PROTOTYPE-GUIDED TOPOLOGY AUGMENTATION FOR EDGE-IMBALANCED GRAPH DOMAIN ADAPTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Graph Domain Adaptation (GDA) transfers knowledge from labeled source graphs to unlabeled target graphs, alleviating label scarcity in real-world graph data. Class imbalance is prevalent in graphs and can lead to biased knowledge transfer in GDA. However, existing methods for class-imbalanced GDA primarily focus on imbalanced number of nodes across source classes, while overlooking edge imbalance. Empirically, we observe pronounced intra-domain edge imbalance and substantial differences in edge distributions across domains. Moreover, we reveal that larger cross-domain edge-distribution discrepancies are associated with lower GDA performance. Inspired by these findings, we propose ProTA, a plug-and-play topology augmentation framework for edge-imbalanced GDA. ProTA constructs topology-aware source prototypes through class-preserving diffusion to guide candidate edge selection. Augmentation budgets are allocated according to intra-domain edge deficits and cross-domain edge-distribution discrepancies. The selected edges strengthen relational support for minority-class nodes in both domains, reducing intra-domain edge imbalance and cross-domain distribution discrepancies. Experiments on 44 transfer tasks across four benchmarks demonstrate that ProTA improves the performance of six representative GDA models.

1 INTRODUCTION

Large-scale graphs with node attributes and topological structure provide rich contexts for numerous real-world applications, including citation networks(Lachaud et al., 2022), social and knowledge graphs(Ga et al., 2025; Liu et al., 2025). However, such graph data often suffer from label scarcity problems, since annotating complex structured data remains costly and difficult. (Cai et al., 2024). To tackle this challenge, graph domain adaptation (GDA) has emerged as a promising solution to transfer knowledge from a labeled source graph to an unlabeled target graph (Chen et al., 2019).

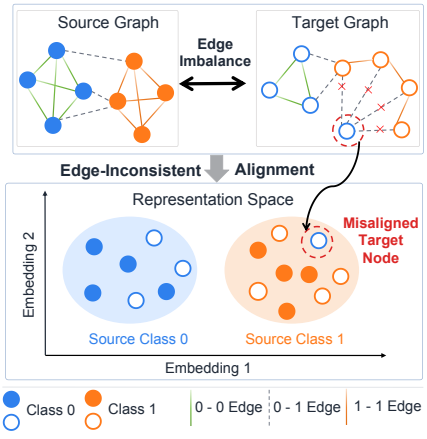
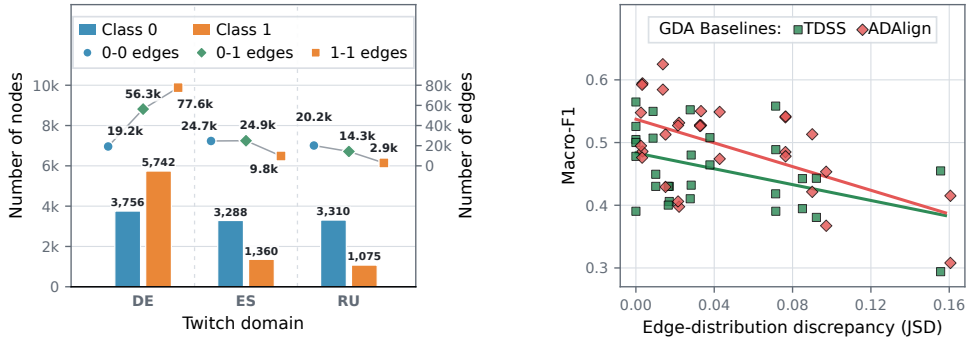


Figure 1: Illustration of misalignment induced by edge imbalance in GDA under balanced numbers of nodes across class.

Early works on GDA often assume a balanced label distribution in source graph (Wu et al., 2020; Dai et al., 2022; Wu et al., 2023; You et al., 2023), but class imbalance is pervasive in real-world graphs (Ma et al., 2025c; Pu et al., 2025). To address this issue, several recent studies consider source-domain class imbalance during knowledge transfer from source domain to target domain. Although these methods (Ma et al., 2025a;b) mitigate bias induced by node imbalance (i.e., unequal numbers of nodes across classes), they overlook edge imbalance both in source domain and target domain. As illustrated in Figure 1, imbalanced class-pair edge may distort neighborhood aggregation and shift node representations, resulting in aligning target nodes with incorrect source classes. Therefore, edge imbalance represents a distinct yet overlooked challenge to effective alignment in GDA.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.



(a) Graph imbalance across Twitch domains

(b) GDA performance and JSD

Figure 2: This presents node and edge distributions across three Twitch domains: DE, ES, and RU, together with Macro-F1 scores of TDSS and ADAlign across 30 transfer tasks. It shows pronounced node and intra-domain edge imbalance, substantial cross-domain discrepancies in edge distributions, and a negative association between cross-domain edge-distribution discrepancy and GDA performance, measured by Jensen–Shannon divergence (JSD)[1].

To verify our argument, we investigate node imbalance and edge imbalance on Twitch benchmark and empirically reveal the association between edge imbalance and GDA performance, as shown in Figure 2. We can conclude that (1) edge imbalance consisits of intra-domain edge imbalance and cross-domain edge distribution discrepancies; (2) source graph and target graph are characterized by pronounced node and edge imbalance, which capture distinct dimensions of graph imbalance namely class composition and topological structure respectively; (3) GDA performance exhibits a negative correlation with edge distribution discrepancy cross domain.

Motivated by this observation, we propose Prototype-guided Topology Augmentation (ProTA), a plug-and-play framework for edge-imbalanced GDA, as shown in Figure 3. ProTA constructs topology-aware source prototypes through multi-hop diffusion to guide minority-class nodes to connect to reliable candidate edges. It further allocates class-pair-specific augmentation budgets according to intra-domain edge deficits and cross-domain edge-distribution discrepancies, thereby to strengthen minority-class connectivity within each domain and balance class-pair edge ratios across domains. The augmented source and target graphs can be directly integrated with existing GDA backbones to mitigate cross-domain misalignment induced edge imbalance.

Our main contributions are summarized as follows:

- We first highlight the significance of edge imbalance in GDA and reveal that it manifests at two levels: intra-domain edge imbalance and cross-domain edge-distribution discrepancy.
- Motivated by our empirical findings, we propose ProTA, a plug-and-play topology augmentation framework that addresses edge imbalance in GDA by rebalancing class-pair edge distributions within domains and aligning them across domains.
- Extensive experiments on four benchmarks with six representative GDA backbones across 44 transfer tasks demonstrate the broad applicability and effectiveness and ProTA.

2 Related Work

Class-Imbalanced Graph Learning. Class-imbalanced graph learning aims to mitigate learning bias caused by uneven class distributions within a graph. Existing methods employ data augmentation, loss reweighting, or topology-aware correction to improve the representation of minority classes (Zhao et al., 2021; Park et al., 2022; Chen et al., 2021; Song et al., 2022; Zhao et al., 2022; Liu et al., 2024c). Although these methods address imbalanced numbers of nodes across

[1]Jensen–Shannon divergence (JSD) is a symmetric and bounded measure of the discrepancy between two probability distributions.

classes and topology-related biases within individual graphs, they primarily focus on single-graph settings and do not account for cross-domain edge-distributions discrepancies.

Graph Domain Adaptation. GDA transfers knowledge from a labeled source graph to an unlabeled target graph by mitigating distribution shifts across domains (Wu et al., 2020; Shi et al., 2025). Early methods learn domain-invariant node representations through discrepancy minimization or adversarial training (Zhang et al., 2019; Shen et al., 2020; Wu et al., 2020; Dai et al., 2022). Recent studies further account for graph-specific distribution shifts involving spectral, structural, conditional, and homophily information, while ADAlign develops scenario-adaptive alignment strategies for different transfer settings (You et al., 2023; Wu et al., 2023; Liu et al., 2023; 2024b; Yang et al., 2025a; Fang et al., 2025a; Chen et al., 2026a). However, these methods primarily address cross-domain distribution shifts and neglect class imbalance in real-world graphs.

Class-Imbalanced Graph Domain Adaptation. Several studies have started to explore class-imbalance learning in GDA. PALA and ImGDA address biased knowledge transfer from class-imbalanced source graphs through prototype learning and class-aware contrastive adaptation, respectively (Ma et al., 2025a;b). However, they primarily focus on imbalanced nodes while overlooking imbalanced edge. ProTA extends this line of research to account for both intra-domain edge deficits and cross-domain edge-distribution discrepancies when augmenting the topology around minority-class nodes. (Full related work is provided in Appendix D.)

3 Problem Definition

3.1 Graph Domain Adaptation Setting

The standard unsupervised GDA setting consists of a labeled source graph $\mathcal{G}^s = (\mathcal{V}^s, \mathcal{E}^s, \mathbf{A}^s, \mathbf{X}^s, \mathbf{Y}^s)$ and an unlabeled target graph $\mathcal{G}^t = (\mathcal{V}^t, \mathcal{E}^t, \mathbf{A}^t, \mathbf{X}^t)$. For each domain $d \in \{s, t\}$, $\mathcal{V}^d$ and $\mathcal{E}^d$ denote the sets of nodes and edges respectively, and $N_d = |\mathcal{V}^d|$ represents the number of nodes. The adjacency matrix and node feature matrix of domain d are denoted by $\mathbf{A}^d = [a^d_{ij}] \in \{0,1\}^{N_d \times N_d}$ and $\mathbf{X}^d \in \mathbb{R}^{N_d \times F}$, respectively. Here, $a^d_{ij} = 1$ if $(i,j) \in \mathcal{E}^d$ and $a^d_{ij} = 0$ otherwise, while F denotes the node feature dimension. The source label vector is denoted by $\mathbf{Y}^s = (y^s_1, \ldots, y^s_{N_s})^\top$, where $y^s_i \in \mathcal{C}$ is the label of source node i. The source and target domains share the label space $\mathcal{C} = \{1, \ldots, C\}$, where C denotes the number of classes. Using $\mathcal{G}^s$ and $\mathcal{G}^t$, GDA aims to learn a transferable model that accurately predicts target node labels despite distribution shifts in node attributes and graph structure, without access to target labels during training (Shen et al., 2020; Wu et al., 2023; Yang et al., 2024).

3.2 Edge Imbalance in Graph Domains

Edge imbalance manifests at two levels: intra-domain edge imbalance and cross-domain edge-distribution discrepancy. For undirected graphs, let $\mathcal{R} = \{(c, r) \in \mathcal{C} \times \mathcal{C} \mid c \leq r\}$ denotes the set of class-pair edge. For domain $d \in \{s, t\}$, let $\boldsymbol{\pi}^d = (\pi^d_{cr})_{(c,r)\in\mathcal{R}}$ denote the global class-pair edge distribution in domain d, where π^d_{cr} is the proportion of edges between classes c and r.

Intra-domain edge imbalance of domain d is quantified by Jensen–Shannon divergence (JSD) (Lin, 1991):

$$I^d_{\text{edge}} = \text{JSD}\left(\boldsymbol{\pi}^d \| \mathbf{u}_{\mathcal{R}}\right). \tag{1}$$

where $\mathbf{u}_{\mathcal{R}} = (1/|\mathcal{R}|)_{(c,r)\in\mathcal{R}}$ is the uniform distribution over all class-pair edge types. Cross-domain edge distribution discrepancy is quantified as

$$D^{s,t}_{\text{edge}} = \text{JSD}\left(\boldsymbol{\pi}^s \| \boldsymbol{\pi}^t\right). \tag{2}$$

4 THE PROPOSED METHODOLOGY

We propose Prototype-Guided Topology Augmentation (ProTA) to address edge imbalance at two levels in GDA, including intra-domain deficits in connectivity of minority-class nodes and cross-domain discrepancies in class-pair edge distributions. As shown in Figure 3, ProTA augments

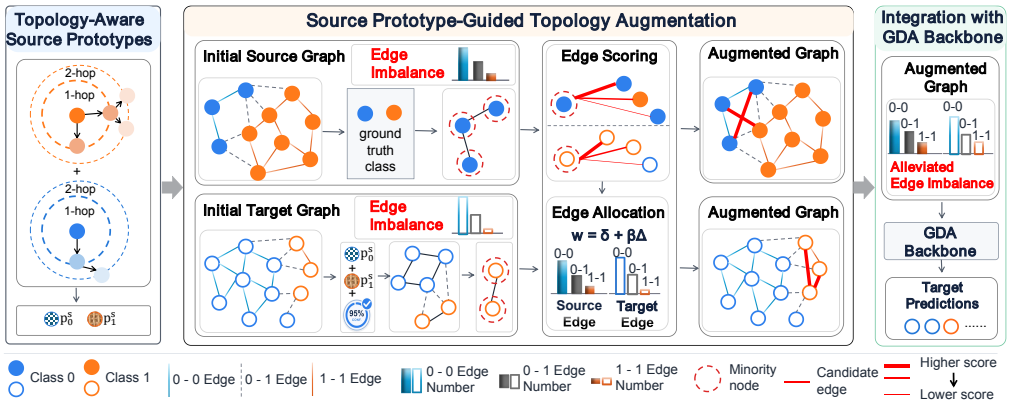


Figure 3: An Overview of ProTA. ProTA uses topology-aware source prototypes to guide topology augmentation, strengthening relational support for minority-class nodes within each domain and balancing edge-ratio distributions across domains before integration with existing GDA backbones.

the topology of source graph and target graph around minority-class nodes. For intra-domain edge imbalance, ProTA adds informative edges to minority-class nodes to compensate for their insufficient relational support. For cross-domain edge-distribution discrepancies, ProTA allocates class-pair-specific augmentation budgets to balance edge-ratio distributions across domains.

