From Small to Large: Cross-Scale Graph Domain Adaptation via Local-Global Structural Alignment

Ruiyi Fang[1] **Gezheng Xu**[1] **Qiuhao Zeng**[1] **Zihao Jing**[1] **Jiale Cai**[1]
Zhihao Li[1] **Hao Zheng**[2] **Xuanting Xie**[3] **Shuo Wang**[3] **Bingheng Li**[4]
Ruizhi Pu[5] **Zhao Kang**[3] **Charles Ling**[1] **Boyu Wang**[1,6,*]

[1]Western University [2]Shenzhen Campus of Sun Yat-sen University
[3]University of Electronic Science and Technology of China
[4]Michigan State University [5]Southeast University [6]Vector Institute

{rfang32,jcai336,zli3446,charles.ling}@uwo.ca ruizhipu@seu.edu.cn
zhengh239@mail.sysu.edu.cn libinghe@msu.edu zkang@uestc.edu.cn
bwang@csd.uwo.ca

Abstract

Graph domain adaptation (GDA) aims to transfer knowledge from a labeled source graph to an unlabeled target graph. Existing GDA methods typically assume that the source graph provides sufficiently complete information for reliable cross-domain alignment. However, this assumption is often unrealistic in practice, where only a small sampled source graph may be available due to privacy, annotation, storage, or data-collection constraints. In this paper, we study Cross-Scale Graph Domain Adaptation (**CSGDA**), where adaptation is performed from a limited-scale source graph to a larger target graph. Our empirical results show that CSGDA exhibits larger local and global discrepancies than conventional GDA, as the source graph provides only limited local structures and limited coverage of the global structure. To address this challenge, we propose a local-global structural alignment framework that, at the local level, augments neighborhood structures to compensate for limited local structural information. At the global level, it extracts global structural patterns and aligns source and target distributions. An adaptive GCN-MLP module then integrates the compensated local structures and extracted global patterns for domain alignment and target prediction. Experiments across multiple graph adaptation benchmarks and sampling ratios demonstrate that our method consistently improves adaptation in the presence of severe source-graph incompleteness. The source code is available at `https://github.com/RuiyiFang/LGSA`.

1 Introduction

Graphs are widely used to model relational data in real-world applications, such as citation networks, social networks, recommender systems [1, 2], traffic systems, biological systems [3], and molecules [4, 5, 6]. In many scenarios, however, obtaining node labels for a target graph is expensive or even infeasible, while labeled graphs from related domains may be available. Graph domain adaptation (GDA) has therefore been studied to transfer knowledge from a labeled source graph to an unlabeled target graph by reducing cross-domain distribution shifts [7, 8, 9].

Despite recent progress, most existing GDA methods implicitly assume that the source graph is sufficiently complete for reliable domain alignment. This assumption is often violated in practice. In practice, the available source graph often covers only a limited portion of the underlying graph, as the

*Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

full source graph may be inaccessible due to data collection, privacy [10, 11, 12], storage, annotation, or access constraints. As illustrated in Fig. 1(a), this setting leads to a new and challenging problem: adapting from a source graph with limited node and structural coverage to a larger target graph. We refer to this problem as cross-scale graph domain adaptation (**CSGDA**). CSGDA is fundamentally

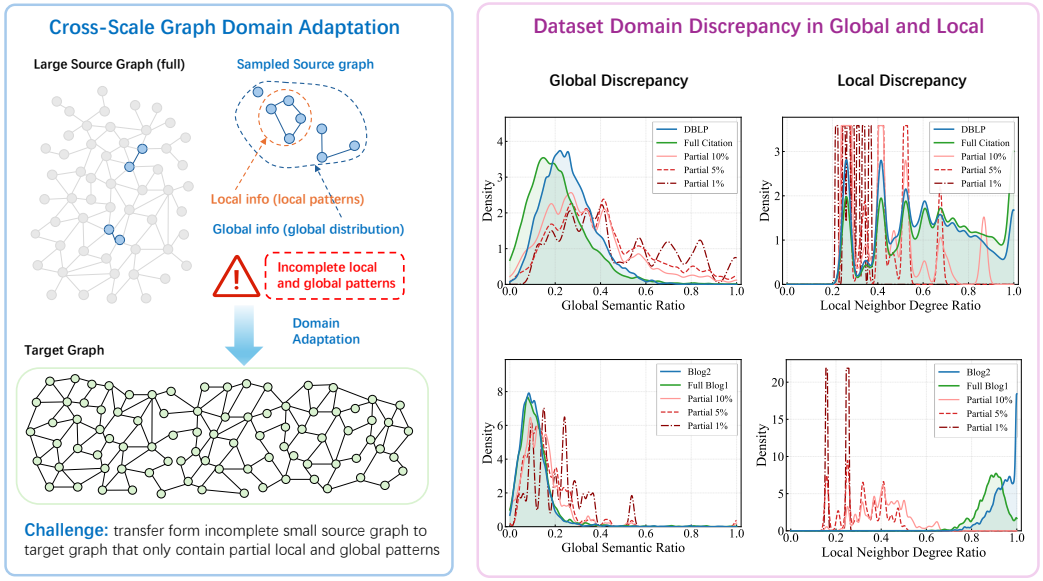


Figure 1: Overview of cross-scale graph domain adaptation.(a) CSGDA transfers knowledge from a sampled small source graph with incomplete local and global patterns to a larger target graph. (b) Cross-scale graph pairs exhibit enlarged global and local discrepancies, motivating our local–global structural alignment framework.

more difficult than conventional GDA, as the available source graph contains limited local and global structural information. At the local level, limited node-edge information in the source graph weakens the neighborhood contexts required by message passing [13, 14, 15]. At the global level, it yields sparse or biased coverage of source-side structural distributions, making cross-network alignment more challenging. To verify this challenge, we conduct a statistical analysis of cross-scale graph pairs to investigate their discrepancy[2]. As shown in Fig. 1(b), it can be observe that source graph is reduced to a limited-scale graph, both local and global information exhibit much larger discrepancies from the target graph than those observed in conventional source-target graph pairs. This makes direct source-target alignment unreliable, since the limited-scale source graph no longer provides a faithful structural basis for matching the target graph.

Motivated by this observation, we propose novel **Local-Global Structural Alignment** (LGSA), which jointly compensates for local structural incompleteness and mitigates global distributional discrepancy. LGSA contains three key modules. The local branch augments the limited-scale source graph with semantically related target nodes to compensate for missing neighborhood contexts and strengthen local structural encoding. The global branch captures sparse source-side structural distributions through structural descriptors and an MLP encoder [16, 17], avoiding over-reliance on the incomplete source adjacency. The adaptive alignment module further fuses the local and global branches according to their node-wise reliability [18, 19], enabling robust target prediction under severe source-side incompleteness. The overall framework is summarized in Fig. 2.

Our main contributions are summarized as follows:

- We introduce cross-scale graph domain adaptation (**CSGDA**), a new and challenging task that adapts from a small sampled source graph to a larger unlabeled target graph, and empirically verify that it induces more pronounced local and global discrepancies than conventional GDA.

[2]Details on the construction of project features are presented in SectionA of the Appendix.

- We propose LGSA, a local-global structural alignment framework that combines local structural augmentation with global structural representation alignment, enabling robust adaptation from limited-scale source graphs.
- Experiments on four cross-network benchmarks under source ratios of 1%, 5%, and 10% show that LGSA achieves the best accuracy in 47 of 48 settings, with gains of up to 8.6% on Citation, while on Blog all methods, LGSA included, remain close to chance.

2 Related Work

Unsupervised domain adaptation (UDA) is a widely studied transfer learning paradigm that aims to reduce the distribution discrepancy between labeled source domains and unlabeled target domains. Most UDA methods learn domain-invariant representations by mapping different domains into a shared feature space [20], while recent studies further consider challenging scenarios such as domain imbalance, label distribution shift [21, 22], fairness-aware adaptation [23, 24], and source-free domain adaptation (SFDA) [25, 26]. For graph-structured data, graph domain adaptation (GDA) has been proposed to transfer knowledge across graphs by addressing both feature and structural discrepancies [7, 27, 28]. Early methods mainly align source and target node representations. For example, ACDNE [29] captures high-order proximity with a k-hop PPMI matrix, while CDNE [30] minimizes the maximum mean discrepancy (MMD) between cross-network embeddings. Later methods exploit richer graph structures for adaptation. UDAGCN [8] captures both local and global information through dual graph convolution and adversarial training. ASN [31] separates domain-specific and domain-invariant factors to learn transferable features. SOGA [32] extends SFDA to graphs by enforcing structural consistency among target nodes, and GraphAE [33] mitigates degree-distribution-induced structural shifts by aligning router embeddings. Recent studies further investigate more fine-grained structural discrepancies in GDA [34, 35, 36]. SpecReg [37] derives an optimal-transport-based GDA bound and regularizes GNNs via spectral smoothness and maximum frequency response. JHGDA [9] models hierarchical structural shifts across multiple graph levels. ALEX [38] constructs a label-shift-enhanced augmented graph view using low-rank adjacency approximation, while SGDA [39] introduces trainable perturbations to reduce marginal distribution shifts through adversarial learning. GraphAlign [40] studies UGDA from a data-centric perspective, generating a small transferable source graph under alignment and rescaling principles so that a standard GNN trained with ERM can better generalize to the unlabeled target graph. GAA [41] reveals node attribute discrepancy as a critical factor in GDA and proposes a cross-channel module to jointly fuse and align attribute and topology views across graphs. HGDA [42] identifies homophily discrepancy as an overlooked source of domain shift and mitigates it by using mixed filters to smooth graph signals for homophily alignment. RSGDA [43] addresses GDA under unknown homophily levels by reconstructing homophilic and heterophilic graph variants and aligning corresponding structures across domains. DGSDA [44] disentangles attribute and topology alignment in the graph spectral domain and employs Bernstein polynomial filters to flexibly align graph spectral characteristics without expensive eigen decomposition. SLOGAN [45] improves UGDA by discovering sparse causal features, performing generative interventions to break spurious cross-domain correlations, and calibrating target pseudo-labels adaptively. Beyond domain adaptation, graph representation learning has also advanced through graph transformers [46, 47, 48], graph clustering [49, 50, 51, 52], and graph foundation models [53, 35, 54].

3 Methodology

3.1 Problem Formulation

Notations. An undirected graph is denoted by $G = \{\mathcal{V}, \mathcal{E}, \mathbf{A}, \mathbf{X}, \mathbf{Y}\}$, where $\mathcal{V}$ and $\mathcal{E}$ are the node and edge sets. The adjacency matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$ encodes the graph topology ($\mathbf{A}_{ij} = 1$ iff $(i, j) \in \mathcal{E}$); the feature matrix $\mathbf{X} \in \mathbb{R}^{N \times d}$ stores d-dimensional node features; for a labeled graph, $\mathbf{Y} \in \mathbb{R}^{N \times C}$ denotes the one-hot label matrix over C classes. Throughout the paper, $\widehat{\mathbf{A}}$ denotes the symmetric normalization of the input adjacency matrix.

Definition 1 (Cross-Scale Graph Domain Adaptation)**.** Let $G^{\mathrm{S}}_{\mathrm{full}} = \{\mathcal{V}^{\mathrm{S}}_{\mathrm{full}}, \mathcal{E}^{\mathrm{S}}_{\mathrm{full}}, \mathbf{A}^{\mathrm{S}}_{\mathrm{full}}, \mathbf{X}^{\mathrm{S}}_{\mathrm{full}}, \mathbf{Y}^{\mathrm{S}}_{\mathrm{full}}\}$ denote a fully labeled source graph with $N^{\mathrm{S}}_{\mathrm{full}} = |\mathcal{V}^{\mathrm{S}}_{\mathrm{full}}|$ nodes, and let $G^{\mathrm{T}} = \{\mathcal{V}^{\mathrm{T}}, \mathcal{E}^{\mathrm{T}}, \mathbf{A}^{\mathrm{T}}, \mathbf{X}^{\mathrm{T}}\}$ denote an unlabeled target graph with $N^{\mathrm{T}} = |\mathcal{V}^{\mathrm{T}}|$ nodes. In CSGDA, the source-side supervision

available at training time is not the entire $G^{\mathrm{S}}_{\mathrm{full}}$, but a node-induced subgraph

$$G^{\mathrm{S}} = \mathcal{S}_\rho(G^{\mathrm{S}}_{\mathrm{full}}) = \{\mathcal{V}^{\mathrm{S}}, \mathcal{E}^{\mathrm{S}}, \mathbf{A}^{\mathrm{S}}, \mathbf{X}^{\mathrm{S}}, \mathbf{Y}^{\mathrm{S}}\}, \qquad \rho = N^{\mathrm{S}}/N^{\mathrm{S}}_{\mathrm{full}} \in (0, 1], \tag{1}$$

where $\mathcal{S}_\rho$ denotes a node-induced subgraph sampling operator that selects source nodes and retains the edges, features, and labels induced on the sampled node set, and details can be found in Appendix B. We focus on the cross-scale regime $N^{\mathrm{S}} \ll N^{\mathrm{T}}$ induced by a small sample rate ρ. The goal is to learn a node classifier $f_\theta : (\mathbf{A}, \mathbf{X}) \mapsto \widehat{\mathbf{Y}}$ using only G^{S} and G^{T}, and to predict $\widehat{\mathbf{Y}}^{\mathrm{T}} \in \mathbb{R}^{N^{\mathrm{T}} \times C}$. In our experiments, we instantiate this cross-scale regime with $\rho \in \{1\%, 5\%, 10\%\}$, corresponding to increasingly limited source-side supervision.

3.2 Framework Overview

We propose **LGSA**, a two-branch framework for CS-GDA, illustrated in Fig. 2. The **local branch** compensates for the missing neighborhood structure of G^{S} by augmenting it with cross-domain edges to semantically similar target-side nodes, propagating source labels along these edges as input signals, and encoding the augmented graph with a GCN. The **global branch** extracts per-node structural descriptors and encodes them with an MLP without graph message passing, thereby decoupling representation learning from the incomplete source adjacency at the propagation level, and aligns the descriptor distributions across domains. An **adaptive GCN–MLP alignment** module then estimates per-node reliability of each branch, fuses the two embeddings via an adaptive gate [55], and enforces local-global consistency on target-side embeddings, while confident pseudo-labels provide additional target supervision for the fused classifier. Superscripts S and T denote source and target domain quantities, respectively.

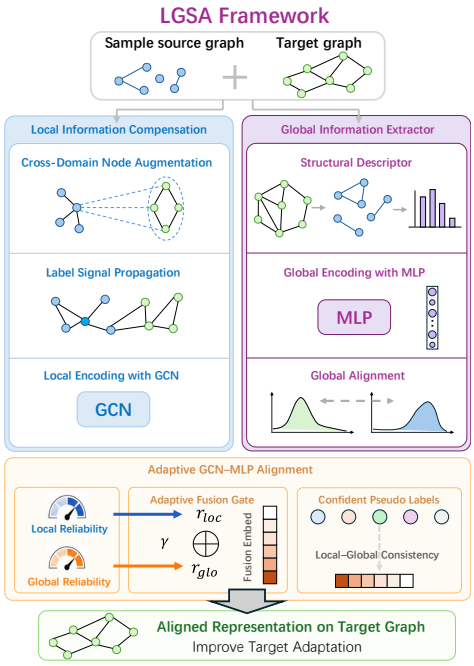


Figure 2: Overview of **LGSA**: a local GCN branch and a global MLP branch fused by an adaptive alignment module.

