Stage-Aware Dual Alignment for Covariate Shift in Graph Domain Adaptation

Hongwei Wen
University of Sydney
hongwei.wen@sydney.edu.au

Can Zhang
Beijing University of Technology
zhangcan1019@emails.bjut.edu.cn

Haoyu He
George Washington University
haoyuhe66@gmail.com

Xintao Zhao
University of New South Wales
z5520512@ad.unsw.edu.au

Hanyuan Hang
FUCA AI Labs
hanghy@fuca-china.com

Minglong Lei *
Beijing University of Technology
leiml@bjut.edu.cn

Abstract

The performance of *Graph Domain Adaptation* (*GDA*) is fundamentally limited by *covariate-driven* discrepancies between source and target graphs, captured by *Covariate Shift* (*CS*) in the joint feature–structure space beyond label-related formulations. We show that CS induces bias at two non-interchangeable stages of *Graph Neural Network* (*GNN*) computation: *Feature Shift* (*FS*) distorts representations before message passing, while *Feature-Conditional Structure Shift* (*FCSS*) biases propagation, rendering single-stage alignment intrinsically insufficient. Based on this, we propose *Dual Alignment for Covariate Shift* (*DACS*), a stage-aware GDA framework that aligns discrepancies at their source. DACS mitigates FS via adversarial feature alignment and addresses FCSS through layer-wise reweighting to correct propagation bias, followed by final adversarial alignment for residual mismatch. This design follows directly from the stage-wise structure of CS rather than heuristic combinations of alignment modules. We further show that representation and propagation biases correspond to distinct components of target-domain error that cannot be eliminated in isolation. Experiments on synthetic and real-world benchmarks demonstrate that DACS consistently outperforms prior methods, especially under complex and coupled distribution shifts.

1 Introduction

Graph Domain Adaptation (GDA), also known as *cross-network classification* [20, 23], aims to transfer knowledge from a labeled source graph to an unlabeled or sparsely labeled target graph. It is a fundamental problem in graph learning, with broad applications in social networks [17, 31], biological systems [13], materials science [4], and spatiotemporal forecasting [32, 19], where distribution shifts [33, 26, 27] naturally arise across domains, platforms, and time. A large body of prior work [5, 7] characterizes domain discrepancy through label-related formulations, such as label shift ($P_S(Y) \neq P_T(Y)$) and label-conditional structure shift ($P_S(A \mid Y) \neq P_T(A \mid Y)$) [15, 16]. While effective when reliable target labels or pseudo-labels are available [28], these approaches implicitly assume that labels govern graph connectivity. However, in many real-world graphs, connectivity is

*Corresponding author: Minglong Lei <leiml@bjut.edu.cn>.

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

primarily driven by node attributes rather than labels [12, 9, 8, 6], and domain discrepancy may arise directly from differences in node features and feature-conditioned connectivity patterns, which are not captured by label-centric formulations.

This motivates studying domain shift directly in the observable feature–structure space. Recent work [14] formalizes this setting as *covariate shift* (CS) in graphs, defined as a mismatch in the joint distribution $P(X, A)$. Unlike label-dependent formulations, CS does not rely on target-domain labels and instead captures discrepancies intrinsic to the data generation process. Understanding how such discrepancies propagate through *graph neural networks* (GNNs) is therefore critical for effective adaptation. This challenge stems from the inherently compositional nature of GNNs, which first encode node features into representations and then propagate them through graph structure via message passing. Inspired by this computation pipeline, we show that the joint mismatch $P_S(X, A) \neq P_T(X, A)$ under CS admits a stage-wise decomposition into two distinct components: *feature shift* (FS), $P_S(X) \neq P_T(X)$, and *feature-conditioned structure shift* (FCSS), $P_S(A \mid X) \neq P_T(A \mid X)$. These two shifts enter the GNN computation at distinct stages: FS distorts initial representations before message passing, while FCSS biases the subsequent propagation mechanism.

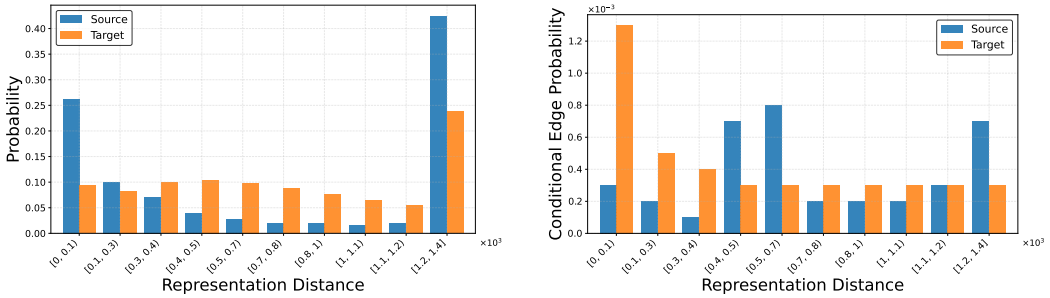
This structural distinction implies that CS induces a stage-wise failure mode in GNNs, where domain discrepancy manifests as a composition of initial representation bias and propagation bias. Crucially, these two sources are inherently non-interchangeable, as they affect different components of the model and interact across layers, so correcting one does not, in general, eliminate the other, rendering single-stage alignment intrinsically insufficient under CS. Despite this structure, most existing GDA methods adopt a single-stage alignment paradigm, typically aligning final-layer representations via adversarial learning [21, 22, 37, 6], feature matching [36, 29, 3], or regularization [34]. While effective for the types of shifts they are designed to address, these approaches entangle distinct sources of discrepancy into a single objective, obscuring their origin and failing to account for domain-dependent message passing, leading to suboptimal adaptation under CS.

In this work, we study GDA from a stage-aware perspective and show that CS induces bias at two non-interchangeable stages of GNN computation, corresponding to the FS/FCSS decomposition above, which gives rise to representation bias and propagation bias as distinct components of target-domain error. Based on this, we propose *Dual Alignment for Covariate Shift* (DACS), a principled stage-aware framework that aligns discrepancies at their source. DACS mitigates FS via adversarial feature alignment and addresses FCSS through layer-wise reweighting to correct propagation bias, followed by final adversarial alignment to remove residual mismatch. Importantly, this design is not heuristic but directly follows from the stage-wise structure of CS. We further provide a theoretical analysis showing that representation bias and propagation bias correspond to distinct components of target-domain error that cannot be eliminated in isolation. Extensive experiments on both controlled synthetic settings and real-world benchmarks validate our analysis and demonstrate that DACS consistently outperforms existing methods, particularly under complex and coupled shifts.

Our contributions are summarized as follows. *(i)* We show that CS in GNNs can be decomposed into two non-interchangeable error sources, FS and FCSS, which act at different computational stages, motivating DACS, a stage-aware dual-alignment framework. *(ii)* We establish a theoretical link between these two error sources and target-domain generalization, showing that FS induces representation bias while FCSS induces propagation bias, and that correcting either alone is insufficient to control the target risk. *(iii)* We conduct extensive experiments demonstrating that DACS consistently outperforms existing methods, especially under complex and coupled distribution shifts.