4.1 Topology-Aware Source Prototypes via Diffusion

Class-Preserving Graph Construction. To capture the characteristics of each source-domain class, ProTA constructs class prototypes that incorporate both node attributes and topological structure. Neighborhood aggregation on the original graph may introduce cross-class information into the prototypes, reducing their discriminability(Klicpera et al., 2019; Lim et al., 2021), obscuring the class-specific characteristics that the prototypes are intended to capture. Therefore, we construct a class-preserving source graph by retaining only same-class edges. Its adjacency matrix is defined as

$$\widetilde{\mathbf{A}}_y^s = \left[(a_{ij}^s + \delta_{ij})\mathbb{I}(y_i^s = y_j^s)\right]_{i,j=1}^{N_s} \in \mathbb{R}^{N_s \times N_s}. \tag{3}$$

Here, a_{ij}^s is the (i,j)-th entry of the source adjacency matrix, δ_{ij} is the Kronecker delta that adds self-loops, and $\mathbb{I}(\cdot)$ is the indicator function. Normalizing $\widetilde{\mathbf{A}}_y^s$ yields the propagation matrix $\widehat{\mathbf{A}}_y^s$. Details are provided in Appendix B.1.

Graph Diffusion Mechanism. Real-world graphs may contain noisy node attributes and edges attributes, so ProTA applies graph diffusion mechanism to smooth node attributes over class-preserving source graph(Gasteiger et al., 2019). Starting from $\mathbf{H}_{(0)}^s = \mathbf{X}^s$, source-node representations are updated using $\widehat{\mathbf{A}}_y^s$ as the propagation operator (Klicpera et al., 2019):

$$\mathbf{H}_{(k)}^s = \alpha \mathbf{X}^s + (1-\alpha)\widehat{\mathbf{A}}_y^s \mathbf{H}_{(k-1)}^s, \qquad k = 1, \ldots, K. \tag{4}$$

Parameter $\alpha \in (0,1)$ balances the contribution of original node attributes and information propagated within each class. Unrolling Eq. 4 reveals the contributions from different propagation depths:

$$\mathbf{H}_{(K)}^s = \alpha \sum_{r=0}^{K-1} (1-\alpha)^r (\widehat{\mathbf{A}}_y^s)^r \mathbf{X}^s + (1-\alpha)^K (\widehat{\mathbf{A}}_y^s)^K \mathbf{X}^s. \tag{5}$$

Each term $(\widehat{\mathbf{A}}_y^s)^r \mathbf{X}^s$ combines node attributes with structural context through r steps of intra-class propagation. The weights $\alpha(1-\alpha)^r$ decay geometrically with propagation depth for $r = 0, \ldots, K-1$.

Source Prototype Construction. For each class c, the diffused representations of its source nodes are averaged and normalized to obtain topology-aware prototype $\mathbf{p}_c^s$. The detailed calculation is provided in Appendix B.2.

4.2 SOURCE PROTOTYPE-GUIDED TOPOLOGY AUGMENTATION

Minority-Class Identification. To strengthen relational support for minority-class nodes, minority classes in each domain are first identified as augmentation targets. For domain $d \in \{s, t\}$, the minority-class set is defined as

$$\mathcal{C}_{\min}^{d} = \left\{ c \in \mathcal{C} \;\middle|\; n_c^d < \xi \max_{r \in \mathcal{C}} n_r^d \right\} \tag{6}$$

where n_c^d denotes the number of class-c nodes in domain d, and ξ is the minority-class threshold.

Prior to topology augmentation, we train the adopted GDA backbone on the original source and target graphs for a small number of epochs to generate pseudo-labels for target-domain nodes (Zhang et al., 2021). For each target node i, $p_{i,c}^t$ denotes the predicted probability of class c. The pseudo-label and prediction confidence are given by $\widehat{y}_i^t = \arg\max_{c \in \mathcal{C}} p_{i,c}^t$ and $q_i^t = \max_{c \in \mathcal{C}} p_{i,c}^t$.

As prediction confidence alone does not guarantee pseudo-label correctness, source prototypes provide an additional criterion for reliable target pseudo-label selection. Agreement with source prototypes is measured by the similarity margin between the predicted class and the most similar alternative class:

$$\mu_i^t = \operatorname{sim}\left(\mathbf{x}_i^t, \mathbf{p}_{\widehat{y}_i^t}^s\right) - \max_{c' \in \mathcal{C} \setminus \{\widehat{y}_i^t\}} \operatorname{sim}\left(\mathbf{x}_i^t, \mathbf{p}_{c'}^s\right) \tag{7}$$

where $\mathbf{x}_i^t$ and $\operatorname{sim}(\cdot,\cdot)$ denote target-node attributes vector and cosine similarity respectively. The set of target nodes with reliable pseudo-labels is defined as

$$\mathcal{V}_{\text{rel}}^t = \left\{ i \in \mathcal{V}^t \;\middle|\; q_i^t \geq \tau,\ \mu_i^t \geq \gamma \right\} \tag{8}$$

where τ and γ are the confidence and prototype-margin thresholds, respectively. $\widehat{n}_c^t$ denotes the number of nodes in $\mathcal{V}_{\text{rel}}^t$ assigned pseudo-label c. The target minority-class set is defined as $\mathcal{C}_{\min}^t = \{c \in \mathcal{C} \mid \widehat{n}_c^t < \xi \max_{r \in \mathcal{C}} \widehat{n}_r^t\}$. Source minority classes $\mathcal{C}_{\min}^s$ are identified by ground-truth class numbers and the same threshold criterion, as detailed in Appendix B.3.

Prototype-Guided Candidate Edge Scoring. For each identified minority-class node, candidate edges are scored by combining pairwise attribute similarity with class-level consistency measured against the source prototypes. For a target minority node i with pseudo-label $\widehat{y}_i^t = c$, candidate neighbors are drawn from the reliable target-node set $\mathcal{V}_{\text{rel}}^t$ and grouped by their pseudo-labels. For each candidate class $c' \in \mathcal{C}$, the candidate node set is defined as

$$\mathcal{Q}_{i,c'}^t = \left\{ j \in \mathcal{V}_{\text{rel}}^t \;\middle|\; \widehat{y}_j^t = c',\ (i,j) \notin \mathcal{E}^t,\ j \neq i \right\}. \tag{9}$$

For each candidate $j \in \mathcal{Q}_{i,c'}^t$, the edge score combines pairwise node attribute similarity with the consistency of both endpoints with their corresponding source prototypes:

$$s_{ij}^{t,c \to c'} = \lambda \operatorname{sim}\left(\mathbf{x}_i^t, \mathbf{x}_j^t\right) + \frac{1-\lambda}{2}\Big[\operatorname{sim}\left(\mathbf{x}_i^t, \mathbf{p}_c^s\right) + \operatorname{sim}\left(\mathbf{x}_j^t, \mathbf{p}_{c'}^s\right)\Big]. \tag{10}$$

The vectors $\mathbf{x}_i^t$ and $\mathbf{x}_j^t$ denote target-node attributes, and $\mathbf{p}_c^s$ and $\mathbf{p}_{c'}^s$ are the corresponding source class prototypes. The parameter $\lambda \in [0,1]$ balances node attribute similarity and prototype consistency. The source candidate edges are scored analogously by ground-truth labels in place of pseudo-labels, as detailed in Appendix B.4.

Discrepancy-Aware Edge Budget Allocation. While candidate-edge scoring determines which edges are more reliable, it does not determine how many edges should be added for each class-pair type. A uniform allocation may still leave underrepresented edge types insufficiently augmented and fail to reduce cross-domain differences in edge distributions. Therefore, augmentation budgets are allocated across class-pair types according to both intra-domain edge deficits and cross-domain edge-distribution discrepancies.

For a minority class c, the allocation weight for connections to class c' is

$$w_{c,c'}^d = \delta_{c,c'}^d + \beta \Delta_{c,c'}^d \tag{11}$$

where $\delta_{c,c'}^d$ and $\Delta_{c,c'}^d$ denote the intra-domain edge deficit and cross-domain edge-distribution discrepancy for class pair (c, c') in domain d respectively, and $\beta \geq 0$ controls the contribution

of cross-domain correction. The deficit definitions and budget allocation procedure are provided in Appendix B.5.

Given an augmentation budget of B edges per minority node, the budget assigned to class-c' candidates for a class-c node in domain d is

$$B^d_{c\to c'} = \mathrm{Round}\left(B\frac{w^d_{c,c'}}{\sum_{q\in\mathcal{C}} w^d_{c,q}}\right). \tag{12}$$

Within each candidate class, the highest-scoring candidates are selected up to the allocated budget, and the corresponding edges are added to the source graph or target graph.[2]

4.3 Integration with GDA Backbones

The augmented source graph $\widetilde{\mathcal{G}}^s$ and target graph $\widetilde{\mathcal{G}}^t$, are directly fed into the adopted GDA backbone without modifying its architecture or domain-alignment objective. The backbone retains its original training objective:

$$\mathcal{L} = \mathcal{L}^s_{\mathrm{cls}} + \lambda_{\mathrm{GDA}}\mathcal{L}_{\mathrm{GDA}} \tag{13}$$

where $\mathcal{L}^s_{\mathrm{cls}}$ is the supervised classification loss on labeled source nodes, $\mathcal{L}_{\mathrm{GDA}}$ is domain-adaptation objective of the backbone, and λ_{GDA} balances the source classification loss and the domain-adaptation objective.

5 Experiments

5.1 Experimental Setup

Datasets. We evaluate ProTA on four widely used GDA benchmarks: Citation, Twitch, Blog, and MAG. Citation contains three academic networks covering five research categories, including ACMv9 (A), Citationv1 (C), and DBLPv7 (D). Twitch consists of six regional social networks from Germany (DE), England (EN), Spain (ES), France (FR), Portugal (PT), and Russia (RU), with users classified according to whether they use explicit language. Blog comprises two disjoint social networks extracted from BlogCatalog, including Blog1 (B1) and Blog2 (B2). MAG contains six country-specific academic networks from the United States (US), China (CN), Germany (DE), Japan (JP), Russia (RU), and France (FR), covering 20 research categories. Detailed dataset statistics are provided in Appendix C.1.

Baselines. We compare our proposed ProTA with six representative GDA methods from two categories: four kinds of general GDA methods, including **UDAGCN**, **TDSS**, **DGSDA**, and **ADAlign** (Wu et al., 2020; Chen et al., 2025; Yang et al., 2025a; Chen et al., 2026a), and two kinds of class-imbalanced GDA methods, **PALA** and **ImGDA** (Ma et al., 2025b;a). For each baseline, we summerize the experimental results before and after integrating ProTA without modifying its original architecture or alignment objective. Detailed baseline descriptions are provided in Appendix C.2. Experimental Setup and hyperparameter settings are provided in Appendix C.3.

5.2 Performance Comparison

The experimental results on 12 transfer tasks are summarized in Table 1 and Table2. The Δ rows indicate the absolute performance gains in percentage points after integrating ProTA, with the largest and second-largest gains in each column highlighted in **bold** and underlined respectively. We have the following findings: It can be seen that integrating **ProTA** boosts the performance of representative GDA methods across most evaluation metrics, demonstrating its effectiveness. On average, **ProTA** improves all six GDA backbones, yielding gains in both Micro-F1 and Macro-F1 across the reported transfer tasks. Specifically, averaged over the 12 reported tasks, **ProTA** yields the largest Micro-F1 gain with DGSDA at **3.67%** and the largest Macro-F1 gain with ImGDA at **4.66%**. Notably, the largest individual improvements are **31.63%** in Micro-F1 for DGSDA on DE→RU and **17.45%** in Macro-F1 for UDAGCN on D→A. A slight decrease in Micro-F1 is observed in one task, while the corresponding Macro-F1 still improves. This can be attributed

[2]If all allocation weights for a class are zero, no edges are added for nodes of that class.

to ProTA's emphasis on improving class-balanced performance. In addition, B1→B2 and B2→B1 show relatively smaller performance gains, which is consistent with their smaller edge-imbalance discrepancies. This observation aligns well with the design of our method. Results for the remaining 32 transfer tasks are provided in Appendix E.1.