3.3 Local Information Compensation

Cross-Domain Node Augmentation. Because G^{S} retains only a small fraction of $G^{\mathrm{S}}_{\mathrm{full}}$, the local topology around each source node can be severely incomplete. For each $v_s \in \mathcal{V}^{\mathrm{S}}$, we augment its neighborhood [56, 57] with the top-M target nodes by feature-space cosine similarity (M is a hyperparameter):

$$\mathcal{N}_{\mathrm{aug}}(v_s) = \mathrm{TopM}_{v_t \in \mathcal{V}^{\mathrm{T}}} \cos\big(\mathbf{x}^{\mathrm{S}}_{v_s}, \mathbf{x}^{\mathrm{T}}_{v_t}\big)\,. \tag{2}$$

The augmented edge set and the augmented source-side graph are

$$\mathcal{E}_{\mathrm{aug}} = \big\{\{v_s, v_t\} : v_s \in \mathcal{V}^{\mathrm{S}},\ v_t \in \mathcal{N}_{\mathrm{aug}}(v_s)\big\}, \qquad \widetilde{G}^{\mathrm{S}} = \{\widetilde{\mathcal{V}}^{\mathrm{S}}, \mathcal{E}^{\mathrm{S}} \cup \mathcal{E}_{\mathrm{aug}}, \widetilde{\mathbf{A}}^{\mathrm{S}}, \widetilde{\mathbf{X}}^{\mathrm{S}}\}, \tag{3}$$

where $\widetilde{\mathcal{V}}^{\mathrm{S}} = \mathcal{V}^{\mathrm{S}} \cup \bigcup_{v_s} \mathcal{N}_{\mathrm{aug}}(v_s)$, the adjacency $\widetilde{\mathbf{A}}^{\mathrm{S}} \in \mathbb{R}^{|\widetilde{\mathcal{V}}^{\mathrm{S}}| \times |\widetilde{\mathcal{V}}^{\mathrm{S}}|}$ is induced by $\mathcal{E}^{\mathrm{S}} \cup \mathcal{E}_{\mathrm{aug}}$, and $\widetilde{\mathbf{X}}^{\mathrm{S}}$ stacks the corresponding features.

Label Signal Propagation. To inject source supervision into the augmented neighborhood, we propagate label signals [58, 59] on $\widetilde{\mathbf{A}}^{\mathrm{S}}$ for T iterations, initialized by the source labels $\mathbf{Y}^{\mathrm{S}}$ on the original source nodes and by zero vectors on the augmented target-side nodes. The propagation uses a teleport coefficient $\alpha \in [0, 1]$ to preserve the original source supervision during diffusion, and produces the soft label-signal matrix $\widetilde{\mathbf{Y}}^{\mathrm{S}}$ over $\widetilde{\mathcal{V}}^{\mathrm{S}}$. The rows of $\widetilde{\mathbf{Y}}^{\mathrm{S}}$ over augmented target-side nodes serve as soft label signals, while the original source labels $\mathbf{Y}^{\mathrm{S}}$ are retained for supervised training. $\widetilde{\mathbf{Y}}^{\mathrm{S}}$ enters the local branch **only** as an input-side signal on $\widetilde{G}^{\mathrm{S}}$.

Local Encoding with GCN. An L-layer GCN with parameters $\{\mathbf{W}_{\text{loc}}^{(l)}\}_{l=0}^{L-1}$ shared across domains encodes both graphs:

$$\mathbf{H}^{(l+1)} = \mathrm{ReLU}\Big(\widehat{\mathbf{A}}\mathbf{H}^{(l)}\mathbf{W}_{\text{loc}}^{(l)}\Big), \qquad (\widehat{\mathbf{A}}, \mathbf{H}^{(0)}) \in \big\{(\widehat{\widetilde{\mathbf{A}}}^{\mathrm{S}}, [\widetilde{\mathbf{X}}^{\mathrm{S}} \,\|\, \widetilde{\mathbf{Y}}^{\mathrm{S}}]),\ (\widehat{\mathbf{A}}^{\mathrm{T}}, [\mathbf{X}^{\mathrm{T}} \,\|\, \mathbf{0}])\big\}. \tag{4}$$

For the target-side forward pass, zero label-signal channels are appended to keep the input dimension consistent. We retain the original source-node embeddings from the source-side forward pass as $\mathbf{H}_{\text{loc}}^{\mathrm{S}}$ and discard the augmented target-side outputs. The target embeddings $\mathbf{H}_{\text{loc}}^{\mathrm{T}}$ are obtained from the forward pass on G^{T}. A shared linear classifier produces the local logits $\mathbf{Z}_{\text{loc}}^{\mathrm{S}}$ and $\mathbf{Z}_{\text{loc}}^{\mathrm{T}}$.

Local Loss Alignment. Let $\bar{\mathbf{h}}_{\text{loc},c}^{\mathrm{S}}$ and $\bar{\mathbf{h}}_{\text{loc},c}^{\mathrm{T}}$ denote the class-c embedding prototypes computed on the source (using $\mathbf{Y}^{\mathrm{S}}$) and on the confident target subset $\mathcal{V}_{\text{conf}}^{\mathrm{T}}$ (using current pseudo labels; see §3.5). The local branch is trained by source CE plus class-prototype matching:

$$\mathcal{L}_{\text{loc}} = \mathrm{CE}\big(\mathbf{Z}_{\text{loc}}^{\mathrm{S}},\, \mathbf{Y}^{\mathrm{S}}\big) + \sum_{c=1}^{C} \big\|\bar{\mathbf{h}}_{\text{loc},c}^{\mathrm{S}} - \bar{\mathbf{h}}_{\text{loc},c}^{\mathrm{T}}\big\|_2^2. \tag{5}$$

Classes without confident target pseudo labels skip the corresponding prototype term.

3.4 Global Information Extractor

Sampling distorts not only local neighborhoods but also node-level structural statistics: degree and k-hop neighborhood size are systematically smaller on G^{S} than on G^{T}. The global branch encodes such statistics as a structural descriptor $\mathbf{\Phi}$ and feeds it, together with node features, into an MLP. Without message passing over the incomplete adjacency, representation learning is decoupled from edge incompleteness at the propagation level, and the mean-embedding alignment below directly absorbs the sampling-induced first-moment shift in $\mathbf{\Phi}$.

Structural Descriptor. For each node v, we stack four scalar topological statistics into $\boldsymbol{\phi}(v) = \big(\deg(v),\, c(v),\, n_2(v),\, n_3(v)\big)^{\top}$, where $\deg(v)$ is the degree, $c(v)$ the clustering coefficient, and $n_k(v)$ the size of the k-hop neighborhood. Stacking $\boldsymbol{\phi}(v)$ over the source and target node sets yields $\mathbf{\Phi}^{\mathrm{S}}$ and $\mathbf{\Phi}^{\mathrm{T}}$.

Global Encoding with MLP. A shared MLP encodes node features concatenated with descriptors:

$$\mathbf{H}_{\text{glo}}^{\mathrm{S}} = \mathrm{MLP}\big([\mathbf{X}^{\mathrm{S}} \,\|\, \mathbf{\Phi}^{\mathrm{S}}]\big), \qquad \mathbf{H}_{\text{glo}}^{\mathrm{T}} = \mathrm{MLP}\big([\mathbf{X}^{\mathrm{T}} \,\|\, \mathbf{\Phi}^{\mathrm{T}}]\big), \tag{6}$$

where $[\,\cdot \,\|\, \cdot\,]$ denotes column-wise concatenation. A linear classifier $(\mathbf{W}_{\text{glo}}^{\text{cls}}, \mathbf{b}_{\text{glo}}^{\text{cls}})$ shared across domains produces the global logits $\mathbf{Z}_{\text{glo}}^{\mathrm{S}}$ and $\mathbf{Z}_{\text{glo}}^{\mathrm{T}}$.

Global Alignment Loss. Letting $\bar{\mathbf{h}}_{\text{glo}}^{\mathrm{S}}$ and $\bar{\mathbf{h}}_{\text{glo}}^{\mathrm{T}}$ denote the mean global embeddings over $\mathcal{V}^{\mathrm{S}}$ and $\mathcal{V}^{\mathrm{T}}$, the global branch is trained by source CE plus marginal-mean alignment:

$$\mathcal{L}_{\text{glo}} = \mathrm{CE}\big(\mathbf{Z}_{\text{glo}}^{\mathrm{S}},\, \mathbf{Y}^{\mathrm{S}}\big) + \big\|\bar{\mathbf{h}}_{\text{glo}}^{\mathrm{S}} - \bar{\mathbf{h}}_{\text{glo}}^{\mathrm{T}}\big\|_2^2. \tag{7}$$

3.5 Adaptive GCN–MLP Alignment

Both branches now produce embeddings on the same node sets $\mathcal{V}^{\mathrm{S}}$ and $\mathcal{V}^{\mathrm{T}}$, which we adaptively fuse using a per-node reliability estimate.

Branch Reliability Estimation. For each node v, branch reliability is given by entropy-normalized confidence [60, 61]:

$$r_{\text{loc},v} = 1 - \frac{H(\mathrm{softmax}(\mathbf{z}_{\text{loc},v}))}{\log C}, \qquad r_{\text{glo},v} = 1 - \frac{H(\mathrm{softmax}(\mathbf{z}_{\text{glo},v}))}{\log C}, \tag{8}$$

where $H(\cdot)$ is the Shannon entropy.

Adaptive Fusion Gate. The fusion weight is generated by a sigmoid gate [62, 63] over the reliability gap: $\gamma_v = \sigma(\mathrm{softplus}(\omega_g)\,(r_{\text{loc},v} - r_{\text{glo},v}) + b_g)$, with learnable scalars ω_g, b_g; the softplus parameterization keeps the gate monotonically increasing in the local-branch reliability advantage. The fused embedding and logit at node v are $\mathbf{h}_v = \gamma_v \mathbf{h}_{\text{loc},v} + (1-\gamma_v)\mathbf{h}_{\text{glo},v}$ and $\mathbf{z}_v = \mathbf{h}_v \mathbf{W}_o + \mathbf{b}_o$,

where $(\mathbf{W}_o, \mathbf{b}_o)$ is an independent fusion classifier. Stacking $\mathbf{z}_v$ over $\mathcal{V}^{\mathrm{S}}$ and $\mathcal{V}^{\mathrm{T}}$ gives $\mathbf{Z}^{\mathrm{S}}$ and $\mathbf{Z}^{\mathrm{T}}$. Since pseudo labels can be noisy [26, 64], we retain only confident target nodes and their hard pseudo labels:

$$\mathcal{V}^{\mathrm{T}}_{\mathrm{conf}} = \big\{v \in \mathcal{V}^{\mathrm{T}} : \max_c \big[\mathrm{softmax}(\mathbf{z}^{\mathrm{T}}_v)\big]_c \geq \tau_p\big\}, \qquad \widetilde{y}^{\mathrm{T}}_v = \arg\max_c (\mathbf{z}^{\mathrm{T}}_v)_c, \quad v \in \mathcal{V}^{\mathrm{T}}_{\mathrm{conf}}. \tag{9}$$

Both $\mathcal{V}^{\mathrm{T}}_{\mathrm{conf}}$ and the pseudo labels $\{\widetilde{y}^{\mathrm{T}}_v\}_{v\in\mathcal{V}^{\mathrm{T}}_{\mathrm{conf}}}$ are refreshed once per epoch using detached predictions, so that gradients are not back-propagated through the pseudo-label selection process. The same pseudo-labels are used to compute $\bar{\mathbf{h}}^{\mathrm{T}}_{\mathrm{loc},c}$ in §3.3. Let $\widetilde{\mathbf{Y}}^{\mathrm{T}}_{\mathrm{conf}}$ denote the corresponding pseudo-label vector/matrix obtained by stacking $\{\widetilde{y}^{\mathrm{T}}_v\}_{v\in\mathcal{V}^{\mathrm{T}}_{\mathrm{conf}}}$. The fused logits are supervised by source labels and confident target pseudo labels:

$$\mathcal{L}_{\mathrm{fuse}} = \mathrm{CE}\big(\mathbf{Z}^{\mathrm{S}},\, \mathbf{Y}^{\mathrm{S}}\big) + \mathrm{CE}\Big(\mathbf{Z}^{\mathrm{T}}_{\mathrm{conf}},\, \widetilde{\mathbf{Y}}^{\mathrm{T}}_{\mathrm{conf}}\Big), \tag{10}$$

where $\mathbf{Z}^{\mathrm{T}}_{\mathrm{conf}}$ stacks the rows of $\mathbf{Z}^{\mathrm{T}}$ indexed by $\mathcal{V}^{\mathrm{T}}_{\mathrm{conf}}$; the second term is omitted when $\mathcal{V}^{\mathrm{T}}_{\mathrm{conf}} = \varnothing$.

Local–Global Consistency. To prevent the two branches from drifting on the unlabeled target graph, we enforce contrastive consistency [65, 66, 67] between their target-side embeddings. Each branch embedding is mapped through a 2-layer projection head and ℓ_2-normalized:

$$\mathbf{q}^{\mathrm{T}}_{\mathrm{loc},v} = \frac{g_{\mathrm{loc}}(\mathbf{h}^{\mathrm{T}}_{\mathrm{loc},v})}{\|g_{\mathrm{loc}}(\mathbf{h}^{\mathrm{T}}_{\mathrm{loc},v})\|_2}, \qquad \mathbf{q}^{\mathrm{T}}_{\mathrm{glo},v} = \frac{g_{\mathrm{glo}}(\mathbf{h}^{\mathrm{T}}_{\mathrm{glo},v})}{\|g_{\mathrm{glo}}(\mathbf{h}^{\mathrm{T}}_{\mathrm{glo},v})\|_2}. \tag{11}$$

Treating the two views of the same target node as a positive pair [68, 69, 70] (other target nodes as negatives), the InfoNCE consistency loss is

$$\mathcal{L}_{\mathrm{cons}} = -\frac{1}{|\mathcal{V}^{\mathrm{T}}|}\sum_{v\in\mathcal{V}^{\mathrm{T}}} \log \frac{\exp\big(\mathbf{q}^{\mathrm{T}\top}_{\mathrm{loc},v}\mathbf{q}^{\mathrm{T}}_{\mathrm{glo},v}/\tau_c\big)}{\sum_{u\in\mathcal{V}^{\mathrm{T}}}\exp\big(\mathbf{q}^{\mathrm{T}\top}_{\mathrm{loc},v}\mathbf{q}^{\mathrm{T}}_{\mathrm{glo},u}/\tau_c\big)}, \tag{12}$$

where τ_c is the contrastive temperature and the denominator includes $u = v$.

3.6 Overall Training Objective

The full objective combines the local and global branches with a balance parameter μ, together with fused supervision and target-side consistency:

$$\mathcal{L} = \mathcal{L}_{\mathrm{fuse}} + \mu\mathcal{L}_{\mathrm{glo}} + (1-\mu)\mathcal{L}_{\mathrm{loc}} + \eta\mathcal{L}_{\mathrm{cons}}, \tag{13}$$

where μ and η are trade-off hyper-parameters. The parameters of the whole framework are updated via backpropagation.