2 Stage-wise Bias in GNNs under Covariate Shift

Node Classification. Let $G = (V, E)$ denote an undirected graph, where V is the set of nodes with $|V| = n$ and E is the set of edges. The adjacency matrix is denoted by $A \in \mathbb{R}^{n \times n}$. Each node $u \in V$ is associated with a feature vector $X_u \in \mathbb{R}^F$, and all node features form $X \in \mathbb{R}^{n \times F}$. We consider node-level classification, which aims to predict the label vector $Y = (Y_u)_{u \in V}$. For clarity, we focus on the binary setting $Y_u \in \{0, 1\}$, while all results extend to multi-class cases. Let $P(X, A, Y)$ denote the joint distribution over features, structure, and labels, with marginal $P(X, A)$ and conditional distribution $P(Y \mid X, A)$. We define the ground-truth labeling function as $f_P : \mathbb{R}^{n \times F} \times \mathcal{A} \to \{0, 1\}^n$. The classification risk of a predictor f under distribution P is $\mathcal{R}_P(f) := \frac{1}{n}\mathbb{E}_P \big\| f(X, A) - f_P(X, A) \big\|_1$.



(a) FS (b) FCSS

Figure 1: Visualization of CS components between datasets DBLP and ACM. (a) Cross-domain variation in pairwise feature distances illustrates FS. (b) Cross-domain variation in edge probabilities conditioned on feature distance illustrates FCSS.

Covariate Shift in Graph Data. We adopt the notion of *Covariate Shift* (CS) in graph data [14], which characterizes domain discrepancy directly in the observable feature–structure space.

Definition 2.1 (Covariate Shift)**.** *Covariate shift occurs when the joint distribution of node features and graph structure differs between the source and target domains:*

$$P_S(X, A) \neq P_T(X, A). \tag{1}$$

A key advantage of this formulation is that it operates purely on observable covariates, without requiring target-domain labels. Importantly, the joint distribution admits the factorization

$$P(X, A) = P(X)\, P(A \mid X), \tag{2}$$

which separates node attributes from the feature-conditioned graph generation mechanism. This factorization gives rise to two complementary sources of discrepancy:

$$P_S(X) \neq P_T(X), \qquad P_S(A \mid X) \neq P_T(A \mid X). \tag{3}$$

In this work, we interpret this decomposition through the lens of GNN computation, and refer to these two components as *feature shift* (FS) and *feature-conditioned structure shift* (FCSS), respectively. FS captures discrepancies in node attributes, while FCSS captures discrepancies in how attributes induce graph connectivity. Unlike marginal structure shift $P_S(A) \neq P_T(A)$, FCSS explicitly models the conditional structural mechanism governing message passing, and therefore directly influences how information is aggregated in GNNs. To illustrate the presence of FS and FCSS in practice, we consider the real-world citation datasets DBLP and ACM [24, 29]. Figure 1(a) compares the distributions of pairwise feature distances across domains, visualizing FS. Figure 1(b) shows, for each feature-distance interval, the probability that two nodes are connected, visualizing FCSS. These results indicate that even when feature similarity profiles are comparable, the induced connectivity patterns can differ substantially across domains, highlighting the necessity of modeling FCSS.

GNN computation as a stage-wise pipeline. To understand how such discrepancies affect graph neural networks, we explicitly formulate GNN computation as a stage-wise pipeline. A GNN computes node representations via iterative message passing. Let H_l denote the representation at layer l, with propagation rule $H_{l+1} = \phi_l(AH_l)$, where ϕ_l is the l-th layer encoder. The final representation is $H_L = \phi(X, A)$, and predictions are given by $\widehat{Y} = g(H_L)$. To expose where domain discrepancy is introduced, we decouple feature encoding from propagation by defining an explicit encoder f such that $H_1 = f(X)$. This yields a two-stage computation pipeline: initial representation learning ($X \to H_1$) followed by structure-dependent propagation ($H_1 \to H_L$).

Stage-wise bias induced by FS and FCSS. The factorization of $P(X, A)$ aligns naturally with this computational pipeline. Specifically, FS perturbs the distribution of initial representations $H_1 = f(X)$, while FCSS alters the conditional propagation from H_1 to H_L. This establishes a direct correspondence between the statistical decomposition of CS and the computational structure of GNNs. Consequently, CS induces two distinct sources of bias: *initial representation bias* at the encoding stage and *propagation bias* during message passing. These two biases are inherently non-interchangeable, as they affect different stages of computation and interact across layers. Correcting one does not, in general, eliminate the other, rendering single-stage alignment intrinsically insufficient under CS. As we will show in the next section, these biases cannot be resolved in isolation. This observation motivates a stage-aware treatment that explicitly intervenes at their respective stages, forming the foundation of our proposed framework.

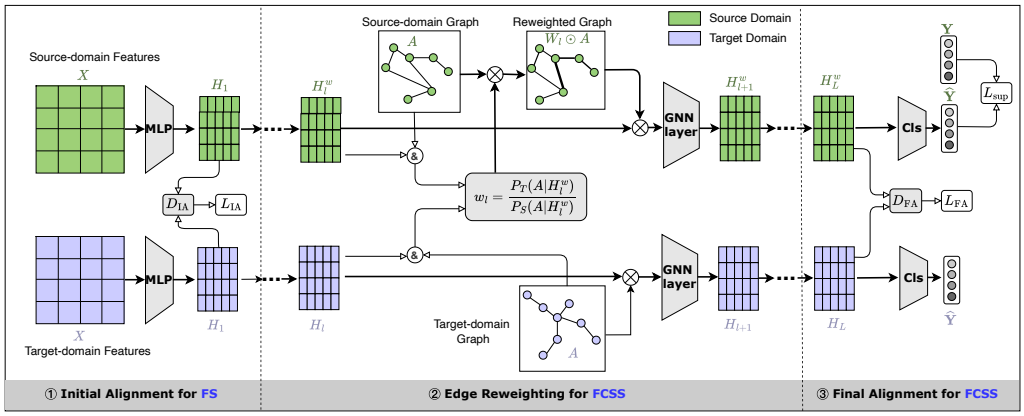


Figure 2: Stage-aware dual-alignment framework for GDA under CS. Green components correspond to the source domain, while blue components correspond to the target domain. **Left:** Initial representation alignment mitigates FS by enforcing domain-invariant marginal distributions through D_{IA}. **Middle:** Layer-wise edge reweighting rescales source edges using feature-conditioned probability ratios to correct propagation bias in expectation. **Right:** Final-layer alignment via D_{FA} suppresses residual higher-order discrepancies. Together, these components realize dual alignment at the level of bias sources.

3 Stage-aware Dual Alignment for Covariate Shift

We propose *Dual Alignment for Covariate Shift* (DACS), a stage-aware GDA framework for CS in GNNs. Building on the stage-wise bias characterization in the previous section, the central idea is to intervene at the computation stages where discrepancies are introduced, rather than aligning only the final representation distribution. Under CS, the final representation entangles discrepancies injected before message passing with those induced by the propagation mechanism. DACS therefore formulates adaptation as *dual alignment*, i.e., jointly controlling two coupled but non-interchangeable sources of bias: *initial representation bias* before message passing and *propagation bias* during message passing.

Motivation from the GNN Computation Chain. Under CS, the source and target domains differ in the joint feature–structure distribution as defined in (1). By the factorization in (2), this mismatch decomposes into the two complementary sources in (3). A GNN computes predictions through the stage-wise pipeline

$$X \xrightarrow{f} H_1 \xrightarrow{\phi(\cdot,A)} H_L \xrightarrow{g} \widehat{Y}, \tag{4}$$

where $H_1 = f(X)$ is the initial representation and $H_L = \phi(H_1, A)$ is the final representation after message passing. FS induces an initial discrepancy $P_S(H_1) \neq P_T(H_1)$, while FCSS induces a conditional discrepancy $P_S(H_L \mid H_1) \neq P_T(H_L \mid H_1)$. These discrepancies act on different stages of the computation and therefore cannot, in general, be resolved by a single alignment objective.