5.3 Edge Imbalance Analysis

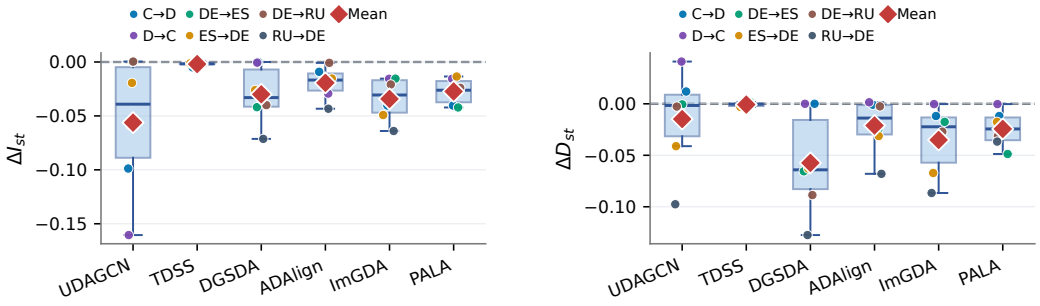
To further examine the effect of ProTA on edge imbalance, we analyze the changes after integrating it with the six GDA backbones, as shown in Figure 4. Overall, ProTA reduces both intra-domain edge imbalance and cross-domain edge distribution discrepancy on average across the six In a few cases, such as UDAGCN on C→D, the cross-domain edge-distribution discrepancy increases slightly, while intra-domain edge imbalance is substantially alleviated and the classification performance

Table 1: Cross-network node classification on selected Citation and MAG transfer tasks.

Method	A→D		D→A		C→D		D→C		DE→FR		FR→DE	
	Mi-F1	Ma-F1	Mi-F1	Ma-F1	Mi-F1	Ma-F1	Mi-F1	Ma-F1	Mi-F1	Ma-F1	Mi-F1	Ma-F1
UDAGCN	71.15	58.51	53.63	41.14	67.25	55.26	59.22	45.03	29.45	20.03	27.71	18.54
UDAGCN+P	71.23	58.57	59.95	58.59	69.86	65.09	67.76	57.62	29.82	21.61	30.17	23.39
Δ	+0.08	+0.06	+6.32	**+17.45**	+2.61	**+9.83**	**+8.54**	**+12.59**	+0.37	+1.58	**+2.46**	+4.85
TDSS	77.96	74.66	74.28	75.82	77.81	74.58	80.88	78.33	57.44	11.01	49.22	5.32
TDSS+P	78.05	75.51	74.59	76.14	78.63	74.94	81.60	79.09	57.50	11.02	51.44	11.34
Δ	+0.09	+0.85	+0.31	+0.32	+0.82	+0.36	+0.72	+0.76	+0.06	+0.01	+2.22	+6.02
DGSDA	76.55	74.06	73.87	75.09	78.37	76.46	83.03	81.85	55.49	11.69	52.29	7.45
DGSDA+P	77.57	75.50	74.21	75.13	78.65	76.95	83.21	82.08	55.52	11.69	52.71	7.28
Δ	**+1.02**	+1.44	+0.34	+0.04	+0.28	+0.49	+0.18	+0.23	+0.03	+0.00	+0.42	-0.17
ADAlign	78.28	76.07	73.92	75.59	79.65	78.68	82.54	80.88	59.10	20.09	56.18	17.07
ADAlign+P	78.39	76.23	74.22	75.82	79.89	78.71	82.76	81.57	59.03	22.45	56.30	17.64
Δ	+0.11	+0.16	+0.30	+0.23	+0.24	+0.03	+0.22	+0.69	-0.07	+2.36	+0.12	+0.57
PALA	74.42	61.52	65.60	61.57	69.53	67.76	74.45	70.94	58.77	9.19	56.07	13.34
PALA+P	74.67	64.10	68.55	68.78	72.63	70.83	77.63	72.93	58.81	13.39	56.11	14.79
Δ	+0.25	**+2.58**	+2.95	+7.21	+3.10	+3.07	+3.18	+1.99	+0.04	**+4.20**	+0.04	+1.45
ImGDA	72.23	66.53	59.57	57.92	68.14	56.77	70.31	68.53	56.05	20.33	52.26	11.47
ImGDA+P	72.61	67.69	67.94	67.24	73.07	61.24	74.63	71.34	56.58	23.55	52.74	25.49
Δ	+0.38	+1.16	**+8.37**	+9.32	**+4.93**	+4.47	+4.32	+2.81	**+0.53**	+3.22	+0.48	**+14.02**

Table 2: Cross-network node classification on selected Twitch and Blog transfer tasks.

Method	DE→ES		ES→DE		DE→RU		RU→DE		B1→B2		B2→B1	
	Mi-F1	Ma-F1	Mi-F1	Ma-F1	Mi-F1	Ma-F1	Mi-F1	Ma-F1	Mi-F1	Ma-F1	Mi-F1	Ma-F1
UDAGCN	69.60	42.56	64.08	59.64	74.71	44.30	62.90	54.93	28.04	21.24	25.09	17.21
UDAGCN+P	70.07	42.75	64.60	60.13	75.10	44.70	63.44	59.09	28.39	21.66	25.51	17.58
Δ	+0.47	+0.19	+0.52	+0.49	+0.39	+0.40	+0.54	+4.16	+0.35	+0.42	+0.42	+0.37
TDSS	51.53	50.39	48.98	48.17	60.94	48.81	45.61	42.43	24.18	17.36	21.63	13.85
TDSS+P	57.44	55.10	51.04	50.65	61.16	49.80	49.25	48.00	24.56	17.67	22.01	14.19
Δ	+5.91	+4.71	**+2.06**	+2.48	+0.22	+0.99	**+3.64**	+5.57	+0.38	+0.31	+0.38	+0.34
DGSDA	60.56	55.71	43.45	38.49	40.94	40.88	42.95	37.46	27.51	17.40	26.43	16.59
DGSDA+P	67.69	57.80	44.43	40.54	72.57	49.49	44.26	40.27	27.87	17.69	26.78	16.71
Δ	**+7.13**	+2.09	+0.98	+2.05	**+31.63**	**+8.61**	+1.31	+2.81	+0.36	+0.29	+0.35	+0.12
ADAlign	68.66	58.06	61.15	60.31	75.46	53.89	60.71	58.50	27.13	20.04	24.36	16.74
ADAlign+P	70.40	59.52	62.11	60.63	75.46	54.22	61.91	59.65	27.55	20.43	24.81	17.10
Δ	+1.74	+1.46	+0.96	+0.32	+0.00	+0.33	+1.20	+1.15	+0.42	+0.39	+0.45	+0.36
ImGDA	70.37	42.14	62.62	56.26	75.05	44.01	62.58	56.54	23.24	11.56	19.43	8.43
ImGDA+P	71.51	50.75	64.02	59.85	75.07	44.19	63.77	60.52	23.93	11.79	24.87	12.74
Δ	+1.14	**+8.61**	+1.40	**+3.59**	+0.02	+0.18	+1.19	+3.98	**+0.69**	+0.23	**+5.44**	**+4.31**
PALA	71.09	56.11	62.70	58.96	75.48	43.01	62.36	52.75	21.27	10.21	21.48	10.79
PALA+P	71.17	57.07	62.78	60.53	75.51	43.20	62.51	58.73	21.75	12.81	21.74	10.98
Δ	+0.08	+0.96	+0.08	+1.57	+0.03	+0.19	+0.15	**+5.98**	+0.48	**+2.60**	+0.26	+0.19



(a) Intra-domain edge imbalance.

(b) Cross-domain edge distribution discrepancy.

Figure 4: Changes in edge imbalance after integrating ProTA with six GDA backbones across six transfer tasks. $\Delta I_{st} = \Delta I_s + \Delta I_t$ and ΔD_{st} denote the changes in intra-domain edge imbalance and cross-domain edge-distribution discrepancy respectively. Negative values indicate reductions in the corresponding metric.

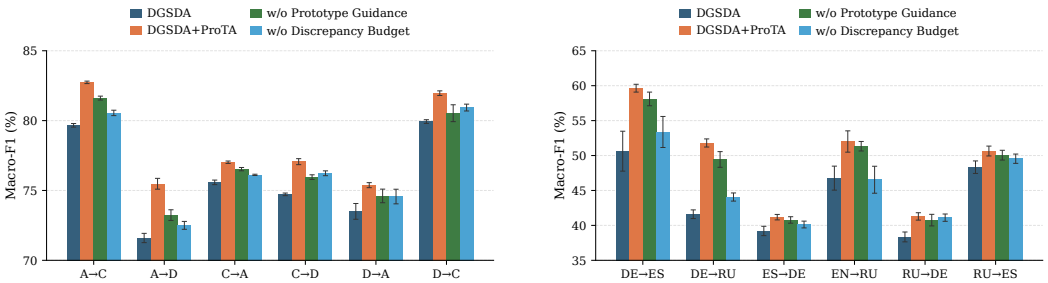
remains improvedas shown in Tables 1. This observation suggests that intra-domain edge imbalance and cross-domain edge distribution discrepancy characterize distinct aspects of graph imbalance and therefore be addressed jointly. Results for the remaining transfer tasks are provided in Appendix E.2.

5.4 Ablation Study

To evaluate the contributions of prototype-guided candidate selection and discrepancy-aware edge-budget allocation, we construct two ProTA variants with DGSDA as the backbone (Yang et al., 2025a). The *w/o Prototype Guidance* variant replaces prototype-guided candidate ranking with random ranking over the same candidate set, while keeping the class-pair budgets and the number of added edges unchanged. The *w/o Discrepancy Budget* variant replaces discrepancy-aware edge-budget allocation with deterministic uniform allocation across candidate class pairs, while retaining prototype-guided candidate ranking and all other settings. As shown in Figure 5, the full ProTA generally achieves the best Macro-F1 across the Citation and Twitch transfer tasks. Removing either prototype guidance or discrepancy-aware budget allocation leads to performance degradation on most tasks, demonstrating that both components contribute to the effectiveness of ProTA. Additional ablation study results are provided in Appendix E.4.

5.5 Visualization

To qualitatively examine the effect of ProTA on representation learning, we use t-SNE (Van der Maaten & Hinton, 2008) to visualize the node representations learned by DGSDA and TDSS before and after integrating ProTA. As shown in Figure 6, the original representations exhibit noticeable



(a) Citation

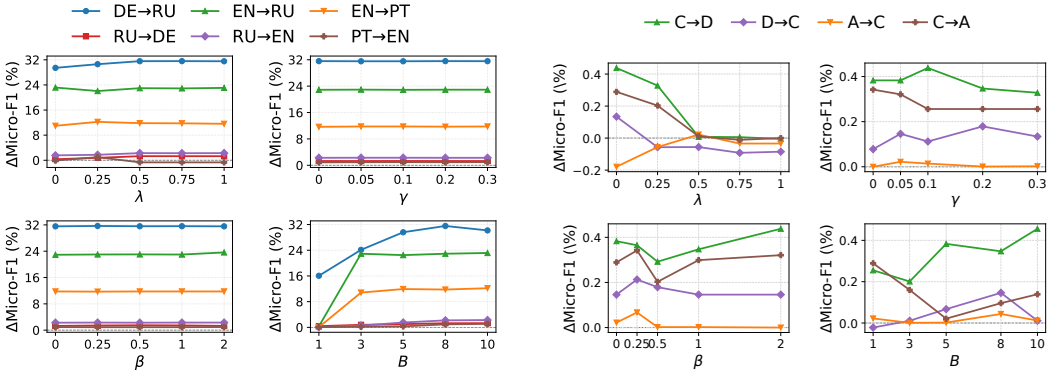
(b) Twitch

Figure 5: Macro-F1 ablation results of ProTA with DGSDA on (a) six Citation transfer tasks and (b) six Twitch transfer tasks. Bars and error bars denote the mean and standard deviation over five runs.



(a) DGSDA (b) DGSDA+P (c) TDSS (d) TDSS+P

Figure 6: t-SNE visualization before and after integrating ProTA into DGSDA and TDSS. Panels (a)–(b) show results on the DBLPv7→ACMv9 task, while panels (c)–(d) show results on the DBLPv7→Citationv1 task. Different colors denote node classes, while dark and light shades of the same color represent source- and target-domain nodes, respectively.



(a) Twitch (b) Citation

Figure 7: The influence of parameters λ, γ, β, and B on Twitch and Citation.

overlap between different classes and relatively diffuse class-wise distributions. After integrating ProTA, nodes of the same class form more compact and distinguishable clusters, while source and target nodes within each class remain broadly intermingled. This visualization suggests that ProTA improves class-discriminative representation structures while preserving cross-domain consistency. Additional visualization results are provided in Appendix E.5.

5.6 Hyperparameter Sensitivity Analysis

We examine the sensitivity of ProTA to λ, γ, β, and B, as shown in Figure 7. On Twitch, the performance is generally insensitive to λ, γ, and β, while increasing B leads to more noticeable changes on several tasks. On Citation, the effects of the four hyperparameters are more task-dependent, with moderate variations observed as their values change. These results indicate that ProTA is generally robust to hyperparameter choices, while the optimal augmentation configuration may vary with the structural characteristics of different graph domains. Additional results are provided in Appendix E.3.

6 Conclusion and Future Work

In this paper, we proposed ProTA framework to solve edge imbalanced in GDA. We identify intra-domain edge imbalance and cross-domain edge-distribution discrepancies, which ProTA mitigates by strengthening minority-class connectivity within each domain and balancing class-pair edge ratios across domains respectively. Comprehensive experiments verify the superiority of our approach. Future work will extend ProTA to more general GDA settings involving non-identical label spaces and complex graph relations, as discussed in Appendix F.

7 AI USE STATEMENT

In this work, we used generative AI tools solely for language polishing and improving the clarity and fluency of the manuscript. We have not used generative AI tools for research ideation, experimental design, code generation, data analysis, or literature review, and other disclosure-related tasks are not applicable to this work. All AI-assisted revisions were carefully reviewed by the authors to ensure that they preserved the intended technical meaning. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

REFERENCES

Ruichu Cai, Fengzhu Wu, Zijian Li, Pengfei Wei, Lingling Yi, and Kun Zhang. Graph domain adaptation: A generative view. *ACM Transactions on Knowledge Discovery from Data*, 18(3): 1–24, 2024.