4 Experiments

We empirically evaluate **LGSA** on the standard cross-network node classification benchmarks under our **cross-scale** setting, where only a small fraction of the source graph is observed at training time, while the target graph is fully accessible (without labels). We aim to answer three research questions: **(RQ1)** How does **LGSA** compare with state-of-the-art GDA methods when the source graph is reduced to 1%, 5%, or 10% of its original size? **(RQ2)** What is the contribution of each component in **LGSA**? **(RQ3)** How sensitive is **LGSA** to its hyperparameters?

We evaluate on four groups of GDA benchmarks: Airport, Citation, Social, and ACM [71]. Airport contains the USA (U), Brazil (B), and Europe (E) air-traffic networks, where nodes are airports, edges are flight routes, and labels are four activity levels; these graphs have no native node attributes, so input features are derived from the graph structure. Citation includes DBLPv7 (D), ACMv9 (A), and Citationv1 (C), three citation networks collected from different sources and time periods, with bag-of-words paper features and five research-area labels. Social consists of two disjoint BlogCatalog networks, Blog1 (B1) and Blog2 (B2), where nodes are bloggers, edges are friendships, and labels are six interest groups. To examine homophily distribution shift, we also use two graphs built on the same set of ACM papers with three research-field labels: ACM3 (A3) links papers that share a subject (Paper-Subject-Paper) [72], and ACM4 (A4) links papers that share an author (Paper-Author-Paper) [73]. Because A3 and A4 share nodes, features, and labels, transfer between them isolates the

Table 1: Cross-scale source statistics on Citation and Blog at ratio r.

Source → Target	Full Source N	Full Source E	Target N	Target E	r=1% N	r=1% E	r=5% N	r=5% E	r=10% N	r=10% E
D → A	5,484	16,234	9,360	31,112	55	42	274	248	548	658
D → C	5,484	16,234	8,935	30,196	55	42	274	248	548	658
A → D	9,360	31,112	5,484	16,234	94	78	468	462	936	1,248
A → C	9,360	31,112	8,935	30,196	94	78	468	462	936	1,248
C → D	8,935	30,196	5,484	16,234	89	72	447	432	894	1,186
C → A	8,935	30,196	9,360	31,112	89	72	447	432	894	1,186
B1 → B2	2,300	66,942	2,896	107,672	23	38	115	712	230	2,342
B2 → B1	2,896	107,672	2,300	66,942	29	64	145	1,186	290	3,768

shift in edge structure and homophily. Every ordered pair of graphs within a group forms a transfer task, giving 16 tasks in total. Detailed statistics are deferred to Appendix H. For each task, we keep the target graph complete and unlabeled and replace the source with a sampled subgraph that retains a fraction $r \in \{1\%, 5\%, 10\%\}$ of the original source nodes. The node budget is split evenly between two samplers. Half is filled by breadth-first expansion from K random seed nodes, which preserves a few dense local neighborhoods; the other half is filled by depth-first probes from random seeds, which scatter nodes across the graph and broaden class and feature coverage. The source graph is induced on the sampled nodes, so an edge is kept only when both of its endpoints are sampled. The two samplers reflect two common ways partial graphs are collected in practice: crawling outward from a few seed entities, and gathering records scattered across the graph. The choice of K is discussed in Appendix C.3.

4.1 Experimental Setup

All experiments are implemented in PyTorch on an Intel(R) Xeon(R) Silver 4210R CPU @ 2.40 GHz with a single NVIDIA RTX A5000 24 GB GPU. We train **LGSA** using the Adam optimiser [74] with learning rate in $[10^{-4}, 5\times10^{-4}]$ and weight decay in $[10^{-4}, 5\times10^{-3}]$. Both the GCN and MLP branches are two layers wide with hidden width 128. The projection heads used for local–global consistency have projection dimension 16, and dropout is set to 0.5. We repeatedly sampled our graph 10 times and trained and tested our model five times with the same partition of each dataset, and then reported the average of ACC.

5 Baselines

We compare against six representative methods spanning two families: (i) a graph encoder without domain alignment, **GCN** [75]; and (ii) state-of-the-art GDA methods **UDAGCN** [8], **GRADE-N** [76], **JHGDA** [9], **SpecReg** [37], **HGDA** [42]. For fairness, every baseline is provided with the **same** sampled source $\widetilde{G}_s$ and full target G^T that **LGSA** uses; no method receives privileged access to the un-sampled portion of G_s.

Cross-scale source construction. For every transfer task $G^{\mathrm{S}}_{\mathrm{full}} \to G^{\mathrm{T}}$, we keep G^{T} fully observed and construct the available source graph G^{S} by sampling a fraction $\rho \in \{1\%, 5\%, 10\%\}$ of the original source nodes. To preserve both **local cohesion** and **global coverage**, sampling follows a 50:50 hybrid of breadth-first search (BFS) and depth-first / random node selection: half of the budget grows as K BFS clusters rooted at randomly seeded nodes (yielding dense local sub-graphs that retain structural signatures), while the remaining half is drawn as scattered DFS probes (providing diverse class and feature coverage). Edges between sampled nodes that exist in G^S are retained to form the induced sub-graph $\widetilde{G}^S$. Table 1 summarises the resulting source sizes on the Citation (ArnetMiner), and Blog benchmarks alongside the fully-observed targets; even at r=10%, the sampled source is one to two orders of magnitude smaller than the target in both nodes and edges, creating a substantial cross-scale gap that the model must bridge. Statistics for the remaining benchmark (Airport) and a detailed study of the BFS:DFS mixing ratio are in Appendix C.

Table 2: Cross-scale node classification accuracy on the Citation and ACM networks under varying source-graph sampling ratios (1%, 5%, 10%).

Ratio	Methods	A→D	D→A	A→C	C→A	C→D	D→C	A3 → A4	A4 → A3
1%	GCN	0.279	0.261	0.275	0.313	0.271	0.295	0.291	0.312
	UDAGCN	0.316	0.310	0.328	0.314	0.315	0.325	0.323	0.311
	GRADE-N	0.310	0.299	0.333	0.304	0.312	0.341	0.315	0.313
	JHGDA	0.345	0.254	0.354	0.313	0.239	0.240	0.304	0.321
	SpecReg	0.363	0.298	0.337	0.203	0.290	0.347	0.313	0.332
	HGDA	0.379	0.312	0.351	0.315	0.322	0.349	0.342	0.382
	LGSA	**0.384**	**0.350**	**0.367**	**0.325**	**0.369**	**0.350**	**0.367**	**0.391**
5%	GCN	0.313	0.309	0.341	0.343	0.322	0.419	0.429	0.416
	UDAGCN	0.369	0.355	0.394	0.362	0.346	**0.461**	0.401	0.432
	GRADE-N	0.357	0.352	0.375	0.368	0.346	0.458	0.421	0.391
	JHGDA	0.382	0.299	0.396	0.388	0.387	0.406	0.383	0.399
	SpecReg	0.361	0.329	0.372	0.343	0.361	**0.461**	0.411	0.422
	HGDA	0.411	0.344	0.415	0.375	0.364	0.277	0.433	0.472
	LGSA	**0.429**	**0.376**	**0.428**	**0.400**	**0.414**	0.428	**0.467**	**0.493**
10%	GCN	0.519	0.491	0.552	0.520	0.512	0.535	0.504	0.595
	UDAGCN	0.577	0.558	0.602	0.579	0.553	0.575	0.581	0.597
	GRADE-N	0.589	0.542	0.625	0.583	0.554	0.589	0.544	0.601
	JHGDA	0.614	0.581	0.652	0.591	0.584	0.626	0.552	0.594
	SpecReg	0.566	0.534	0.598	0.573	0.556	0.581	0.612	0.615
	HGDA	0.607	0.585	0.650	0.605	0.579	0.618	0.597	0.601
	LGSA	**0.615**	**0.602**	**0.689**	**0.619**	**0.670**	**0.645**	**0.619**	**0.632**

Table 3: Cross-scale node classification accuracy on the Airport and Blog networks under varying source-graph sampling ratios.

Ratio	Methods	U→B	U→E	B→U	B→E	E→U	E→B	B1 → B2	B2 → B1
1%	GCN	0.142	0.178	0.184	0.184	0.212	0.196	0.192	0.186
	UDAGCN	0.131	0.188	0.195	0.202	0.223	0.184	0.123	0.174
	GRADE-N	0.124	0.191	0.213	0.228	0.209	0.261	0.187	0.116
	JHGDA	0.116	0.228	0.239	0.193	0.297	0.250	0.161	0.121
	SpecReg	0.128	0.203	0.215	0.204	0.299	0.242	0.137	0.181
	HGDA	0.211	0.214	0.226	0.231	0.213	0.193	0.183	0.141
	LGSA	**0.232**	**0.244**	**0.252**	**0.252**	**0.318**	**0.275**	**0.202**	**0.200**
5%	GCN	0.310	0.305	0.286	0.198	0.205	0.278	0.155	0.171
	UDAGCN	0.307	0.317	0.230	0.154	0.275	0.223	0.174	0.191
	GRADE-N	0.310	0.348	0.285	0.195	0.199	0.275	0.163	0.198
	JHGDA	0.366	0.320	0.298	0.177	0.276	0.201	0.176	0.179
	SpecReg	0.342	0.338	0.311	0.201	0.225	0.290	0.151	0.203
	HGDA	0.312	0.316	0.289	0.176	0.251	0.294	0.145	0.208
	LGSA	**0.377**	**0.363**	**0.315**	**0.228**	**0.290**	**0.305**	**0.178**	**0.216**
10%	GCN	0.250	0.288	0.292	0.220	0.313	0.325	0.117	0.192
	UDAGCN	0.300	0.265	0.349	0.276	0.341	0.294	0.102	0.211
	GRADE-N	0.290	0.311	0.391	0.222	0.303	0.272	0.118	0.188
	JHGDA	0.244	0.314	0.390	0.219	0.348	0.310	0.163	0.211
	SpecReg	0.318	0.325	0.333	0.270	0.301	0.319	0.163	0.225
	HGDA	0.314	0.313	0.374	0.296	0.319	0.317	0.113	0.215
	LGSA	**0.326**	**0.359**	**0.405**	**0.308**	**0.366**	**0.326**	**0.175**	**0.228**

5.1 Cross-scale Cross-network Node Classification (RQ1)

Tables 2 and 3 report ACC under source ratios $r \in \{1\%, 5\%, 10\%\}$; in each column, the best result is in bold and the second best is underlined. Three observations can be made. First, CSGDA is challenging for existing methods. On Citation, baseline accuracy drops from 0.49–0.65 at $r{=}10\%$ to 0.20–0.38 at $r{=}1\%$, and on Airport at $r{=}1\%$, 32 of the 36 baseline results fall below the four-class chance level of 0.25. This confirms that the sampled source graph is not merely smaller, but incomplete in both nodes and induced edges, making source–target alignment substantially harder. Second, **LGSA** is the best method in 47 of the 48 transfer–ratio settings. On Citation, its average

margin over the strongest baseline of each transfer is 1.9%, 1.0%, and 2.9% at $r{=}1\%$, 5%, and 10%. The gain is largest at $r{=}10\%$, where the average rises from 0.611 to 0.640, **LGSA** reaches 0.689 on A→C, and the largest single gain is on C→D ($0.584 \rightarrow 0.670$). The only exception is D→C at $r{=}5\%$, where UDAGCN and SpecReg reach 0.461 against 0.428 for **LGSA**. On ACM3↔ACM4, which exhibits homophily shift, **LGSA** is best at every ratio by 0.7%–3.4%. Third, the gains are smaller on Airport and Blog. On Airport, which has no native node features, **LGSA** is best in all 18 settings, but its average margin per ratio is only 1.4%–1.7%, and at $r{=}1\%$ it reaches only 0.232–0.318, close to chance. On Blog, every method stays close to the six-class chance level of $1/6 \approx 0.167$ at every ratio, and several baselines fall below it; **LGSA** is nominally best (0.175–0.228), so we do not read the Blog results as evidence of successful adaptation for any method.

5.2 Ablation Study (RQ2)

We compare the full **LGSA** with three ablated variants on the Airport benchmark across $\rho \in \{1\%, 5\%, 10\%\}$; Detailed results on the other benchmarks, which exhibit the same trends, are provided in Appendix G.

- **LGSA**$_1$ removes cross-domain node augmentation and label-signal injection (Eq. 3), reducing the local branch to a vanilla GCN over the truncated G^{S}.
- **LGSA**$_2$ removes the global branch (the MLP and the structural descriptor $\boldsymbol{\phi}(v)$); prediction is then read off the local logits.
- **LGSA**$_3$ replaces the entropy-based fusion gate γ_v with uniform 0.5 averaging and removes the consistency loss $\mathcal{L}_{\mathrm{cons}}$.

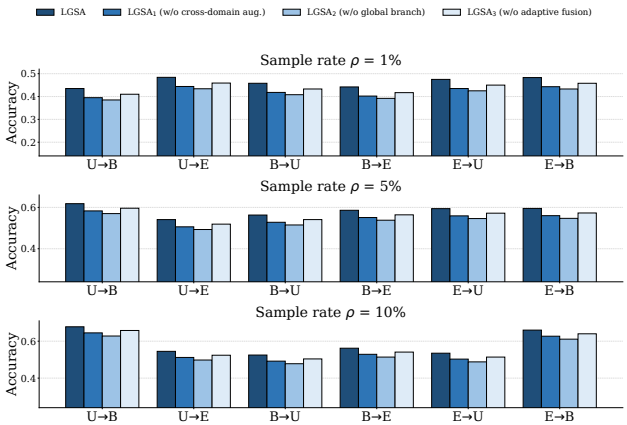
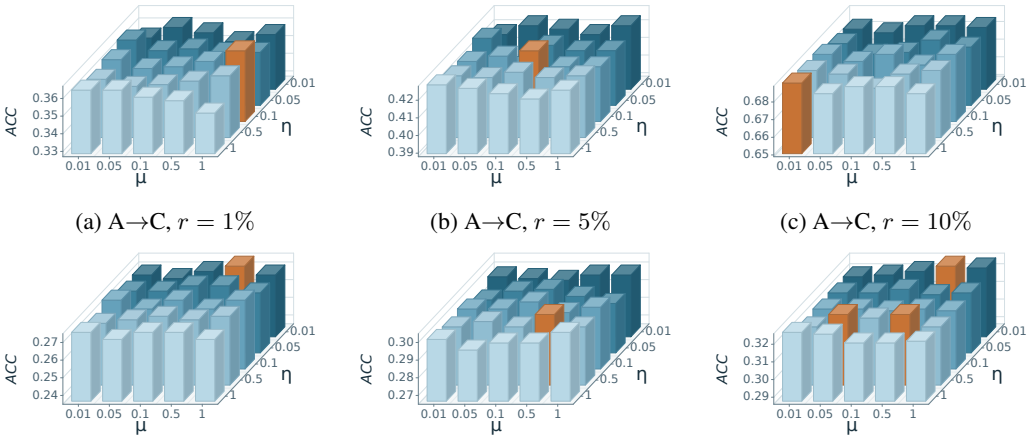


Figure 3: Ablation study on the Airport benchmark across sample rates $\rho \in \{1\%, 5\%, 10\%\}$. The full **LGSA** dominates every variant on every transfer, and the gap of each component widens monotonically as ρ shrinks.

