Gradual Adversarial Graph Transfer: Stabilizing Alignment with Dual Transitional Domains

Xing Wei
Tongji University
Shanghai, China
xing627@tongji.edu.cn

Jiayi Yang
Tongji University
Shanghai, China
2111125@tongji.edu.cn

Chunchun Chen
Tongji University
Shanghai, China
c2chen@tongji.edu.cn

Chenrun Wang
Tongji University
Shanghai, China
wcr0905@tongji.edu.cn

Xin Sun
City University of Macau
Macao, China
sunxin1984@ieee.org

Ming Li
Zhejiang Normal University
Jinhua, China
mingli@zjnu.edu.cn

Rui Fan
Tongji University
Shanghai, China
rui.fan@ieee.org

Hanli Wang
Tongji University
Shanghai, China
hanliwang@tongji.edu.cn

Wei Ye*
Tongji University
Shanghai, China
yew@tongji.edu.cn

Abstract

Graph domain adaptation under large source-target shifts is challenging because direct endpoint alignment often causes unstable adversarial optimization, while explicit intermediate-graph generation may introduce topology distortion and semantic artifacts. To address this problem, we propose Gradual Adversarial Graph Transfer (GAGT), which formulates cross-domain graph transfer as constructing a semantically feasible latent transition pathway. GAGT first learns domain-invariant and domain-specific representations through contrastive disentangled representation learning with tolerance-bounded reconstruction, where the reconstruction tolerance regularizes a semantically feasible region rather than forcing exact recovery. Based on this, GAGT synthesizes dual transitional domains as bidirectional bridge states and performs gradual adversarial alignment along the pathway. In this way, a large source-target discrepancy is decomposed into smaller and semantically constrained local discrepancies, yielding smoother adversarial signals and more stable transfer. Extensive experiments on diverse graph domain adaptation benchmarks demonstrate that GAGT outperforms nearly all state-of-the-art methods, especially under large structural and feature distribution shifts. Our code is available at https://github.com/GEEX-Weixing/GAGT-CIKM.

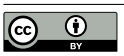
CCS Concepts

• **Computing methodologies** → **Neural networks**.

Keywords

Graph Domain Adaptation, Graph Transfer, Stable Alignment, Adversarial Training

*Corresponding author.



This work is licensed under a Creative Commons Attribution 4.0 International License.
CIKM '26, Rome, Italy.
© 2026 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-2539-5/2026/11
https://doi.org/10.1145/3799682.3840648

ACM Reference Format:
Xing Wei, Jiayi Yang, Chunchun Chen, Chenrun Wang, Xin Sun, Ming Li, Rui Fan, Hanli Wang, and Wei Ye. 2026. Gradual Adversarial Graph Transfer: Stabilizing Alignment with Dual Transitional Domains. In *Proceedings of the 35th ACM International Conference on Information and Knowledge Management (CIKM '26), November 7–11, 2026, Rome, Italy.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3799682.3840648

1 Introduction

Graph domain shift [26] refers to discrepancies between graph domains in topology and/or node feature distributions, where a model trained on a labeled source graph may not generalize reliably to an unlabeled target graph at deployment. Graph inductive learning [15, 45] or graph domain generalization [30, 40] methods improve models' performance by learning transferable inductive biases across graphs. But their failure to leverage knowledge from the unlabeled target graph during training prevents them from effectively handling source-target discrepancies. To this end, graph domain adaptation (GDA) [36–38] is proposed to leverage the unlabeled target graph to reduce distribution discrepancy, typically by aligning source and target representations, so that underlying knowledge learned from the labeled source graph can be transferred more reliably to the shifted target graph. It enables graph neural networks (GNNs) [4, 5, 21, 46, 47, 50, 55, 58] to be applied to more scenarios, particularly cross-network node classification.

A popular paradigm for alignment is adversarial training [2, 43, 44], where a domain discriminator distinguishes source from target while the generator is optimized to confuse it. However, under substantial domain shifts, one-step source-target alignment often yields unstable min-max dynamics and shortcut matching, in which domain-correlated but task-irrelevant cues dominate optimization, leading to negative transfer [42, 53, 54]. A natural remedy is to construct an adaptation pathway by decomposing the large domain gap into smaller transitions. Existing methods synthesize intermediate graphs through optimal-transport-based coupling, soft-to-hard adjacency discretization, or diffusion-based graph reconstruction [7, 16, 51]. However, these explicit graph-generation

strategies may introduce dense couplings, discretization artifacts, generation noise, or topology distortion. Once such artifacts become entangled with task semantics, alignment may be dominated by domain-correlated shortcut cues rather than transferable semantics, thereby exacerbating negative transfer under large shifts.

This raises a central question: **how can we construct a transition pathway that is both semantically valid and free from explicit topology distortion under large graph domain shifts?** To answer this question, we propose Gradual Adversarial Graph Transfer (GAGT), which constructs a semantically feasible latent transition pathway instead of generating explicit intermediate graphs. The key idea is to make the transfer process decomposable, constrained, and locally alignable. Specifically, GAGT first learns a disentangled latent space in which the domain-invariant component preserves task-relevant semantics while the domain-specific component captures domain-conditioned variations. A tolerance-bounded reconstruction objective further regularizes a feasible semantic region, ensuring that the latent space remains sufficiently informative for prediction while avoiding over-constrained reconstruction. Within this constrained space, a knowledge transition module synthesizes two bidirectional transitional domains, forming latent bridge states between the source and target domains. Gradual adversarial alignment is then performed along this pathway, so that the model aligns several smaller and semantically valid local discrepancies rather than one large and unconstrained endpoint discrepancy. Our contributions are summarized as follows:

- We formulate large-shift graph domain adaptation as the construction of a semantically feasible latent transition pathway, which decomposes a large source-target discrepancy into smaller and locally alignable transfer steps.
- We propose a tolerance-bounded contrastive disentanglement mechanism to build the semantic basis of this pathway. It preserves task-relevant semantics without forcing exact reconstruction and provides a controllable semantic tolerance budget that guides stable transition synthesis.
- We design a bidirectional knowledge transition module to synthesize dual transitional domains and perform gradual adversarial alignment along the resulting pathway. Extensive experiments verify its effectiveness, stability, and efficiency across diverse graph domain adaptation tasks.

2 Related Work

2.1 Adversarial Alignment

Traditional adversarial alignment methods operate in a shared embedding space, where a domain discriminator distinguishes the source from the target and the generator is trained to fool it [12, 56]. Many variants stabilize or refine this paradigm by introducing graph-specific inductive biases: topology proximity constraints [9, 35], dual encoders for consistency [44], spectral regularization [49], adaptive propagation for improved theoretical adaptation bounds [25], or adaptive spectral discrepancy alignment [6].

2.2 Gradual Graph Domain Adaptation

Beyond one-step alignment, several methods construct gradual transitions by synthesizing intermediate graphs or modifying graphs to narrow the domain shifts. Graph mixup approaches [23, 41, 51, 52] typically apply knowledge-preserving transformations (e.g., Fused Gromov-Wasserstein distance [31]) and interpolate between transformed graphs. In addition, DiffGDA [7] models gradual adaptation as a continuous evolution via a diffusion process [17].

2.3 Topology-aware Alignment

Since graph domain shifts often involve structural discrepancies, some methods consider topology-aware alignment. They match local substructures or enforce structure consistency across graphs, including subtree-based discrepancy estimation [43], source-side edge reweighting [28], edge recalibration to handle conditional structure shifts [29], target-side reconstruction to preserve neighborhood coherence [8], and homophily-aware signal alignment [10].

2.4 Representation Disentanglement

Representation disentanglement for GDA [27] separates transferable semantics from domain-specific factors. Subspace decomposition [54] uses orthogonality, while DIVA [19] models domain-specific, class-relevant, and residual factors with conditional priors. GraphDA [2], DisEncDR [3], and DGSDA [48] further use maximum-entropy, mutual-information, and attribute-topology spectral constraints, respectively. Contrastive approaches [5] likewise separate shared semantics from domain-specific noise in cross-domain recommendation [14, 24].

Different from these methods, GAGT formulates large-shift graph domain adaptation as a semantically constrained latent transition process rather than direct endpoint alignment. It constructs task-relevant, locally alignable intermediate states to progressively reduce the source-target discrepancy, avoiding explicit intermediate-graph generation and associated topology artifacts while suppressing domain-specific shortcuts.

3 Methodology

Preliminaries: We consider two graph domains: a labeled source graph $\mathcal{G}^S = (\mathcal{V}^S, \mathcal{E}^S, \mathbf{X}^S, \mathbf{A}^S, \mathbf{Y}^S)$ and an unlabeled target graph $\mathcal{G}^T = (\mathcal{V}^T, \mathcal{E}^T, \mathbf{X}^T, \mathbf{A}^T)$. Both graphs share the same label space, but the target labels are unavailable during training. A graph is denoted by $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{X}, \mathbf{A}, \mathbf{Y})$, where $\mathcal{V} = \{v_1, \ldots, v_N\}$ is the node set and $\mathcal{E}$ is the edge set. $\mathbf{X} = [\mathbf{x}_1^\top; \ldots; \mathbf{x}_N^\top] \in \mathbb{R}^{N \times F}$ is the node feature matrix, and $\mathbf{A} \in \mathbb{R}^{N \times N}$ is the adjacency matrix with $\mathbf{A}_{i,j} = 1$ if an edge $e_{i,j} \in \mathcal{E}$ exists between v_i and v_j, and $\mathbf{A}_{i,j} = 0$ otherwise. $\mathbf{Y} \in \mathbb{R}^{N \times C}$ is the node label matrix, where N is the number of nodes and C is the number of classes. Given the labeled source graph and the unlabeled target graph, our goal is to learn a node classifier that transfers task-discriminative knowledge from the source to the target and predicts target-node labels without accessing target labels during training.