Deli Chen, Yankai Lin, Guangxiang Zhao, Xuancheng Ren, Peng Li, Jie Zhou, and Xu Sun. Topology-imbalance learning for semi-supervised node classification. In *Advances in Neural Information Processing Systems*, volume 34, pp. 29885–29897, 2021.

Wei Chen, Guo Ye, Yakun Wang, Zhao Zhang, Libang Zhang, Daixin Wang, Zhiqiang Zhang, and Fuzhen Zhuang. Smoothness really matters: A simple yet effective approach for unsupervised graph domain adaptation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 15875–15883, 2025.

Wei Chen, Xingyu Guo, Shuang Li, Zhao Zhang, Yan Zhong, Fuzhen Zhuang, and Deqing Wang. Learning adaptive distribution alignment with neural characteristic function for graph domain adaptation. In *International Conference on Learning Representations*, 2026a.

Wei Chen, Xingyu Guo, Shuang Li, Yan Zhong, Zhao Zhang, Fuzhen Zhuang, Hongrui Liu, Libang Zhang, Guo Ye, and Huimei He. Learning structure-semantic evolution trajectories for graph domain adaptation. *International Conference on Learning Representations*, 2026b.

Yiming Chen, Shiji Song, Shuang Li, and Cheng Wu. A graph embedding framework for maximum mean discrepancy-based domain adaptation algorithms. *IEEE Transactions on Image Processing*, 29:199–213, 2019.

Quanyu Dai, Xiao-Ming Wu, Jiaren Xiao, Xiao Shen, and Dan Wang. Graph transfer learning via adversarial domain adaptation with graph convolution. *IEEE Transactions on Knowledge and Data Engineering*, 35(5):4908–4922, 2022. doi: 10.1109/TKDE.2022.3144250.

Ruiyi Fang, Bingheng Li, Jingyu Zhao, Ruizhi Pu, Qiuhao Zeng, Gezheng Xu, Charles Ling, and Boyu Wang. Homophily enhanced graph domain adaptation. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267, pp. 16006–16028. PMLR, 2025a.

Ruiyi Fang, Bingheng Li, zhao kang, Qiuhao Zeng, Nima Hosseini Dashtbayaz, Ruizhi Pu, Charles Ling, and Boyu Wang. On the benefits of attribute-driven graph domain adaptation. In *The Thirteenth International Conference on Learning Representations*, 2025b.

Sangmin Ga, Paul Hyunbin Cho, Gordon Euhyun Moon, and Sungwon Jung. Efficient gnn-based social recommender systems through social graph refinement: S. ga et al. *The Journal of Supercomputing*, 81(1):215, 2025.

Johannes Gasteiger, Stefan Weiß enberger, and Stephan Günnemann. Diffusion improves graph learning. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

Renhong Huang, Jiarong Xu, Xin Jiang, Ruichuan An, and Yang Yang. Can modifying data address graph domain adaptation? In *Proceedings of the 30th ACM SIGKDD conference on knowledge discovery and data mining*, pp. 1131–1142, 2024.

Johannes Klicpera, Aleksandar Bojchevski, and Stephan Günnemann. Predict then propagate: Graph neural networks meet personalized PageRank. In *International Conference on Learning Representations*, 2019.

Guillaume Lachaud, Patricia Conde-Cespedes, and Maria Trocan. Graph neural networks-based multilabel classification of citation network. In *Asian Conference on Intelligent Information and Database Systems*, pp. 128–140. Springer, 2022.

Wen-Zhi Li, Chang-Dong Wang, Hui Xiong, and Jian-Huang Lai. Graphsha: Synthesizing harder samples for class-imbalanced node classification. In *Proceedings of the 29th ACM SIGKDD conference on knowledge discovery and data mining*, pp. 1328–1340, 2023.

Jongin Lim, Daeho Um, Hyung Jin Chang, Dae Ung Jo, and Jin Young Choi. Class-attentive diffusion network for semi-supervised classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 8601–8609, 2021. doi: 10.1609/aaai.v35i10.17043.

Chuanyun Lin, Xi Zhou, and Xiao Shen. Hierarchical graph contrastive domain adaptation for multi-source cross-network node classification. *Expert Systems with Applications*, 284:127900, 2025.

Jianhua Lin. Divergence measures based on the shannon entropy. *IEEE Transactions on Information theory*, 37(1):145–151, 1991.

Meihan Liu, Zeyu Fang, Zhen Zhang, Ming Gu, Sheng Zhou, Xin Wang, and Jiajun Bu. Rethinking propagation for unsupervised graph domain adaptation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 13963–13971, 2024a.

Shikun Liu, Tianchun Li, Yongbin Feng, Nhan Tran, Han Zhao, Qiang Qiu, and Pan Li. Structural re-weighting improves graph domain adaptation. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 21778–21793. PMLR, 2023.

Shikun Liu, Deyu Zou, Han Zhao, and Pan Li. Pairwise alignment improves graph domain adaptation. In *Forty-first International Conference on Machine Learning*, 2024b.

Yating Liu, Xin Zheng, Yi Li, and Yanqing Guo. Test-time adaptation on recommender system with data-centric graph transformation. In *Thirty-Fourth International Joint Conference on Artificial Intelligence (IJCAI-25)*. International Joint Conferences on Artificial Intelligence Organization, 2025.

Zhining Liu, Ruizhong Qiu, Zhichen Zeng, Hyunsik Yoo, David Zhou, Zhe Xu, Yada Zhu, Kommy Weldemariam, Jingrui He, and Hanghang Tong. Class-imbalanced graph learning without class rebalancing. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 31747–31772. PMLR, 2024c.

Junyu Luo, Zhiping Xiao, Yifan Wang, Xiao Luo, Jingyang Yuan, Wei Ju, Langechuan Liu, and Ming Zhang. Rank and align: Towards effective source-free graph domain adaptation. In Kate Larson (ed.), *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24*, pp. 4706–4714. International Joint Conferences on Artificial Intelligence Organization, 8 2024. doi: 10.24963/ijcai.2024/520. URL https://doi.org/10.24963/ijcai.2024/520. Main Track.

Junyu Luo, Yuhao Tang, Yiwei Fu, Xiao Luo, Zhizhuo KOU, Zhiping Xiao, Wei Ju, Wentao Zhang, and Ming Zhang. Sparse causal discovery with generative intervention for unsupervised graph domain adaptation. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=Rab1LXJKLp.

Junyu Luo, Haoyu Tao, Xiao Luo, Yusheng Zhao, Zhiping Xiao, Dailan He, Wei Ju, Chong Chen, Xian-Sheng Hua, and Ming Zhang. Robust cross supervision with target mining for source-free graph domain adaptation. *IEEE Transactions on Knowledge and Data Engineering*, 2026.

Xin Ma, Yifan Wang, Siyu Yi, Wei Ju, Junyu Luo, Yusheng Zhao, Xiao Luo, and Jiancheng Lv. Dual prototype-enhanced contrastive framework for class-imbalanced graph domain adaptation. In *Advances in Neural Information Processing Systems*, volume 38, 2025a.

Xin Ma, Yifan Wang, Siyu Yi, Wei Ju, Bei Wu, Ziyue Qiao, Chenwei Tang, and Jiancheng Lv. Pala: Class-imbalanced graph domain adaptation via prototype-anchored learning and alignment. In *IJCAI*, pp. 3198–3207, 2025b.

Yihong Ma, Yijun Tian, Nuno Moniz, and Nitesh V Chawla. Class-imbalanced learning on graphs: A survey. *ACM Computing Surveys*, 57(8):1–16, 2025c.

Joonhyung Park, Jaeyun Song, and Eunho Yang. GraphENS: Neighbor-aware ego network synthesis for class-imbalanced node classification. In *International Conference on Learning Representations*, 2022.

Ruizhi Pu, Gezheng Xu, Ruiyi Fang, Bing-Kun Bao, Charles Ling, and Boyu Wang. Leveraging group classification with descending soft labeling for deep imbalanced regression. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 19978–19985, 2025.

Ziyue Qiao, Xiao Luo, Meng Xiao, Hao Dong, Yuanchun Zhou, and Hui Xiong. Semi-supervised domain adaptation in graph transfer learning. In Edith Elkind (ed.), *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI-23*, pp. 2279–2287. International Joint Conferences on Artificial Intelligence Organization, 8 2023. doi: 10.24963/ijcai.2023/253. URL https://doi.org/10.24963/ijcai.2023/253. Main Track.

Ziyue Qiao, Meng Xiao, Weiyu Guo, Xiao Luo, and Hui Xiong. Information filtering and interpolating for semi-supervised graph domain adaptation. *Pattern Recognition*, 153:110498, 2024.

Jiawen Qin, Haonan Yuan, Qingyun Sun, Lyujin Xu, Jiaqi Yuan, Pengfeng Huang, Zhaonan Wang, Xingcheng Fu, Hao Peng, Jianxin Li, and Philip S. Yu. Igl-bench: Establishing the comprehensive benchmark for imbalanced graph learning. In *ICLR*, 2025.

Xiao Shen, Quanyu Dai, Fu-lai Chung, Wei Lu, and Kup-Sze Choi. Adversarial deep network embedding for cross-network node classification. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 2991–2999, 2020.

Xiao Shen, Chuanyun Lin, Zhihao Chen, Junwei Gong, and Xi Zhou. Prototypical multi-source graph contrastive domain adaptation. *Pattern Recognition*, pp. 114312, 2026.

Bo-Shen Shi, Yong-Qing Wang, Fang-Da Guo, Bing-Bing Xu, Hua-Wei Shen, and Xue-Qi Cheng. Domain adaptation for graph representation learning: Challenges, progress, and prospects. *Journal of Computer Science and Technology*, 40(2):283–300, 2025.

Jaeyun Song, Joonhyung Park, and Eunho Yang. Tam: topology-aware margin loss for class-imbalanced node classification. In *International conference on machine learning*, pp. 20369–20383. PMLR, 2022.

Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

Mengzhu Wang, Wenhao Ren, Yu Zhang, Yanlong Fan, Dianxi Shi, Luoxi Jing, and Nan Yin. Gaussian mixture model for graph domain adaptation. In *IJCAI*, pp. 1963–1972, 2025.

Hongwei Wen, Can Zhang, Haoyu He, Hanyuan Hang, and Minglong Lei. Progressive graph structure adjustment for homophily shift adaptation. In *Forty-third International Conference on Machine Learning*.

Jun Wu, Jingrui He, and Elizabeth Ainsworth. Non-iid transfer learning on graphs. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pp. 10342–10350, 2023.

Man Wu, Shirui Pan, Chuan Zhou, Xiaojun Chang, and Xingquan Zhu. Unsupervised domain adaptive graph convolutional networks. In *Proceedings of The Web Conference 2020*, pp. 1457–1467, 2020. doi: 10.1145/3366423.3380219.

Liang Yang, Xin Chen, Jiaming Zhuo, Di Jin, Chuan Wang, Xiaochun Cao, Zhen Wang, and Yuanfang Guo. Disentangled graph spectral domain adaptation. In *Forty-second International Conference on Machine Learning*, 2025a.

Liu Yang, Mengni Chen, Tingxuan Chen, Jinqi Hu, and Zidong Wang. Graphdeh: Graph diffusion enhanced hypergrpah method for class-imbalanced node classification. In *2025 IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1–6. IEEE, 2025b.

Niya Yang, Ye Wang, Zhizhi Yu, Dongxiao He, Xin Huang, and Di Jin 0001. Joint domain adaptive graph convolutional network. In *IJCAI*, pp. 2496–2504, 2024.

Nan Yin, Mengzhu Wang, Zhenghan Chen, Li Shen, Huan Xiong, Bin Gu, and Xiao Luo. Dream: Dual structured exploration with mixup for open-set graph domain adaption. In *International conference on learning representations*, volume 2024, pp. 19776–19797, 2024.

Nan Yin, Li Shen, Mengzhu Wang, Xinwang Liu, Chong Chen, and Xian-Sheng Hua. Dream: a dual variational framework for unsupervised graph domain adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025a.

Nan Yin, Xiao Teng, Zhiguang Cao, and Mengzhu Wang. Coupling category alignment for graph domain adaptation. In James Kwok (ed.), *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence, IJCAI-25*, pp. 3561–3569. International Joint Conferences on Artificial Intelligence Organization, 8 2025b. doi: 10.24963/ijcai.2025/396. Main Track.

Yuning You, Tianlong Chen, Zhangyang Wang, and Yang Shen. Graph domain adaptation via theory-grounded spectral regularization. In *International Conference on Learning Representations*, 2023.

Haodong Zhang, Tao Ren, Changhu Wang, Yifan Wang, Wei Ju, Huaizhi Tang, Junyu Luo, Zimo Wang, Ziyue Qiao, Xian-Sheng Hua, et al. Disco: Diffusion-guided unbiased discriminative learning for unsupervised graph domain adaptation. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1*, pp. 1892–1903, 2026.