Moreover, the distribution of H_L satisfies $P(H_L) = \int P(H_L \mid H_1)P(H_1)\, dH_1$, which entangles representation bias and propagation bias. As a result, aligning $P(H_L)$ alone neither identifies nor eliminates the source of target-domain error. This yields the stage-wise requirement

$$P_S(H_1) \approx P_T(H_1), \qquad P_S(H_L \mid H_1) \approx P_T(H_L \mid H_1). \tag{5}$$

Dual Alignment. DACS directly instantiates (5). The term *dual alignment* refers to the simultaneous control of the two non-interchangeable discrepancy sources: initial alignment of H_1 for addressing FS and conditional propagation alignment of H_L given H_1 for addressing FCSS. While initial alignment can be achieved at the representation level via adversarial training, propagation bias arises from discrepancies in the feature-conditioned aggregation mechanism, i.e., $P_S(H_L \mid H_1) \neq P_T(H_L \mid H_1)$, which cannot be corrected by representation alignment alone. To address this, we explicitly modify the propagation operator via reweighted message passing. Let $H_1^w := H_1$ and define recursively

$$H_{l+1}^w = \phi_l\big((W_l \odot A)\, H_l^w\big), \qquad l = 1, \ldots, L-1, \tag{6}$$

where W_l is a feature-conditioned edge reweighting matrix. Intuitively, W_l acts as an importance weighting term that compensates for discrepancies in feature-conditioned edge distributions. In

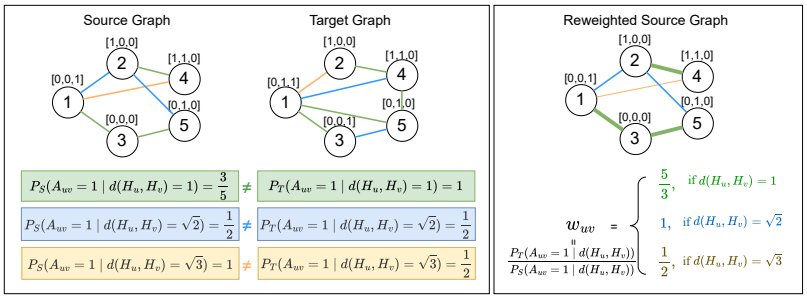


Figure 3: A numerical illustration of FCSS and edge reweighting.

practice, W_l is instantiated by an empirical estimator $\widehat{W}_l$, which will be introduced later in (8). The resulting representation after reweighted propagation is denoted by H_L^w.

As illustrated in Figure 2, DACS incorporates three key components to realize the two alignment objectives. ① **Initial representation alignment** enforces $P_S(H_1) \approx P_T(H_1)$ and constitutes the representation-side component of dual alignment. ② **Layer-wise reweighting** modifies the propagation operator to approximate $\mathbb{E}_{P_S}(H_L^w \mid H_1) \approx \mathbb{E}_{P_T}(H_L \mid H_1)$, thereby correcting the systematic component of propagation bias. ③ **Final-layer reweighted alignment** enforces $P_S(H_L^w) \approx P_T(H_L)$ to suppress residual higher-order mismatch after reweighting. The latter two components jointly instantiate the propagation-side alignment. Together, DACS provides a structured approximation to (5), separating representation alignment from conditional propagation correction.

① Addressing FS via Initial Representation Alignment. FS induces a discrepancy in the distribution of H_1. We mitigate it via domain-adversarial learning [10]. A discriminator $D_{\mathrm{IA}} : \mathbb{R}^d \to [0,1]$ is trained to distinguish source and target embeddings, while the encoder f is optimized to fool it:

$$L_{\mathrm{IA}} := \max_{D_{\mathrm{IA}}} \Big[\mathbb{E}_{P_S}[\log D_{\mathrm{IA}}(f(X_u))] + \mathbb{E}_{P_T}[\log(1 - D_{\mathrm{IA}}(f(X_v)))] \Big]. \tag{7}$$

This enforces $P_S(H_1) \approx P_T(H_1)$ and directly reduces representation bias.

② Correcting FCSS via Layer-wise Representation Reweighting. Even when H_1 is aligned, FCSS induces bias in the propagation process. We correct this by instantiating the abstract reweighting operator W_l with an empirical estimator $\widehat{W}_l$. Let $H_1^{\widehat{W}} := H_1$ and define

$$H_{l+1}^{\widehat{W}} = \phi_l\big((\widehat{W}_l \odot A) H_l^{\widehat{W}}\big), \qquad l = 1, \ldots, L-1, \tag{8}$$

where $\widehat{W}_l = (\widehat{w}_l^{uv})$ with $\widehat{w}_l^{uv} = \widehat{P}_T(A_{uv} \mid H_l^{\widehat{W}}, H_1) / \widehat{P}_S(A_{uv} \mid H_l^{\widehat{W}}, H_1)$. DACS depends only on the induced likelihood ratio, allowing flexible instantiation of $\widehat{P}_S$ and $\widehat{P}_T$; implementation details are deferred to Appendix A, and an illustration is shown in Figure 3. This reweighting modifies the propagation operator such that $\mathbb{E}_{P_S}(H_L^{\widehat{W}} \mid H_1) \approx \mathbb{E}_{P_T}(H_L \mid H_1)$, thereby correcting the dominant systematic component of propagation bias.

③ Final-layer Reweighted Representation Alignment. Layer-wise reweighting aligns first-order propagation statistics by matching $\mathbb{E}(H_L \mid H_1)$, but does not eliminate higher-order discrepancies in $P(H_L \mid H_1)$. To suppress these residual mismatches, we apply adversarial alignment at the final representation layer. A discriminator $D_{\mathrm{FA}} : \mathbb{R}^d \to [0,1]$ is trained to distinguish reweighted source embeddings $H_L^{\widehat{W}}$ from target embeddings H_L, while the feature extractor is optimized to fool it:

$$L_{\mathrm{FA}} := \max_{D_{\mathrm{FA}}} \Big[\mathbb{E}_{P_S}[\log D_{\mathrm{FA}}(H_L^{\widehat{W},u})] + \mathbb{E}_{P_T}[\log(1 - D_{\mathrm{FA}}(H_L^u))] \Big], \tag{9}$$

where $H_L^{\widehat{W},u}$ is defined in (8). This enforces $P_S(H_L^{\widehat{W}}) \approx P_T(H_L)$, complementing reweighting by removing higher-order discrepancies. Together, reweighting (RW) and final-layer alignment (FA) constitute the propagation-side component of dual alignment.

Source Supervision. To preserve discriminative power on the source graph, we train the classifier using the standard cross-entropy loss:

$$L_{\mathrm{sup}} = \mathbb{E}_{P_S}\big[L_{\mathrm{CE}}(g(H_L^{\widehat{W},u}), Y_u)\big]. \tag{10}$$

The Full Objective. We integrate source supervision with the two alignment objectives:

$$\min_{f,\phi,g} \quad L_{\text{sup}} + \gamma_{\text{IA}}\, L_{\text{IA}} + \gamma_{\text{FA}}\, L_{\text{FA}}. \tag{11}$$

At inference time, predictions on the target graph are obtained using its native structure:

$$\widehat{Y}_T = g(\phi(f(X_T), A_T)).$$

4 Theoretical Analysis

This section provides a theoretical foundation for the proposed stage-aware dual-alignment framework. Building on the stage-wise bias formulation in Section 3, we show that under CS, target-domain error in GNNs decomposes into two distinct and non-interchangeable components that arise at different stages of the computation pipeline. This analysis not only explains the limitations of single-stage alignment methods, but also establishes dual alignment as a necessary consequence of the problem structure, rather than a heuristic design choice.