Fig. 3 shows that the full model dominates every variant on every transfer. Removing the global branch (LGSA$_2$) is the most damaging ($\sim$4.9% average drop), confirming that $\boldsymbol{\phi}(v)$ and its mean-embedding alignment are indispensable for absorbing the sampling-induced first-moment shift on $\boldsymbol{\Phi}$. Removing cross-domain augmentation (LGSA$_1$) and adaptive fusion (LGSA$_3$) cause smaller but consistent drops of $\sim 3.6\%$ and $\sim 2.3\%$, respectively. A monotone trend further emerges: every component contributes more as ρ shrinks —from $\rho{=}10\%$ to 1%, the gaps widen from 4.8% to 5.0% (global branch), 3.3% to 4.0% (augmentation), and 2.1% to 2.5% (fusion), showing that two-branch compensation is most beneficial precisely where CSGDA is hardest.

5.3 Parameter Analysis (RQ3)

In Eq. (13), $\mathcal{L}_{\mathrm{fuse}}$ serves as the anchor with unit weight, so **LGSA** only requires tuning two loss-level hyperparameters: the local–global balance $\mu \in [0, 1]$, which shifts weight between $\mathcal{L}_{\mathrm{glo}}$ and $\mathcal{L}_{\mathrm{loc}}$, and the consistency weight η. To examine how sensitive performance is to these choices, we jointly sweep them on Citation (A→C) and Airport (E→B) under $r \in \{1\%, 5\%, 10\%\}$ over a 5×5 grid $\mu, \eta \in \{0.01, 0.05, 0.1, 0.5, 1\}$. Figure 4 reports the resulting accuracy surfaces. All six response surfaces in Figure 4 are nearly flat, with the best-worst gap in each panel within $\sim 1{-}2\%$, showing that **LGSA** is largely insensitive to the exact loss weighting. The optimum shifts only mildly with the sampling ratio, and the shift is interpretable. When $r{=}1\%$, the sampled source graph is highly sparse and locally truncated, so neighborhood aggregation in the local GCN suffers from missing edges; a relatively larger μ helps stabilize training by leaning on the global MLP branch, which uses the structural descriptor $\boldsymbol{\phi}(v)$ and is unaffected by edge truncation. As r increases to 10%, more source nodes and induced edges become available, the local GCN recovers more meaningful neighborhoods, and a smaller μ suffices. For the consistency weight, moderate values $\eta \in [0.05, 0.1]$ are consistently



(a) A→C, $r = 1\%$ (b) A→C, $r = 5\%$ (c) A→C, $r = 10\%$

(d) E→B, $r = 1\%$ (e) E→B, $r = 5\%$ (f) E→B, $r = 10\%$

Figure 4: Sensitivity of **LGSA** to the hyperparameters μ and η in Airport and Citation Benchmark.

effective across both datasets and all sampling ratios, while $\eta = 1$ tends to overemphasize the InfoNCE consistency term and weakens the supervised signal from $\mathcal{L}_{\text{fuse}}$. Conversely, very small values ($\eta = 0.01$) reduce $\mathcal{L}_{\text{cons}}$ to a nearly inactive regularizer, removing the cross-view alignment that ties the two branches together on the target side. Overall, the broad plateau within $\mu, \eta \in [0.05, 0.5]$ confirms that **LGSA** requires little tuning across datasets and sampling ratios, a practical advantage in the CSGDA setting where validation labels on the large target graph are unavailable.

6 Conclusion

We study cross-scale graph domain adaptation (CSGDA), where only a small sampled source graph is available for adaptation to a substantially larger unlabeled target. Source-side sampling introduces both local neighborhood truncation and global distribution distortion, undermining existing GDA methods. We propose LGSA: a local branch that compensates missing structure via cross-domain node augmentation and label-signal injection, an adjacency-decoupled MLP branch that captures sparse global patterns, and an adaptive GCN–MLP module with target-side consistency that integrates the two. Across four benchmarks and three source ratios, LGSA achieves the best accuracy in 47 of 48 settings, with the largest gains on Citation at a 10% source ratio. The gains are small on the featureless Airport graphs at 1% and on Blog, where all methods, LGSA included, remain close to chance. Characterising when structural compensation can recover enough source information to help is left as an open problem.

Limitations. Although **LGSA** leads in 47 of 48 settings, it trails two baselines on D→C at $r{=}5\%$, and 11 of its leads are below 1%, which may fall within run-to-run variation. On Blog, and on Airport at $r{=}1\%$, all methods stay near or below chance, so a nominal lead there does not indicate successful adaptation; the density and ratio thresholds below which structural compensation stops helping are not yet characterised. Beyond this, LGSA targets the closed-set, single-source CSGDA setting, where the label space is shared and only one source domain is available, and its global descriptor is restricted to four hand-crafted topological statistics, which may underrepresent more nuanced structural patterns [77, 54, 78]. Diagnosing the near-chance results on Blog, extensions to open-set [79, 80, 81, 82] and multi-source [83] variants, and richer learnable structural encodings [84, 85] are left for future work.

7 ACKNOWLEDGMENTS

This work is supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) Discovery Grants program.

References

[1] Di Jin, Luzhi Wang, Yizhen Zheng, Guojie Song, Fei Jiang, Xiang Li, Wei Lin, and Shirui Pan. Dual intent enhanced graph neural network for session-based new item recommendation. In *Proceedings of the ACM Web Conference 2023*, pages 684–693, 2023.

[2] Hourun Li, Yifan Wang, Zhiping Xiao, Jia Yang, Changling Zhou, Ming Zhang, and Wei Ju. DisCo: graph-based disentangled contrastive learning for cold-start cross-domain recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 12049–12057, 2025.

[3] Jiaming Zhuo, Shuai Zhai, Ziyi Ma, Kun Fu, Chuan Wang, Di Jin, Zhen Wang, Xiaochun Cao, Huazhu Fu, and Liang Yang. On the spectral unreachability of brain graph learning. In *Forty-third International Conference on Machine Learning*, 2026.

[4] Zihao Jing, Qiuhao Zeng, Ruiyi Fang, Yan Sun, Boyu Wang, and Pingzhao Hu. Entropy-guided dynamic tokens for graph-LLM alignment in molecular understanding. In *The Fourteenth International Conference on Learning Representations*, 2026.

[5] Zihao Jing, Qiuhao Zeng, Ruiyi Fang, Yan Yi Li, Yan Sun, Boyu Wang, and Pingzhao Hu. Scaling-aware adapter for structure-grounded LLM reasoning. In *Forty-third International Conference on Machine Learning*, 2026.

[6] Ruiyi Fang, Changhong Chen, Qiang Wang, Charles Ling, Boyu Wang, and Hong Zhang. Transfer learning with graph neural networks to predict polymer solubility parameters. *npj Computational Materials*, 2026.

[7] Xiao Shen and Fu Lai Chung. Network embedding for cross-network node classification. *arXiv preprint arXiv:1901.07264*, 2019.

[8] Man Wu, Shirui Pan, Chuan Zhou, Xiaojun Chang, and Xingquan Zhu. Unsupervised domain adaptive graph convolutional networks. In *Proceedings of The Web Conference 2020*, pages 1457–1467, 2020.

[9] Boshen Shi, Yongqing Wang, Fangda Guo, Jiangli Shao, Huawei Shen, and Xueqi Cheng. Improving graph domain adaptation with network hierarchy. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 2249–2258, 2023.

[10] Zhihao Li, Gezheng Xu, Jiale Cai, Ruiyi Fang, Di Wu, Qicheng Lao, Charles Ling, and Boyu Wang. When priors backfire: On the vulnerability of unlearnable examples to pretraining. In *The Fourteenth International Conference on Learning Representations*, 2026.

[11] Jiale Cai, Gezheng Xu, Zhihao Li, Ruiyi Fang, Ruizhi Pu, Di Wu, Qicheng Lao, Charles Ling, and Boyu Wang. FUSE: Full-spectrum unlearnable examples via spectral equalization. In *Forty-third International Conference on Machine Learning*, pages 10764–10788, 2026.

[12] Hao Zheng, Shiyu Song, Zhigang Hu, Meiguang Zheng, Liu Yang, Aikun Xu, Rongchang Zhao, Ruizhi Pu, Ruiyi Fang, and Boyu Wang. Intra-class unbiased prototype aggregation and classifier collaboration for personalized federated learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, 2026.

[13] Ruiyi Fang, Liangjian Wen, Zhao Kang, and Jianzhuang Liu. Structure-preserving graph representation learning. In *2022 IEEE International Conference on Data Mining (ICDM)*, pages 927–932. IEEE, 2022.

[14] Bingheng Li, Xuanting Xie, Haoxiang Lei, Ruiyi Fang, and Zhao Kang. Simplified PCNet with robustness. *Neural Networks*, 184:107099, 2025.

[15] Jincheng Huang, Lun Du, Xu Chen, Qiang Fu, Shi Han, and Dongmei Zhang. Robust mid-pass filtering graph convolutional networks. In *Proceedings of the ACM Web Conference 2023*, pages 328–338, 2023.

[16] Xuanting Xie, Bingheng Li, Erlin Pan, Xinyi Wang, Keren He, Wenyu Chen, and Zhao Kang. Aggregation-aware MLP: An unsupervised approach for graph message-passing. *Pattern Recognition*, page 113877, 2026.

[17] Liang Yang, Yuwei Liu, Jiaming Zhuo, Di Jin, Chuan Wang, Zhen Wang, and Xiaochun Cao. Do we really need message passing in brain network modeling? In *Forty-second International Conference on Machine Learning*, 2025.

[18] Jincheng Huang, Jialie Shen, Xiaoshuang Shi, and Xiaofeng Zhu. On which nodes does GCN fail? enhancing GCN from the node perspective. In *Forty-first International Conference on Machine Learning*, 2024.

[19] Jincheng Huang, Yujie Mo, Ping Hu, Xiaoshuang Shi, Shangbo Yuan, Zeyu Zhang, and Xiaofeng Zhu. Exploring the role of node diversity in directed graph representation learning. In *IJCAI*, pages 2072–2080, 2024.

[20] Lei Zhu, Qi She, Qian Chen, Yunfei You, Boyu Wang, and Yanye Lu. Weakly supervised object localization as domain adaption. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14637–14646, 2022.

[21] Taotao Jing, Bingrong Xu, and Zhengming Ding. Towards fair knowledge transfer for imbalanced domain adaptation. *IEEE Transactions on Image Processing*, 30:8200–8211, 2021.

[22] Ruizhi Pu, Gezheng Xu, Ruiyi Fang, Binkun Bao, Charles X Ling, and Boyu Wang. Leveraging group classification with descending soft labeling for deep imbalanced regression. *AAAI*, 2025.

[23] Hourun Li, Yifan Wang, Qinghua Ran, Junyu Luo, Jia Yang, Changling Zhou, Zhiping Xiao, Wei Ju, Xiao Luo, and Ming Zhang. CELL: A causal perspective for fairness-aware graph adaptation. In *Forty-third International Conference on Machine Learning*, 2026.

[24] Yifan Wang, Hourun Li, Ling Yue, Zhiping Xiao, Jia Yang, Changling Zhou, Wei Ju, Ming Zhang, and Xiao Luo. DANCE: Dual unbiased expansion with group-acquired alignment for out-of-distribution graph fairness learning. In *Forty-second International Conference on Machine Learning*, 2025.

[25] Shiqi Yang, Yaxing Wang, Joost Van De Weijer, Luis Herranz, and Shangling Jui. Generalized source-free domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8978–8987, 2021.

[26] Gezheng Xu, Li Yi, Pengcheng Xu, Jiaqi Li, Ruizhi Pu, Changjian Shui, Charles Ling, A Ian McLeod, and Boyu Wang. Unraveling the mysteries of label noise in source-free domain adaptation: Theory and practice. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(5):4007–4023, 2025.

[27] Xiao Shen, Mengqiu Shao, Shirui Pan, Laurence T. Yang, and Xi Zhou. Domain-adaptive graph attention-supervised network for cross-network edge classification. *IEEE Transactions on Neural Networks and Learning Systems*, 35(12):17842–17855, 2024.

[28] 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 *The Fourteenth International Conference on Learning Representations*, pages 17271–17294, 2026.

[29] 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, pages 2991–2999, 2020.

[30] Xiao Shen, Quanyu Dai, Sitong Mao, Fu-lai Chung, and Kup-Sze Choi. Network together: Node classification via cross-network deep network embedding. *IEEE Transactions on Neural Networks and Learning Systems*, 32(5):1935–1948, 2020.

[31] Xiaowen Zhang, Yuntao Du, Rongbiao Xie, and Chongjun Wang. Adversarial separation network for cross-network node classification. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 2618–2626, 2021.

[32] Haitao Mao, Lun Du, Yujia Zheng, Qiang Fu, Zelin Li, Xu Chen, Shi Han, and Dongmei Zhang. Source free unsupervised graph domain adaptation. *The 17th ACM International Conference on Web Search and Data Mining*, 2024.

[33] Bencheng Yan and Chaokun Wang. Graphae: adaptive embedding across graphs. In *2020 IEEE 36th International Conference on Data Engineering (ICDE)*, pages 1958–1961. IEEE, 2020.

[34] Ruiyi Fang, Jingyu Zhao, Shuo Wang, Ruizhi Pu, Bingheng Li, Jiale Cai, Zhihao Li, Zihao Jing, Jian Zhu, Song Tang, et al. SAGA: Structural aggregation guided alignment with dynamic view and neighborhood order selection for multiview graph domain adaptation. In *The Fourteenth International Conference on Learning Representations*, pages 95585–95607, 2026.

[35] Shuo Wang, Bokui Wang, Zhixiang Shen, Boyan Deng, and Zhao Kang. Multi-domain graph foundation models: Robust knowledge transfer via topology alignment. In *Forty-second International Conference on Machine Learning*, pages 64806–64821, 2025.

[36] 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. In *The Fourteenth International Conference on Learning Representations*, pages 109348–109368, 2026.

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

[38] Jingyang Yuan, Xiao Luo, Yifang Qin, Zhengyang Mao, Wei Ju, and Ming Zhang. Alex: Towards effective graph transfer learning with noisy labels. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 3647–3656, 2023.

[39] Ziyue Qiao, Xiao Luo, Meng Xiao, Hao Dong, Yuanchun Zhou, and Hui Xiong. Semi-supervised domain adaptation in graph transfer learning. *IJCAI*, 2023.

[40] 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*, pages 1131–1142, 2024.

[41] Ruiyi Fang, Bingheng Li, Zhao Kang, Qiuhao Zeng, Nima Hosseini Dashtbayaz, Ruizhi Pu, Boyu Wang, and Charles Ling. On the benefits of attribute-driven graph domain adaptation. *ICLR*, 2025.

[42] Ruiyi Fang, Bingheng Li, Jingyu Zhao, Ruizhi Pu, Qiuhao Zeng, Gezheng Xu, Charles Ling, and Boyu Wang. Homophily enhanced graph domain adaptation. In *Forty-second International Conference on Machine Learning*, 2025.