Pan Zhang, Bo Zhang, Ting Zhang, Dong Chen, Yong Wang, and Fang Wen. Prototypical pseudo label denoising and target structure learning for domain adaptive semantic segmentation. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12409–12419, 2021. doi: 10.1109/CVPR46437.2021.01223.

Yizhou Zhang, Guojie Song, Lun Du, Shuwen Yang, and Yilun Jin. DANE: Domain adaptive network embedding. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, pp. 4362–4368, 2019. doi: 10.24963/ijcai.2019/606.

Zhen Zhang and Bingsheng He. Towards unsupervised open-set graph domain adaptation via dual reprogramming. *Advances in Neural Information Processing Systems*, 38:53320–53343, 2026.

Zhen Zhang, Meihan Liu, Anhui Wang, Hongyang Chen, Zhao Li, Jiajun Bu, and Bingsheng He. Collaborate to adapt: Source-free graph domain adaptation via bi-directional adaptation. In *Proceedings of the ACM Web Conference 2024*, pp. 664–675, 2024.

Tianxiang Zhao, Xiang Zhang, and Suhang Wang. Graphsmote: Imbalanced node classification on graphs with graph neural networks. In *Proceedings of the 14th ACM international conference on web search and data mining*, pp. 833–841, 2021.

Tianxiang Zhao, Dongsheng Luo, Xiang Zhang, and Suhang Wang. Topoimb: Toward topology-level imbalance in learning from graphs. In Bastian Rieck and Razvan Pascanu (eds.), *Proceedings of the First Learning on Graphs Conference*, volume 198 of *Proceedings of Machine Learning Research*, pp. 37:1–37:18. PMLR, 09–12 Dec 2022.

Mengting Zhou and Zhiguo Gong. Graphsr: A data augmentation algorithm for imbalanced node classification. In *Proceedings of the AAAI Conference on artificial intelligence*, volume 37, pp. 4954–4962, 2023.

A NOTATION

Table 1: Notation and description.

Notation	Description
$\mathcal{G}^s, \mathcal{G}^t$	Source and target graphs, respectively
$\mathcal{V}^s, \mathcal{V}^t$	Node sets of the source and target graphs, respectively
$\mathcal{E}^s, \mathcal{E}^t$	Edge sets of the source and target graphs, respectively
$\mathbf{A}^s, \mathbf{A}^t$	Adjacency matrices of the source and target graphs, respectively
$\mathbf{X}^s, \mathbf{X}^t$	Node feature matrices of the source and target graphs, respectively
$\mathbf{Y}^s$	Source-domain node labels
$\mathcal{C}$	Shared class set across the two domains
c, c'	Minority class and candidate-neighbor class, respectively
$d \in \{s, t\}$	Domain index, denoting the source or target domain
$\mathcal{R}$	Set of class-pair edge types
$\boldsymbol{\pi}^d$	Normalized class-pair edge distribution in domain d
$\mathbf{u}_{\mathcal{R}}$	Uniform distribution over the class-pair edge types in $\mathcal{R}$
I^d_{edge}	Intra-domain edge imbalance of domain d
$D^{s,t}_{\text{edge}}$	Cross-domain edge-distribution discrepancy
n^d_c	Number of nodes belonging to class c in domain d
$\mathcal{C}^d_{\min}$	Minority-class set in domain d
ξ	Threshold for identifying minority classes
$p^t_{i,c}$	Predicted probability that target node i belongs to class c
$\widetilde{y}^t_i$	Pseudo-label of target node i
q^t_i	Prediction confidence of target node i
μ^t_i	Prototype-consistency margin of target node i
$\mathcal{V}^t_{\text{rel}}$	Set of target nodes with reliable pseudo-labels
$\widetilde{n}^t_c$	Estimated number of reliable target nodes assigned to class c
$\widetilde{\mathbf{A}}^s_y$	Adjacency matrix of the class-preserving source graph
$\widehat{\mathbf{A}}^s_y$	Normalized propagation matrix of the class-preserving source graph
$\mathbf{H}^s_{(k)}$	Source-node representations after k diffusion steps
$\mathbf{p}^s_c$	Topology-aware source prototype of class c
α	Residual coefficient in graph diffusion
K	Number of graph-diffusion steps
$\mathcal{Q}^d_{i,c'}$	Candidate nodes of class c' for node i in domain d
$s^{d,c\to c'}_{ij}$	Candidate-edge score between nodes i and j
λ	Trade-off between attribute similarity and prototype consistency
$\pi^d_{c'\|c}$	Class-conditional edge-distribution ratio from class c to c'
$\delta^d_{c,c'}$	Intra-domain edge deficit of class-pair type (c, c')
$\Delta^d_{c,c'}$	Directional cross-domain edge-distribution deficit of type (c, c')
$w^d_{c,c'}$	Allocation weight of class-pair edge type (c, c')
β	Weight of cross-domain edge-distribution correction
B	Total augmentation budget for each minority-class node
$B^d_{c\to c'}$	Edge budget allocated from class c to class c' in domain d
$\widetilde{\mathcal{G}}^s, \widetilde{\mathcal{G}}^t$	Augmented source and target graphs, respectively

B ALGORITHMS

Algorithm 1 Prototype-guided Topology Augmentation (ProTA)

Require: Source graph $\mathcal{G}^s$ with labels, target graph $\mathcal{G}^t$, diffusion steps K, restart coefficient α, minority threshold ξ, confidence threshold τ, prototype margin threshold γ, edge scoring coefficient λ, cross-domain coefficient β, edge budget B

Ensure: Enhanced graphs $\widetilde{\mathcal{G}}^s$ and $\widetilde{\mathcal{G}}^t$

// Stage 1: Class-Preserving Diffusion for Source Prototypes

Construct class-preserving adjacency matrix $\widetilde{\mathbf{A}}_y^s$ using source labels
Normalize $\widetilde{\mathbf{A}}_y^s$ to obtain $\widehat{\mathbf{A}}_y^s$
Initialize $\mathbf{H}_{(0)}^s \leftarrow \mathbf{X}^s$
for $k = 1$ to K **do**
    $\mathbf{H}_{(k)}^s \leftarrow \alpha \mathbf{X}^s + (1-\alpha)\widehat{\mathbf{A}}_y^s \mathbf{H}_{(k-1)}^s$
end for
Compute source prototypes $\{\mathbf{p}_c^s\}_{c\in\mathcal{C}}$ by averaging diffused representations

// Stage 2: Prototype-Guided Topology Augmentation

Identify source minority classes $\mathcal{C}_{\min}^s$ using ground-truth labels
Warm up the GDA backbone and obtain target predictions
Select reliable target nodes $\mathcal{V}_{\text{rel}}^t$ using confidence and prototype consistency
Identify target minority classes $\mathcal{C}_{\min}^t$ from reliable pseudo-labels
for $d \in \{s, t\}$ **do**
    for each minority node i with class c **do**
        Construct candidate sets $\mathcal{Q}_{i,c'}^d$
        for each candidate node $j \in \mathcal{Q}_{i,c'}^d$ **do**
            Compute edge score $s_{ij}^{d,c\to c'}$
        end for
        Estimate class-conditional edge distribution $\boldsymbol{\pi}^d$
        Compute edge deficits $\delta_{c,c'}^d$ and $\Delta_{c,c'}^d$
        Compute allocation weights $w_{c,c'}^d = \delta_{c,c'}^d + \beta\Delta_{c,c'}^d$
        Allocate edge budget $B_{c\to c'}^d$
        Add selected candidate edges
    end for
end for

// Stage 3: Integration with GDA Backbones

Construct enhanced graphs $\widetilde{\mathcal{G}}^s$ and $\widetilde{\mathcal{G}}^t$
Feed the enhanced graphs into the adopted GDA backbone
Optimize the GDA model with $\mathcal{L} = \mathcal{L}_{\text{cls}}^s + \lambda\mathcal{L}_{\text{GDA}}$
return $\widetilde{\mathcal{G}}^s, \widetilde{\mathcal{G}}^t$

B.1 NORMALIZATION OF ADJACENCY MATRIX

We define $\widetilde{\mathbf{D}}_y^s$ as degree matrix of the class-preserving adjacency matrix $\widetilde{\mathbf{A}}_y^s$. The normalized class-preserving propagation matrix is defined as

$$\widehat{\mathbf{A}}_y^s = \big(\widetilde{\mathbf{D}}_y^s\big)^{-\frac{1}{2}} \widetilde{\mathbf{A}}_y^s \big(\widetilde{\mathbf{D}}_y^s\big)^{-\frac{1}{2}}. \tag{1}$$

B.2 NORMALIZATION OF SOURCE PROTOTYPE

For each class $c \in \mathcal{C}$, let $\mathcal{V}_c^s = \{i \in \mathcal{V}^s : y_i^s = c\}$. We construct its topology-aware source prototype by averaging the diffused source representations belonging to that class:

$$\overline{\mathbf{p}}_c^s = \frac{1}{|\mathcal{V}_c^s|} \sum_{i\in\mathcal{V}_c^s} \mathbf{h}_i^s. \tag{2}$$

We then normalize each prototype and collect all class prototypes as

$$\mathbf{p}_c^s = \frac{\overline{\mathbf{p}}_c^s}{\|\overline{\mathbf{p}}_c^s\|_2 + \epsilon}, \qquad \mathbf{P}^s = [\mathbf{p}_1^s, \ldots, \mathbf{p}_C^s]^\top, \tag{3}$$

where $\epsilon > 0$ ensures numerical stability and $\mathbf{P}^s \in \mathbb{R}^{C \times F}$ contains one prototype per class. Because the prototypes aggregate multi-hop same-class context, they provide class-level semantic references for subsequent topology augmentation in both domains.

B.3 Source-Domain Minority Classes

For the source domain, minority classes can be directly identified using the ground-truth node labels. Specifically, the number of nodes belonging to class c is calculated as

$$n_c^s = \sum_{i \in \mathcal{V}^s} \mathbb{I}(y_i^s = c), \tag{4}$$

where $\mathbb{I}(\cdot)$ denotes the indicator function.

Given a minority threshold $\xi \in (0, 1)$, the source-domain minority-class set is defined as

$$\mathcal{C}_{\min}^s = \left\{ c \in \mathcal{C} \mid n_c^s < \xi \max_{r \in \mathcal{C}} n_r^s \right\}. \tag{5}$$

Classes whose node numbers are significantly smaller than the dominant class are considered minority classes.

B.4 Source-Domain Candidate Edge Selection

Source prototypes provide class-specific semantic references for assessing candidate connections. Each candidate edge is scored by combining the attribute similarity of its endpoints with their consistency with the corresponding class prototypes.

For a source minority node i of class c, let $\mathcal{Q}_{i,c'}^s$ denote the set of non-neighboring source nodes belonging to class c'. For each candidate $j \in \mathcal{Q}_{i,c'}^s$, the edge score is defined as

$$s_{ij}^{s,c \to c'} = \lambda \operatorname{sim}\left(\mathbf{x}_i^s, \mathbf{x}_j^s\right) + \frac{1-\lambda}{2}\Big[\operatorname{sim}\left(\mathbf{x}_i^s, \mathbf{p}_c^s\right) + \operatorname{sim}\left(\mathbf{x}_j^s, \mathbf{p}_{c'}^s\right)\Big], \tag{6}$$

where $\mathbf{x}_i^s$ and $\mathbf{x}_j^s$ are node attribute vectors, and $\mathbf{p}_c^s$ and $\mathbf{p}_{c'}^s$ are the corresponding source class prototypes. The function $\operatorname{sim}(\cdot, \cdot)$ denotes cosine similarity, and $\lambda \in [0, 1]$ balances attribute similarity and prototype consistency. The notation $c \to c'$ identifies the classes of the minority node and its candidate.

The obtained candidate scores are subsequently used for topology augmentation according to the discrepancy-aware edge budget allocation strategy described in the main paper.

B.5 Computation of Edge Deficits and Distribution Discrepancies

This section details the class-conditional edge distributions used to compute the intra-domain edge deficits $\delta_{c,c'}^d$ and directional cross-domain edge-distribution discrepancies $\Delta_{c,c'}^d$.

Class-Conditional Edge Distributions. For each class c, the conditional distribution describes the relative frequencies of neighboring classes, aggregated over class-c nodes. Source distributions are computed from ground-truth labels, whereas target distributions are estimated from reliable pseudo-labels.

Let $\mathcal{V}_{\text{rel}}^t$ denote the target-node set defined in Eq. 8. Only target edges whose endpoints both belong to this set are used for estimation:

$$\mathcal{E}_{\text{rel}}^t = \left\{ \{i, j\} \in \mathcal{E}^t \;\middle|\; i, j \in \mathcal{V}_{\text{rel}}^t \right\}. \tag{7}$$

For classes $c, c' \in \mathcal{C}$, the corresponding adjacency count is

$$\widehat{m}^t_{c,c'} = \sum_{\{i,j\}\in\mathcal{E}^t_{\text{rel}}} \Big[\mathbb{I}(\widehat{y}^t_i = c)\mathbb{I}(\widehat{y}^t_j = c') + \mathbb{I}(\widehat{y}^t_i = c')\mathbb{I}(\widehat{y}^t_j = c)\Big], \tag{8}$$

where $\mathbb{I}(\cdot)$ is the indicator function. Each undirected edge appears once in the summation, and the two terms account for its two orientations. A same-class edge contributes twice to $\widehat{m}^t_{c,c}$, once for each endpoint acting as the central node.