Theorem 4.1 (Target Risk Decomposition)**.** *Let $H_1 = f(X)$ denote the initial representation and $H_L = \phi(H_1, A)$ denote the final representation after message passing. Under CS, the target-domain classification risk satisfies*

$$\begin{aligned}\mathcal{R}_T(g) \le \mathcal{R}_S(g) &+ \sqrt{2}\cdot D_{\text{KL}}^{1/2}\big(P_S(H_1), P_T(H_1)\big) \\ &+ \sqrt{2}\cdot \mathbb{E}_{H_1\sim P_T}\, D_{\text{KL}}^{1/2}\big(P_S(H_L \mid H_1), P_T(H_L \mid H_1)\big),\end{aligned} \tag{12}$$

where $D_{\text{KL}}(P_S, P_T)$ is the KL divergence between the distributions P_S and P_T.

The bound reveals that the target-domain error decomposes into three components, each corresponding to a distinct stage of the GNN pipeline and directly aligning with a specific module in DACS.

(i) Source-domain error. The first term $\mathcal{R}_S(g)$ reflects intrinsic task difficulty. This term is controlled by the source supervision objective L_{sup} in (10), which anchors the learned representations to the downstream prediction task and prevents degenerate domain-invariant solutions.

(ii) Initial representation bias. The second term measures the discrepancy between $P_S(H_1)$ and $P_T(H_1)$, induced by FS. This bias is injected before message passing and perturbs the input representation space. It is explicitly reduced by the initial representation alignment module (7), which enforces $P_S(H_1) \approx P_T(H_1)$ so that both domains share a common input distribution for propagation.

(iii) Propagation bias. The third term measures the discrepancy between $P_S(H_L \mid H_1)$ and $P_T(H_L \mid H_1)$, induced by FCSS. This bias is injected during message passing and alters the aggregation mechanism. It is addressed in a structurally decomposed manner: the layer-wise reweighting module (8) aligns the dominant first-order propagation behavior via $\mathbb{E}_{P_S}(H_L^{\widehat{W}} \mid H_1) \approx \mathbb{E}_{P_T}(H_L \mid H_1)$, while the final-layer alignment module (9) further reduces residual higher-order discrepancies by enforcing $P_S(H_L^{\widehat{W}}) \approx P_T(H_L)$.

These components do not merely correspond to the terms in Theorem 4.1; rather, they constitute a minimal and necessary set of operations that jointly control all sources of target-domain error.

Crucially, representation and propagation biases arise at non-interchangeable stages. Controlling either one alone is insufficient for generalization under CS, thereby necessitating dual alignment.

Theorem 4.2 (Non-interchangeability of Alignment Stages)**.** *There exist source and target graph-domain distributions under CS such that:*

(i) $P_S(H_1) = P_T(H_1)$ *but* $P_S(H_L \mid H_1) \neq P_T(H_L \mid H_1)$,

(ii) $P_S(H_L \mid H_1) = P_T(H_L \mid H_1)$ *but* $P_S(H_1) \neq P_T(H_1)$,

(iii) $P_S(H_L) = P_T(H_L)$ *but both discrepancies persist,*

and in each case a source-optimal classifier can incur maximal target error.

Theorem 4.2 formalizes that alignment objectives defined at a single stage are fundamentally insufficient. In particular, the final-layer discrepancy entangles the two discrepancy sources: $P(H_L) = \int P(H_L \mid H_1)\, P(H_1)\, dH_1$, and therefore, final-layer alignment alone cannot identify or correct their origins. This establishes that any method that does not simultaneously control

both $P(H_1)$ and $P(H_L \mid H_1)$ may incur arbitrarily large target error, thereby necessitating the dual-alignment structure of DACS.

The propagation bias identified in Theorem 4.1 arises from the discrepancy in the conditional distribution $P(H_L \mid H_1)$. Directly aligning this conditional distribution is intractable in practice, as it depends on the combinatorial structure of the adjacency matrix. To better understand the origin of this discrepancy, the following result characterizes when propagation bias vanishes.

Proposition 4.3 (Conditional Propagation Invariance)**.** *If $P_S(A \mid H_1) = P_T(A \mid H_1)$, then $P_S(H_L \mid H_1) = P_T(H_L \mid H_1)$.*

This result identifies the origin of propagation bias as a mismatch in the feature-conditioned adjacency mechanism, and implies that aligning $P(A \mid H_1)$ is sufficient to eliminate conditional propagation discrepancy. However, directly aligning $P(A \mid H_1)$ is intractable in practice. To analyze its effect, we consider an oracle importance reweighting scheme: $H^w_{l+1} = \phi_l(w_l A H^w_l)$, $l = 1, \ldots, L-1$, where $w_l = P_T(A \mid H^w_l, H_1)/P_S(A \mid H^w_l, H_1)$. This oracle reweighting characterizes the ideal correction that would align conditional propagation behavior. In practice, DACS replaces w_l with an empirical estimator $\widehat{W}_l$, as defined in (8).

Theorem 4.4 (Reweighted Conditional Expectation Alignment)**.** *If the conditional expectations of the ℓ-th layer representations are matched for any encoder, i.e., $\mathbb{E}_{P_T}(H_l \mid H_1) = \mathbb{E}_{P_S}(H^w_l \mid H_1)$, then $\mathbb{E}_{P_T}(AH_l \mid H_1) = \mathbb{E}_{P_S}(w_l A H^w_l \mid H_1)$.*

By induction and $H^w_1 = H_1$ in the source domain, Theorem 4.4 yields $\mathbb{E}_{P_T}(H_L \mid H_1) = \mathbb{E}_{P_S}(H^w_L \mid H_1)$ under mean aggregation if the conditional expectations of the l-th layer representations are matched, aligning the dominant first-order component of propagation behavior. This provides a direct theoretical justification for the layer-wise reweighting module (8) in DACS. However, expectation alignment alone does not guarantee full conditional distribution alignment, leaving residual higher-order discrepancies in the final representations. The following theorem shows that reducing residual representation discrepancy also mitigates the corresponding joint representation–label discrepancy.

Theorem 4.5 (Representation-to-Joint Alignment)**.** *Let the conditional label discrepancy between the source and target domains be $\Delta_Y := \sup_h d_{\mathrm{TV}}\big(P_S(Y \mid H^w_L = h), P_T(Y \mid H_L = h)\big)$. Assume the prediction loss satisfies $0 \le \mathcal{L}(g(h), y) \le 1$ and let $\mathcal{F}_H$ be a function class over the representation space containing the source conditional expected loss $\mathcal{L}_S(h) := \mathbb{E}_{Y \sim P_S(Y \mid h)}\mathcal{L}(g(h), Y)$. Then*

$$\left|\mathbb{E}_{P_S(H^w_L, Y)}\mathcal{L}(g(H^w_L), Y) - \mathbb{E}_{P_T(H_L, Y)}\mathcal{L}(g(H_L), Y)\right| \le d_{\mathcal{F}_H}\big(P_S(H^w_L), P_T(H_L)\big) + 2\Delta_Y,$$

where $d_{\mathcal{F}_H}(P, Q) := \sup_{f \in \mathcal{F}_H} |\mathbb{E}_P[f] - \mathbb{E}_Q[f]|$ is an IPM over the representation space.