[43] Ruiyi Fang, Shuo Wang, Ruizhi Pu, Qiuhao Zeng, Hao Zheng, Ziyan Wang, Jiale Cai, Zhimin Mei, Song Tang, Charles Ling, et al. Graph domain adaptation via homophily-agnostic reconstructing structure. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 21047–21055, 2026.

[44] 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*, 2025.

[45] 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. *ICML*, 2025.

[46] Jiaming Zhuo, Yuwei Liu, Yintong Lu, Ziyi Ma, Kun Fu, Chuan Wang, Yuanfang Guo, Zhen Wang, Xiaochun Cao, and Liang Yang. DUALFormer: Dual graph transformer. In *The Thirteenth International Conference on Learning Representations*, 2025.

[47] Jiaming Zhuo, Ziyi Ma, Yintong Lu, Yuwei Liu, Kun Fu, Di Jin, Chuan Wang, Wenning Wu, Zhen Wang, Xiaochun Cao, and Liang Yang. A closer look at graph transformers: Cross-aggregation and beyond. In *Advances in Neural Information Processing Systems*, 2025.

[48] Jiaming Zhuo, Ziyi Ma, Kun Fu, Di Jin, Chuan Wang, Zhen Wang, Xiaochun Cao, Huazhu Fu, and Liang Yang. Rethinking graph transformers as graph signal denoisers: The role of block-diagonal priors. In *Forty-third International Conference on Machine Learning*, 2026.

[49] Xuanting Xie, Bingheng Li, Erlin Pan, Zhaochen Guo, Zhao Kang, and Wenyu Chen. One node one model: Featuring the missing-half for graph clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 21688–21696, 2025.

[50] Xuanting Xie, Erlin Pan, Zhao Kang, Wenyu Chen, and Bingheng Li. Provable filter for real-world graph clustering. *IEEE Transactions on Cybernetics*, 2026.

[51] Xuanting Xie, Bingheng Li, Erlin Pan, Rui Hou, Wenyu Chen, and Zhao Kang. Attention beyond neighborhoods: Reviving transformer for graph clustering. *IEEE Transactions on Circuits and Systems for Video Technology*, 2026.

[52] Xuanting Xie, Zhaochen Guo, Bingheng Li, Xingtong Yu, Zhifei Liao, Zhao Kang, and Yuan Fang. Clustering as reasoning: A k-means interpretation of chain-of-thought graph learning. In *Forty-third International Conference on Machine Learning*, 2026.

[53] Xingtong Yu, Shenghua Ye, Ruijuan Liang, Chang Zhou, Hong Cheng, Xinming Zhang, and Yuan Fang. Evaluating progress in graph foundation models: A comprehensive benchmark and new insights. *arXiv preprint arXiv:2603.10033*, 2026.

[54] Xiaodong He, Haolan He, Ruiyi Fang, Ming Sun, and Zhao Kang. Structure-centric graph foundation model via geometric bases. In *Forty-third International Conference on Machine Learning*, 2026.

[55] Jie Xu, Na Zhao, Gang Niu, Masashi Sugiyama, and Xiaofeng Zhu. Robust multi-view learning via representation fusion of sample-level attention and alignment of simulated perturbation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4232–4241, 2025.

[56] Jiaming Zhuo, Yintong Lu, Hui Ning, Kun Fu, Bingxin Niu, Dongxiao He, Chuan Wang, Yuanfang Guo, Zhen Wang, Xiaochun Cao, and Liang Yang. Unified graph augmentations for generalized contrastive learning on graphs. In *Advances in Neural Information Processing Systems*, 2024.

[57] Liang Yang, Zhenna Li, Jiaming Zhuo, Jing Liu, Ziyi Ma, Chuan Wang, Zhen Wang, and Xiaochun Cao. Graph contrastive learning with joint spectral augmentation of attribute and topology. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 21983–21991, 2025.

[58] Jincheng Huang, Yujie Mo, Xiaoshuang Shi, Lei Feng, and Xiaofeng Zhu. Enhancing the influence of labels on unlabeled nodes in graph convolutional networks. In *Forty-second International Conference on Machine Learning*, 2025.

[59] Jiaming Zhuo, Can Cui, Kun Fu, Bingxin Niu, Dongxiao He, Yuanfang Guo, Zhen Wang, Chuan Wang, Xiaochun Cao, and Liang Yang. Propagation is all you need: A new framework for representation learning and classifier training on graphs. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 481–489, 2023.

[60] Jincheng Huang, Jie Xu, Xiaoshuang Shi, Ping Hu, Lei Feng, and Xiaofeng Zhu. Revisiting confidence calibration for misclassification detection in VLMs. In *The Fourteenth International Conference on Learning Representations*, 2026.

[61] Jincheng Huang, Jie Xu, Xiaoshuang Shi, Ping Hu, Lei Feng, and Xiaofeng Zhu. The final layer holds the key: A unified and efficient gnn calibration framework. *arXiv preprint arXiv:2505.11335*, 2025.

[62] Qiuhao Zeng, Jerry Huang, Peng Lu, Ruiyi Fang, Gezheng Xu, Zihao Jing, Yufei Cui, Charles Ling, Gang Niu, and Boyu Wang. Attention with routed-memory for learnable sparse control. In *International Conference on Machine Learning*, pages 153364–153385, 2026.

[63] Quanxin Wang, Xuanting Xie, Bingheng Li, Xingtong Yu, Shuo Wang, Ruiyi Fang, and Zhao Kang. Dynamic graph prompting via topology-routed mixed-curvature experts. *arXiv preprint arXiv:2608.06031*, 2026.

[64] Hui Guo, Grace Y. Yi, and Boyu Wang. Addressing both variable selection and misclassified responses with parametric and semiparametric methods. *Bernoulli*, 32(2):1303–1327, 2026.

[65] Jiaming Zhuo, Can Cui, Kun Fu, Bingxin Niu, Dongxiao He, Chuan Wang, Yuanfang Guo, Zhen Wang, Xiaochun Cao, and Liang Yang. Graph contrastive learning reimagined: Exploring universality. In *Proceedings of the ACM Web Conference 2024*, pages 641–651, 2024.

[66] Liang Yang, Yukun Cai, Hui Ning, Jiaming Zhuo, Di Jin, Ziyi Ma, Yuanfang Guo, Chuan Wang, and Zhen Wang. Universal graph self-contrastive learning. In *IJCAI*, pages 3534–3542, 2025.

[67] Luzhi Wang, Yizhen Zheng, Di Jin, Fuyi Li, Yongliang Qiao, and Shirui Pan. Contrastive graph similarity networks. *ACM Transactions on the Web*, 18(2):1–20, 2024.

[68] Jiaming Zhuo, Feiyang Qin, Can Cui, Kun Fu, Bingxin Niu, Mengzhu Wang, Yuanfang Guo, Chuan Wang, Zhen Wang, Xiaochun Cao, and Liang Yang. Improving graph contrastive learning via adaptive positive sampling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23179–23187, 2024.

[69] Jie Xu, Shuo Chen, Yazhou Ren, Xiaoshuang Shi, Hengtao Shen, Gang Niu, and Xiaofeng Zhu. Self-weighted contrastive learning among multiple views for mitigating representation degeneration. In *Advances in Neural Information Processing Systems*, volume 36, pages 1119–1131, 2023.

[70] Jie Xu, Huayi Tang, Yazhou Ren, Liang Peng, Xiaofeng Zhu, and Lifang He. Multi-level feature learning for contrastive multi-view clustering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16051–16060, 2022.

[71] Zhixiang Shen, Haolan He, and Zhao Kang. Balanced multi-relational graph clustering. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 4120–4128, 2024.

[72] Shaohua Fan, Xiao Wang, Chuan Shi, Emiao Lu, Ken Lin, and Bai Wang. One2multi graph autoencoder for multi-view graph clustering. In *proceedings of the web conference 2020*, pages 3070–3076, 2020.

[73] Xinyu Fu, Jiani Zhang, Ziqiao Meng, and Irwin King. Magnn: Metapath aggregated graph neural network for heterogeneous graph embedding. In *Proceedings of the web conference 2020*, pages 2331–2341, 2020.

[74] Diederick P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

[75] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[76] Jun Wu, Jingrui He, and Elizabeth Ainsworth. Non-iid transfer learning on graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 10342–10350, 2023.

[77] Shuo Wang, Xiangyu Wang, Quanxin Wang, Bailin Wu, Bokui Wang, Shunyang Huang, Boyan Deng, Haonan Liu, Ruiyi Fang, Zhenxiang Xu, et al. The post-GCN decade revisited: Curvature-stratified evaluation of relational learning. *arXiv preprint arXiv:2606.06397*, 2026.

[78] Xiangyu Wang, Shuo Wang, Ruiyi Fang, and Zhao Kang. Preserving geometric integrity in graph prompting via measure-constrained optimal transport. *arXiv preprint arXiv:2609.23547*, 2026.

[79] Xiao Shen, Zhihao Chen, Shirui Pan, Shuang Zhou, Laurence T. Yang, and Xi Zhou. Open-set cross-network node classification via unknown-excluded adversarial graph domain alignment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 20398–20408, 2025.

[80] Junwei Gong, Xiao Shen, Zhihao Chen, Shirui Pan, Xiao Wang, and Xi Zhou. negMIX: Negative mixup for OOD generalization in open-set node classification. In *Proceedings of the ACM Web Conference 2026*, pages 707–718, 2026.

[81] Luzhi Wang, Dongxiao He, He Zhang, Yixin Liu, Wenjie Wang, Shirui Pan, Di Jin, and Tat-Seng Chua. GOODAT: Towards test-time graph out-of-distribution detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 15537–15545, 2024.

[82] Luzhi Wang, Xuanshuo Fu, He Zhang, Chuang Liu, Xiaobao Wang, and Hongbo Liu. From subtle to significant: Prompt-driven self-improving optimization in test-time graph OOD detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 15851–15859, 2026.

[83] Xiao Shen, Chuanyun Lin, Zhihao Chen, Junwei Gong, and Xi Zhou. Prototypical multi-source graph contrastive domain adaptation. *Pattern Recognition*, 180:114312, 2026.

[84] Zhixiang Shen, Shuo Wang, and Zhao Kang. Beyond redundancy: Information-aware unsupervised multiplex graph structure learning. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

[85] Xuanting Xie, Wenyu Chen, and Zhao Kang. Robust graph structure learning under heterophily. *Neural Networks*, 185:107206, 2025.

[86] Leonardo FR Ribeiro, Pedro HP Saverese, and Daniel R Figueiredo. struc2vec: Learning node representations from structural identity. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 385–394, 2017.

[87] Jundong Li, Xia Hu, Jiliang Tang, and Huan Liu. Unsupervised streaming feature selection in social media. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 1041–1050, 2015.

Appendix Contents

A Empirical Analysis of Local and Global Distribution Gaps 17

B Subgraph Sampling Strategy for the Cross-Scale Source Graph 18

C Cross-scale Sampling: Airport Statistics and BFS:DFS Mixing Ablation 20

C.1 Cross-scale source statistics on Airport 20

C.2 Ablation on the BFS:DFS mixing ratio 20

C.3 Choice of the number of BFS clusters K 21

D Model Efficient Experiment 21

E Parameter Analysis 22

F Description of Algorithm LGSA 24

G Additional Ablation Results on Citation and ACM/Blog Benchmarks . . 25

H Datasets 25

A Empirical Analysis of Local and Global Distribution Gaps

In this appendix, we provide an additional empirical analysis to support the local–global motivation of LGSA. Different from the downstream node classification results in the main text, this analysis does not rely on target accuracy. Instead, it directly measures how the sampled source graph $G^S = \mathcal{S}_\rho(G^S_{\text{full}})$ deviates from the target graph G^T in terms of global semantic coverage and local structural consistency. Recall that $\mathcal{S}_\rho$ selects only a small fraction of source nodes and keeps the node-induced edges among them. Therefore, when $\rho \in \{1\%, 5\%, 10\%\}$, the observed source graph contains only sparse node support and severely truncated neighborhoods.

Global semantic score. To characterize global information without using target labels, we define a one-dimensional semantic projection score for each node. For a node v in domain $D \in \{S, T\}$, we first construct an input-level global descriptor

$$\psi^D(v) = \left[\mathbf{x}_v^D \,\|\, \phi^D(v)\right], \qquad \phi^D(v) = \left(\deg^D(v), c^D(v), n_2^D(v), n_3^D(v)\right)^\top, \tag{14}$$

where $\mathbf{x}_v^D$ is the node feature, $\deg^D(v)$ is the degree, $c^D(v)$ is the clustering coefficient, and $n_k^D(v)$ denotes the size of the k-hop neighborhood. This descriptor is consistent with the structural statistics used by the global branch in Eq. (6) of the main text. We then standardize $\psi^D(v)$ over the union of the source and target nodes and project it onto the first principal direction $\mathbf{u}_1$:

$$s^D_{\text{glo}}(v) = \text{Norm}_{[0,1]}\left(\mathbf{u}_1^\top \psi^D(v)\right), \tag{15}$$

where $\text{Norm}_{[0,1]}(\cdot)$ linearly rescales the projected values into $[0, 1]$. The distribution of $\{s^D_{\text{glo}}(v) : v \in V^D\}$ summarizes how the nodes of each graph cover the global feature–structure space. Since this score is computed from node attributes and structural descriptors rather than graph message passing, it reflects graph-level semantic coverage and is directly related to the global MLP branch.

Local Neighbor Degree Score. To characterize local structural information without relying on labels, we define a node-wise neighbor-degree consistency score. For a node v in domain D, let $\mathcal{N}^D(v)$ denote its one-hop neighborhood in G^D. The local score is

$$s^D_{\deg}(v) = \begin{cases} \dfrac{1}{|\mathcal{N}^D(v)|} \displaystyle\sum_{u \in \mathcal{N}^D(v)} \dfrac{\min\{\deg^D(v), \deg^D(u)\}}{\max\{\deg^D(v), \deg^D(u)\} + \epsilon}, & |\mathcal{N}^D(v)| > 0, \\ 0, & |\mathcal{N}^D(v)| = 0, \end{cases} \tag{16}$$

where ϵ is a small constant for numerical stability. This score lies in $[0, 1]$ and measures whether a node is connected to neighbors with similar degree profiles. It is a local structural statistic because it depends on both the retained edges and the degrees of the one-hop neighbors. When source nodes and their incident edges are heavily removed, both $\mathcal{N}^S(v)$ and $\deg^S(v)$ become unstable, making the distribution of $s^S_{\deg}(v)$ fragmented. This directly reflects the local message-passing difficulty faced by the GCN branch.

Distribution-level discrepancy. For each transfer task, we compare the empirical distributions of the above scores between the target graph, the full source graph, and the sampled source graph. Let

$$P^D_{\text{glo}} = \left\{s^D_{\text{glo}}(v) : v \in V^D\right\}, \qquad P^D_{\deg} = \left\{s^D_{\deg}(v) : v \in V^D\right\}. \tag{17}$$

We further quantify the source–target mismatch by the one-dimensional Wasserstein distance:

$$\Delta_{\text{glo}}(G^S, G^T) = W_1\left(P^S_{\text{glo}}, P^T_{\text{glo}}\right), \qquad \Delta_{\text{loc}}(G^S, G^T) = W_1\left(P^S_{\deg}, P^T_{\deg}\right). \tag{18}$$

Here, Δ_{glo} measures the global semantic discrepancy, while Δ_{loc} measures the local structural discrepancy.