The smoothed target class-conditional distribution is

$$\widehat{\pi}^t_{c'|c} = \frac{\widehat{m}^t_{c,c'} + \epsilon_e}{\sum_{q\in\mathcal{C}} \left(\widehat{m}^t_{c,q} + \epsilon_e\right)}, \tag{9}$$

where $\epsilon_e > 0$ assigns nonzero probability to unobserved connections and keeps the distribution well-defined when no reliable edges are available for class c.

The source numbers $m^s_{c,c'}$ follow the same two-orientation counting rule, using $\mathcal{E}^s$ and ground-truth labels y^s_i in place of $\mathcal{E}^t_{\text{rel}}$ and $\widehat{y}^t_i$, respectively. The source conditional distribution is

$$\pi^s_{c'|c} = \frac{m^s_{c,c'} + \epsilon_e}{\sum_{q\in\mathcal{C}} \left(m^s_{c,q} + \epsilon_e\right)}. \tag{10}$$

Intra-Domain Edge Deficits. A uniform distribution over candidate classes serves as the reference for measuring intra-domain imbalance. With $u = 1/|\mathcal{C}|$, the deficits are defined as

$$\delta^s_{c,c'} = \Big[u - \pi^s_{c'|c}\Big]_+, \qquad \delta^t_{c,c'} = \Big[u - \widehat{\pi}^t_{c'|c}\Big]_+, \tag{11}$$

where $[x]_+ = \max(x, 0)$. A positive deficit indicates that the corresponding neighboring class is underrepresented relative to the uniform reference.

Cross-Domain Edge-Distribution Discrepancies. Directional discrepancies identify connections that are less prevalent in one domain than in the other:

$$\begin{aligned} \Delta^s_{c,c'} &= \Big[\widehat{\pi}^t_{c'|c} - \pi^s_{c'|c}\Big]_+, \\ \Delta^t_{c,c'} &= \Big[\pi^s_{c'|c} - \widehat{\pi}^t_{c'|c}\Big]_+. \end{aligned} \tag{12}$$

A positive $\Delta^s_{c,c'}$ indicates a lower conditional frequency in the source domain; $\Delta^t_{c,c'}$ captures the reverse direction. These quantities enter the allocation weights in Eq. 11.ß

C EMPIRICAL DETAILS

C.1 DATASET DESCRIPTION

We evaluate ProTA on four widely used GDA benchmarks: Citation, Twitch, Blog, and MAG. Citation contains three academic networks, ACMv9 (A), Citationv1 (C), and DBLPv7 (D), covering five research categories. Twitch consists of six regional social networks from Germany (DE), England (EN), Spain (ES), France (FR), Portugal (PT), and Russia (RU), with users classified according to whether they use explicit language. Blog comprises two disjoint social networks, Blog1 (B1) and Blog2 (B2), extracted from BlogCatalog, while MAG contains six country-specific academic networks from the United States (US), China (CN), Germany (DE), Japan (JP), Russia (RU), and France (FR), covering 20 research categories.

For each benchmark, every ordered pair of distinct domains is treated as a source-to-target adaptation task. This produces 6, 30, 2, and 30 transfer tasks on Citation, Twitch, Blog, and MAG, respectively.

C.2 BASELINES

We select six representative methods from two categories. For general GDA, **UDA-GCN** captures local and global graph information through adversarial adaptation (Wu et al., 2020); **TDSS**

smooths the target graph through neighborhood sampling to mitigate structural shifts (Chen et al., 2025); **DGSDA** disentangles attribute and topology alignment using graph spectral filters (Yang et al., 2025a); and **ADAlign** adaptively aligns transfer-specific distribution discrepancies (Chen et al., 2026a). For class-imbalanced GDA, **PALA** employs prototype-anchored learning and alignment (Ma et al., 2025b), while **ImGDA** combines dual-branch prototypes with class-aware contrastive adaptation (Ma et al., 2025a). For each method, we compare its original version with its ProTA-enhanced counterpart while keeping the original architecture and alignment objective unchanged.

Below, we provide detailed descriptions of the six baselines used in our experiments. These methods are grouped into general GDA methods and class-imbalanced GDA methods.

General GDA Methods.

- **UDAGCN** (Wu et al., 2020) integrates local and global graph information through a dual graph convolutional architecture. It adopts adversarial domain adaptation to reduce the distribution discrepancy between source and target node representations, thereby encouraging the learning of domain-invariant features.
- **TDSS** (Chen et al., 2025) mitigates structural distribution shifts by smoothing the target graph through neighborhood sampling. By refining the target-side structural information before alignment, it aims to improve the transferability of node representations across graph domains.
- **DGSDA** (Yang et al., 2025a) performs graph domain adaptation from the spectral perspective. It disentangles attribute and topology information through graph spectral filters and aligns the corresponding components separately, thereby reducing different sources of distribution shift between domains.
- **ADAlign** (Chen et al., 2026a) develops an adaptive alignment strategy for different graph transfer scenarios. Instead of applying a fixed alignment objective to all tasks, it adjusts the alignment mechanism according to transfer-specific distribution discrepancies.

Class-Imbalanced GDA Methods.

- **PALA** (Ma et al., 2025b) addresses class-imbalanced graph domain adaptation through prototype-anchored learning and alignment. It constructs and recalibrates class prototypes in the source domain to alleviate the biased transfer caused by imbalanced class distributions.
- **ImGDA** (Ma et al., 2025a) combines dual-branch prototype learning with class-aware contrastive adaptation. The learned prototypes provide class-level guidance for cross-domain adaptation, while the contrastive objective further reduces class-imbalance-induced bias during knowledge transfer.

For each baseline, we compare its original version with the corresponding ProTA-enhanced variant. ProTA only augments the graph topology, while the original model architecture and domain-alignment objective of each baseline remain unchanged.

C.3 Implementation Notes

All experiments are implemented using PyTorch 2.1.1 with CUDA 12.1 and conducted on a workstation equipped with a 13th Gen Intel(R) Core(TM) i9-13980HX CPU and an NVIDIA GeForce RTX 4090 Laptop GPU with 16GB memory. For fair comparison, ProTA and all GDA baselines adopt the same backbone architectures, optimization settings, and training configurations, while the search ranges and selected values of ProTA-specific hyperparameters are provided in the Appendix. Before applying ProTA, the underlying GDA model is trained on the original graphs for w warm-up epochs to generate reliable pseudo-labels for target nodes. We repeatedly train and test each model five times using the same dataset partition and report the average Micro-F1 and Macro-F1.

Table 2: Statistics of the benchmark datasets used in our experiments. The number of edges denotes the number of stored edges in the processed graph.

Type	Dataset	#Nodes	#Edges	#Feat. Dims	#Classes
Citation	ACMv9 (A)	9,360	31,112	6,775	5
	Citationv1 (C)	8,935	30,196	6,775	5
	DBLPv7 (D)	5,484	16,234	6,775	5
Twitch	Germany (DE)	9,498	153,138	3,170	2
	England (EN)	7,126	35,324	3,170	2
	Spain (ES)	4,648	59,382	3,170	2
	France (FR)	6,549	112,666	3,170	2
	Portugal (PT)	1,912	31,299	3,170	2
	Russia (RU)	4,385	37,304	3,170	2
Blog	Blog1 (B1)	2,300	66,942	8,189	6
	Blog2 (B2)	2,896	107,672	8,189	6
MAG	China (CN)	101,952	285,991	128	20
	Germany (DE)	43,032	127,704	128	20
	France (FR)	29,262	79,182	128	20
	Japan (JP)	37,498	91,412	128	20
	Russia (RU)	32,833	68,294	128	20
	United States (US)	132,558	702,482	128	20

D Related Work

D.1 Class-Imbalanced Graph Learning

Existing methods for class-imbalanced graph learning broadly fall into two categories: class rebalancing and topology-aware correction (Liu et al., 2024c; Qin et al., 2025). Class-rebalancing methods compensate for unequal class sizes, whereas topology-aware methods mitigate structural biases that induced by class imbalance during message passing. Other methods alleviate class imbalance by synthesizing minority nodes and edges (Zhao et al., 2021), constructing minority-centered networks (Park et al., 2022), leveraging reliable unlabeled nodes (Zhou & Gong, 2023), or generating harder minority samples (Li et al., 2023). Topology-aware methods mitigate class-imbalance bias by exploiting class-boundary information, local connectivity, topology-group structure, and message-passing patterns, or through diffusion-enhanced structural augmentation (Chen et al., 2021; Song et al., 2022; Zhao et al., 2022; Liu et al., 2024c; Yang et al., 2025b).

These methods account for both unequal class sizes and topology-induced structural bias, while do not explicitly model imbalance in class-pair edge distributions. Moreover, most methods are developed for single-graph settings and therefore do not consider how graph imbalance affect distribution shifts across domains.

D.2 Graph Domain Adaptation

Graph domain adaptation (GDA) aims to transfer knowledge from a labeled source graph to an unlabeled target graph by mitigating distribution shifts across domains (Wu et al., 2020; Shi et al., 2025). Early methods mainly learn transferable node representations that integrate attributes and graph structure through distribution-discrepancy minimization or adversarial training (Zhang et al., 2019; Shen et al., 2020; Wu et al., 2020; Dai et al., 2022).

Subsequent studies start to characterize graph-specific distribution shifts. Representative approaches model spectral discrepancies, structural shifts, conditional structure shifts, and pairwise relations (You et al., 2023; Wu et al., 2023; Liu et al., 2023; 2024b; Yang et al., 2025a). Other methods revisit graph propagation, refine target-graph topology, model attribute–topology interactions and homophily shifts, perform class-level alignment, or develop scenario-adaptive alignment strategies (Liu et al., 2024a; Chen et al., 2025; Huang et al., 2024; Fang et al., 2025b;a; Yin et al., 2025b; Chen et al., 2026a). Recent studies further explore diffusion-guided adaptation, causal structure intervention, variational modeling, distribution modeling, structure–semantic evolution,

and progressive graph-structure adjustment (Zhang et al., 2026; Luo et al., 2025; Yin et al., 2025a; Wang et al., 2025; Chen et al., 2026b; Wen et al.).

Beyond the standard unsupervised setting, GDA has also been extended to different adaptation scenarios. Source-free GDA removes access to source-domain data during adaptation (Luo et al., 2024; Zhang et al., 2024; Luo et al., 2026), while semi-supervised GDA exploits limited target supervision (Qiao et al., 2023; 2024). Other GDA methods further relaxes the assumption of a fully shared label space (Yin et al., 2024; Zhang & He, 2026), and multi-source GDA transfers knowledge from multiple source graphs (Lin et al., 2025; Shen et al., 2026).

Despite substantial progress in modeling different forms of domain shift and extending GDA to diverse adaptation settings, these methods primarily focus on discrepancies between graph domains and rarely explicitly account for class imbalance.

D.3 Class-Imbalanced Graph Domain Adaptation

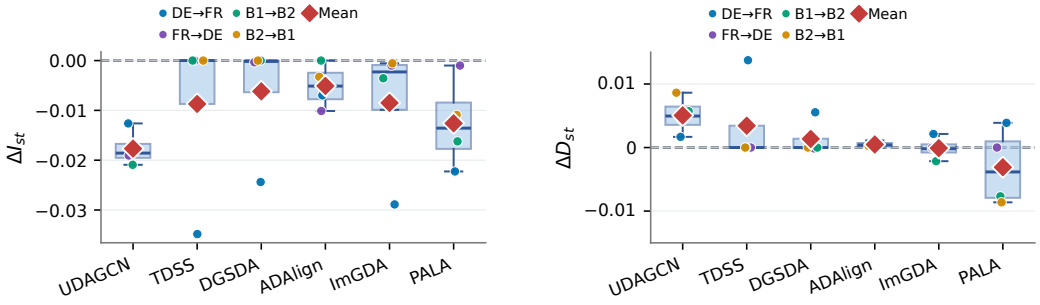
Recent studies have begun to investigate class imbalance in GDA. To alleviate biased knowledge transfer from class-imbalanced source graphsPALA learns and recalibrates source-domain prototypes (Ma et al., 2025b), while ImGDA combines dual-branch prototype learning with class-aware contrastive adaptation (Ma et al., 2025a). Although both methods exploit target pseudo-labels during adaptation, their imbalance modeling remains primarily focused on source-domain node distributions.

As a result, intra-domain imbalance in class-pair edge and their distributions discrepancies across domains remain largely unexplored. ProTA extends class-imbalanced GDA from node imbalance to edge imbalance by jointly considering intra-domain edge deficits and cross-domain edge-distribution discrepancies during topology augmentation.

E Additional Results

E.1 Performance Comparison

The experimental results on the 24 additional transfer tasks are reported in Tables **??**–**??**. The Δ rows report performance changes in percentage points after integrating ProTA, with the largest and second-largest gains in each column highlighted in **bold** and underlined, respectively. Overall, ProTA yields positive average gains in both Micro-F1 and Macro-F1 for all six GDA backbones, improving 127 of the 144 backbone–task combinations in Micro-F1 and 141 combinations in Macro-F1. Averaged over the 24 tasks, ProTA achieves the largest Micro-F1 gain with DGSDA at **1.51** percentage points and the largest Macro-F1 gain with PALA at **5.14** points. The largest individual improvements reach **17.70** Micro-F1 points for DGSDA on EN→RU and **20.41** Macro-F1 points for PALA on PT→EN. Although Micro-F1 decreases in a limited number of cases, the corresponding Macro-F1 improves in all of them, indicating a trade-off between overall and class-balanced performance rather than uniform degradation. In addition, DE→US and US→DE exhibit relatively modest gains, consistent with their smaller improvements in edge imbalance. These results further demonstrate that ProTA can complement diverse domain-alignment objectives across different graph domains.