Theorem 4.5 formalizes the role of the final-layer alignment module (9). While layer-wise reweighting mainly corrects the dominant propagation bias, final-layer adversarial alignment reduces the remaining representation mismatch. The theorem further shows that when the learned representations preserve consistent label information across domains, reducing the representation discrepancy also mitigates the induced representation–label discrepancy, thereby improving target-domain generalization.

5 Experiments

Our experiments are designed to validate the design of DACS. Section 4 shows that CS induces two non-interchangeable sources of target-domain error: initial representation bias before message passing and propagation bias during message passing. Accordingly, DACS performs dual alignment by controlling both sources at their corresponding stages. The experiments therefore evaluate three aspects: whether FS and FCSS induce distinct failure modes, whether stage-specific correction improves over single-stage alignment, and whether jointly addressing both sources remains effective on real-world GDA benchmarks where multiple shifts are entangled.

Baselines. We compare DACS with two groups of representative GDA methods. *(i)* Feature-aggregation-based methods, including UDA-GCN [29], ASN [35], AdaGCN [3], CWGCN [25], JHGDA [22], GAA [6], and Adalign [2], which mainly reduce domain discrepancy through representation-level alignment, adversarial learning, or aggregation-based adaptation. *(ii)* Structure-aware methods, including StruRW [15] and PairAlign [16], which explicitly model structural discrepancy, typically under label-related assumptions. These baselines allow us to examine whether

existing representation-level or label-related adaptation strategies are sufficient under CS, or whether stage-aware dual alignment is necessary. Implementation details are provided in Appendix C.3.

5.1 Synthetic Experiments

The synthetic experiments are constructed to isolate the two discrepancy terms identified in Theorem 4.1. Unlike real-world graphs, where multiple shifts are entangled, the synthetic setting allows us to control whether the target domain contains *(i)* only FS, *(ii)* only FCSS, or *(iii)* both FS and FCSS. These regimes correspond respectively to initial representation bias, propagation bias, and their coupled effect.

In the source domain, node features are sampled from a uniform mixture of three Gaussian components. Node labels are obtained via k-means clustering on features, ensuring that labels are determined entirely by the geometry of X. Edges are generated using a *Stochastic Block Model* (SBM) [1] conditioned on these labels, where two nodes sharing the same label are connected with probability p, and with probability q otherwise. This construction preserves a feature-determined label mechanism while allowing the feature-conditioned edge distribution to be controlled through (p, q), thereby enabling precise manipulation of FS and FCSS.

FS-only. To induce FS without FCSS, we apply orthogonal rotations to target-domain features while keeping SBM parameters fixed. This changes $P(X)$ while preserving $P(A \mid X)$. Rotation angles of 0°, 30°, and 60° correspond to no FS, mild FS, and severe FS, respectively.

FCSS-only. To induce FCSS without FS, we modify the SBM parameters in the target domain while keeping feature distributions unchanged. This changes $P(A \mid X)$ while preserving $P(X)$. Specifically, we set $(p, q) = (0.02, 0.002)$ in the source domain, and $(0.02, 0.002)$, $(0.02, 0.010)$, and $(0.015, 0.016)$ in the target domain to represent no FCSS, mild FCSS, and severe FCSS.

FS + FCSS. We further combine feature rotation and SBM parameter modification to create settings where representation distortion and propagation bias coexist. Details of synthetic data generation are provided in Appendix C.1.

Table 1: Performance on synthetic datasets.

FS Level	Mild FS	Severe FS	No FS		Mild FS		Severe FS	
FCSS Level	No FCSS		Mild FCSS	Severe FCSS	Mild FCSS	Severe FCSS	Mild FS	Severe FS
UDAGCN	86.46±1.65	80.34±1.90	86.72±1.62	86.71±0.93	85.53±0.29	79.91±1.95	76.23±1.17	72.09±0.86
ASN	93.29±0.13	86.69±1.10	96.28±0.44	93.45±0.22	91.17±2.13	89.42±1.64	84.54±0.69	80.77±1.52
AdaGCN	86.71±1.62	81.43±1.07	86.73±1.65	86.69±1.62	86.60±1.64	86.44±1.02	80.72±1.64	78.84±3.23
CWGCN	93.24±0.26	90.08±1.15	96.87±0.62	93.46±1.30	87.01±1.28	86.95±1.24	<u>85.14±1.59</u>	79.83±2.84
JHGDA	80.20±2.65	78.38±2.54	81.57±2.37	79.94±2.64	79.91±2.51	78.38±3.15	74.15±2.23	73.65±2.47
GAA	92.53±1.57	90.69±1.48	89.07±1.47	86.59±1.88	85.11±1.82	83.63±2.02	81.50±2.56	79.93±1.43
StruRW	85.29±1.60	82.45±2.14	86.07±1.67	84.81±1.78	84.47±1.74	82.58±1.32	81.64±2.04	<u>80.78±2.38</u>
PairAlign	96.10±0.22	79.43±1.69	<u>98.93±0.09</u>	<u>97.91±0.13</u>	<u>95.27±0.24</u>	<u>94.93±0.27</u>	75.12±0.57	72.45±1.48
Adalign	<u>99.21±0.48</u>	<u>92.86±0.79</u>	90.65±1.16	53.73±3.29	85.43±1.68	52.20±0.75	76.99±5.09	35.19±0.41
DACS	**99.81±0.15**	**94.93±0.26**	**100±0.00**	**99.82±0.12**	**99.23±0.14**	**98.74±0.19**	**93.55±0.78**	**93.18±0.89**

The best and second-best performances are marked as **bold** and <u>underline</u>, respectively.

Table 1 reports performance under all controlled shift combinations. The results reveal a clear non-interchangeability pattern. When only FS is present, representation-level adaptation methods remain competitive; however, their performance decreases once FCSS is introduced, indicating that representation alignment alone cannot correct propagation-induced bias. Conversely, structure-aware methods such as PairAlign perform strongly under FCSS-only settings, but degrade sharply when FS and FCSS co-occur, showing that structure correction alone cannot compensate for feature-induced representation mismatch. These complementary failures are consistent with Theorem 4.2: controlling one source of discrepancy does not, in general, control the other. DACS consistently achieves the best performance across all synthetic settings. These results validate the central theoretical insight: dual alignment is necessary because FS and FCSS induce distinct error sources, and effective adaptation requires jointly controlling initial representation bias and propagation bias.

5.2 Real-world Experiments

Real-world benchmarks test whether the stage-wise behavior observed in the controlled setting also appears in practical GDA tasks. We consider citation networks (`DBLP/ACM` [24, 30] and `ArXiv` [11]), where the task is to predict paper topics, and `Airport` traffic graphs [18], where node features are one-hot encodings of node degrees following [22]. These datasets naturally contain covariate-driven

shifts arising from temporal evolution, geographic variation, and platform differences. Dataset details are provided in Appendix C.2.

Table 2: Performance on DBLP/ACM and Airport datasets.