Overview: We propose the Gradual Adversarial Graph Transfer (GAGT) framework, shown in Figure 1, for graph domain adaptation under large shifts. GAGT first learns domain-invariant and domain-specific representations through contrastive disentanglement with tolerance-bounded reconstruction. It then synthesizes dual transitional domains in the latent space and aligns adjacent states adversarially, decomposing the source-target gap into smaller local

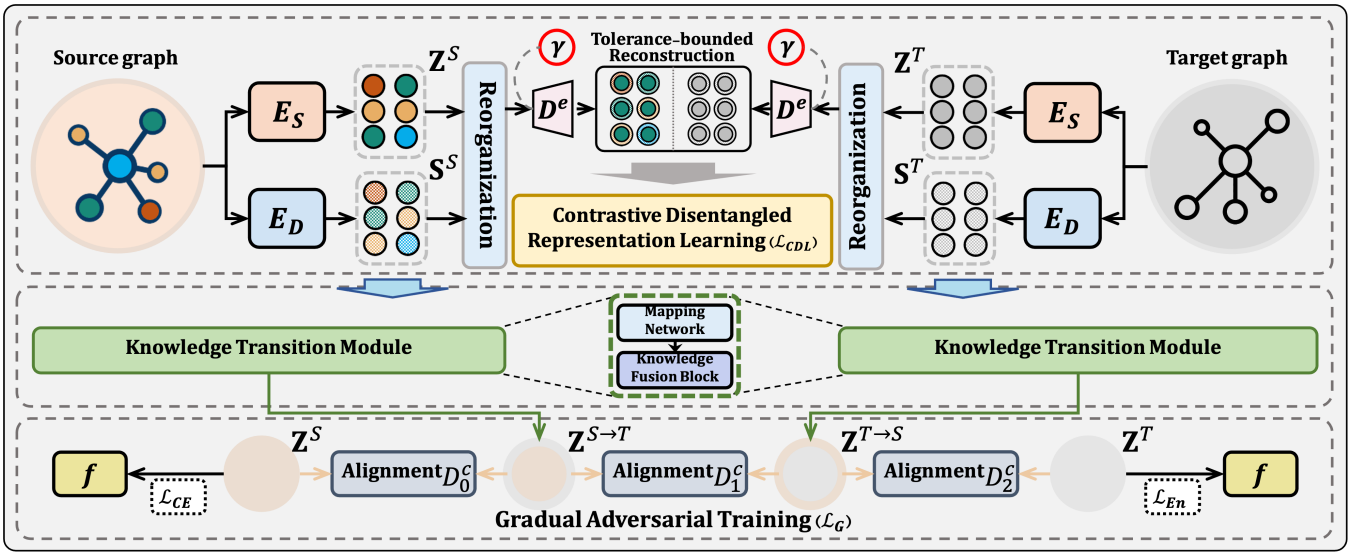


Figure 1: Overview of GAGT. GAGT first learns disentangled source and target representations through contrastive learning and tolerance-bounded reconstruction, preserving graph-semantic information in the latent space while allowing controlled transition flexibility. Based on this semantically feasible latent space, GAGT synthesizes dual transitional domains to construct an adaptation pathway and performs gradual adversarial alignment between adjacent latent states for stable target prediction.

discrepancies. A conservative tolerance bound motivates, rather than directly determines, the practical semantic-flexibility budget, enabling smoother and more stable adaptation.

3.1 Contrastive Disentangled Representation Learning

Before constructing the latent adaptation pathway, GAGT first learns a disentangled representation space in which task-relevant semantics are separated from domain-induced variations. This step is crucial because transitional domains synthesized from entangled representations may inherit both task-relevant semantics and domain-correlated noise, causing the adaptation pathway to be dominated by shortcut cues and leading to negative transfer. Existing disentangling methods usually separate domain-invariant and domain-specific factors through orthogonality [2] or cosine similarity [34] constraints. However, these constraints do not explicitly guarantee that the domain-invariant representation preserves sufficient task-discriminative semantics, which may cause subsequent alignment to rely on spurious domain cues [53]. To address this limitation, we propose contrastive disentangled representation learning with tolerance-bounded reconstruction. The contrastive objective guides the separation of domain-invariant and domain-specific factors through representation reorganization, while the reconstruction tolerance preserves semantic sufficiency without over-constraining the latent space.

To achieve this goal, we employ an asymmetric two-stream encoder architecture. The domain-invariant encoder (E_S) has stronger modeling capacity to capture stable semantic patterns from node attributes and graph topology, whereas the domain-specific encoder (E_D) is deliberately kept lightweight to absorb domain-specific knowledge and reduce semantic leakage into the domain-specific branch. This design encourages the two branches to encode complementary factors rather than duplicating the same information:

$$\mathbf{Z} = E_S(\mathbf{X}, \mathbf{A}), \quad \mathbf{S} = E_D(\mathbf{X}, \mathbf{A}), \tag{1}$$

where each encoder outputs the mean and log-variance of a Gaussian posterior, and $\mathbf{Z} = [\mathbf{z}_1, \ldots, \mathbf{z}_N]$ and $\mathbf{S} = [\mathbf{s}_1, \ldots, \mathbf{s}_N]$ denote the reparameterized domain-invariant and domain-specific representations, respectively. We use $\{\mathbf{Z}^S, \mathbf{S}^S\}$ and $\{\mathbf{Z}^T, \mathbf{S}^T\}$ to distinguish representations from different domains.

We then construct contrastive reconstruction views by reorganizing the disentangled representations. For node v_i, the positive view $\hat{\mathbf{x}}_{i,i}$ is reconstructed from its matched pair $[\mathbf{z}_i, \mathbf{s}_i]$, while the negative view $\hat{\mathbf{x}}_{i,k}$ is reconstructed by replacing $\mathbf{s}_i$ with the domain-specific representation $\mathbf{s}_k$ of another node from the same domain. This reorganization creates contrastive pairs that test whether the learned domain-invariant representation can preserve stable semantics while domain-specific variations are separated into S. For the positive view, we impose a tolerance-bounded feature reconstruction constraint:

$$\mathcal{L}_R^f = \frac{1}{|\mathcal{V}|} \sum_{i=1}^{|\mathcal{V}|} \max\left\{ \left\| \mathbf{x}_i - \hat{\mathbf{x}}_{i,i} \right\|_2 - \gamma,\ 0 \right\}, \tag{2}$$

where γ is the reconstruction tolerance. The gradient becomes zero when the reconstruction error falls below γ, ensuring that $\mathbf{z}_i$ retains stable semantic information when combined with contrastive reorganization without over-constraining the representation.

We deploy the variational structure reconstruction constraint [22] as:

$$\mathcal{L}_R^s = -\mathbb{E}_{q_\theta(\mathbf{H}|\mathbf{X},\mathbf{A})}[\log p_\theta(\mathbf{A}|\mathbf{H})] + \mathrm{KL}\left[q_\theta(\mathbf{H}|\mathbf{X},\mathbf{A}) \| p_\theta(\mathbf{H})\right], \tag{3}$$

where $\mathbf{H} = [\mathbf{Z}, \mathbf{S}]$ denotes the concatenated latent representation. Following VGAE [22], $q_\theta(\mathbf{H}|\mathbf{X}, \mathbf{A})$ is modeled as a factorized Gaussian whose parameters are produced by the encoders, and $p(\mathbf{H})$ is set to a standard Gaussian prior. The likelihood term reconstructs topology with a Bernoulli inner-product decoder, i.e., $\hat{\mathbf{A}}_{ij} = \sigma(\mathbf{h}_i^\top \mathbf{h}_j)$, where $\mathbf{h}_i = [\mathbf{z}_i; \mathbf{s}_i]$. The KL term regularizes the posterior toward the prior to prevent an irregular latent space.

After reconstruction, the contrastive disentanglement is applied to decompose the representations by encouraging similarity between the positive pairs and dissimilarity between the negative pairs. The positive view should be closer to the original node's feature, while the negative views should be pushed further apart:

$$\mathcal{L}_{CDL} = -\frac{1}{|\mathcal{V}|}\sum_{i=1}^{|\mathcal{V}|}\log\frac{e^{\mathrm{sim}(\mathbf{x}_i,\hat{\mathbf{x}}_{i,i})/\tau}}{e^{\mathrm{sim}(\mathbf{x}_i,\hat{\mathbf{x}}_{i,i})/\tau}+\sum_{k=1}^{K}e^{\mathrm{sim}(\mathbf{x}_i,\hat{\mathbf{x}}_{i,k})/\tau}}, \quad (4)$$

where $\mathrm{sim}(\cdot,\cdot)$ denotes the cosine similarity function. $\hat{\mathbf{x}}_{i,i}$ and $\hat{\mathbf{x}}_{i,k}$ are the positive and negative views. τ is the temperature coefficient (set to 0.5). K is the number of negative samples.

Minimizing $\mathcal{L}_{CDL}$ pulls the original node feature closer to its positive reconstruction while pushing away reconstructions produced by mismatched domain-specific factors. In this way, the model is encouraged to preserve stable semantic information in the domain-invariant representation and to assign domain-dependent variations to the domain-specific representation. This intended separation is empirically supported by the probing diagnosis in Appendix B. The tolerance-bounded reconstruction further prevents the encoder from overfitting nuisance details, allowing the learned domain-invariant representation to remain semantically sufficient but not overly constrained. $\mathbf{Z}$ serves as a reliable semantic anchor for later transitional domain synthesis, while $\mathbf{S}$ provides domain-conditioned information for modeling domain transition.

3.2 Latent Domain Adaptation Pathway Construction

After separating domain-invariant knowledge from domain-specific knowledge, the discrepancy between the source and target domains may persist in the latent space, which requires further alignment to facilitate knowledge transfer. Direct adversarial alignment between the two domains can be unstable under large shifts, potentially leading to poor graph transfer. To stabilize this process, we decompose the large source-target gap into a sequence of smaller, well-conditioned transitions by introducing transitional domains.

Instead of explicitly generating intermediate graphs, which may distort topology and introduce structural artifacts, we construct the adaptation pathway directly in the latent space. The tolerance-bounded reconstruction constraint regularizes a semantically feasible region for this pathway: smaller γ values enforce stricter semantic faithfulness, while larger γ values provide more flexibility for bridging larger shifts. Based on this, the proposed knowledge transition module synthesizes two transitional domains, $\mathbf{Z}^{S\to T}$ and $\mathbf{Z}^{T\to S}$, by fusing domain-conditioned information into the domain-invariant representations. These two latent bridge states connect the source and target domains from opposite directions, allowing the subsequent adversarial alignment to operate on adjacent and smaller discrepancies, while improving the smoothness and stability of the graph domain adaptation under large domain shifts.

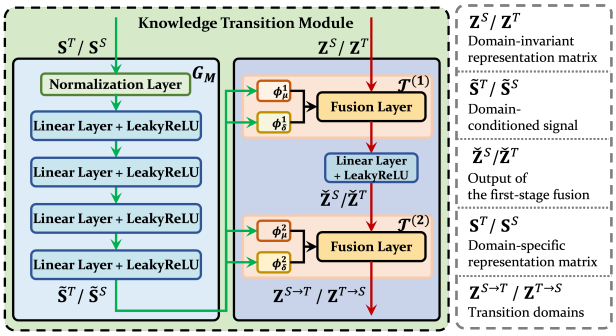


Figure 2: Architecture of the Knowledge Transition Module.

As shown in Figure 2, the knowledge transition module comprises a mapping network and two-stage knowledge fusion blocks. Given the domain-specific representation matrix ($\mathbf{S}^S$ or $\mathbf{S}^T$), the mapping network (G_M) first transforms it into a compact domain signal:

$$\widetilde{\mathbf{S}} = G_M(\mathbf{S}), \quad (5)$$

where $G_M(\cdot)$ consists of a normalization layer followed by four linear layers. The resulting $\widetilde{\mathbf{S}}$ serves as a domain-conditioned signal to parameterize two fusion stages.