(a) Changes in intra-domain edge imbalance.

(b) Changes in cross-domain edge-distribution discrepancy.

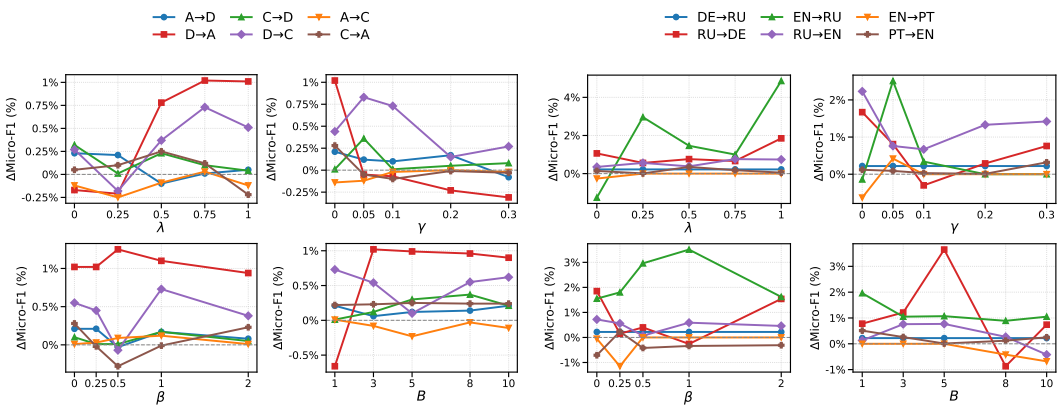
Figure 1: Changes in edge imbalance after integrating ProTA with six GDA backbones across four additional transfer tasks. The vertical axes report the aggregated change in source- and target-domain intra-domain edge imbalance, $\Delta I_{st} = \Delta I_s + \Delta I_t$, and the change in cross-domain edge-distribution discrepancy, ΔD_{st}, in (a) and (b), respectively. Negative values indicate reductions in the corresponding metric. Colored points denote individual transfer tasks, boxes summarize their distributions, and red diamonds indicate the mean across the four tasks.

Table 3: Cross-network node classification on the remaining Twitch transfer tasks (1/4).

Method	DE→EN		DE→FR		DE→PT		EN→DE		EN→ES		EN→FR	
	Mi-F1	Ma-F1	Mi-F1	Ma-F1	Mi-F1	Ma-F1	Mi-F1	Ma-F1	Mi-F1	Ma-F1	Mi-F1	Ma-F1
UDAGCN	59.30	58.15	62.96	39.36	65.90	54.72	63.85	55.37	70.37	41.79	63.12	38.81
UDAGCN+P	59.32	58.20	63.05	39.66	66.74	57.69	65.16	61.46	70.57	41.79	63.17	38.87
Δ	+0.02	+0.05	+0.09	+0.30	+0.84	+2.97	+1.31	+6.09	+0.20	+0.00	+0.05	+0.06
TDSS	56.44	55.62	55.70	55.29	48.90	48.72	60.22	56.31	61.60	57.47	56.13	54.84
TDSS+P	57.54	56.66	58.56	56.87	50.78	50.77	60.42	56.47	63.55	57.60	59.93	55.51
Δ	+1.10	+1.04	+2.86	+1.58	+1.88	+2.05	+0.20	+0.16	+1.95	+0.13	**+3.80**	+0.67
DGSDA	59.80	56.99	55.52	55.21	56.17	55.54	64.30	60.66	59.25	57.10	60.99	57.67
DGSDA+P	60.95	60.53	64.77	58.02	64.64	56.63	64.37	60.77	68.50	60.99	64.15	59.17
Δ	+1.15	+3.54	**+9.25**	+2.81	**+8.47**	+1.09	+0.07	+0.11	**+9.25**	+3.89	+3.16	+1.50
ADAlign	56.96	56.49	54.61	54.39	59.76	59.56	61.94	58.76	66.38	56.86	60.01	54.73
ADAlign+P	57.24	57.11	55.39	55.53	61.45	60.05	62.70	59.98	66.78	57.03	60.02	54.85
Δ	+0.28	+0.62	+0.78	+1.14	+1.69	+0.49	+0.76	+1.22	+0.40	+0.17	+0.01	+0.12
PALA	56.62	48.19	63.14	38.70	65.43	39.70	61.74	57.82	70.74	41.43	63.14	38.70
PALA+P	58.01	57.56	63.22	56.70	65.79	54.52	61.46	59.73	71.30	51.98	63.44	51.68
Δ	**+1.39**	**+9.37**	+0.08	**+18.00**	+0.36	**+14.82**	-0.28	+1.91	+0.56	**+10.55**	+0.30	+12.98
ImGDA	56.40	49.12	62.90	38.92	64.38	43.00	61.89	50.73	70.61	41.95	62.90	38.92
ImGDA+P	56.72	55.61	62.91	38.97	64.23	43.61	63.63	59.48	70.76	50.42	64.03	53.06
Δ	+0.32	+6.49	+0.01	+0.05	-0.15	+0.61	**+1.74**	**+8.75**	+0.15	+8.47	+1.13	**+14.14**

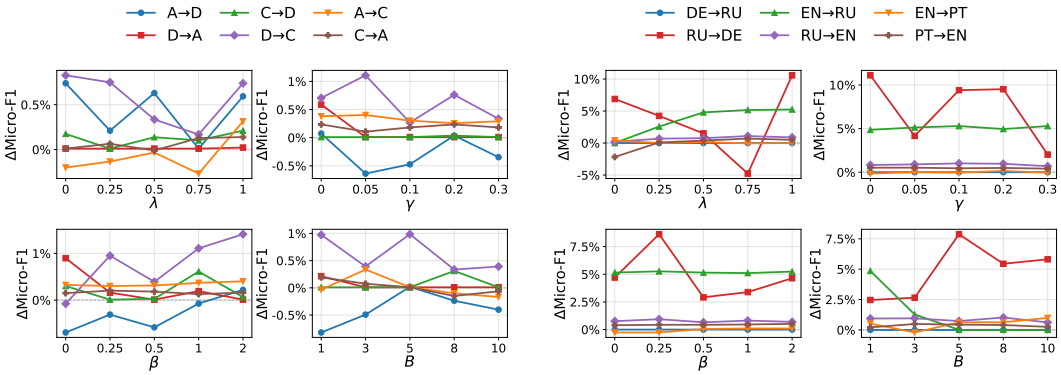
E.2 EDGE IMBALANCE ANALYSIS

Results on four additional transfer tasks are presented in Figure 1. Averaged over the 24 backbone–task combinations, ProTA yields a modest reduction in intra-domain edge imbalance, with a mean ΔI_{st} of -0.0098 and negative average changes for all six backbones. Cross-domain edge-distribution discrepancy remains nearly unchanged overall, with a mean ΔD_{st} of $+0.0012$. Consistent with these relatively small changes in edge imbalance. These results complement the larger gains reported in the main experiments and reinforce the need to evaluate intra-domain edge imbalance and cross-domain edge-distribution discrepancy jointly.



(a) Twitch (b) Citation

Figure 2: Sensitivity analysis of ProTA hyperparameters λ, γ, β, and B when integrated with TDSS on (a) six Twitch transfer tasks and (b) six Citation transfer tasks. The vertical axis reports the change in Micro-F1 relative to the TDSS backbone, where positive values indicate performance improvements after incorporating ProTA.



(a) Results on the six Citation transfer tasks. (b) Results on the six Twitch transfer tasks.

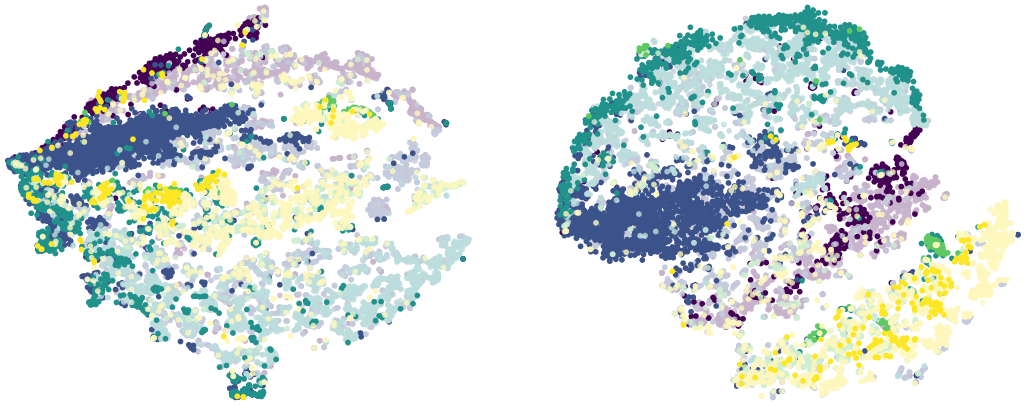
Figure 3: Sensitivity of ADAlign+ProTA to λ, γ, β, and B on Citation and Twitch transfer tasks. The vertical axis reports the Micro-F1 difference relative to the corresponding ADAlign baseline, in percentage points.

E.3 Hyperparameter Sensitivity Analysis

Figures 2 and 3 show the sensitivity of ProTA to λ, γ, β, and B when integrated with TDSS and ADAlign. Overall, ProTA maintains stable performance over a broad range of hyperparameter values, while the sensitivity varies across transfer tasks and backbones. This indicates that ProTA is generally robust to hyperparameter choices, although the preferred settings may differ across transfer scenarios.

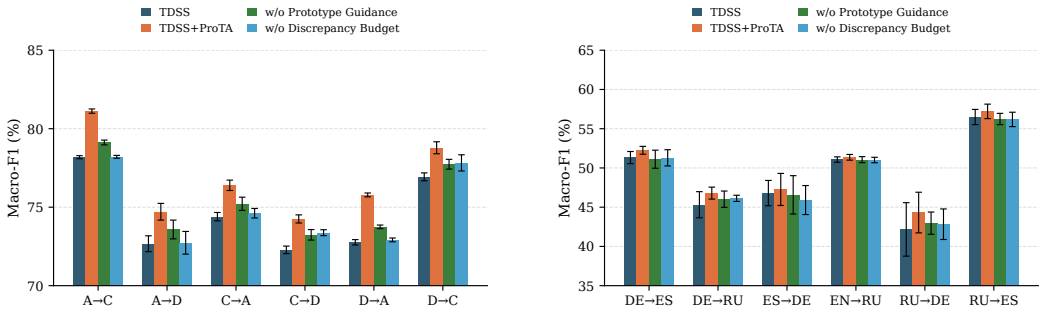
E.4 Ablation Study

E.5 Visualization



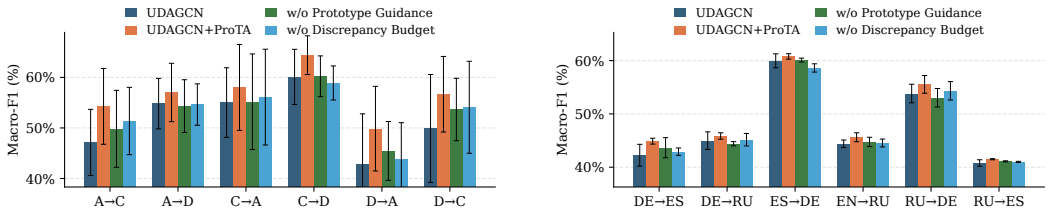
(a) PALA (b) PALA+P

Figure 6: t-SNE visualization of node representations learned by PALA and PALA+P on the DBLPv7→ACMv9 task. Different colors denote node classes, while dark and light shades of the same color represent source- and target-domain nodes, respectively.



(a) Citation (b) Twitch

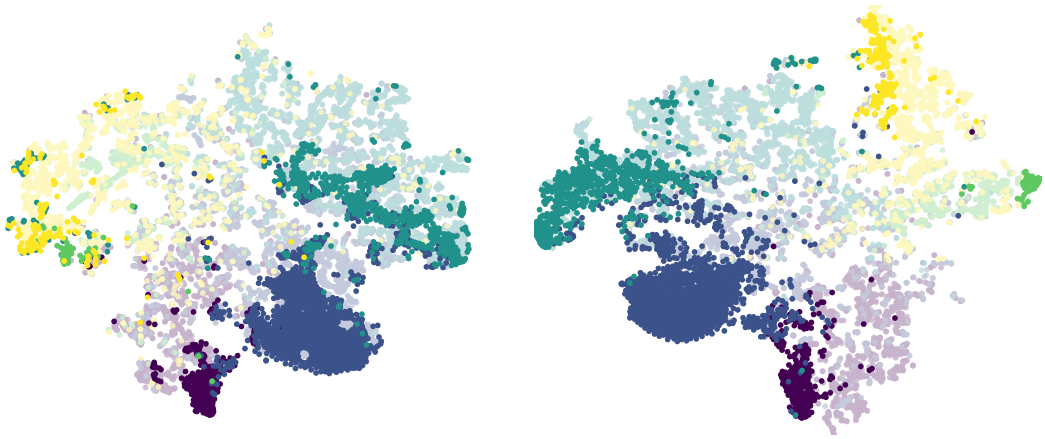
Figure 4: Macro-F1 ablation results of ProTA with TDSS on (a) six Citation transfer tasks and (b) six Twitch transfer tasks. Bars and error bars denote the mean and standard deviation over five runs.