Models	DBLP/ACM		Airport					
	A-D	D-A	U-E	E-U	B-E	E-B	B-U	U-B
UDAGCN	68.86±1.08	63.91±1.16	48.87±1.45	43.41±0.52	50.77±0.70	47.62±1.13	49.78±0.55	61.22±1.25
ASN	72.70±0.38	71.62±0.78	46.45±0.34	46.25±2.65	49.62±1.04	59.03±1.56	49.86±2.37	51.91±0.89
AdaGCN	68.59±1.12	66.75±1.59	43.64±0.82	45.57±0.41	48.52±0.63	56.17±1.22	50.30±1.45	47.24±1.20
CWGCN	76.74±0.11	72.75±0.45	47.03±0.85	45.74±0.95	48.77±0.52	63.96±0.89	49.82±1.46	62.29±0.78
JHGDA	75.58±1.65	73.22±0.68	50.75±3.01	52.27±6.25	56.64±1.14	**73.13±1.37**	50.20±0.54	69.27±3.32
GAA	78.37±7.43	74.76±1.57	55.96±3.69	53.33±1.24	56.64±0.75	68.30±2.08	53.22±1.25	69.81±4.92
StruRW	70.19±2.10	66.57±0.42	53.77±0.98	49.67±2.88	56.06±1.31	65.65±0.15	52.19±2.01	62.84±0.11
PairAlign	75.24±1.06	74.77±1.57	55.39±0.94	54.28±2.07	55.72±0.97	52.90±1.48	52.78±1.62	67.86±0.50
Adalign	78.49±7.70	73.47±1.08	48.62±2.17	49.38±0.33	53.08±0.89	60.15±0.57	50.81±0.91	70.99±5.29
DACS	**86.21±0.24**	**79.14±0.31**	**56.58±0.36**	**55.72±0.08**	**57.39±0.15**	70.76±0.19	**54.09±0.13**	**74.34±0.30**

Table 3: Performance on Arxiv dataset.

Models	1950-2007		1950-2009		1950-2011	
	2014-2016	2016-2018	2014-2016	2016-2018	2014-2016	2016-2018
UDAGCN	38.10 ± 1.62	OOM	42.85 ± 2.09	OOM	53.13 ± 0.31	OOM
AdaGCN	37.69 ± 1.19	43.01 ± 1.87	45.50 ± 0.68	47.53 ± 1.12	51.37 ± 1.12	53.34 ± 1.60
CWGCN	32.53 ± 0.52	35.10 ± 1.95	42.05 ± 0.37	36.97 ± 2.31	46.08 ± 0.99	41.56 ± 0.59
StruRW	38.56 ± 0.77	37.17 ± 2.75	43.55 ± 2.37	43.55 ± 2.37	53.19 ± 0.45	53.64 ± 0.65
PairAlign	39.98 ± 0.77	41.14 ± 2.07	44.60 ± 0.42	44.43 ± 0.34	53.75 ± 0.48	52.83 ± 0.98
Adalign	41.14 ± 0.22	45.07 ± 1.63	43.65 ± 0.48	48.15 ± 0.69	49.95 ± 2.62	52.91 ± 0.69
DACS	$\mathbf{42.94 \pm 2.25}$	$\mathbf{45.33 \pm 4.26}$	$\mathbf{45.87 \pm 0.97}$	$\mathbf{50.35 \pm 2.03}$	$\mathbf{54.80 \pm 0.75}$	$\mathbf{55.01 \pm 1.68}$

The best and second-best performances are marked as **bold** and underline, respectively.

Results on DBLP/ACM and Airport are reported in Table 2, and results on the large-scale ArXiv benchmark are reported in Table 3. For ArXiv, baselines that are out-of-memory (OOM) across all tasks are omitted. To ensure scalability, DACS uses a k-nearest neighbor strategy with $k = 100$ for estimating edge reweighting statistics. Across all benchmarks, DACS consistently outperforms competing methods, with an improvement of up to 7.72% on A-D. These gains indicate that covariate-induced discrepancies are a substantial source of target-domain error in real-world GDA. Methods based on label-related shift assumptions or single-stage representation alignment can perform well on some transfer tasks, but their performance is less stable across diverse domains. In contrast, DACS provides consistent improvements by jointly addressing FS and FCSS at their corresponding stages, supporting the claim that practical graph shift is not fully captured by a single final-representation discrepancy. Complexity analysis and additional model diagnostics are provided in Appendix A and Appendix C.4, respectively.

5.3 Ablation Analysis

We examine the role of each component in DACS by removing initial alignment (IA), layer-wise reweighting (RW), or final-layer alignment (FA). As shown in Table 4, the full model consistently achieves the best performance. Removing any component leads to clear degradation, indicating that the three modules play complementary rather than redundant roles.

Table 4: Ablation results on the variants of the proposed DACS method.

Models	DBLP/ACM		Airport					
	A-D	D-A	U-E	E-U	B-E	E-B	B-U	U-B
DACS w/o IA	71.91±0.89	62.44±0.26	51.30±0.16	49.18±0.14	55.65±0.06	69.62±0.32	52.12±0.42	71.13±0.51
DACS w/o RW	75.10±0.59	64.15±0.32	55.71±0.23	52.64±0.14	56.77±0.22	69.81±0.65	53.45±0.25	71.88±0.61
DACS w/o FA	66.28±5.21	60.83±2.06	46.95±2.41	48.11±1.65	46.02±3.87	65.47±2.95	47.82±1.90	63.58±3.39
DACS	**86.21±0.24**	**79.14±0.31**	**56.58±0.36**	**55.72±0.08**	**57.39±0.15**	**70.76±0.19**	**54.09±0.13**	**74.34±0.30**

Specifically, removing IA weakens the mitigation of FS-induced representation bias, while removing RW reduces the ability to correct FCSS-induced propagation bias. The largest degradation is observed when FA is removed, indicating that residual higher-order mismatch remains important even after propagation correction. Notably, on the A-D task with severe FCSS, removing RW reduces accuracy from 86% to 75%, substantially weakening the competitiveness of DACS against strong baselines.

These findings highlight the complementary roles of RW and FA in addressing FCSS: RW mitigates propagation bias arising from feature-conditioned structural differences, while FA reduces the residual representation discrepancy after propagation. Overall, these results are consistent with the theoretical analysis in Section 4, demonstrating that IA, RW, and FA address distinct yet complementary stages of adaptation rather than constituting a heuristic combination of alignment losses.

6 Conclusion and Future Work

We show that CS in GDA induces a stage-wise failure mode driven by two non-interchangeable sources of error: initial representation bias before message passing and propagation bias during message passing. This perspective exposes a fundamental limitation of existing methods that rely on single-stage alignment. Motivated by this analysis, we propose DACS, a stage-aware dual-alignment framework that explicitly targets both sources of discrepancy. By aligning initial representations, correcting propagation behavior, and refining residual mismatch at the final layer, DACS provides a principled alternative to heuristic alignment strategies. Both theory and experiments demonstrate that controlling only one discrepancy source is insufficient, whereas stage-aware dual alignment yields robust improvements across controlled and real-world settings. Overall, our results suggest that effective GDA should be formulated at the level of computation stages, rather than treating domain discrepancy as a single representation-level mismatch. One limitation of the current framework is that it focuses on static graph domains. A promising future direction is to extend the proposed stage-aware dual-alignment framework to evolving graphs, where both feature distributions and propagation mechanisms change dynamically over time.

Acknowledgments and Disclosure of Funding

The authors declare that no specific funding was received for this work and that they have no competing interests.

References

[1] Emmanuel Abbe. Community detection and stochastic block models: Recent developments. *Journal of Machine Learning Research*, 18(177):1–86, 2018.

[2] Wei Chen, Xingyu Guo, Shuang Li, Zhao Zhang, Yan Zhong, Fuzhen Zhuang, et al. Learning adaptive distribution alignment with neural characteristic function for graph domain adaptation. In *The Fourteenth International Conference on Learning Representations*, 2026.