Although the fusion operation uses graph-level statistics, it does not operate on topology-free features. Both $\mathbf{Z}$ and $\mathbf{S}$ are produced by graph encoders conditioned on $\mathbf{A}$ in Eq. (1), while their joint representation is further regularized by topology reconstruction in Eq. (3). Thus, the graph structure is encoded before transition synthesis, while the module deliberately avoids explicitly generating intermediate adjacency matrices. Given $\mathbf{Z}$, we compute $\mu(\mathbf{Z})$ and $\mathrm{std}(\mathbf{Z})$ over nodes and broadcast the domain-conditioned parameters from $\widetilde{\mathbf{S}}$ to all nodes. Concretely, each fusion stage is conditioned on $\widetilde{\mathbf{S}}$ through two lightweight predictors $\phi_\mu^{(t)}(\cdot)$ and $\phi_\delta^{(t)}(\cdot)$ for $t \in \{1, 2\}$. The first-stage fusion ($\mathcal{T}^{(1)}$) is

$$\check{\mathbf{Z}} = \mathcal{T}^{(1)}(\mathbf{Z}; \widetilde{\mathbf{S}}) = \phi_\delta^{(1)}(\widetilde{\mathbf{S}}) \odot \frac{\mathbf{Z} - \mu(\mathbf{Z})}{\mathrm{std}(\mathbf{Z})} + \phi_\mu^{(1)}(\widetilde{\mathbf{S}}), \quad (6)$$

where $\mu(\mathbf{Z}) = \frac{1}{|\mathcal{V}|}\sum_{i=1}^{|\mathcal{V}|}\mathbf{z}_i$ and $\mathrm{std}(\mathbf{Z}) = \sqrt{\frac{1}{|\mathcal{V}|}\sum_{i=1}^{|\mathcal{V}|}(\mathbf{z}_i - \mu(\mathbf{Z}))^{\odot 2} + \zeta}$ denote the mean and standard deviation computed over all nodes in a graph, where ζ is the numerical stability term. The outputs of $\phi_\mu^{(1)}(\widetilde{\mathbf{S}})$ and $\phi_\delta^{(1)}(\widetilde{\mathbf{S}})$ are broadcast to all nodes. The operator $\odot$ denotes element-wise multiplication, and $^{\odot 2}$ denotes element-wise square.

Similarly, the second-stage fusion ($\mathcal{T}^{(2)}$) is applied to obtain the transitional domain:

$$\mathbf{Z}^{S\to T} = \mathcal{T}^{(2)}(\check{\mathbf{Z}}^S; \widetilde{\mathbf{S}}^T) = \phi_\delta^{(2)}(\widetilde{\mathbf{S}}^T) \odot \frac{\check{\mathbf{Z}}^S - \mu(\check{\mathbf{Z}}^S)}{\mathrm{std}(\check{\mathbf{Z}}^S)} + \phi_\mu^{(2)}(\widetilde{\mathbf{S}}^T), \quad (7)$$

and $\mathbf{Z}^{T\to S}$ is obtained analogously by swapping (S, T).

Unlike conditional normalization methods such as FiLM [33] and AdaIN [18], our module operates on topology-aware latent representations and derives graph-domain conditioning from $\mathbf{S}$ and graph-level statistics rather than external/style signals. This supports bidirectional latent transition synthesis without explicit intermediate-topology generation. In this way, we progressively

synthesize the transitional domains and construct the following latent adaptation pathway: $\mathcal{R} = \{\mathbf{Z}^S, \mathbf{Z}^{S\to T}, \mathbf{Z}^{T\to S}, \mathbf{Z}^T\}$.

Note that the reconstruction tolerance γ is not directly imposed on the dual transitional domains but is introduced in the disentanglement stage. It regularizes the semantic neighborhood within which valid latent transitions can be synthesized. A smaller γ yields a more faithful but less flexible semantic space, whereas a larger γ provides more transition flexibility but may introduce semantic slack. This trade-off is theoretically analyzed in Sec. 3.4.

3.3 Gradual Adversarial Training

To reduce the discrepancy in the semantic space while preserving domain-invariant knowledge learned by the disentanglement objective, we perform gradual adversarial training along the latent pathway $\mathcal{R}$. Instead of directly aligning the two endpoints $\mathbf{Z}^S$ and $\mathbf{Z}^T$, GAGT decomposes the large gap into several local adversarial alignment steps between adjacent latent states. This design provides smoother adversarial signals and reduces the risk of shortcut matching under large graph domain shifts.

Let $\mathbf{R}_0 = \mathbf{Z}^S$, $\mathbf{R}_1 = \mathbf{Z}^{S\to T}$, $\mathbf{R}_2 = \mathbf{Z}^{T\to S}$, and $\mathbf{R}_3 = \mathbf{Z}^T$. The gradual adversarial training objective is defined as:

$$\mathcal{L}_G = \min_{E_S,E_D} \max_{D_m^c} \sum_{m=0}^{2} \Big[\mathbb{E}_{\mathbf{z}\sim\mathbf{R}_m} \log D_m^c(\mathbf{z}) + \mathbb{E}_{\mathbf{z}\sim\mathbf{R}_{m+1}} \log(1 - D_m^c(\mathbf{z}))\Big], \tag{8}$$

where $\mathbf{z} \sim \mathbf{R}_m$ denotes uniformly sampling node representations from the corresponding latent state, and D_m^c is the local domain discriminator for the m-th adjacent pair. In practice, each local adversarial game is implemented with a gradient reversal layer (GRL) [12], where D_m^c distinguishes two adjacent latent states while the encoders E_S and E_D are optimized to confuse it. Therefore, the pathway is gradually aligned through $\mathbf{Z}^S \leftrightarrow \mathbf{Z}^{S\to T}$, $\mathbf{Z}^{S\to T} \leftrightarrow \mathbf{Z}^{T\to S}$, and $\mathbf{Z}^{T\to S} \leftrightarrow \mathbf{Z}^T$, making each adversarial game easier than one-step source-target alignment.

To prevent adversarial alignment from collapsing task-discriminative structures, we introduce a supervised cross-entropy loss on the labeled source graph and an entropy minimization regularizer on the unlabeled target graph:

$$\mathcal{L}_{CE} = -\frac{1}{|\mathcal{V}^S|} \sum_{i=1}^{|\mathcal{V}^S|} \sum_{c=1}^{C} \mathbf{y}_{i,c}^S \log \hat{\mathbf{y}}_{i,c}^S, \mathcal{L}_{En} = -\frac{1}{|\mathcal{V}^T|} \sum_{i=1}^{|\mathcal{V}^T|} \sum_{c=1}^{C} \hat{\mathbf{y}}_{i,c}^T \log \hat{\mathbf{y}}_{i,c}^T, \tag{9}$$

where $|\mathcal{V}^S|$ and $|\mathcal{V}^T|$ denote the numbers of nodes on the source and target graphs. $\mathbf{y}_{i,c}^S$ is the source label of node i for class c, and $\hat{\mathbf{y}}_{i,c}^S$ and $\hat{\mathbf{y}}_{i,c}^T$ are the predicted class probabilities produced by the classifier f after softmax normalization. The source cross-entropy loss preserves semantic discriminability, while the target entropy loss encourages confident predictions and pushes the decision boundary away from high-density regions in the target space [13].

By integrating all losses, the total objective of GAGT is:

$$\mathcal{L}_{Total} = \eta_1\mathcal{L}_{CDL} + \eta_2(\mathcal{L}_R^f + \mathcal{L}_R^s) + \eta_3\mathcal{L}_G + \eta_4\mathcal{L}_{CE} + \frac{e_c}{e_t}\mathcal{L}_{En}, \tag{10}$$

where $\{\eta_1, \eta_2, \eta_3, \eta_4\}$ are trade-off hyperparameters. e_c and e_t denote the current and total numbers of training epochs, respectively. The factor $\frac{e_c}{e_t}$ linearly ramps up the target entropy regularizer during training.

3.4 Theoretical Analysis

The reconstruction tolerance γ regulates the semantic flexibility of the latent adaptation pathway. A smaller γ preserves stricter semantic faithfulness, whereas a larger γ allows a wider feasible region for bridging domain shifts but may also introduce semantic relaxation. We therefore analyze how γ can be upper-bounded by connecting the γ-induced feasible region with the standard domain adaptation bound [1].

Proposition 1 (Conservative upper bound of γ). *For node v_i, define the γ-feasible semantic region as*

$$\mathcal{F}_\gamma(\mathbf{x}_i, \mathbf{s}_i) = \left\{\mathbf{z} \mid \left\|\mathbf{x}_i - D^e([\mathbf{z}, \mathbf{s}_i])\right\|_2 \le \gamma\right\}. \tag{11}$$

Assume that D^e is locally Lipschitz with respect to $\mathbf{z}$ with constant $L_d > 0$, and let $r_i = \|\mathbf{x}_i - D^e([\mathbf{z}_i, \mathbf{s}_i])\|_2$. If $\gamma \ge r_i$, then any perturbation $\Delta\mathbf{z}$ satisfying

$$\|\Delta\mathbf{z}\|_2 \le \frac{\gamma - r_i}{L_d} \tag{12}$$

remains within $\mathcal{F}_\gamma(\mathbf{x}_i, \mathbf{s}_i)$.

Moreover, under the target-risk bound

$$\epsilon_T(f) \le \epsilon_S(f) + \frac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}(\mathbf{Z}^S, \mathbf{Z}^T; \gamma) + C_\gamma\gamma + \lambda, \tag{13}$$

and a conservative first-order relaxation [1] that enlarging the feasible region can reduce the achievable divergence only at a bounded linear rate, γ should satisfy

$$\gamma \le \bar{\gamma} = \frac{d_0 L_d}{2(1 + C_\gamma L_d)}, \tag{14}$$

where $d_0 = d_{\mathcal{H}\Delta\mathcal{H}}(\mathbf{Z}^S, \mathbf{Z}^T; 0)$ is the initial distribution divergence.

The proof is provided in Appendix F. Proposition 1 establishes a conservative relation between the admissible tolerance and the initial domain discrepancy under the stated assumptions. Importantly, Eq. (14) constrains the scale of γ rather than providing a directly computable tuning rule, since d_0, L_d, and C_γ are generally unavailable in practice. It therefore motivates a monotonic principle: larger initial discrepancies may permit wider semantic tolerance, while the $C_\gamma\gamma$ slack discourages excessive relaxation along the latent pathway.

Because $d_{\mathcal{H}\Delta\mathcal{H}}$, L_d, and C_γ are not directly available, the node-level bound in Proposition 1 cannot be instantiated numerically. We therefore use Maximum Mean Discrepancy (MMD [32]) only as a dataset-level proxy for the relative ordering of initial source-target discrepancies, rather than as an estimator of the theoretical bound or γ itself. This approximation transfers only the monotonic relation suggested by Eq. (14) to practice: each dataset $\mathcal{B}$ receives a conservative tolerance budget according to its worst-case initial discrepancy:

$$D_{\mathcal{B}} = \max_{(S,T)\in\mathcal{B}} d_0(S,T), \quad \gamma_{\max}(\mathcal{B}) = \min\{\kappa D_{\mathcal{B}}, 0.1\}. \tag{15}$$

where κ is a dataset-shared scaling coefficient fixed empirically. The 0.1 cap prevents the reconstruction constraint from becoming overly permissive [53]. Therefore, Eq. (15) should be viewed as a conservative dataset-level heuristic motivated by, but not equivalent to, Proposition 1: it restricts the candidate range of γ, whereas the final value is selected empirically as described in Sec. 4.3.2. The resulting settings are reported in Sec. 4.3.2 and Appendix A.

Table 1: Node classification accuracies on ArnetMiner, Airport Networks, and Social Networks. d_{MMD} denotes the MMD-based proxy of the initial divergence between two graphs. The best result is in bold, and the second-best is underlined.