(a) Citation transfer tasks.

(b) Twitch transfer tasks.

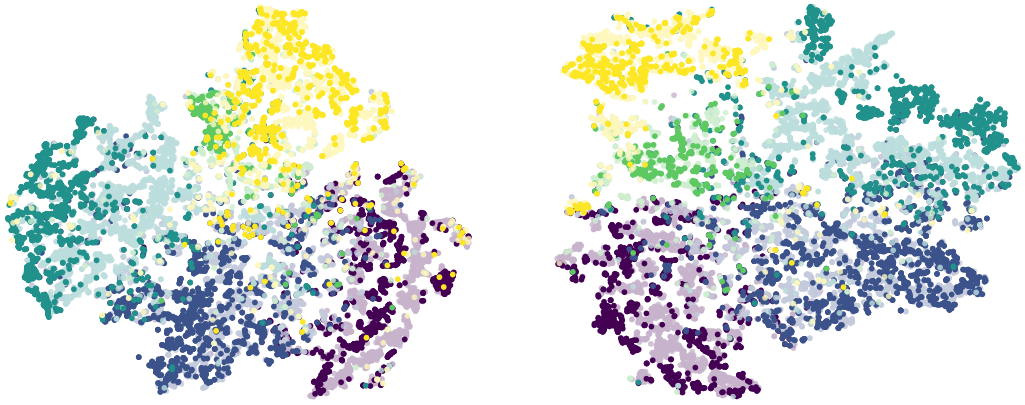
Figure 5: Ablation study of ProTA on the Citation and Twitch transfer tasks using UDAGCN as the backbone. We compare the original UDAGCN, UDAGCN equipped with the complete ProTA module, ProTA without prototype-guided candidate selection, and ProTA without discrepancy-aware edge-budget allocation. The bars report the mean Macro-F1 over five random seeds, while the error bars indicate the corresponding standard deviation. Removing either component generally degrades performance relative to the complete ProTA model, demonstrating that prototype-guided candidate selection and discrepancy-aware budget allocation both contribute to effective topology augmentation.



(a) ImGDA

(b) ImGDA+P

Figure 7: t-SNE visualization of node representations learned by ImGDA and ImGDA+P on the DBLPv7→ACMv9 task. Different colors denote node classes, while dark and light shades of the same color represent source- and target-domain nodes, respectively.



(a) ADAlign (b) ADAlign+P

Figure 8: t-SNE visualization of node representations learned by ADAlign and ADAlign+P on the DBLPv7→Citationv1 task. Different colors denote node classes, while dark and light shades of the same color represent source- and target-domain nodes, respectively.

F Future Work

The current formulation of ProTA considers source and target graphs with a shared label space and defines edge distributions using class-pair relations in homogeneous graphs. A promising direction is to extend ProTA to settings with partially shared label spaces, where domain-specific classes make direct prototype guidance and cross-domain edge-distribution comparison less reliable. This extension could identify shared semantic classes while accounting for uncertainty in domain-specific classes, allowing edge selection and budget allocation to focus on semantically compatible categories across domains. ProTA could also be generalized to heterogeneous or directed graphs by constructing relation- and direction-aware prototypes and modeling edge distributions separately for different relation types. Such extensions would enable ProTA to capture richer cross-domain topological shifts without conflating structurally distinct relations.

Table 4: Task-specific ProTA hyperparameter settings on Twitch. For a bidirectional task $A \leftrightarrow B$, each entry reports the ordered pair $(A \to B,\ B \to A)$.

Dataset	Transfer scenario	Hyper-parameter	UDAGCN	TDSS	DGSDA	ADAlign	PALA	ImGDA
Twitch	DE↔EN	γ	0/0	-0.10/0	0.3/0.02	0/-.10	0/0.02	0.20/0
		λ	0/0	0.75/0.75	0.75/0.25	1/1	0/1	0/0.75
		β	0/0	0/0	0/0	.50/0	0/0	0.25/1
		B	2/5	3/2	10/8	1/1	8/3	8/10
Twitch	DE↔ES	γ	0.02/0.1	-0.10/-0.20	0.3/0.3	.10/.10	0.2/0.03	0.20/0.20
		λ	0.75/0.75	0.75/0.50	0.75/1	.75/.75	0.75/0.1	0/0.25
		β	2/0	0/1	0/0	1/2	0.25/0.375	1/0
		B	2/5	3/1	10/10	5/3	10/2	3/10
Twitch	DE↔FR	γ	0.01/0.1	0.05/-0.10	0.3/0.2	.05/.05	0.3/0.02	0.30/0.05
		λ	0.25/0.25	0.25/0.25	0.75/0.25	.75/1	0.5/1	0.75/0.50
		β	0.25/0.5	0.25/0.50	0/0	0/.25	0.5/2	0.50/0
		B	3/2	4/1	10/8	3/3	5/5	10/5
Twitch	DE↔PT	γ	0.05/0.02	-0.10/0.10	0.2/0.1	0/0	0.2/0.02	0.10/-0.05
		λ	0/0.25	0.75/0	0.25/1	.50/1	0.75/1	1/0.75
		β	1/0	0/0	0/0	.50/0	0.5/0.5	0/2
		B	2/8	3/1	8/3	2/3	1/10	2/10
Twitch	DE↔RU	γ	0.02/0.05	-0.20/-0.10	0.3/0	-.30/.10	0/0.1	0.05/0.20
		λ	0.75/0	0.75/0.75	0.5/0.75	.75/.50	1/0.25	0.75/0.50
		β	1/1	0/0	0/0	1/1	2/1	2/0
		B	1/5	1/2	8/10	5/3	5/3	5/10
Twitch	EN↔ES	γ	0/0.01	-0.20/-0.10	0.3/0.2	0/-.10	0.2/0.1	0.05/0.02
		λ	1/0.25	0.25/0.75	0.75/0.25	1/1	0.75/0	0.50/0.50
		β	0.25/0.25	0/0.25	0/0	.25/0	0.25/1	0.25/0
		B	1/5	1/4	10/3	3/2	5/8	10/5
Twitch	EN↔FR	γ	0/0	0/-0.10	0.3/0	-.10/-.10	0.3/0.3	0.10/0
		λ	0.5/0	0.50/0.50	0.75/1	1/1	0.25/0.5	0.25/0.25
		β	0/0	0/0.50	0/0	.50/0	0.5/0.5	0.25/0
		B	5/3	3/1	10/1	3/2	5/8	8/3
Twitch	EN↔PT	γ	0/0.1	0.10/0.05	0.3/0.2	-.10/-.10	0.3/0	0.10/-0.20
		λ	0.5/0	0.50/0.75	0.75/0.25	.50/1	0/1	0.25/0
		β	0/0	1/0.50	0/0	0/.25	0.25/0.25	2/0
		B	5/2	1/2	10/8	2/3	10/8	1/1
Twitch	EN↔RU	γ	0.01/0.1	-0.20/-0.10	0.3/0.3	0/.05	-0.1/0.02	0.30/0.10
		λ	0.5/0	0.25/0.75	0.5/1	.75/1	0.25/0.5	0/0.75
		β	0/0	0.50/0	0.25/0	0/.50	0.05/0.25	1/0
		B	1/5	2/3	8/10	2/2	8/5	5/10
Twitch	ES↔FR	γ	0.02/0	-0.10/-0.10	0/0	0/0	0.3/0	0/-0.20
		λ	1/0.5	0.25/1	0.25/0.5	1/.75	0.5/0	0/0
		β	1/0.25	0/0.50	0/0	.25/0	0.25/2	1/0
		B	2/1	2/2	3/5	2/3	10/3	10/1
Twitch	ES↔PT	γ	0.05/0.05	-0.10/0	0.1/0.1	0/.05	0/0	0.10/-0.20
		λ	0.5/1	0.75/0.50	0.25/1	1/1	0/0	0/0.75
		β	0/0	0/0.50	0/0	.25/.25	0/0	0.25/2
		B	8/8	3/1	3/3	3/2	1/3	8/2
Twitch	ES↔RU	γ	0.05/0	-0.30/0	0.3/0.2	-.10/.05	0.3/0.1	0.30/0.20
		λ	0.75/0	0.25/0.75	0.5/1	1/.75	0/0.75	0.50/0.50
		β	0/0.5	0/0.50	0/0.25	0/.50	2/1	0.50/1
		B	1/1	2/1	8/12	2/3	5/10	8/8
Twitch	FR↔PT	γ	0.1/0	0.10/0.05	0/0	0/0	0/0.3	0.05/0.20
		λ	0.25/0	1/1	0.75/0	.50/.75	0.75/0.5	0/1
		β	0.5/0	0.25/0.50	0/0	.50/.25	0/0.25	0.50/2
		B	2/1	2/3	1/1	3/2	3/10	8/5
Twitch	FR↔RU	γ	0/0	0/-0.10	0/0	0/-.10	0.02/0.1	0.20/0.05
		λ	0.75/0	0.25/0.75	1/0.25	.50/.50	0.75/1	0.25/0.25
		β	0/1	0.25/0	0/0	0/0	0/2	2/0.50
		B	5/1	3/3	10/3	2/3	5/8	10/8

Continued on the next page

Table 4: Task-specific ProTA hyperparameter settings on Twitch (continued)

Dataset	Transfer scenario	Hyper-parameter	UDAGCN	TDSS	DGSDA	ADAlign	PALA	ImGDA
Twitch	PT↔RU	γ	0.1/0	-0.20/-0.10	0.3/0.2	.05/0	0.3/0	-0.05/0.30
		λ	0.5/0	0.75/0.75	0.5/0.25	.75/1	0.5/0.5	0/0
		β	0.5/0	0.25/0.25	1/0	.25/.25	1/0	1/1
		B	5/1	1/1	8/3	1/3	8/10	3/10

Table 5: Task-specific ProTA hyperparameter settings on Citation. For a bidirectional task $A \leftrightarrow B$, each entry reports the ordered pair $(A \to B,\ B \to A)$.

Dataset	Transfer scenario	Hyper-parameter	UDAGCN	TDSS	DGSDA	ADAlign	PALA	ImGDA
Citation	A↔C	γ	0/0.02	0/0	0.05/0.02	-.10/0	0.02/0	0.20/0.20
		λ	0.5/0.25	1/0.50	0.5/0	1/1	0/0	0/0.75
		β	0/0	0.75/0	0/0	.25/.50	0.5/0.125	0/0
		B	5/1	3/5	1/1	2/3	1/2	3/1
Citation	A↔D	γ	0.1/0.1	0/0.02	0/0.05	.05/-.10	0.05/0	0.10/0.10
		λ	0.5/0	0.25/0.25	0/0.75	.25/1	0/0.25	0/0
		β	1/0.5	0.25/0.25	0/0	.50/2	0/0.25	2/2
		B	1/5	1/1	10/1	5/2	1/1	8/1
Citation	C↔D	γ	0/0.1	0/0.10	0.05/0.02	-.30/.05	0/0.1	0/0.02
		λ	0.5/0	0/0.75	0/0	.25/1	0.5/0.25	0.25/0
		β	0/0.25	0.25/1	0/0	.50/0	0/0.25	0.50/1
		B	1/5	5/1	5/8	3/2	2/3	3/1

Table 6: Task-specific ProTA hyperparameter settings on MAG and Blog. For a bidirectional task $A \leftrightarrow B$, each entry reports the ordered pair $(A \to B,\ B \to A)$.

Dataset	Transfer scenario	Hyper-parameter	UDAGCN	TDSS	DGSDA	ADAlign	PALA	ImGDA
MAG	DE↔FR	γ	-0.2/-0.1	-0.30/-0.30	0/-0.3	.10/-.20	-0.1/-0.1	-0.05/-0.20
		λ	0.75/0	0.25/0.75	0.5/0	.50/.75	0.75/0.75	0.25/0
		β	1/0	0.50/0.25	0.25/0.25	1/4	0.5/0.25	1/0
		B	2/1	3/1	8/5	5/2	2/2	8/1
MAG	DE↔US	γ	-0.10/-0.20	0/-0.10	-0.20/-0.20	-0.10/0	0/-0.20	-0.10/-0.10
		λ	0.50/0.25	0.25/0.50	0.50/0.25	0.75/0.50	0.25/0.75	0.50/0
		β	0/0.25	0.25/0	0.25/0.25	0.50/0.25	0/0.50	0.25/0
		B	3/5	2/3	3/3	5/3	3/2	5/5
Blog	Blog1↔Blog2	γ	0.02/0.02	0/0.02	0/0.02	-.30/-.10	-0.2/0.02	0.30/0.30
		λ	0/0	1/0.75	1/0.75	.75/.50	1/0.75	0/1
		β	1/0.25	0/1	0/0	.25/.50	0.05/1	0.50/1
		B	2/3	1/1	1/1	3/5	1/1	8/10