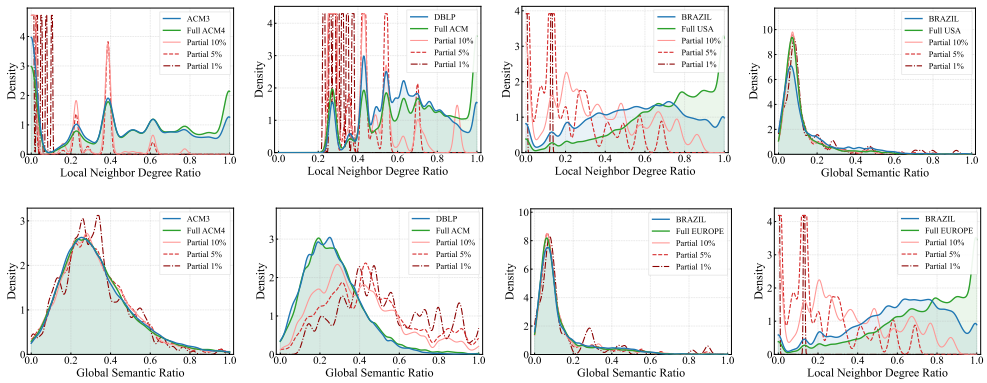


Figure 5: Empirical illustration of source–target discrepancy under partial source-graph sampling. The full source and target graphs have relatively smooth global and local score distributions, while the sampled source graph with $\rho \in \{1\%, 5\%, 10\%\}$ shows fragmented support and large blank regions. Left: the global semantic score captures the sparse coverage of the sampled source graph in the feature–structure space. Right: the Local Neighbor Degree Score captures the distortion of local structural patterns after source nodes and induced edges are removed.

Empirical observations. As shown in Fig. 5, the sampled source graph exhibits clear distributional gaps compared with the target graph. For the global semantic score, the partial source distribution becomes sparse and discontinuous when only 1%–10% of source nodes are retained. This indicates that source-side sampling removes a large portion of the global feature structure support, causing the observed source graph to cover only a biased subset of the semantic space. This observation explains why an adjacency-independent MLP branch is needed: it can model the global distributional pattern from node features and structural descriptors without over-relying on the incomplete source adjacency.

For the Local Neighbor Degree Score, the partial source distribution is also highly fragmented. Since the sampled graph keeps only the induced edges among the retained nodes, many original neighbors are removed and the one-hop degree profiles are substantially changed. As a result, the local structural distribution of G^S deviates from that of G^T, indicating that direct GCN message passing on the truncated source graph is unreliable. This motivates the local compensation branch of LGSA, which augments the sampled source graph with semantically related target nodes and then uses GCN propagation to recover more informative local structural signals.

Overall, these empirical results show that the difficulty of cross-scale graph domain adaptation is not merely caused by having fewer labeled source nodes. Rather, source sampling simultaneously induces global semantic sparsity and local structural fragmentation. The Global Semantic Score and Local Neighbor Degree Score provide two complementary diagnostics for these effects, and they directly support the local–global design of LGSA.

B Subgraph Sampling Strategy for the Cross-Scale Source Graph

In CSGDA, the training-time source graph is not the full labeled source graph $G^{\mathrm{S}}_{\mathrm{full}} = \{\mathcal{V}^{\mathrm{S}}_{\mathrm{full}}, \mathcal{E}^{\mathrm{S}}_{\mathrm{full}}, A^{\mathrm{S}}_{\mathrm{full}}, X^{\mathrm{S}}_{\mathrm{full}}, Y^{\mathrm{S}}_{\mathrm{full}}\}$. Instead, it is generated by a node-induced sampling operator $\mathcal{S}_\rho$:

$$G^{\mathrm{S}} = \mathcal{S}_\rho(G^{\mathrm{S}}_{\mathrm{full}}) = \{\mathcal{V}^{\mathrm{S}}, \mathcal{E}^{\mathrm{S}}, A^{\mathrm{S}}, X^{\mathrm{S}}, Y^{\mathrm{S}}\}, \qquad |\mathcal{V}^{\mathrm{S}}| = N^{\mathrm{S}} = \lfloor \rho |\mathcal{V}^{\mathrm{S}}_{\mathrm{full}}| \rfloor, \tag{19}$$

where $\rho \in \{1\%, 5\%, 10\%\}$ is the source sampling ratio used in our experiments. The target graph G^{T} is kept fully observed, while only the sampled source graph G^{S} and its labels are available during training. Thus, $\mathcal{S}_\rho$ is applied only to the source graph.

Overview of $\mathcal{S}_\rho$. The goal of $\mathcal{S}_\rho$ is to construct a small source graph that is locally coherent while still covering diverse regions of the full source graph. A purely uniform sampler may destroy local neighborhoods, while a single BFS or random-walk sampler can over-concentrate on one dense community. Therefore, we use a 50:50 hybrid BFS–DFS/random sampling strategy. Given the total

node budget N^{S}, we split it into two parts:

$$N^{\mathrm{S}}_{\mathrm{BFS}} = \left\lfloor \frac{N^{\mathrm{S}}}{2} \right\rfloor, \qquad N^{\mathrm{S}}_{\mathrm{DFS}} = N^{\mathrm{S}} - N^{\mathrm{S}}_{\mathrm{BFS}}. \tag{20}$$

The BFS part preserves dense local substructures, while the DFS/random part adds scattered probes to improve global coverage.

Step 1: BFS clusters for local structural preservation. We first select K seed nodes uniformly at random from $\mathcal{V}^{\mathrm{S}}_{\mathrm{full}}$:

$$\mathcal{A} = \{a_1, a_2, \ldots, a_K\} \subset \mathcal{V}^{\mathrm{S}}_{\mathrm{full}}. \tag{21}$$

Each seed a_k initializes a BFS queue and grows one local cluster. The BFS budget is divided approximately evenly across the K clusters:

$$q_k = \left\lceil \frac{N^{\mathrm{S}}_{\mathrm{BFS}}}{K} \right\rceil. \tag{22}$$

For each seed a_k, we run BFS on $G^{\mathrm{S}}_{\mathrm{full}}$ until the cluster reaches size q_k or no new neighbor can be reached. Specifically, when a node is popped from the queue, its unvisited neighbors are randomly shuffled and added to the cluster until the cluster budget is filled. If a BFS cluster terminates early because its connected region is exhausted, we restart BFS from another unsampled node. The union of all BFS clusters is denoted by $\mathcal{V}^{\mathrm{S}}_{\mathrm{BFS}}$.

This step keeps locally dense neighborhoods around multiple seeds, so the sampled source graph still contains meaningful ego-network structures, homophily patterns, and short-range connectivity required by message passing.

Step 2: DFS/random probes for global coverage. After the BFS stage, we sample the remaining $N^{\mathrm{S}}_{\mathrm{DFS}}$ nodes using scattered DFS/random probes. We repeatedly pick a starting node uniformly from $\mathcal{V}^{\mathrm{S}}_{\mathrm{full}} \setminus \mathcal{V}^{\mathrm{S}}_{\mathrm{BFS}}$ and perform a short DFS-style walk. At each step, the sampler follows one random unvisited neighbor if such a neighbor exists; otherwise, it restarts from another unsampled node. Formally, if the current node is u, the next node is sampled from

$$\mathcal{N}_{\mathrm{unvis}}(u) = \{v \in \mathcal{N}(u) : v \notin \mathcal{V}^{\mathrm{S}}_{\mathrm{BFS}} \cup \mathcal{V}^{\mathrm{S}}_{\mathrm{DFS}}\}. \tag{23}$$

If $\mathcal{N}_{\mathrm{unvis}}(u) = \emptyset$, a new unsampled node is selected uniformly at random as the next probe root. The process stops once $|\mathcal{V}^{\mathrm{S}}_{\mathrm{DFS}}| = N^{\mathrm{S}}_{\mathrm{DFS}}$.

Different from the BFS stage, this stage does not aim to form dense local balls. Instead, it inserts relatively scattered nodes from different graph regions. These DFS/random probes help the sampled source graph retain broader class, feature, and structural diversity, which is important for cross-scale alignment.

Step 3: Node union and induced subgraph construction. The final sampled source node set is

$$\mathcal{V}^{\mathrm{S}} = \mathcal{V}^{\mathrm{S}}_{\mathrm{BFS}} \cup \mathcal{V}^{\mathrm{S}}_{\mathrm{DFS}}, \qquad |\mathcal{V}^{\mathrm{S}}| = N^{\mathrm{S}}. \tag{24}$$

The source edge set is then constructed as the node-induced subgraph over $\mathcal{V}^{\mathrm{S}}$:

$$\mathcal{E}^{\mathrm{S}} = \{(i,j) \in \mathcal{E}^{\mathrm{S}}_{\mathrm{full}} \mid i,j \in \mathcal{V}^{\mathrm{S}}\}. \tag{25}$$

The adjacency, feature, and label matrices are obtained by restricting the full source graph to the sampled node set:

$$A^{\mathrm{S}} = A^{\mathrm{S}}_{\mathrm{full}}[\mathcal{V}^{\mathrm{S}}, \mathcal{V}^{\mathrm{S}}], \qquad X^{\mathrm{S}} = X^{\mathrm{S}}_{\mathrm{full}}[\mathcal{V}^{\mathrm{S}}, :], \qquad Y^{\mathrm{S}} = Y^{\mathrm{S}}_{\mathrm{full}}[\mathcal{V}^{\mathrm{S}}, :]. \tag{26}$$

No edges outside the sampled node set are retained, and no labels from the target graph are used. Therefore, all methods are evaluated under the same cross-scale setting: a small labeled source graph G^{S} and a fully observed but unlabeled target graph G^{T}.

Discussion. The operator $\mathcal{S}_\rho$ creates source-side incompleteness in two ways. First, because only a small fraction of nodes is kept, many local neighbors of sampled source nodes are removed, leading to truncated local message-passing contexts. Second, because the sampled source graph contains only a partial view of the full source distribution, its global structural statistics, such as degree, clustering coefficient, and multi-hop neighborhood sizes, can deviate substantially from those of the target graph. These two effects are precisely the local and global information losses studied in CSGDA.

Table 4: Cross-scale source statistics on Airport at ratio r. Brazil at $r{=}1\%$ is floored to two seed nodes (see text).

Source → Target	Full Source #N	Full Source #E	Target #N	Target #E	$r{=}1\%$ #N	$r{=}1\%$ #E	$r{=}5\%$ #N	$r{=}5\%$ #E	$r{=}10\%$ #N	$r{=}10\%$ #E
U → B	1,190	13,599	131	1,038	12	18*	60	92*	119	214*
U → E	1,190	13,599	399	5,995	12	18*	60	92*	119	214*
B → U	131	1,038	1,190	13,599	2	1*	7	8*	13	22*
B → E	131	1,038	399	5,995	2	1*	7	8*	13	22*
E → U	399	5,995	1,190	13,599	4	4*	20	34*	40	86*
E → B	399	5,995	131	1,038	4	4*	20	34*	40	86*

Table 5: Effect of the BFS:DFS mixing ratio at $r{=}5\%$. α is the share of the budget assigned to BFS clusters; the remainder is drawn by DFS / random probes. Best per column in **bold**.

α (%)	D → A (Citation) Acc. (%)	D → A (Citation) Macro-F1	B1 → B2 (Blog) Acc. (%)	B1 → B2 (Blog) Macro-F1	Structural diagnostics C_y	Structural diagnostics $\bar{C}$
0 (pure DFS)	71.2	70.4	64.8	63.9	**1.00**	0.04
25	73.6	72.7	66.5	65.7	1.00	0.11
50	**74.9**	**74.0**	**67.8**	**67.0**	1.00	0.19
75	73.4	72.5	66.9	66.1	0.92	0.27
100 (pure BFS)	70.1	69.0	63.7	62.5	0.71	**0.34**

The BFS component of $\mathcal{S}_\rho$ preserves local cohesion, whereas the DFS/random component improves global coverage. Together, they produce a small but nontrivial source graph for evaluating whether a GDA method can transfer from incomplete source-side information to a much larger target graph.

Hyperparameters. Unless otherwise stated, we use the same 50:50 BFS–DFS/random split for all cross-scale experiments. The sampling ratio is $\rho \in \{1\%, 5\%, 10\%\}$, and the BFS seeds are divided into K local clusters. In practice, K is chosen so that each BFS cluster has a reasonable local budget under the given ρ. For all baselines and LGSA, the same sampled source graph G^{S} is used under each transfer task and sampling ratio, ensuring a fair comparison.

C Cross-scale Sampling: Airport Statistics and BFS:DFS Mixing Ablation

This appendix complements Section 3 by providing (i) per-task sampled-source statistics on the **Airport** benchmark and (ii) an ablation on the BFS:DFS mixing ratio of our hybrid sampler.

C.1 Cross-scale source statistics on Airport

Table 4 reports source/target sizes for all six transfer tasks in Airport at $r \in \{1\%, 5\%, 10\%\}$, mirroring the format of Table 1. Compared with Citation and Blog, Airport graphs have an order-of-magnitude larger average degree ($\bar{d}_{\mathrm{USA}} \approx 11.4$, $\bar{d}_{\mathrm{Eur}} \approx 15.0$, $\bar{d}_{\mathrm{Bra}} \approx 7.9$), so even a small node budget already induces a non-trivial number of edges in $\widetilde{G}_s$. For Brazil, $r = 1\%$ corresponds to a single seed node and is therefore degenerate; we floor the budget to two nodes to keep at least one candidate edge.

Two observations follow: **(a) Cross-scale ratio.** For `B→U` and `E→U`, the target is up to two orders of magnitude larger than the sampled source in both nodes and edges, matching the regime of interest in the main paper. **(b) Asymmetry.** Tasks with USA as source give the largest sampled sub-graphs; tasks with Brazil as source impose the most extreme data scarcity, providing a clean stress test for the proposed alignment.

C.2 Ablation on the BFS:DFS mixing ratio

Our hybrid sampler splits the budget evenly between BFS expansion (local cohesion) and DFS / random probing (global coverage). To justify this 50:50 choice, we sweep the mixing ratio $\alpha \in \{0, 25, 50, 75, 100\}$, where α denotes the percentage of the budget allocated to BFS, and report (i) the average target-side accuracy on two representative tasks and (ii) two structural diagnostics of $\widetilde{G}_s$: *class coverage* C_y (fraction of source classes appearing in $\widetilde{G}_s$) and the average local clustering coefficient $\bar{C}$. All numbers are averaged over five sampling seeds at $r = 5\%$.

Table 6: Sensitivity to the number of BFS clusters K at $r = 5\%$, $\alpha = 50\%$. We compare $K = 1$, $K_{1/2} = \lceil \sqrt{b_{\mathrm{BFS}}}/2 \rceil$, $K^\star = \lceil \sqrt{b_{\mathrm{BFS}}} \rceil$ (default, Eq. (27), underlined), $K_2 = \lceil 2\sqrt{b_{\mathrm{BFS}}} \rceil$, and $K = b_{\mathrm{BFS}}$ (pure random). Best per task in **bold**.