Domain	ArnetMiner					Airport Networks					Social Networks	
Model	A → C	C → A	C → D	D → A	D → C	U → B	U → E	B → U	E → U	E → B	B1 → B2	B2 → B1
d_{MMD}	0.0360	0.0360	0.0427	0.0425	0.0427	0.9723	0.7180	0.9723	0.7180	0.4603	0.0685	0.0685
UDA-GCN$_{2020}$	79.57$_{\pm 0.44}$	72.37$_{\pm 2.23}$	76.25$_{\pm 0.20}$	56.95$_{\pm 9.28}$	73.59$_{\pm 3.38}$	32.51$_{\pm 5.88}$	34.18$_{\pm 4.73}$	24.50$_{\pm 1.24}$	26.48$_{\pm 2.94}$	28.85$_{\pm 3.90}$	65.27$_{\pm 0.96}$	66.14$_{\pm 0.37}$
ACDNE$_{2020}$	82.20$_{\pm 0.37}$	77.34$_{\pm 0.45}$	75.25$_{\pm 0.59}$	64.52$_{\pm 9.87}$	72.71$_{\pm 7.70}$	26.71$_{\pm 0.26}$	25.26$_{\pm 0.36}$	25.12$_{\pm 0.16}$	25.12$_{\pm 1.68}$	26.71$_{\pm 2.51}$	86.96$_{\pm 0.08}$	**88.31$_{\pm 0.37}$**
AdaGCN$_{2023}$	76.00$_{\pm 0.68}$	69.59$_{\pm 2.73}$	74.55$_{\pm 1.37}$	61.39$_{\pm 3.76}$	72.15$_{\pm 0.18}$	42.29$_{\pm 0.61}$	41.30$_{\pm 0.82}$	46.18$_{\pm 0.26}$	43.41$_{\pm 0.57}$	**56.41$_{\pm 1.60}$**	65.21$_{\pm 3.72}$	60.18$_{\pm 0.37}$
SpecReg$_{2023}$	76.72$_{\pm 5.03}$	76.18$_{\pm 1.30}$	75.85$_{\pm 1.46}$	65.78$_{\pm 4.08}$	66.08$_{\pm 12.69}$	36.94$_{\pm 6.30}$	39.54$_{\pm 1.78}$	27.54$_{\pm 1.65}$	26.72$_{\pm 2.97}$	36.64$_{\pm 8.54}$	72.17$_{\pm 4.65}$	70.71$_{\pm 4.15}$
A2GNN$_{2024}$	81.90$_{\pm 0.14}$	77.36$_{\pm 0.75}$	74.87$_{\pm 0.92}$	73.01$_{\pm 2.03}$	77.68$_{\pm 0.85}$	38.93$_{\pm 10.43}$	49.82$_{\pm 0.66}$	38.05$_{\pm 2.17}$	40.50$_{\pm 0.51}$	48.24$_{\pm 5.49}$	70.19$_{\pm 3.80}$	69.76$_{\pm 2.53}$
ADAlign$_{2026}$	81.29$_{\pm 0.75}$	76.76$_{\pm 0.60}$	**77.82$_{\pm 0.29}$**	71.55$_{\pm 1.16}$	77.68$_{\pm 1.58}$	47.94$_{\pm 2.83}$	48.97$_{\pm 1.84}$	29.93$_{\pm 4.92}$	25.95$_{\pm 1.78}$	24.89$_{\pm 0.92}$	25.79$_{\pm 1.63}$	27.25$_{\pm 1.55}$
GGDA$_{2025}$	79.47$_{\pm 0.14}$	70.66$_{\pm 0.41}$	73.12$_{\pm 0.25}$	66.19$_{\pm 0.58}$	77.45$_{\pm 0.52}$	33.59$_{\pm 4.58}$	39.25$_{\pm 2.96}$	41.38$_{\pm 1.36}$	38.74$_{\pm 0.28}$	45.19$_{\pm 6.30}$	66.18$_{\pm 0.48}$	63.49$_{\pm 0.78}$
GADGET$_{2025}$	62.96$_{\pm 0.21}$	59.75$_{\pm 0.20}$	65.29$_{\pm 2.18}$	55.02$_{\pm 0.16}$	60.30$_{\pm 0.06}$	54.73$_{\pm 0.61}$	48.09$_{\pm 2.09}$	43.36$_{\pm 3.06}$	40.05$_{\pm 0.89}$	50.07$_{\pm 1.03}$	67.96$_{\pm 0.04}$	68.17$_{\pm 0.14}$
DiffGDA$_{2026}$	74.99$_{\pm 3.17}$	69.20$_{\pm 7.69}$	75.84$_{\pm 1.61}$	54.96$_{\pm 9.62}$	58.08$_{\pm 6.87}$	26.72$_{\pm 0.00}$	25.56$_{\pm 0.00}$	45.03$_{\pm 5.26}$	28.40$_{\pm 3.05}$	26.72$_{\pm 0.00}$	18.74$_{\pm 1.23}$	20.63$_{\pm 3.73}$
GRADE$_{2023}$	76.97$_{\pm 0.96}$	70.57$_{\pm 1.56}$	74.10$_{\pm 1.67}$	65.37$_{\pm 0.90}$	72.95$_{\pm 0.80}$	56.88$_{\pm 0.91}$	48.52$_{\pm 2.67}$	**49.86$_{\pm 0.60}$**	48.10$_{\pm 0.45}$	38.93$_{\pm 17.58}$	71.25$_{\pm 0.38}$	69.55$_{\pm 0.29}$
StruRW$_{2023}$	54.35$_{\pm 3.16}$	54.35$_{\pm 1.84}$	57.10$_{\pm 1.88}$	49.31$_{\pm 7.96}$	53.66$_{\pm 7.60}$	24.88$_{\pm 1.03}$	24.86$_{\pm 0.40}$	25.60$_{\pm 0.42}$	25.12$_{\pm 0.50}$	27.17$_{\pm 2.13}$	86.11$_{\pm 0.77}$	86.08$_{\pm 0.76}$
PairAlign$_{2024}$	45.24$_{\pm 6.47}$	45.65$_{\pm 3.88}$	48.37$_{\pm 4.92}$	48.21$_{\pm 2.42}$	50.82$_{\pm 3.48}$	42.74$_{\pm 1.60}$	44.81$_{\pm 2.03}$	46.57$_{\pm 0.19}$	44.95$_{\pm 0.28}$	44.42$_{\pm 1.48}$	64.01$_{\pm 1.27}$	62.58$_{\pm 2.76}$
TDSS$_{2025}$	81.06$_{\pm 0.17}$	78.07$_{\pm 0.38}$	74.17$_{\pm 0.25}$	74.85$_{\pm 0.21}$	78.43$_{\pm 0.36}$	44.27$_{\pm 6.74}$	41.80$_{\pm 8.23}$	40.35$_{\pm 7.91}$	40.49$_{\pm 8.12}$	30.53$_{\pm 7.63}$	70.97$_{\pm 0.39}$	72.97$_{\pm 0.95}$
HGDA$_{2025}$	17.11$_{\pm 8.20}$	16.07$_{\pm 5.52}$	14.00$_{\pm 7.06}$	15.99$_{\pm 5.49}$	17.23$_{\pm 8.21}$	25.65$_{\pm 3.19}$	25.42$_{\pm 1.66}$	26.95$_{\pm 1.68}$	25.67$_{\pm 1.39}$	30.00$_{\pm 6.65}$	53.15$_{\pm 6.54}$	41.47$_{\pm 3.37}$
ASN$_{2021}$	81.04$_{\pm 0.12}$	74.84$_{\pm 0.50}$	76.53$_{\pm 0.65}$	70.02$_{\pm 1.32}$	78.62$_{\pm 0.29}$	30.53$_{\pm 4.97}$	25.63$_{\pm 7.48}$	29.68$_{\pm 4.37}$	28.89$_{\pm 4.79}$	33.28$_{\pm 10.35}$	86.78$_{\pm 0.11}$	86.32$_{\pm 0.48}$
GraphDA$_{2024}$	55.84$_{\pm 10.46}$	62.55$_{\pm 1.89}$	61.77$_{\pm 3.58}$	53.74$_{\pm 8.86}$	64.73$_{\pm 2.71}$	47.63$_{\pm 2.62}$	41.35$_{\pm 1.69}$	44.45$_{\pm 1.61}$	47.63$_{\pm 2.62}$	42.59$_{\pm 1.12}$	68.73$_{\pm 2.86}$	71.63$_{\pm 2.21}$
DGSDA$_{2025}$	76.73$_{\pm 0.92}$	74.10$_{\pm 0.43}$	74.75$_{\pm 0.45}$	71.88$_{\pm 0.53}$	80.11$_{\pm 0.50}$	37.39$_{\pm 5.40}$	34.03$_{\pm 3.38}$	49.63$_{\pm 1.17}$	42.16$_{\pm 1.69}$	43.48$_{\pm 5.99}$	79.48$_{\pm 0.59}$	78.17$_{\pm 1.41}$
GAGT	**82.62$_{\pm 0.38}$**	**78.53$_{\pm 0.18}$**	77.73$_{\pm 1.02}$	**75.27$_{\pm 1.27}$**	**80.63$_{\pm 0.80}$**	**58.02$_{\pm 0.89}$**	**50.38$_{\pm 1.35}$**	49.50$_{\pm 0.19}$	**50.67$_{\pm 1.27}$**	55.85$_{\pm 0.81}$	**87.98$_{\pm 0.69}$**	87.56$_{\pm 1.09}$

4 Experimental Evaluation

4.1 Experimental Setup

4.1.1 ***Datasets.*** We evaluate GAGT on five graph domain adaptation datasets: **ArnetMiner** [39] includes ACMv9 (A), Citationv1 (C), and DBLPv7 (D), **Airport Networks** [25] contains airport graphs for the USA (U), Brazil (B), and Europe (E), **Social Networks** [35] consists of Blog1 (B1) and Blog2 (B2), **MAG** [29] uses country-level academic graphs from France (FR), Japan (JP), and Russia (RU), and **Elliptic** [57] is a temporal bitcoin transaction graph containing a sequence of graph snapshots (S10, M10, E10).

4.1.2 ***Baselines.*** For the cross-network node classification task, we use the following baselines: **(i)** GDA methods based on adversarial alignment: UDA-GCN [44], ACDNE [35], AdaGCN [9], SpecReg [49], A2GNN [25], ADAlign [6]. **(ii)** Gradual graph domain adaptation: GGDA [23], GADGET [51], DiffGDA [7]. **(iii)** GDA methods based on topology-aware alignment: GRADE [43], StruRW [28], PairAlign [29], TDSS [8], HGDA [10]. **(iv)** GDA methods based on representation disentanglement: ASN [54], GraphDA [2], DGSDA [48]. All these GDA baselines follow the settings in their public codes and the benchmark in [37].

4.1.3 ***Implementation Details.*** In each task, one graph serves as the labeled source domain, while another serves as the unlabeled target domain. Target labels are only used for evaluation. For the domain-invariant and domain-specific encoders, we adopt the 2-layer GATConv and GCNConv provided by PyTorch Geometric [11]. The hidden and output dimensions are set to 512 and 64, respectively, and the dropout rate is set to 0.5. For the feature decoder (D^e), local domain discriminators (D_m^c), and node classifier (f), we adopt 2-layer MLPs. For all tasks, we train our model for 400 epochs using the Adam optimizer [20] with a learning rate of 1×10^{-3} and a weight decay of 5×10^{-4}. We present the average accuracies and standard deviations on the target graph over five runs with different random seeds. We also provide the pseudo-code, time complexity analysis, balancing hyperparameter settings, and their sensitivity analysis in Appendices A and E.