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

[4] 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.

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

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

[7] 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.

[8] Yuqi Fang, Junhao Zhang, Linmin Wang, Qianqian Wang, and Mingxia Liu. Action: Augmentation and computation toolbox for brain network analysis with functional mri. *NeuroImage*, 305:120967, 2025.

[9] Bailey K Fosdick and Peter D Hoff. Testing and modeling dependencies between a network and nodal attributes. *Journal of the American Statistical Association*, 110(511):1047–1056, 2015.

[10] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario March, and Victor Lempitsky. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(59):1–35, 2016.

[11] Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. In *Advances in Neural Information Processing Systems*, volume 33, pages 22118–22133. Curran Associates, Inc., 2020.

[12] Myunghwan Kim and Jure Leskovec. Modeling social networks with node attributes using the multiplicative attribute graph model. In *Proceedings of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence*, pages 400–409, 2011.

[13] Hui Li, Bin Wu, Miaomiao Sun, Zhenfeng Zhu, Kuisheng Chen, and Hong Ge. Cross-domain contrastive graph neural network for lncrna–protein interaction prediction. *Knowledge-Based Systems*, 296:111901, 2024.

[14] Meihan Liu, Zhen Zhang, Jiachen Tang, Jiajun Bu, Bingsheng He, and Sheng Zhou. Revisiting, benchmarking and understanding unsupervised graph domain adaptation. In *The Thirty-eighth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024.

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

[16] Shikun Liu, Deyu Zou, Han Zhao, and Pan Li. Pairwise alignment improves graph domain adaptation. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pages 32552–32575. PMLR, 2024.

[17] Ziqi Liu, Yue Shen, Xiaocheng Cheng, Qiang Li, Jianping Wei, Zhiqiang Zhang, Dong Wang, Xiaodong Zeng, Jinjie Gu, and Jun Zhou. Learning representations of inactive users: A cross domain approach with graph neural networks. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 3278–3282, 2021.

[18] Leonardo F.R. Ribeiro, Pedro H.P. 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*, page 385–394, 2017.

[19] Junqi Shao, Shen Li, Ke Zhang, Anyou Wang, and Meng Li. Cross-city traffic prediction based on deep domain adaptive transfer learning. *Transportation Research Part C: Emerging Technologies*, 176:105152, 2025.

[20] 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.

[21] 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, 2021.

[22] 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.

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

[24] Jie Tang, Jing Zhang, Limin Yao, Juanzi Li, Li Zhang, and Zhong Su. Arnetminer: extraction and mining of academic social networks. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, page 990–998, New York, NY, USA, 2008.

[25] Wei Wang, Gaowei Zhang, Hongyong Han, and Chi Zhang. Correntropy-induced Wasserstein GCN: Learning graph embedding via domain adaptation. *IEEE Transactions on Image Processing*, 32:3980–3993, 2023.

[26] Hongwei Wen, Annika Betken, and Hanyuan Hang. Class probability matching with calibrated networks for label shift adaption. In *International Conference on Learning Representations*, volume 2024, pages 29337–29367, 2024.

[27] Hongwei Wen, Annika Betken, and Hanyuan Hang. Optimal learning of kernel logistic regression for complex classification scenarios. In *International Conference on Learning Representations*, volume 2025, pages 53918–53958, 2025.

[28] Hongwei Wen, Can Zhang, Haoyu He, Hanyuan Hang, and Minglong Lei. Progressive graph structure adjustment for homophily shift adaptation. In *Forty-third International Conference on Machine Learning*, 2026.

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

[30] Qitian Wu, Hengrui Zhang, Junchi Yan, and David Wipf. Handling distribution shifts on graphs: An invariance perspective. In *The Tenth International Conference on Learning Representations*, 2022.

[31] Yunpeng Xiao, Jinsong Yang, Wanjing Zhao, Qian Li, and Yucai Pang. Cross-domain social rumor-propagation model based on transfer learning. *IEEE Transactions on Neural Networks and Learning Systems*, 36(4):6529–6543, 2025.

[32] Chen Xu, Qiang Wang, Wenqi Zhang, and Chen Sun. Spatiotemporal ego-graph domain adaptation for traffic prediction with data missing. *IEEE Transactions on Intelligent Transportation Systems*, 25(12):20804–20819, 2024.

[33] Gezheng Xu, Li Yi, Pengcheng Xu, Jiaqi Li, Ruizhi Pu, Changjian Shui, A Ian McLeod, Boyu Wang, and Charles Ling. 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.

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

[35] 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*, page 2618–2626, New York, NY, USA, 2021.

[36] Yizhou Zhang, Guojie Song, Lun Du, Shuwen Yang, and Yilun Jin. DANE: Domain adaptive network embedding. In *International Joint Conference on Artificial Intelligence*, pages 4362–4368, 2019.

[37] Tianxiang Zhao, Dongsheng Luo, Xiang Zhang, and Suhang Wang. Multi-source unsupervised domain adaptation on graphs with transferability modeling. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 4479–4489, 2024.

Appendix

This appendix extends the stage-aware analysis of CS in GDA and the proposed DACS framework. It further substantiates the decomposition of target-domain error into two non-interchangeable components—representation mismatch and propagation mismatch—and their implications for adaptation. Section A describes the practical estimation of feature-conditioned edge reweighting for propagation correction. Section B provides complete proofs of all theoretical results, including target risk decomposition, non-interchangeability of stage-wise alignment, and additional analysis on propagation bias. Section C reports extended experimental results, including implementation details, ablations, and sensitivity analyses.

A Weight Estimation for Representation Reweighting

This section describes the practical estimation of the feature-conditioned edge reweighting introduced in Section 3. This component implements propagation correction under CS by approximating the conditional alignment objective derived in Section 4. As shown in Section 4, the ideal correction targets the mismatch in feature-conditioned edge formation across domains. The corresponding importance weight is

$$w_l^{uv} = \frac{P_T(A_{uv} \mid H_l^w, H_1)}{P_S(A_{uv} \mid H_l^w, H_1)}, \tag{A.1}$$

which measures the relative likelihood of edge formation conditioned on node representations. Direct estimation is intractable due to its dependence on global graph structure and high-dimensional features. We therefore derive a sequence of approximations that preserve its statistical meaning while enabling efficient estimation.

Local pairwise approximation. The full conditional $P(A_{uv} \mid H_l^w, H_1)$ depends on all node representations. We approximate it using only the two incident nodes:

$$P(A_{uv} \mid H_l^w, H_1) \approx P(A_{uv} \mid H_l^{w,u}, H_l^{w,v}), \tag{A.2}$$

implicitly assuming that higher-order dependencies are captured by the learned embeddings.

Distance-based reduction. We further assume that connectivity depends on representations through their geometric proximity:

$$P(A_{uv} \mid H_l^{w,u}, H_l^{w,v}) \approx P\left(A_{uv} \mid d(H_l^{w,u}, H_l^{w,v})\right), \tag{A.3}$$

where $d(\cdot, \cdot)$ denotes Euclidean distance. This reduces estimation to a one-dimensional conditional density, improving stability while remaining consistent with representation geometry in GNNs.