	D → A ($b_{\mathrm{BFS}} = 137$)		B1 → B2 ($b_{\mathrm{BFS}} = 58$)	
K	Acc. (%)	$\bar{R}$	Acc. (%)	$\bar{R}$
1	71.0	4.6	63.9	3.8
$K_{1/2}$	73.8	2.7	66.4	2.3
$K^\star$	**74.9**	1.9	**67.8**	1.6
K_2	74.1	1.4	67.0	1.1
b_{BFS}	71.5	1.0	64.6	1.0

C.3 Choice of the number of BFS clusters K

Given a sampling budget $b = \lceil r|V_s| \rceil$ and the 50:50 mixing rule of Sec. C.2, the BFS half of the budget, $b_{\mathrm{BFS}} = b/2$, is partitioned into K clusters of average size $s = b_{\mathrm{BFS}}/K$. The two extremes are again pathological: $K = 1$ collapses all BFS effort into a single dense ego-graph (high local cohesion, no diversity), whereas $K = b_{\mathrm{BFS}}$ degenerates into pure random seeding (no neighbourhood expansion). We therefore set

$$K \;=\; \left\lceil \sqrt{b_{\mathrm{BFS}}} \right\rceil \;=\; \left\lceil \sqrt{r|V_s|/2} \right\rceil, \tag{27}$$

which makes both the number of clusters and the per-cluster size scale as $\Theta(\sqrt{b})$, jointly trading off coverage against cohesion. As concrete instances, Eq. (27) yields $K = 12$ for D→A and $K = 8$ for B1→B2 at $r = 5\%$.

Sensitivity. We sweep K around the default of Eq. (27) on the same two tasks at $r = 5\%$ ($\alpha = 50\%$, five sampling seeds). Table 6 reports target-side accuracy and the average per-cluster radius $\bar{R}$ (BFS hops to consume the cluster budget). Performance is essentially flat within a factor of two around the default and degrades only at the extremes, confirming that $K = \lceil \sqrt{b/2} \rceil$ is a safe, hyper-parameter-free choice.

D Model Efficient Experiment

Model Complexity. Here, we analyze the computational complexity of the proposed LGSA. The overall cost decomposes naturally over the three modules introduced in Sec. 3: (i) the local GCN branch on the augmented source graph $\widetilde{G}^{\mathrm{S}}$ and the target graph G^{T}, (ii) the global MLP branch on per-node structural descriptors, and (iii) the adaptive GCN–MLP alignment module. Let N^{S} and N^{T} denote the numbers of source and target nodes, $|\widetilde{\mathcal{E}}^{\mathrm{S}}|$ and $|\mathcal{E}^{\mathrm{T}}|$ the edge counts of $\widetilde{G}^{\mathrm{S}}$ and G^{T}, d the input feature dimension, d_h the hidden width, C the number of classes, and M the number of cross-domain neighbors augmented per source node.

Local Branch. Each GCN layer costs $O\big((|\widetilde{\mathcal{E}}^{\mathrm{S}}| + |\mathcal{E}^{\mathrm{T}}|)\, d_h \;+\; (N^{\mathrm{S}}(1+M) + N^{\mathrm{T}})\, d_h^2\big)$, dominated by sparse message passing and per-node feature transformation. Cross-domain node augmentation performs a one-shot top-M cosine similarity search at $O(N^{\mathrm{S}} N^{\mathrm{T}} d)$, executed only once before training.

Global Branch. The structural descriptor $\mathbf{\Phi} \in \mathbb{R}^{N \times 4}$ is precomputed once at a cost of $O(|\mathcal{E}^{\mathrm{S}}| + |\mathcal{E}^{\mathrm{T}}|)$ for degree and clustering coefficients and $O((N^{\mathrm{S}} + N^{\mathrm{T}})\bar{d}^{\,k})$ for k-hop neighborhood sizes, where $\bar{d}$ is the average degree. The MLP encoder is then graph-free and costs $O((N^{\mathrm{S}} + N^{\mathrm{T}})(d+4)\, d_h)$ per layer.

Adaptive Alignment. Branch-reliability estimation (Eq. 8) and the sigmoid fusion gate are linear in the number of nodes: $O((N^{\mathrm{S}} + N^{\mathrm{T}})(C + d_h))$.

Putting these together, the per-iteration training cost of LGSA is

$$O\Big((|\widetilde{\mathcal{E}}^{\mathrm{S}}| + |\mathcal{E}^{\mathrm{T}}|)\, d_h \;+\; (N^{\mathrm{S}} + N^{\mathrm{T}})\, d_h^2 \;+\; (N^{\mathrm{S}} + N^{\mathrm{T}})(d + C)\Big),$$

which is comparable to a single GCN backbone augmented with a lightweight MLP head. Crucially, LGSA does *not* introduce the heavy inter-graph operations adopted by recent GDA methods (e.g.,

Dataset	Method	Training Time (Norm. w.r.t. UDAGCN)	Memory Usage (Norm. w.r.t. UDAGCN)	Accuracy (%)
U→B	UDAGCN	1	1	0.307
	JHGDA	1.341	1.452	0.366
	GRADE-N	0.812	0.876	0.310
	SpecReg	0.483	<u>0.498</u>	0.342
	HGDA	<u>0.471</u>	0.503	<u>0.312</u>
	$LGSA$	**0.462**	**0.487**	**0.377**
U→E	UDAGCN	1	1	0.317
	JHGDA	1.402	1.498	0.320
	GRADE-N	0.823	0.881	<u>0.348</u>
	SpecReg	<u>0.469</u>	0.512	0.338
	HGDA	0.486	<u>0.501</u>	0.316
	$LGSA$	**0.458**	**0.491**	**0.363**
B→E	UDAGCN	1	1	0.154
	JHGDA	1.367	1.483	0.177
	GRADE-N	0.804	0.872	0.195
	SpecReg	0.478	0.509	<u>0.201</u>
	HGDA	<u>0.467</u>	<u>0.496</u>	0.176
	$LGSA$	**0.461**	**0.488**	**0.228**
A→D	UDAGCN	1	1	0.369
	JHGDA	1.389	1.521	0.382
	GRADE-N	0.831	0.894	0.357
	SpecReg	0.491	<u>0.521</u>	0.361
	HGDA	<u>0.482</u>	0.526	<u>0.411</u>
	$LGSA$	**0.476**	**0.503**	**0.429**
A→C	UDAGCN	1	1	0.394
	JHGDA	1.412	1.538	0.396
	GRADE-N	0.842	0.901	0.375
	SpecReg	0.503	<u>0.532</u>	0.372
	HGDA	<u>0.494</u>	0.541	<u>0.415</u>
	$LGSA$	**0.488**	**0.518**	**0.428**
C→D	UDAGCN	1	1	0.346
	JHGDA	1.378	1.512	0.382
	GRADE-N	0.827	0.889	0.345
	SpecReg	0.498	<u>0.527</u>	0.361
	HGDA	<u>0.487</u>	0.534	<u>0.364</u>
	$LGSA$	**0.481**	**0.512**	**0.414**

Table 7: Comparison of training time, GPU memory usage, and target-domain accuracy on the Airport (U/B/E) and Citation (A/C/D) benchmarks under source sampling ratio $\rho = 5\%$. Training time and memory are normalized w.r.t. UDAGCN. **Bold**: best; <u>underlined</u>: second best.

dense optimal-transport plans in PA, full eigen-decomposition in SpecReg, or hierarchical pooling in JHGDA), so its overhead remains close to the simplest UDAGCN backbone, while the global branch is essentially free relative to a GCN forward pass.

Model Efficient Experiment. To further evaluate the efficiency of LGSA, Table 7 reports a running-time and GPU-memory comparison across representative GDA baselines, including UDAGCN [8], JHGDA [9], GRADE-N [76], SpecReg [37], and HGDA [42], which align graph-domain discrepancies in fundamentally different ways (dual GCN + adversary, hierarchical pooling, non-IID transfer, spectral regularization, mixed homophilic/heterophilic filters). All numbers are measured at the default source sampling ratio $\rho = 5\%$ (same setting as the main experiments) on a single NVIDIA RTX A5000 24 GB GPU. Training time and memory usage are normalized w.r.t. UDAGCN on each task; the best result is in **bold** and the second-best is <u>underlined</u>. As shown in the table, the evaluations on both Airport and Citation benchmarks demonstrate that LGSA achieves the best target-domain accuracy across all six cross-scale tasks, while keeping training time and memory comparable to the most efficient baselines, confirming that the proposed local–global structural alignment incurs only tolerable computational and storage overhead.

E Parameter Analysis

The effectiveness of cross-scale graph domain adaptation (CSGDA) is highly influenced by *how incomplete the sampled source graph is* and by the local/global structural discrepancies between source and target graphs. LGSA jointly trains a local GCN branch (to compensate for missing neighborhood structure) and a global MLP branch (to capture sparse distributional patterns), which

Types	Tasks	μ	η
Airport	U→B	0.5	0.5
	U→E	0.5	0.1
	B→U	0.5	0.5
	B→E	0.5	0.5
	E→U	0.5	0.1
	E→B	0.5	0.5
Citation	A→C	0.1	0.1
	A→D	0.1	0.1
	C→A	0.1	0.5
	C→D	0.1	0.1
	D→A	0.1	0.1
	D→C	0.1	0.1
Social	B1→B2	0.05	0.1
	B2→B1	0.05	0.1
ACM	A3→A4	0.5	0.1
	A4→A3	0.1	0.5

Table 8: Hyperparameter setting of μ (global-branch loss weight) and η (local–global consistency weight) on each CSGDA task at sampling ratio $\rho = 5\%$.

are fused by an adaptive gate. Accordingly, its overall objective combines three terms,

$$\mathcal{L} \;=\; \mathcal{L}_{\mathrm{loc}} \;+\; \mu\,\mathcal{L}_{\mathrm{glo}} \;+\; \eta\,\mathcal{L}_{\mathrm{cons}},$$

where $\mathcal{L}_{\mathrm{loc}}$ and $\mathcal{L}_{\mathrm{glo}}$ are the local and global branch losses defined in Eq. 5 and Eq. 7, and $\mathcal{L}_{\mathrm{cons}}$ is the local–global consistency loss applied on the confident target pseudo labels produced by the adaptive alignment module (Sec. 3.5). The two hyperparameters μ and η are selected from $\{0.01, 0.05, 0.1, 0.5, 1\}$, providing flexibility to balance the contributions of (i) source label supervision in each branch, (ii) global structural alignment, and (iii) cross-branch agreement on the target side. Their final values per task are summarised in Table 8, and the per-benchmark analysis below explains how each setting reflects the underlying cross-scale property.

Airport Dataset. The Airport benchmark contains very small graphs (131–1,190 nodes) with sparse connectivity and low homophily, so even a moderate sampling ratio removes a substantial fraction of edges and produces severely truncated source neighborhoods. In this regime, the global branch is the only component that still observes a stable, sampling-robust signal (degree, clustering, k-hop neighbourhood size), whereas the local branch is particularly vulnerable to incomplete adjacency. We therefore use a *relatively large* $\mu \in \{0.5\}$ to let the global descriptor distribution drive cross-domain alignment, and a *moderate-to-large* $\eta \in \{0.1, 0.5\}$ to encourage the local branch to agree with the global branch on confident target nodes, preventing the GCN from being misled by sparse and structurally incomplete neighborhoods.

Citation Dataset. The Citation benchmark consists of medium-sized paper-citation networks (5,484–9,360 nodes) that share a stable five-class label space and exhibit relatively high topical homophily across domains. Even after sampling, the residual local topology around each source node is still largely informative, so the local GCN branch alone can already produce reliable embeddings, and over-emphasising the global branch can blur the fine-grained class-prototype matching used in Eq. 5. We therefore use *small* $\mu \in \{0.1\}$ and *small* η ($\{0.1, 0.5\}$, with $\eta = 0.5$ chosen only on C→A, the task with the largest source-to-target scale gap), which keeps the classifier dominated by source supervision while still letting the global branch provide a smooth correction.

Social Dataset (Blog). The Blog graphs are dense, attribute-rich social networks with markedly different connectivity densities between B1 and B2. In this setting, both local neighborhoods and global degree statistics are sensitive to the random sampling of source nodes, and pushing either branch too hard tends to overfit the sampling noise rather than the true structural prior. We accordingly choose *small* values $\mu = 0.05$ and $\eta = 0.1$, which lets the two branches act as mild regularisers for each other and keeps the overall optimisation dominated by reliable source-side supervision and feature-space alignment.

ACM Dataset (Homophily Shift). The ACM benchmark is purposely constructed to expose homophily shift: A3 (Paper–Subject–Paper) is highly homophilic and extremely dense, while A4 (Paper–Author–Paper) is sparser and far less homophilic. The two transfer directions are therefore *asymmetric*, which is directly reflected in the chosen hyperparameters. For A3→A4, the source graph is structurally richer than the target, so the global branch carries most of the cross-domain signal and

we use $\mu=0.5$, $\eta=0.1$ (rely on the global branch, only mildly enforce local–global consistency). For A4→A3, in contrast, the source is sparse and heterophily-dominated while the target is dense; here the local GCN branch needs more freedom to learn its own representation, and the consistency loss becomes the main bridge to the target, so we instead use $\mu=0.1$, $\eta=0.5$. Across both directions, jointly tuning μ and η is essential to absorb the homophily gap that single-branch GDA methods leave unaddressed.

F Description of Algorithm LGSA

Algorithm 1 summarizes the full training procedure of **LGSA**. The cross-domain augmented source graph $\widetilde{G}^{\mathrm{S}}$, the propagated label-signal matrix $\widetilde{\mathbf{Y}}^{\mathrm{S}}$, and the structural descriptors $\mathbf{\Phi}^{\mathrm{S}}$, $\mathbf{\Phi}^{\mathrm{T}}$ depend only on input graphs/labels and are therefore computed once before training. The two branches and the adaptive fusion module are then jointly optimised for E epochs; the confident target set $\mathcal{V}^{\mathrm{T}}_{\mathrm{conf}}$ and the pseudo labels $\{\widehat{y}^{\mathrm{T}}_v\}$ are refreshed once per epoch using *detached* predictions, so no gradient is back-propagated through pseudo-label selection. We use E for the maximum number of epochs to avoid notational clash with the consistency-loss coefficient η in Eq. 13.

Algorithm 1: The proposed algorithm **LGSA**

Input: Subsampled source graph $G^{\mathrm{S}}=\{\mathcal{V}^{\mathrm{S}},\mathcal{E}^{\mathrm{S}},\mathbf{A}^{\mathrm{S}},\mathbf{X}^{\mathrm{S}},\mathbf{Y}^{\mathrm{S}}\}$; target graph $G^{\mathrm{T}}=\{\mathcal{V}^{\mathrm{T}},\mathcal{E}^{\mathrm{T}},\mathbf{A}^{\mathrm{T}},\mathbf{X}^{\mathrm{T}}\}$; augmentation size M; propagation steps T and teleport α; pseudo-label threshold τ_p; contrastive temperature τ_c; trade-offs μ,η; maximum epochs E.