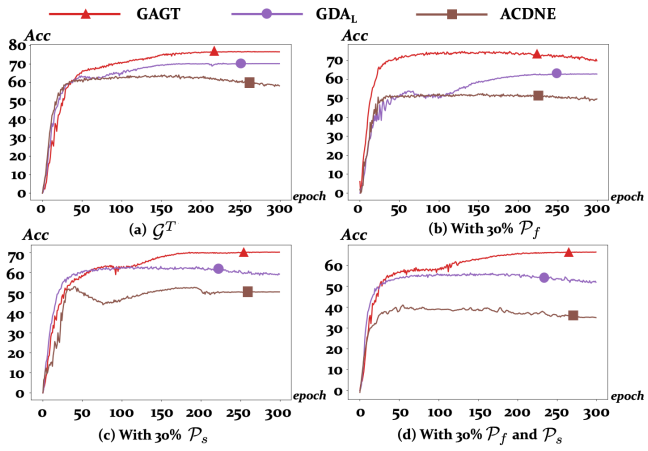


Figure 3: Performance comparison of the three methods under different shifts in the task of ACMv9 → Citationv1. $\mathcal{P}_f$ and $\mathcal{P}_s$ denote feature and structure perturbations.

4.2 Performance Comparison

4.2.1 ***Classification Accuracy.*** The results of the node classification tasks are listed in Tables 1 and 2. GAGT outperforms nearly all baselines. Specifically, we make the following observations: GAGT disentangles domain-invariant knowledge from domain-specific knowledge through contrastive learning with tolerance-bounded reconstruction, mitigating negative transfer caused by domain-correlated cues. By using dual transitional domains synthesized

Table 2: Node classification accuracies on MAG and Elliptic. The best result is in bold and the second-best is underlined.

Domain	MAG						Elliptic					
Model	FR → JP	FR → RU	JP → FR	JP → RU	RU → FR	RU → JP	S10 → M10	S10 → E10	M10 → S10	M10 → E10	E10 → S10	E10 → M10
d_{MMD}	0.1058	0.1254	0.1058	0.1360	0.1254	0.1360	0.2267	0.4166	0.2267	0.2392	0.4166	0.2392
UDA-GCN$_{2020}$	55.20$_{\pm1.19}$	77.76$_{\pm0.16}$	52.67$_{\pm0.47}$	81.28$_{\pm0.23}$	52.69$_{\pm0.63}$	59.73$_{\pm0.92}$	81.80$_{\pm0.10}$	74.31$_{\pm2.41}$	81.08$_{\pm0.97}$	70.47$_{\pm12.44}$	69.70$_{\pm22.09}$	77.80$_{\pm2.08}$
ACDNE$_{2020}$	51.54$_{\pm3.87}$	61.10$_{\pm1.61}$	51.09$_{\pm1.51}$	68.48$_{\pm1.24}$	52.14$_{\pm1.48}$	59.51$_{\pm0.95}$	82.02$_{\pm0.11}$	81.97$_{\pm0.28}$	83.94$_{\pm0.49}$	84.29$_{\pm0.37}$	80.06$_{\pm0.99}$	84.43$_{\pm0.61}$
AdaGCN$_{2023}$	56.00$_{\pm2.98}$	76.71$_{\pm1.23}$	53.00$_{\pm0.89}$	80.23$_{\pm1.07}$	**52.83$_{\pm0.25}$**	59.86$_{\pm1.26}$	82.23$_{\pm0.18}$	77.39$_{\pm0.20}$	85.42$_{\pm0.15}$	81.55$_{\pm0.14}$	83.91$_{\pm0.20}$	81.92$_{\pm0.12}$
SpecReg$_{2023}$	50.93$_{\pm2.39}$	62.66$_{\pm2.02}$	51.26$_{\pm1.12}$	70.80$_{\pm2.59}$	52.49$_{\pm1.28}$	58.58$_{\pm1.45}$	82.01$_{\pm0.60}$	76.81$_{\pm0.19}$	82.43$_{\pm0.89}$	78.04$_{\pm2.74}$	77.47$_{\pm1.88}$	77.82$_{\pm1.17}$
A2GNN$_{2024}$	48.91$_{\pm1.61}$	44.90$_{\pm0.69}$	46.30$_{\pm2.29}$	55.13$_{\pm0.82}$	51.94$_{\pm1.40}$	55.41$_{\pm2.88}$	76.59$_{\pm0.61}$	71.77$_{\pm0.95}$	79.21$_{\pm0.39}$	73.81$_{\pm1.18}$	79.21$_{\pm0.43}$	77.85$_{\pm0.17}$
ADAlign$_{2026}$	54.12$_{\pm0.59}$	75.16$_{\pm0.39}$	52.08$_{\pm0.18}$	85.16$_{\pm1.08}$	51.44$_{\pm2.80}$	60.11$_{\pm0.65}$	78.83$_{\pm0.33}$	73.44$_{\pm0.40}$	81.05$_{\pm0.24}$	75.87$_{\pm0.39}$	80.24$_{\pm0.35}$	79.05$_{\pm0.42}$
GGDA$_{2025}$	50.13$_{\pm0.97}$	60.87$_{\pm1.78}$	49.82$_{\pm0.36}$	71.72$_{\pm1.19}$	52.50$_{\pm0.08}$	59.81$_{\pm0.60}$	80.64$_{\pm0.45}$	74.58$_{\pm0.80}$	76.82$_{\pm1.70}$	75.30$_{\pm0.42}$	76.13$_{\pm1.74}$	76.72$_{\pm0.56}$
GADGET$_{2025}$	54.14$_{\pm0.59}$	65.84$_{\pm2.48}$	52.28$_{\pm1.54}$	81.59$_{\pm0.16}$	52.45$_{\pm1.24}$	59.73$_{\pm1.82}$	82.16$_{\pm0.30}$	79.28$_{\pm1.14}$	82.14$_{\pm0.42}$	80.26$_{\pm0.48}$	69.05$_{\pm3.46}$	79.94$_{\pm0.28}$
DiffGDA$_{2026}$	54.79$_{\pm0.78}$	76.52$_{\pm1.41}$	52.19$_{\pm0.39}$	79.96$_{\pm0.44}$	52.44$_{\pm0.00}$	60.11$_{\pm0.00}$	78.87$_{\pm0.36}$	74.22$_{\pm0.25}$	81.01$_{\pm0.15}$	76.39$_{\pm0.24}$	80.62$_{\pm0.29}$	79.05$_{\pm0.29}$
GRADE$_{2023}$	56.50$_{\pm0.32}$	77.37$_{\pm0.86}$	52.53$_{\pm0.96}$	80.37$_{\pm0.30}$	52.38$_{\pm3.58}$	61.14$_{\pm0.90}$	82.07$_{\pm0.09}$	77.18$_{\pm0.34}$	81.66$_{\pm0.46}$	80.28$_{\pm0.34}$	80.54$_{\pm2.03}$	80.36$_{\pm0.50}$
StruRW$_{2023}$	51.57$_{\pm6.74}$	75.37$_{\pm4.38}$	46.82$_{\pm7.23}$	85.15$_{\pm0.00}$	52.43$_{\pm0.04}$	59.59$_{\pm1.04}$	85.57$_{\pm0.15}$	81.41$_{\pm0.16}$	88.16$_{\pm0.15}$	**85.21$_{\pm0.19}$**	80.66$_{\pm2.15}$	84.58$_{\pm0.40}$
PairAlign$_{2024}$	56.07$_{\pm0.48}$	50.86$_{\pm3.46}$	49.38$_{\pm1.45}$	13.49$_{\pm3.17}$	47.99$_{\pm2.77}$	57.38$_{\pm1.74}$	77.61$_{\pm1.11}$	51.60$_{\pm7.79}$	81.09$_{\pm3.33}$	77.80$_{\pm1.61}$	35.12$_{\pm2.94}$	54.11$_{\pm5.73}$
TDSS$_{2025}$	55.82$_{\pm0.07}$	79.43$_{\pm0.37}$	52.17$_{\pm0.04}$	83.99$_{\pm0.12}$	52.43$_{\pm0.01}$	60.11$_{\pm0.29}$	82.82$_{\pm0.10}$	82.89$_{\pm0.07}$	81.55$_{\pm0.10}$	81.55$_{\pm0.11}$	77.43$_{\pm0.22}$	77.21$_{\pm0.17}$
HGDA$_{2025}$	18.61$_{\pm0.42}$	14.99$_{\pm1.35}$	20.87$_{\pm1.40}$	11.48$_{\pm1.93}$	21.35$_{\pm0.74}$	19.43$_{\pm0.88}$	82.07$_{\pm2.96}$	76.65$_{\pm0.21}$	79.04$_{\pm0.00}$	78.00$_{\pm0.73}$	80.05$_{\pm2.99}$	79.19$_{\pm4.51}$
ASN$_{2021}$	54.48$_{\pm2.09}$	75.58$_{\pm0.87}$	51.50$_{\pm1.39}$	78.44$_{\pm1.15}$	52.59$_{\pm0.76}$	60.26$_{\pm1.06}$	75.56$_{\pm0.00}$	76.02$_{\pm0.00}$	78.54$_{\pm0.00}$	76.02$_{\pm0.00}$	78.54$_{\pm0.00}$	75.56$_{\pm0.00}$
GraphDA$_{2024}$	OOM	OOM	OOM	OOM	OOM	OOM	73.79$_{\pm1.39}$	72.19$_{\pm2.91}$	80.57$_{\pm0.79}$	71.90$_{\pm1.98}$	72.32$_{\pm3.35}$	67.23$_{\pm4.64}$
DGSDA$_{2025}$	22.74$_{\pm0.79}$	13.78$_{\pm1.47}$	22.93$_{\pm0.68}$	16.31$_{\pm1.05}$	24.40$_{\pm0.56}$	22.16$_{\pm0.78}$	83.42$_{\pm0.12}$	78.69$_{\pm0.07}$	83.65$_{\pm0.06}$	80.69$_{\pm0.33}$	81.92$_{\pm0.08}$	79.51$_{\pm0.17}$
GAGT	**57.74$_{\pm0.30}$**	**80.77$_{\pm1.98}$**	**53.99$_{\pm0.60}$**	**86.22$_{\pm3.81}$**	52.78$_{\pm2.67}$	**61.36$_{\pm1.24}$**	**85.76$_{\pm0.34}$**	**83.02$_{\pm0.29}$**	**89.24$_{\pm0.14}$**	84.86$_{\pm0.31}$	**84.94$_{\pm0.14}$**	**85.53$_{\pm0.67}$**

within the tolerance-regularized latent space, GAGT replaces fragile direct adversarial alignment and topology-altering methods, enabling smoother transitions that result in more stable knowledge transfer across domains. Additionally, GAGT preserves semantic discriminability through source supervision and target entropy regularization. In contrast, other baselines, such as those relying on pseudo-labels or topology-preserving transformations, often fail under large shifts, leading to negative transfer, instability, or structural distortions, which ultimately hamper domain adaptation.

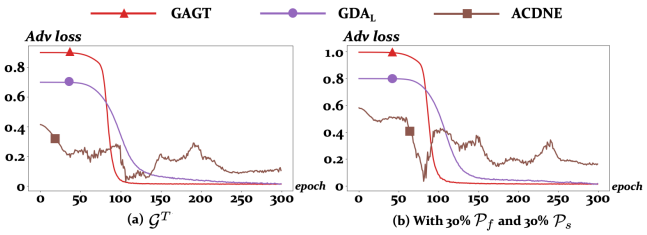


Figure 4: Stability comparison of the three methods under a large shift in the task of ACMv9 → Citationv1.

