2019. Separate to adapt: Open set domain adaptation via progressive separation. In *CVPR*, 2927–2936.

Long, M.; Cao, Z.; Wang, J.; and Jordan, M. I. 2018a. Conditional adversarial domain adaptation. *NeurIPS*, 31.

Long, M.; Cao, Z.; Wang, J.; and Jordan, M. I. 2018b. Conditional adversarial domain adaptation. *NeurIPS*, 31.

Panareda Busto, P.; and Gall, J. 2017. Open set domain adaptation. In *ICCV*, 754–763.

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. 2019. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, volume 32.

Rezayi, S.; Zhao, H.; Zhu, R.; and Li, S. 2023. XDC: Adaptive Cross Domain Short Text Clustering. In *Proceedings of the 2023 SIAM International Conference on Data Mining (SDM)*, 927–935. SIAM.

Saito, K.; Kim, D.; Sclaroff, S.; and Saenko, K. 2020. Universal domain adaptation through self supervision. *NeurIPS*, 33: 16282–16292.

Saito, K.; Yamamoto, S.; Ushiku, Y.; and Harada, T. 2018. Open set domain adaptation by backpropagation. In *ECCV*, 153–168.

Scarselli, F.; Gori, M.; Tsoi, A. C.; Hagenbuchner, M.; and Monfardini, G. 2009. The Graph Neural Network Model. *IEEE Transactions on Neural Networks*, 20(1): 61–80.

Shen, X.; Dai, Q.; Mao, S.; Chung, F.-L.; and Choi, K.-S. 2021. Network Together: Node Classification via Cross-Network Deep Network Embedding. *IEEE Transactions on Neural Networks and Learning Systems*, 32(5): 1935–1948.

Shi, W.; Zhu, R.; and Li, S. 2022. Pairwise adversarial training for unsupervised class-imbalanced domain adaptation. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 1598–1606.

Sun, B.; Feng, J.; and Saenko, K. 2016. Return of frustratingly easy domain adaptation. In *AAAI*, volume 30.

Szklarczyk, D.; Gable, A. L.; Lyon, D.; Junge, A.; Wyder, S.; Huerta-Cepas, J.; Simonovic, M.; Doncheva, N. T.; Morris, J. H.; Bork, P.; Jensen, L. J.; and Mering, C. 2018. STRING v11: protein–protein association networks with increased coverage, supporting functional discovery in genome-wide experimental datasets. *Nucleic Acids Research*, 47(D1): D607–D613.

Tang, J.; Zhang, J.; Yao, L.; Li, J.; Zhang, L.; and Su, Z. 2008. ArnetMiner: Extraction and Mining of Academic Social Networks. In *KDD*, KDD '08, 990–998. New York, NY, USA: Association for Computing Machinery. ISBN 9781605581934.

Wu, M.; Pan, S.; Zhou, C.; Chang, X.; and Zhu, X. 2020. Unsupervised Domain Adaptive Graph Convolutional Networks. In *WWW*, WWW '20, 1457–1467. New York, NY, USA: Association for Computing Machinery. ISBN 9781450370233.

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

Xu, M.; Wang, H.; Ni, B.; Guo, H.; and Tang, J. 2021. Self-supervised graph-level representation learning with local and global structure. In *ICML*, 11548–11558. PMLR.

Zhang, X.; Du, Y.; Xie, R.; and Wang, C. 2021. Adversarial Separation Network for Cross-Network Node Classification. In *CIKM*, 2618–2626.

Zhang, Y.; Deng, W.; Wang, M.; Hu, J.; Li, X.; Zhao, D.; and Wen, D. 2020. Global-local gcn: Large-scale label noise cleansing for face recognition. In *CVPR*, 7731–7740.

Zhu, R.; Guo, D.; Qi, D.; Chu, Z.; Yu, X.; and Li, S. 2023. Trustworthy Representation Learning Across Domains. *arXiv preprint arXiv:2308.12315*.

Zhu, R.; Jiang, X.; Lu, J.; and Li, S. 2021a. Cross-domain graph convolutions for adversarial unsupervised domain adaptation. *IEEE Transactions on Neural Networks and Learning Systems*.

Zhu, R.; Jiang, X.; Lu, J.; and Li, S. 2021b. Transferable feature learning on graphs across visual domains. In *2021 IEEE International Conference on Multimedia and Expo (ICME)*, 1–6. IEEE.

Zhu, R.; and Li, S. 2021. Self-supervised universal domain adaptation with adaptive memory separation. In *2021 IEEE International Conference on Data Mining (ICDM)*, 1547–1552. IEEE.

Zhu, R.; and Li, S. 2022. Self-supervision based semantic alignment for unsupervised domain adaptation. In *Proceedings of the 2022 SIAM International Conference on Data Mining (SDM)*, 1–9. SIAM.

Zhu, R.; Sang, G.; and Zhao, Q. 2016. Discriminative feature adaptation for cross-domain facial expression recognition. In *2016 International Conference on Biometrics (ICB)*, 1–7. IEEE.

Zhu, R.; Tao, Z.; Li, Y.; and Li, S. 2021c. Automated graph learning via population based self-tuning GCN. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2096–2100.

Zhu, R.; Zhang, T.; Zhao, Q.; and Wu, Z. 2015. A transfer learning approach to cross-database facial expression recognition. In *2015 International Conference on Biometrics (ICB)*, 293–298. IEEE.

Zhuang, C.; and Ma, Q. 2018. Dual graph convolutional networks for graph-based semi-supervised classification. In *WWW*, 499–508.