`// One-time pre-processing`

For each $v_s\in\mathcal{V}^{\mathrm{S}}$, compute $\mathcal{N}_{\mathrm{aug}}(v_s)=\mathrm{TopM}_{v_t\in\mathcal{V}^{\mathrm{T}}}\cos(\mathbf{x}^{\mathrm{S}}_{v_s},\mathbf{x}^{\mathrm{T}}_{v_t})$;

Build augmented source graph $\widetilde{G}^{\mathrm{S}}$ via Eq. 3;

Propagate label signals on $\widetilde{\mathbf{A}}^{\mathrm{S}}$ for T steps with teleport α to obtain $\widetilde{\mathbf{Y}}^{\mathrm{S}}$;

Compute structural descriptors $\mathbf{\Phi}^{\mathrm{S}},\mathbf{\Phi}^{\mathrm{T}}$ from $\big(\deg(v),c(v),n_2(v),n_3(v)\big)$;

Initialize $\mathcal{V}^{\mathrm{T}}_{\mathrm{conf}}\leftarrow\varnothing$;

for $it=1$ **to** E **do**

  `// Local branch over` $\widetilde{G}^{\mathrm{S}}$ `and` G^{T}

  $\mathbf{H}^{\mathrm{S}}_{\mathrm{loc}}\leftarrow\mathrm{GCN}_{\mathrm{loc}}\big(\widetilde{\mathbf{A}}^{\mathrm{S}},[\widetilde{\mathbf{X}}^{\mathrm{S}}\,\|\,\widetilde{\mathbf{Y}}^{\mathrm{S}}]\big)$ restricted to $\mathcal{V}^{\mathrm{S}}$; $\mathbf{H}^{\mathrm{T}}_{\mathrm{loc}}\leftarrow\mathrm{GCN}_{\mathrm{loc}}\big(\mathbf{A}^{\mathrm{T}},[\mathbf{X}^{\mathrm{T}}\,\|\,\mathbf{0}]\big)$;

  Local logits $\mathbf{Z}^{\mathrm{S}}_{\mathrm{loc}},\mathbf{Z}^{\mathrm{T}}_{\mathrm{loc}}$ via the shared linear classifier;

  `// Global branch over` $[\mathbf{X}\,\|\,\mathbf{\Phi}]$

  $\mathbf{H}^{\mathrm{S}}_{\mathrm{glo}}\leftarrow\mathrm{MLP}\big([\mathbf{X}^{\mathrm{S}}\,\|\,\mathbf{\Phi}^{\mathrm{S}}]\big)$; $\mathbf{H}^{\mathrm{T}}_{\mathrm{glo}}\leftarrow\mathrm{MLP}\big([\mathbf{X}^{\mathrm{T}}\,\|\,\mathbf{\Phi}^{\mathrm{T}}]\big)$;

  Global logits $\mathbf{Z}^{\mathrm{S}}_{\mathrm{glo}},\mathbf{Z}^{\mathrm{T}}_{\mathrm{glo}}$ via the shared linear classifier;

  `// Adaptive GCN-MLP fusion`

  Compute per-node reliabilities $r_{\mathrm{loc},v},r_{\mathrm{glo},v}$ by Eq. 8;

  $\gamma_v\leftarrow\sigma\big(\mathrm{softplus}(\omega_g)(r_{\mathrm{loc},v}-r_{\mathrm{glo},v})+b_g\big)$;

  $\mathbf{h}_v\leftarrow\gamma_v\mathbf{h}_{\mathrm{loc},v}+(1-\gamma_v)\mathbf{h}_{\mathrm{glo},v}$, $\mathbf{z}_v\leftarrow\mathbf{h}_v\mathbf{W}_o+\mathbf{b}_o$;

  `// Pseudo-label refresh (detached) and prototype update`

  $\mathcal{V}^{\mathrm{T}}_{\mathrm{conf}}\leftarrow\big\{v\in\mathcal{V}^{\mathrm{T}}:\max_c[\mathrm{softmax}(\mathbf{z}^{\mathrm{T}}_v)]_c\ge\tau_p\big\}$, $\widehat{y}^{\mathrm{T}}_v\leftarrow\arg\max_c(\mathbf{z}^{\mathrm{T}}_v)_c$;

  Update target class prototypes $\{\bar{\mathbf{h}}^{\mathrm{T}}_{\mathrm{loc},c}\}_{c=1}^{C}$ on $\mathcal{V}^{\mathrm{T}}_{\mathrm{conf}}$ for use in Eq. 5;

  `// Local-global consistency`

  Compute projections $\mathbf{q}^{\mathrm{T}}_{\mathrm{loc},v},\mathbf{q}^{\mathrm{T}}_{\mathrm{glo},v}$ by Eq. 11;

  `// Loss aggregation and parameter update`

  Compute $\mathcal{L}_{\mathrm{loc}},\mathcal{L}_{\mathrm{glo}},\mathcal{L}_{\mathrm{fuse}},\mathcal{L}_{\mathrm{cons}}$ by Eqs. 5, 7, 10, 12;

  Compute the overall loss $\mathcal{L}\leftarrow\mathcal{L}_{\mathrm{fuse}}+\mu\mathcal{L}_{\mathrm{glo}}+(1-\mu)\mathcal{L}_{\mathrm{loc}}+\eta\mathcal{L}_{\mathrm{cons}}$ (Eq. 13);

  Update all parameters of the framework by back-propagating $\mathcal{L}$;

end

Predict the labels of target-graph nodes:;

$\widehat{\mathbf{Y}}^{\mathrm{T}}\leftarrow\arg\max\ \mathrm{softmax}(\mathbf{Z}^{\mathrm{T}})$ from the trained model;

Output: Classification result $\widehat{\mathbf{Y}}^{\mathrm{T}}$

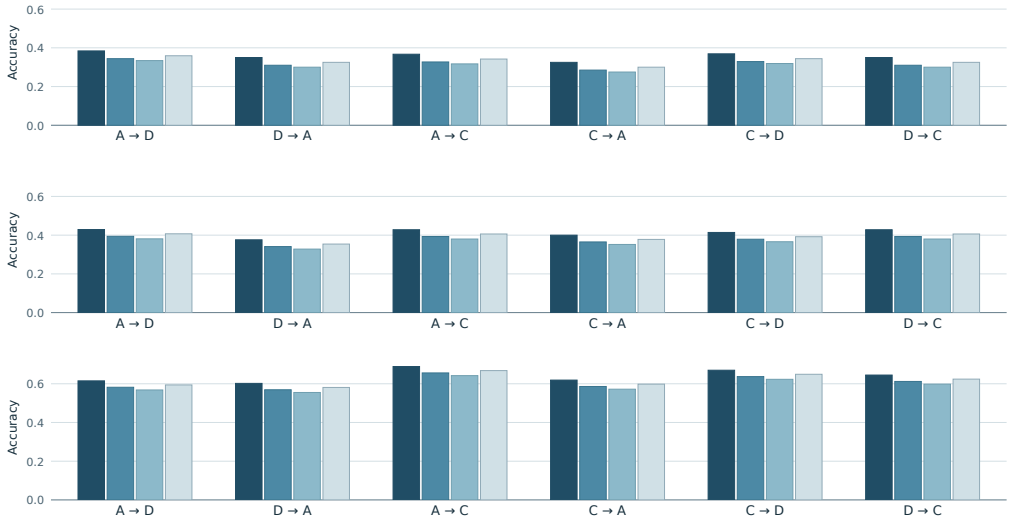


Figure 6: Ablation study on the citation benchmarks across the six transfers among *ACMv9* (A), *DBLPv7* (D), and *Citationv1* (C). From left to right: $\rho = 1\%, 5\%, 10\%$. The full LGSA outperforms LGSA_1 (no cross-domain augmentation), LGSA_2 (no global branch), and LGSA_3 (uniform fusion, no $\mathcal{L}_{\text{cons}}$) on every transfer; LGSA_2 is consistently the most damaging variant, and the gap widens as ρ decreases.

G Additional Ablation Results on Citation and ACM/Blog Benchmarks

We complement the ablation study of Sec. 5.2 (Airport) with results on the remaining benchmarks: six transfers among the citation networks *ACMv9* (A), *DBLPv7* (D), and *Citationv1* (C), and four transfers among two ACM time-splits (A3, A4) and two Blog domains (B1, B2). The three ablated variants LGSA_1, LGSA_2, LGSA_3, the sample rates $\rho \in \{1\%, 5\%, 10\%\}$, and all hyperparameters are identical to the main experiments.

Figures 6 and 7 show that the conclusions drawn on Airport carry over without exception:

- The full **LGSA** dominates all three variants on every $(\text{source} \to \text{target}, \rho)$ pair.
- Removing the global branch (LGSA_2) is again the most damaging ablation, costing on average $\sim 5\%$ accuracy and re-confirming that the descriptor $\phi(v)$ together with its mean-embedding alignment is indispensable for absorbing the sampling-induced first-moment shift on $\mathbf{\Phi}$.
- Removing cross-domain augmentation (LGSA_1) and the adaptive fusion (LGSA_3) yield smaller but consistent drops of $\sim$3–4% and $\sim$2–3%, respectively.
- The same monotone trend in ρ holds: every component contributes more as ρ shrinks, with the largest gaps at $\rho = 1\%$, where CSGDA is hardest.

H Datasets

We evaluate our method on four families of GDA benchmarks—**Airport**, **Citation**, **Social**, and **ACM**—covering air-traffic, citation, blogger-interaction, and academic co-occurrence networks. The detailed statistics are summarized in Table 9.

Airport. The Airport benchmark [86] consists of three air-traffic networks collected from the USA (**U**), Brazil (**B**), and Europe (**E**). Each node corresponds to an airport and each edge denotes the existence of a commercial flight between two airports. Node labels indicate four levels of airport activity, quantized from passenger flow, so that the classification task reflects the structural role of each airport in its national air-traffic system. Since the three networks differ substantially in size

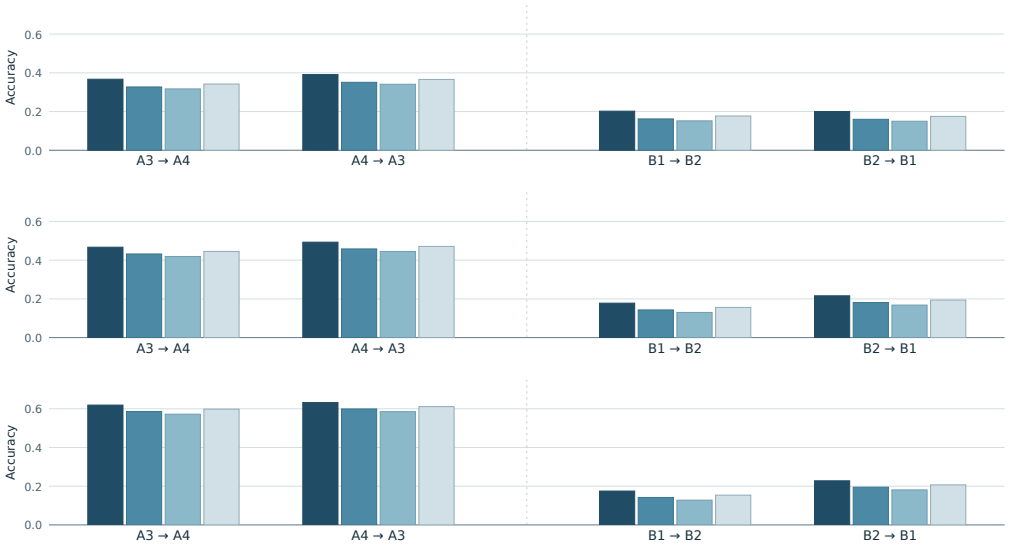


Figure 7: Ablation study on the ACM time-split (A3, A4) and Blog (B1, B2) benchmarks across the four transfers A3 $\rightarrow$ A4, A4 $\rightarrow$ A3, B1 $\rightarrow$ B2, B2 $\rightarrow$ B1. From left to right: $\rho = 1\%, 5\%, 10\%$. Trends match those on Airport and the citation benchmarks: the full LGSA dominates every variant, LGSA_2 is the most damaging ablation, and the per-component contributions grow monotonically as ρ shrinks.

Table 9: Statistics of the datasets used in our experiments.

Types	Datasets	#Node	#Edge	#Label
Airport	USA	1,190	13,599	4
	Brazil	131	1,038	
	Europe	399	5,995	
Citation	ACMv9	9,360	15,556	5
	Citationv1	8,935	15,098	
	DBLPv7	5,484	8,117	
Social	Blog1	2,300	33,471	6
	Blog2	2,896	53,836	
ACM	ACM3	3,025	2,221,699	3
	ACM4	4,019	57,853	

(from 131 to 1,190 nodes) yet share the same label semantics, this benchmark is a canonical testbed for cross-scale and cross-region GDA.

Citation. The Citation benchmark [8] contains three paper-citation networks—**ACMv9** (**A**), **Citationv1** (**C**), and **DBLPv7** (**D**)—extracted from ArnetMiner. Nodes are research papers and edges denote citation relations. Each paper is represented by a bag-of-words feature derived from its title/abstract and is labeled with one of five research areas shared across all three networks. The three graphs are collected from different sources and time periods, leading to non-trivial feature, structure, and label-distribution shifts that are widely used to benchmark cross-domain node classification.

Social. The Social benchmark consists of two BlogCatalog [87] networks, **Blog1** (**B1**) and **Blog2** (**B2**). Nodes correspond to bloggers and edges encode friendship relations on the BlogCatalog platform. Each blogger is associated with one of six interest categories. The two graphs are crawled from disjoint sub-communities of the same platform, sharing the label space but exhibiting different connectivity densities and topical mixtures, which makes them well suited for evaluating transfer between social subgraphs.

ACM (homophily-shift). To probe the behavior of GDA methods under *homophily distribution shift*, we additionally use two ACM-derived graphs constructed on the *same* set of papers but with *different* relations: **ACM3** (**A3**) is built from the ACM Paper-Subject-Paper meta-path [72], and **ACM4** (**A4**) is built from the ACM Paper-Author-Paper meta-path [73]. Both graphs share the same three-class label space (research field), but the subject-based view is highly homophilic and extremely dense (over 2.2×10^6 edges on $\sim$3K nodes), whereas the author-based view is much sparser and exhibits a markedly lower homophily ratio. This pair therefore isolates the effect of homophily shift while controlling for label semantics.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We present the contributions and scope in the abstract and introduction.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We have discussed the limitations in the conclusion section.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

Justification: This paper does not include theoretical results.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: All the information required to reproduce the main results is included in the experiment settings.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: Source code will be released after acceptance.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Relevant details are discussed in the experiment settings.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We report results averaged over five runs with different random seeds.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We describe the compute resources in the implementation details.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conducted in the paper conforms with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: The potential social impacts are discussed in the conclusion.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: This paper aims at data adaptation, and there are no safety risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: Existing assets are properly cited.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: Our code will be released after acceptance with detailed documents.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: This paper does not involve these contents.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The experiments in this paper do not involve human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: LLM is used only for polishing the writing. The core method development in this research does not involve LLMs as important components.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.