4.2.2 ***Robustness.*** As shown in Figures 3 and 4, we present the learning curves (accuracy and adversarial loss vs. epoch) of GAGT, alongside two other approaches: **ACDNE** [35], which performs one-step adversarial alignment in a shared embedding space, and **GDA$_L$**, which facilitates gradual adversarial alignment through intermediate graphs generated by GADGET [51]. When the Citationv1 target graph ($\mathcal{G}^T$) is subjected to feature perturbation (adding standard Gaussian noise with varying intensities) and structure perturbation (randomly deleting edges in varying proportions), GAGT maintains smoother trajectories with reduced oscillation and faster convergence as the perturbation strength increases. This indicates that GAGT achieves more stable knowledge transfer through its dual transitional domains, ensuring reliable domain adaptation even under large shifts in both graph topology and feature distributions.

4.3 Analysis

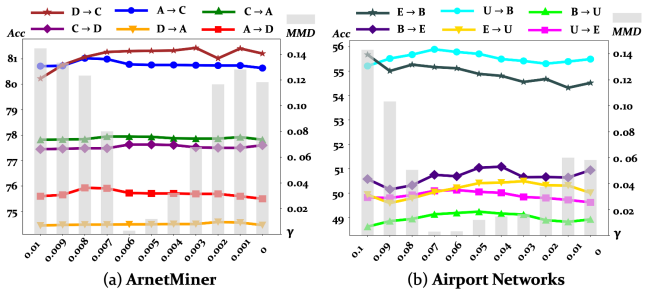
4.3.1 ***Ablation Study.*** To evaluate the effectiveness of each component in our method, GAGT, we conduct an ablation study by testing various combinations of the components: the reconstruction tolerance γ, the contrastive disentanglement loss $\mathcal{L}_{CDL}$, and the dual transitional domains $\mathbf{Z}^{S\to T}/\mathbf{Z}^{T\to S}$. The symbol ✗ indicates removal, while ✓ denotes inclusion. When γ is removed, we set $\gamma = 0$; when $\mathcal{L}_{CDL}$ is removed, we use a single encoder without representation disentanglement; when the transitional domains are removed, adversarial alignment is performed without synthesized intermediate domains. Table 3 shows that all three components contribute to the final performance. The dual transitional domains improve over direct alignment, confirming the benefit of pathway-based adaptation. The contrastive disentanglement loss provides consistent gains by separating transferable semantics from domain-related variations. The reconstruction tolerance further improves performance by balancing semantic sufficiency and transition flexibility. The best results are achieved when all components are jointly used, indicating that semantic disentanglement and gradual pathway alignment are complementary. We also report the GCN-GCN and GAT-GAT variants in Appendix C, showing that the gains are not due to the stronger backbone. Additional comparisons in Appendix D show that tolerance-bounded reconstruction outperforms other common choices, confirming that semantic sufficiency is more suitable than exact recovery for stable latent transition.

Analysis of transition pathway. We further analyze the proposed transition pathway from three aspects, with detailed results reported in Appendix B. First, as shown in Table 5, the single-transition variants already improve over direct no-transition alignment, while the dual-transition design achieves the best overall trade-off; adding more transitional domains brings no consistent

Table 3: Performance of different component combinations in GAGT. ↑ indicates improvement over the first-row baseline.

Components			Tasks		
γ	$\mathcal{L}_{CDL}$	$\mathbf{Z}^{S\to T}/\mathbf{Z}^{T\to S}$	A → C	U → B	S10 → M10
✗	✗	✗	$62.58_{\pm 2.04}$	$39.84_{\pm 2.23}$	$75.55_{\pm 0.41}$
✗	✗	✓	$66.95_{\pm 2.15}$(↑ 4.37)	$42.30_{\pm 1.62}$(↑ 2.46)	$76.29_{\pm 1.28}$(↑ 0.74)
✗	✓	✗	$70.12_{\pm 1.65}$(↑ 7.54)	$45.05_{\pm 1.58}$(↑ 5.21)	$78.67_{\pm 0.99}$(↑ 3.12)
✓	✓	✗	$73.80_{\pm 2.60}$(↑ 11.22)	$49.20_{\pm 0.63}$(↑ 9.36)	$81.07_{\pm 1.08}$(↑ 5.52)
✗	✓	✓	$79.40_{\pm 0.50}$(↑ 16.82)	$55.60_{\pm 1.85}$(↑ 15.76)	$83.48_{\pm 2.15}$(↑ 7.93)
✓	✓	✓	$82.62_{\pm 0.38}$(↑ 20.04)	$58.02_{\pm 0.89}$(↑ 18.18)	$85.76_{\pm 0.34}$(↑ 10.21)

gains and may introduce semantic drift or additional approximation difficulty. Second, in the A → C task, the generated transitional domains form valid latent bridges: the endpoint MMD distance between $\mathbf{Z}^S$ and $\mathbf{Z}^T$ is $0.62_{\pm 0.15}$, whereas the adjacent distances along the pathway are smaller, i.e., $0.42_{\pm 0.01}$, $0.14_{\pm 0.01}$, and $0.30_{\pm 0.07}$. Moreover, the classification confidence for $\mathbf{Z}^{S\to T}$ and $\mathbf{Z}^{T\to S}$ remains high, reaching $91.03_{\pm 3.14}$ and $86.02_{\pm 7.70}$, respectively, suggesting that the transitional states remain compatible with the learned decision function in the A → C task. Finally, probing diagnosis shows that $\mathbf{Z}$ is more label-predictive, whereas $\mathbf{S}$ is more domain-predictive, supporting their roles as the semantic carrier and domain-conditioned signal for synthesis.

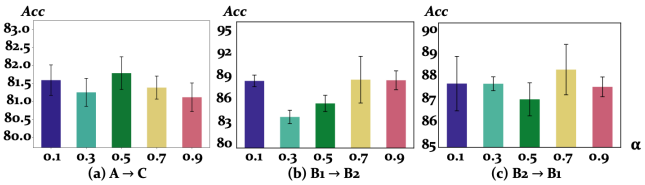


(a) ArnetMiner (b) Airport Networks

Figure 5: Domain adaptation performance with varying γ.

4.3.2 *Hyperparameter Study.*

Study on Reconstruction Tolerance γ. Consistent with Sec. 3.4, we separate the theoretical constraint from practical parameter selection: Eq. (14) provides only a monotonic tolerance principle, while Eq. (15) defines an empirical dataset-level search range. We fix $\kappa = 0.2$ across datasets and allocate only a small fraction of the observed discrepancy to this conservative range. This yields a smaller practical range for ArnetMiner, approximately $\gamma \le 0.01$, and a larger capped range for Airport Networks, $\gamma \le 0.1$, due to its much larger initial discrepancy. Since tasks within the same dataset share the same feature space and label space, we use a dataset-level γ rather than tuning task-specific values. As shown in Figure 5, increasing γ initially reduces post-alignment MMD, but excessively large γ yields little additional alignment gain and may degrade target accuracy. Therefore, we select the smallest γ that reaches the post-alignment MMD plateau. For ArnetMiner and Airport Networks, we set γ to 0.006 and 0.07 based on the MMD performance of D → C and U → B. Others refer to Appendix A.



(a) A → C (b) B1 → B2 (c) B2 → B1

Figure 6: Accuracy of GAGT with varying proportions (α) of negative samples in different tasks.

Study on Negative Samples Number K. We study the effect of the number of negative samples K, where $K = \alpha \cdot |\mathcal{V}|$ and $\alpha \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$. The results show that no single α consistently performs best across all transfer tasks. Since a larger K directly increases the cost of contrastive disentanglement, we choose $\alpha = 0.1$ as a practical trade-off between performance and efficiency. This setting still achieves competitive classification performance while keeping the contrastive learning overhead limited.

4.3.3 *Efficiency Study.*

Figure 7 reports the component-wise runtime and module-wise parameter counts of GAGT in A → C. The knowledge transition module requires only 0.003s per epoch and 0.44MB parameters. Most parameters come from the encoders and reconstruction decoders, while the runtime is mainly dominated by backpropagation, classification, and contrastive learning. Both adversarial training and knowledge transition introduce negligible runtime overhead, showing that the proposed latent transition pathway is not the main computational bottleneck. More detailed runtime and baseline comparisons are provided in Appendix A.

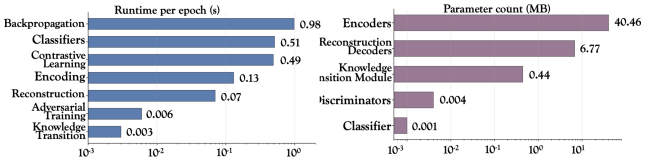


Figure 7: The component-wise runtime and module-wise parameter counts of GAGT.

5 Conclusion

In this paper, we propose GAGT, which enables smooth and stable knowledge transfer between source and target domains, particularly under large domain shifts. By leveraging dual transitional domains and contrastive disentanglement with tolerance-bounded reconstruction, GAGT decomposes the domain gap into smaller and more manageable transitions. The reconstruction tolerance regulates the flexibility of the transitions, balancing semantic faithfulness and adaptation robustness. Gradual adversarial alignment along this pathway mitigates adversarial instability and negative transfer. Extensive experiments and diagnostic analyses show that GAGT achieves strong and stable performance.

A Algorithm and Computational Overhead

Algorithm 1 summarizes the training procedure of GAGT. Given a labeled source graph $\mathcal{G}^S = (\mathcal{V}^S, \mathcal{E}^S, \mathbf{X}^S, \mathbf{A}^S, \mathbf{Y}^S)$ and an unlabeled

target graph $\mathcal{G}^T = (\mathcal{V}^T, \mathcal{E}^T, \mathbf{X}^T, \mathbf{A}^T)$, where $\mathbf{X}^S \in \mathbb{R}^{N_1 \times F}$ and $\mathbf{X}^T \in \mathbb{R}^{N_2 \times F}$, the training complexity is bounded by $O((q+4)Md+2Nd^2+KNd+NdF)$. Here, $N = N_1 + N_2$ and $M = M_1 + M_2$ are the total numbers of nodes and edges, d is the latent dimensionality, K is the number of negative samples, and q is the edge sampling ratio for structure reconstruction. We set γ to 0.01, 0.02, and 0.05 for Social Networks, MAG, and Elliptic, respectively.

Algorithm 1: The training procedure of GAGT.

Input: Source graph $\mathcal{G}^S = (\mathcal{V}^S, \mathcal{E}^S, \mathbf{X}^S, \mathbf{A}^S, \mathbf{Y}^S)$, target graph $\mathcal{G}^T = (\mathcal{V}^T, \mathcal{E}^T, \mathbf{X}^T, \mathbf{A}^T)$, number of negative samples K, number of training epochs E
Output: The trained node classifier f^*.

```
for e ← 1 to E do
    Obtain domain-invariant representations Z^S, Z^T from E_S and
      domain-specific representations S^S, S^T from E_D ;
    Shuffle domain-specific representations K times and generate
      negative reconstructed views ;
    Perform contrastive disentanglement by Eq. (4) ;
    Generate transitional domains by Eqs. (5), (6), and (7) ;
    Obtain source and target logits from the classifier f ;
    Optimize Eq. (10) to update all trainable modules ;
return f*
```

As Table 4 shows, we further compare the per-epoch runtime with representative baselines. GAGT is slower than several lightweight baselines, but remains faster than GADGET and A2GNN under the same setting.

Table 4: Runtime comparison with representative baselines.

Model	GRADE	ASN	ACDNE	AdaGCN	A2GNN	GADGET	GAGT
Time/s	0.155	1.113	1.140	0.416	4.183	2.351	2.1973

B Analysis of the Transition Pathway

We analyze the proposed transitional domains from three aspects: the number and direction of transitions, the quality of generated transitional states, and the functional separation between Z and S.

Number and direction of transitional domains. We first study how the number and direction of transitional domains affect performance. As shown in Table 5, using a single transition already improves over the no-transition variant, showing the benefit of pathway-based alignment. The dual-transition design achieves the best overall trade-off, while adding more transitional domains brings no consistent gains and may introduce semantic drift or additional approximation difficulty.

Diagnosis of Latent Transition Pathway. We further diagnose whether the constructed latent pathway is semantically meaningful and whether the two representation branches play distinct roles. As shown in Figure 8, we report the MMD distances averaged over five independent runs, with the standard deviations shown as subscripts. The endpoint MMD distance between $\mathbf{Z}^S$ and $\mathbf{Z}^T$ is $0.62_{\pm 0.15}$, while the adjacent distances along the pathway are

Table 5: Performance of the different numbers and directions of transitional domains.

Setting	A→C	U→B	B1→B2
No-transition	$62.58_{\pm 2.04}$	$39.84_{\pm 2.23}$	$48.20_{\pm 0.37}$
Only $\mathbf{Z}^{S\to T}$	$76.28_{\pm 0.56}$	$50.27_{\pm 1.56}$	$72.99_{\pm 7.31}$
Only $\mathbf{Z}^{T\to S}$	$75.62_{\pm 1.18}$	$51.02_{\pm 1.29}$	$74.48_{\pm 7.83}$
$\mathbf{Z}^{S\to T}$ and $\mathbf{Z}^{T\to S}$	$\mathbf{82.62_{\pm 0.38}}$	$58.02_{\pm 0.89}$	$\mathbf{87.98_{\pm 0.69}}$
4 transitional domains	$81.71_{\pm 0.61}$	$\mathbf{58.51_{\pm 1.56}}$	$86.90_{\pm 1.73}$
6 transitional domains	$81.77_{\pm 0.74}$	$58.07_{\pm 2.94}$	$85.08_{\pm 1.67}$
8 transitional domains	$79.08_{\pm 0.75}$	$57.77_{\pm 1.86}$	$84.71_{\pm 1.40}$

consistently smaller: $0.42_{\pm 0.01}$ from $\mathbf{Z}^S$ to $\mathbf{Z}^{S\to T}$, $0.14_{\pm 0.01}$ between $\mathbf{Z}^{S\to T}$ and $\mathbf{Z}^{T\to S}$, and $0.30_{\pm 0.07}$ from $\mathbf{Z}^{T\to S}$ to $\mathbf{Z}^T$. Meanwhile, the classification confidence for the transitional domains remains high, reaching $91.03_{\pm 3.14}$ for $\mathbf{Z}^{S\to T}$ and $86.02_{\pm 7.70}$ for $\mathbf{Z}^{T\to S}$. These results indicate that the transitional domains are not arbitrary latent points but semantically meaningful bridge states that decompose the source-target discrepancy into smaller local shifts.

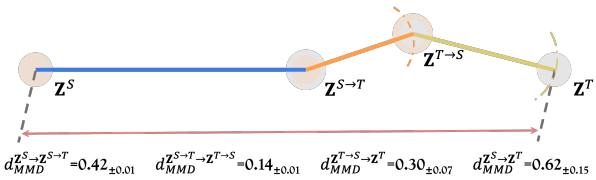


Figure 8: MMD distances along the latent transition pathway.

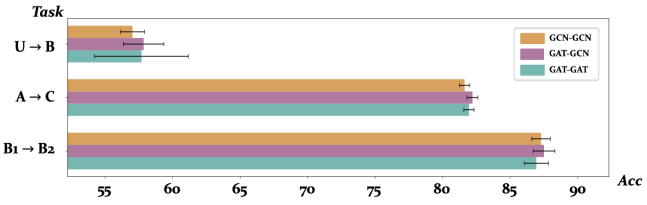


Figure 9: Backbone effect analysis on three transfer tasks.

To examine the representation roles, we freeze the encoders and train linear probes on Z and S, with target labels used only for diagnosis. The label probe performs much better on Z than on S, i.e., $\mathbf{95.58_{\pm 0.49}}$ versus $21.29_{\pm 0.98}$, showing that Z retains stronger task-related information. In contrast, the domain probe performs much better on S than on Z, i.e., $\mathbf{95.87_{\pm 2.43}}$ versus $55.50_{\pm 1.50}$, indicating that S captures more domain-related information. These results support using Z for task information and S for domain conditioning in transition construction.

C Effect of Backbone

To examine whether the improvement comes from a stronger backbone, we compare GCN-GCN, GAT-GCN, and GAT-GAT configurations, where the two terms denote the backbones of the domain-invariant and domain-specific branches. As shown in Figure 9, the

three configurations perform comparably, and strengthening the domain-specific branch does not yield consistent gains. This supports our asymmetric design and shows that the improvement mainly comes from contrastive disentanglement and the latent transition pathway rather than backbone strength.

D Effect of Reconstruction and Disentanglement Constraints

We further evaluate different constraints for reconstruction and representation disentanglement, including MSE, L1, cosine reconstruction, and orthogonality constraints. Unlike these alternatives, the tolerance-bounded reconstruction stops penalizing reconstruction errors once semantic sufficiency is reached, thereby avoiding over-constraining the domain-invariant branch and reducing domain-specific leakage. In Table 6, it achieves the best performance on all three tasks, demonstrating a better balance among semantic preservation, disentanglement, and adaptation flexibility.

Table 6: Comparison of different constraint designs.

Constraint	A→C	U→B	B1→B2
MSE	$79.56_{\pm 0.20}$	$51.04_{\pm 3.43}$	$81.34_{\pm 1.79}$
L1	$78.49_{\pm 0.56}$	$49.51_{\pm 3.43}$	$83.65_{\pm 2.54}$
Cosine	$78.18_{\pm 0.71}$	$50.02_{\pm 3.43}$	$85.21_{\pm 3.99}$
Orthogonal constraint	$77.28_{\pm 2.96}$	$50.53_{\pm 2.55}$	$86.14_{\pm 0.64}$
Tolerance-bounded	$82.62_{\pm 0.38}$	$58.02_{\pm 0.89}$	$87.98_{\pm 0.69}$

E Effect of Balancing Weights

For the sensitivity analysis of η_1, we set $\eta_2 = 1$, $\eta_3 = 1$, $\eta_4 = 1$. This one-at-a-time sensitivity analysis also applies to the sensitivity analysis of the other three hyperparameters to choose the values that achieve consistently high accuracy across representative transfer tasks rather than tuning them separately for each task. As shown in Figure 10, the proposed GAGT can achieve much better performance when $\eta_1 = 0.001$, $\eta_2 = 0.5$, $\eta_3 = 0.5$, $\eta_4 = 5$.

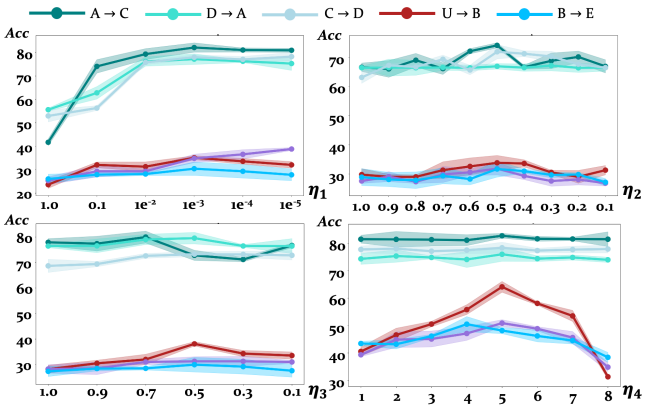


Figure 10: The influences of $\eta_1, \eta_2, \eta_3, \eta_4$.

F Proof of Proposition 1

We prove Proposition 1 in two steps.

Feasible perturbation radius. Recall that

$$r_i = \left\|\mathbf{x}_i - D^e([\mathbf{z}_i, \mathbf{s}_i])\right\|_2.$$

For any perturbation $\Delta\mathbf{z}$, by the triangle inequality and the local Lipschitz condition of D^e,

$$\begin{aligned}&\left\|\mathbf{x}_i - D^e([\mathbf{z}_i + \Delta\mathbf{z}, \mathbf{s}_i])\right\|_2 \\ &\le \left\|\mathbf{x}_i - D^e([\mathbf{z}_i, \mathbf{s}_i])\right\|_2 + \left\|D^e([\mathbf{z}_i, \mathbf{s}_i]) - D^e([\mathbf{z}_i + \Delta\mathbf{z}, \mathbf{s}_i])\right\|_2 \\ &\le r_i + L_d\|\Delta\mathbf{z}\|_2.\end{aligned}$$

Thus, if $\gamma \ge r_i$ and

$$\|\Delta\mathbf{z}\|_2 \le \frac{\gamma - r_i}{L_d},$$

then

$$\left\|\mathbf{x}_i - D^e([\mathbf{z}_i + \Delta\mathbf{z}, \mathbf{s}_i])\right\|_2 \le \gamma,$$

which implies

$$\mathbf{z}_i + \Delta\mathbf{z} \in \mathcal{F}_\gamma(\mathbf{x}_i, \mathbf{s}_i).$$

This proves Eq. (12). If $\gamma < r_i$, the radius is non-positive and no positive feasible perturbation radius can be guaranteed.

Conservative upper bound on γ. Let

$$d(\gamma) = d_{\mathcal{H}\Delta\mathcal{H}}(\mathbf{Z}^S, \mathbf{Z}^T; \gamma).$$

The γ-dependent part of Eq. (13) is

$$B(\gamma) = \frac{1}{2}d(\gamma) + C_\gamma\gamma, \qquad B(0) = \frac{1}{2}d_0.$$

Increasing γ enlarges the feasible semantic region and may reduce the achievable source-target divergence, but this reduction is conservatively bounded. Under a first-order relaxation,

$$d(\gamma) \ge \left[d_0 - \frac{2}{L_d}\gamma\right]_+.$$

Therefore, the divergence term in $B(\gamma)$ can decrease by at most γ/L_d, while the semantic slack term increases by $C_\gamma\gamma$. To keep the tolerance-induced relaxation within the initial discrepancy scale, we require

$$\frac{\gamma}{L_d} + C_\gamma\gamma \le \frac{d_0}{2}.$$

Solving this inequality gives

$$\gamma \le \frac{d_0}{2\left(C_\gamma + \frac{1}{L_d}\right)} = \frac{d_0 L_d}{2(1 + C_\gamma L_d)} = \bar{\gamma}.$$

Thus, $\gamma \le \bar{\gamma}$ gives a conservative upper bound for controlling the feasible semantic region. This justifies the monotonic relation between the initial discrepancy d_0 and the feasible range of γ. □

G Acknowledgments

We thank the anonymous reviewers for their valuable and constructive comments. This work was supported partially by the National Natural Science Foundation of China under Grants 62576253 and 62176184, AI for Science Program, Shanghai Municipal Commission of Economy and Informatization under Grant 2025-GLZ-RGZN-BTBX-02016, the Fundamental Research Funds for the Central Universities, and the Science and Technology Development Fund, Macao SAR No. 0006/2024/RIA1.

H GenAI Usage Disclosure

We used generative AI tools (e.g., ChatGPT) for language editing and assistance in checking clarity and presentation during manuscript revision. These tools were not used to generate experimental results, data, or scientific conclusions. All manuscript content was reviewed and verified by the authors.

References

[1] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. 2010. A theory of learning from different domains. *Machine learning* 79, 1 (2010), 151–175.

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

[3] Jiangxia Cao, Xixun Lin, Xin Cong, Jing Ya, Tingwen Liu, and Bin Wang. 2022. Disencdr: Learning disentangled representations for cross-domain recommendation. In *Proceedings of the 45th International ACM SIGIR conference on research and development in information retrieval*. 267–277.

[4] Chunchun Chen, Xing Wei, Jiayi Yang, Chenrun Wang, Yiwei Fu, Yuxing Zhang, Xin Sun, Rui Fan, and Wei Ye. 2026. Edge self-adversarial augmentation enhances graph contrastive learning against neighborhood inconsistency. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 40. 20005–20013.

[5] Chunchun Chen, Xing Wei, Yuxing Zhang, and Wei Ye. 2026. Adversarial Augmentation With Maximum Discrepancy for Graph Contrastive Learning. *IEEE Transactions on Neural Networks and Learning Systems* (2026).

[6] Wei Chen, Xingyu Guo, Shuang Li, Zhao Zhang, Yan Zhong, Fuzhen Zhuang, et al. 2026. Learning Adaptive Distribution Alignment with Neural Characteristic Function for Graph Domain Adaptation. In *The Fourteenth International Conference on Learning Representations*.

[7] Wei Chen, Xingyu Guo, Shuang Li, Yan Zhong, Zhao Zhang, Fuzhen Zhuang, Hongrui Liu, Libang Zhang, Guo Ye, and HUIMEI HE. 2026. Learning Structure-Semantic Evolution Trajectories for Graph Domain Adaptation. In *The Fourteenth International Conference on Learning Representations*.

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

[9] Quanyu Dai, Xiao-Ming Wu, Jiaren Xiao, Xiao Shen, and Dan Wang. 2023. Graph transfer learning via adversarial domain adaptation with graph convolution. *IEEE Transactions on Knowledge and Data Engineering* 35, 5 (2023), 4908–4922.

[10] Ruiyi Fang, Bingheng Li, Jingyu Zhao, Ruizhi Pu, Qiuhao Zeng, Gezheng Xu, Charles Ling, and Boyu Wang. 2025. Homophily Enhanced Graph Domain Adaptation. In *International Conference on Machine Learning*. 16006–16028.

[11] Matthias Fey and Jan Eric Lenssen. 2019. Fast graph representation learning with PyTorch Geometric. *arXiv preprint arXiv:1903.02428* (2019).

[12] Yaroslav Ganin and Victor Lempitsky. 2015. Unsupervised domain adaptation by backpropagation. In *International conference on machine learning*. 1180–1189.

[13] Yves Grandvalet and Yoshua Bengio. 2004. Semi-supervised learning by entropy minimization. *Advances in neural information processing systems* 17 (2004).

[14] Xiaobo Guo, Shaoshuai Li, Naicheng Guo, Jiangxia Cao, Xiaolei Liu, Qiongxu Ma, Runsheng Gan, and Yunan Zhao. 2023. Disentangled representations learning for multi-target cross-domain recommendation. *ACM Transactions on Information Systems* 41, 4 (2023), 1–27.

[15] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *Proceedings of the Advances in neural information processing systems*, Vol. 30.

[16] Jiajie Hao, Guoji Shen, Xiaofei Zhang, and Haidong Shao. 2025. An adversarial gradual domain adaptation approach for fault diagnosis via intermediate domain generation. *Nondestructive Testing and Evaluation* (2025), 1–21.

[17] Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. In *Advances in neural information processing systems*, Vol. 33. 6840–6851.

[18] Xun Huang and Serge Belongie. 2017. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE international conference on computer vision*. 1501–1510.

[19] Maximilian Ilse, Jakub M Tomczak, Christos Louizos, and Max Welling. 2020. Diva: Domain invariant variational autoencoders. In *Medical Imaging with Deep Learning*. 322–348.

[20] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

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

[22] Thomas N Kipf and Max Welling. 2016. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308* (2016).

[23] Pui Ieng Lei, Ximing Chen, Yijun Sheng, Yanyan Liu, Jingzhi Guo, and Zhiguo Gong. 2025. Gradual Domain Adaptation for Graph Learning. *arXiv preprint arXiv:2501.17443* (2025).

[24] Jing Liu, Lele Sun, Weizhi Nie, Peiguang Jing, and Yuting Su. 2024. Graph disentangled contrastive learning with personalized transfer for cross-domain recommendation. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 38. 8769–8777.

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

[26] Meihan Liu, Zhen Zhang, Jiachen Tang, Jiajun Bu, Bingsheng He, and Sheng Zhou. 2024. Revisiting, benchmarking and understanding unsupervised graph domain adaptation. In *Proceedings of the Advances in Neural Information Processing Systems*, Vol. 37. 89408–89436.

[27] Shuhan Liu and Kaize Ding. 2024. Beyond generalization: A survey of out-of-distribution adaptation on graphs. *arXiv preprint arXiv:2402.11153* (2024).

[28] Shikun Liu, Tianchun Li, Yongbin Feng, Nhan Tran, Han Zhao, Qiang Qiu, and Pan Li. 2023. Structural re-weighting improves graph domain adaptation. In *Proceedings of the International conference on machine learning*. 21778–21793.

[29] Shikun Liu, Deyu Zou, Han Zhao, and Pan Li. 2024. Pairwise alignment improves graph domain adaptation. *arXiv preprint arXiv:2403.01092* (2024).

[30] Bin Lu, Ze Zhao, Xiaoying Gan, Shiyu Liang, Luoyi Fu, Xinbing Wang, and Chenghu Zhou. 2024. Graph out-of-distribution generalization with controllable data augmentation. *IEEE Transactions on Knowledge and Data Engineering* 36, 11 (2024), 6317–6329.

[31] Xinyu Ma, Xu Chu, Yasha Wang, Yang Lin, Junfeng Zhao, Liantao Ma, and Wenwu Zhu. 2023. Fused gromov-wasserstein graph mixup for graph-level classifications. *NeurIPS* 36 (2023), 15252–15276.

[32] Liwen Ouyang and Aaron Key. 2021. Maximum mean discrepancy for generalization in the presence of distribution and missingness shift. *arXiv preprint arXiv:2111.10344* (2021).

[33] Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. 2018. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 32.

[34] Chengxuan Qian, Shuo Xing, Li Li, Yue Zhao, and Zhengzhong Tu. 2026. DecAlign: Hierarchical Cross-Modal Alignment for Decoupled Multimodal Representation Learning. In *The Fourteenth International Conference on Learning Representations*.

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

[36] Boshen Shi, Yongqing Wang, Fangda Guo, Jiangli Shao, Huawei Shen, and Xueqi Cheng. 2023. Improving graph domain adaptation with network hierarchy. In *Proceedings of the 32nd ACM international conference on information and knowledge management*. 2249–2258.

[37] Boshen Shi, Yongqing Wang, Fangda Guo, Jiangli Shao, Huawei Shen, and Xueqi Cheng. 2023. Opengda: Graph domain adaptation benchmark for cross-network learning. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*. 5396–5400.

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

[39] Jie Tang, Jing Zhang, Limin Yao, Juanzi Li, Li Zhang, and Zhong Su. 2008. Arnetminer: extraction and mining of academic social networks. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. 990–998.

[40] Qin Tian, Chen Zhao, Minglai Shao, Wenjun Wang, Yujie Lin, and Dong Li. 2025. Mldgg: Meta-learning for domain generalization on graphs. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1*. 1361–1372.

[41] Vayer Titouan, Nicolas Courty, Romain Tavenard, and Rémi Flamary. 2019. Optimal transport for structured data with application on graphs. In *International Conference on Machine Learning*. PMLR, 6275–6284.

[42] Zirui Wang, Zihang Dai, Barnabás Póczos, and Jaime Carbonell. 2019. Characterizing and avoiding negative transfer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 11293–11302.

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

[44] Man Wu, Shirui Pan, Chuan Zhou, Xiaojun Chang, and Xingquan Zhu. 2020. Unsupervised domain adaptive graph convolutional networks. In *Proceedings of the web conference 2020*. 1457–1467.

[45] Yuankai Wu, Dingyi Zhuang, Aurelie Labbe, and Lijun Sun. 2021. Inductive graph neural networks for spatiotemporal kriging. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 4478–4485.

[46] Jiayi Yang, Sourav Medya, and Wei Ye. 2024. Incorporating heterophily into graph neural networks for graph classification. In *2024 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, 1544–1551.

[47] Jiayi Yang, Xing Wei, Chunchun Chen, Yi Feng, Wengang Guo, Rui Fan, Xiaofeng Cao, Xin Sun, and Wei Ye. 2026. Fast Mixture of Curvature-Aware Experts for

Diverse and Dynamic Graph Topologies. In *Forty-third International Conference on Machine Learning*.

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

[49] Yuning You, Tianlong Chen, Zhangyang Wang, and Yang Shen. 2023. Graph domain adaptation via theory-grounded spectral regularization. In *Proceedings of the eleventh international conference on learning representations*.

[50] Qian Yu, Sizhe Wang, and Yixin Tao. 2025. Enhancing anti-money laundering detection with self-attention graph neural networks. In *SHS Web of Conferences*, Vol. 213. 01016.

[51] Zhichen Zeng, Ruizhong Qiu, Wenxuan Bao, Tianxin Wei, Xiao Lin, Yuchen Yan, Tarek F Abdelzaher, Jiawei Han, and Hanghang Tong. 2025. Pave Your Own Path: Graph Gradual Domain Adaptation on Fused Gromov-Wasserstein Geodesics. *arXiv preprint arXiv:2505.12709* (2025).

[52] Zhichen Zeng, Ruizhong Qiu, Zhe Xu, Zhining Liu, Yuchen Yan, Tianxin Wei, Lei Ying, Jingrui He, and Hanghang Tong. 2024. Graph mixup on approximate gromov–wasserstein geodesics. In *Proceedings of the Forty-first International Conference on Machine Learning*.

[53] An Zhang, Han Wang, Xiang Wang, and Tat-Seng Chua. 2024. Disentangling Masked Autoencoders for Unsupervised Domain Generalization. In *European Conference on Computer Vision*. 126–151.

[54] Xiaowen Zhang, Yuntao Du, Rongbiao Xie, and Chongjun Wang. 2021. Adversarial separation network for cross-network node classification. In *Proceedings of the 30th ACM international conference on information & knowledge management*. 2618–2626.

[55] Yuxing Zhang, Siyuan Meng, Chunchun Chen, Mengyao Peng, Hongyan Gu, and Xinli Huang. 2024. Linkthief: Combining generalized structure knowledge with node similarity for link stealing attack against gnn. In *Proceedings of the 32nd ACM International Conference on Multimedia*. 4947–4956.

[56] Yizhou Zhang, Guojie Song, Lun Du, Shuwen Yang, and Yilun Jin. 2019. Dane: Domain adaptive network embedding. *arXiv preprint arXiv:1906.00684* (2019).

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

[58] Yanping Zheng, Lu Yi, and Zhewei Wei. 2025. A survey of dynamic graph neural networks. *Frontiers of Computer Science* 19, 6 (2025), 196323.