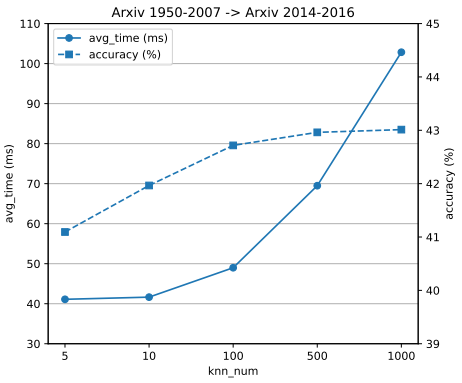
Binning-based estimation. Let d_{uv} denote the pairwise distance. We estimate $P(A = 1 \mid d)$ via a binning-based empirical estimator. We uniformly partition all pairwise distances into $\{[b_j, b_{j+1})\}_{j=1}^J$, and for node pair $(u, v) \in V_s^2$, we define

$$\widehat{P}_T(A_{uv} = 1 \mid d_{uv}) := \sum_{j=1}^{J} \mathbf{1}\{d_{uv} \in [b_j, b_{j+1})\} \cdot \frac{\sum_{(u',v') \in V_t^2} \mathbf{1}\{A_{u'v'} = 1,\ d^t_{u'v'} \in [b_j, b_{j+1})\}}{\sum_{(u',v') \in V_t^2} \mathbf{1}\{d^t_{u'v'} \in [b_j, b_{j+1})\}}, \tag{A.4}$$

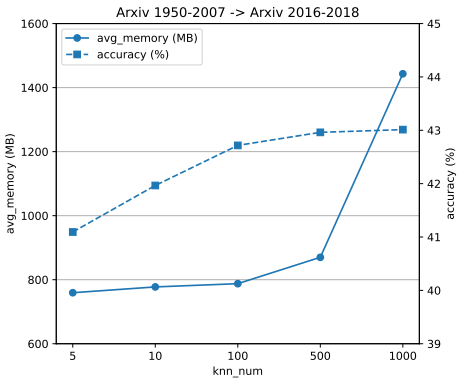
where $d^t_{u',v'}$ is the pairwise distance between the target nodes $(u', v') \in V_t^2$. An analogous estimator is constructed for the source domain, and their ratio yields $\widehat{w}_l^{uv}$.

k-nearest neighbor restriction. Computing all pairwise distances is prohibitive. We restrict estimation to the k nearest neighbors of each node, reducing complexity from quadratic to linear while preserving local statistics relevant for edge formation.

Computational complexity. Let $|V_S|, |V_T|$ denote the number of nodes, $|E_S|$ the number of source edges, d the embedding dimension, L the number of layers, and k the neighborhood size. The overall complexity is $O\big((|V_S| + |V_T|)kLd + |E_S|Ld\big)$, which scales linearly with graph size for fixed k, d, and L. Nearest neighbors are computed with cost $O\big((|V_S| \log |V_S| + |V_T| \log |V_T|)dk\big)$.



(a) Accuracy and average runtime under different k.



(b) Accuracy and average memory under different k.

Figure 4: Accuracy–efficiency tradeoff under different neighborhood sizes k.

Accuracy–efficiency tradeoff. The neighborhood size k controls a tradeoff between estimation quality and efficiency, evaluated on the `ArXiv` task. Figure 4 shows that moderate values of k capture sufficient local statistics, while larger values yield diminishing returns at higher computational cost. We therefore set $k = 100$.

Stabilized weights. To reduce variance, we use a convex combination with a neutral weight:

$$\widehat{w}_l^{uv} = \lambda \cdot \frac{\widehat{P}_T(A_{uv} \mid d_{uv})}{\widehat{P}_S(A_{uv} \mid d_{uv})} + (1 - \lambda), \tag{A.5}$$

where $\lambda \in [0, 1]$ controls the strength of reweighting.

Overall, this procedure provides a scalable and stable approximation to feature-conditioned propagation correction, bridging the theoretical importance weights in (A.1) and the practical implementation used in DACS.

B Proofs Related to Section 4

Proof of Theorem 4.1. We prove the theorem by following the computation path of a GNN and showing that the target-domain risk can be decomposed according to the two stages at which CS enters the model.

Recall that the GNN classifier computes predictions through

$$(X, A) \xrightarrow{f} H_1 \xrightarrow{\phi(\cdot, A)} H_L \xrightarrow{g} \widehat{Y},$$

where $H_1 = f(X)$ is the initial representation before message passing and $H_L = \phi(H_1, A)$ is the final representation after propagation. This decomposition is important because CS affects the two stages differently: feature shift changes the distribution of H_1, while feature-conditioned structure shift changes the conditional distribution of H_L given H_1.

Step 1: Conditional risk on intermediate representations. For fixed intermediate representations (H_1, H_L), define the conditional target-domain risk as

$$r_T(H_1, H_L) := \frac{1}{n} \cdot \mathbb{E}_{P_T}[\|g(H_L) - f_T(X, A)\|_1 \mid H_1, H_L],$$

and analogously define $r_S(H_1, H_L)$ under P_S. This quantity measures the expected classification error once the sample has already been mapped to the representation state (H_1, H_L). Thus, it separates the downstream prediction error from the distributional discrepancies that determine how frequently different representation states are visited.

Since $H_1 = f(X)$ and $H_L = \phi(H_1, A)$ are induced by the data distribution, the source and target domains induce different distributions over representations. In particular,

$$dP_T(H_1) := \int_{\{X:\, f(X) = H_1\}} dP_T(X), \qquad dP_T(H_L \mid H_1) := \int_{\{A:\, \phi(H_1, A) = H_L\}} dP_T(A \mid H_1).$$

The first distribution describes how often initial representations appear in the target domain. The second describes how graph structure transforms a fixed initial representation into a final representation through message passing. Therefore, discrepancies in $P(H_1)$ correspond to representation bias, while discrepancies in $P(H_L \mid H_1)$ correspond to propagation bias.

Step 2: Decomposing target risk by the tower rule. The target risk can be written as

$$\mathcal{R}_T(g \circ \phi) = \mathcal{R}_S(g \circ \phi) + \big(\mathcal{R}_T(g \circ \phi) - \mathcal{R}_S(g \circ \phi)\big).$$

Applying the law of total expectation over H_1 gives

$$\mathcal{R}_T(g \circ \phi) = \mathcal{R}_S(g \circ \phi) + \mathbb{E}_{P_T}[r_T(H_1)] - \mathbb{E}_{P_S}[r_S(H_1)]\,, \tag{A.6}$$

where $r_T(H_1) := \mathbb{E}_{P_T}[r_T(H_1, H_L) \mid H_1]$, and $r_S(H_1)$ is defined analogously.

We now decompose the difference between the two expectations:

$$\mathbb{E}_{P_T}(r_T(H_1)) - \mathbb{E}_{P_S}(r_S(H_1)) = \mathbb{E}_{P_T}\big[r_T(H_1) - r_S(H_1)\big] + \int_{\mathcal{H}_1} r_S(H_1)\big(dP_T(H_1) - dP_S(H_1)\big).$$

The second term is purely caused by the mismatch between the distributions of H_1. It is therefore the representation-stage discrepancy. Taking absolute values and applying the triangle inequality yields

$$\left|\int_{\mathcal{H}_1} r_S(H_1)\big(dP_T(H_1) - dP_S(H_1)\big)\right| \le \int_{\mathcal{H}_1} r_S(H_1)\big|dP_T(H_1) - dP_S(H_1)\big|.$$

It remains to analyze the first term $\mathbb{E}_{P_T}[r_T(H_1) - r_S(H_1)]$. Applying the tower rule again, now over H_L conditioned on H_1, gives

$$\begin{aligned} r_T(H_1) - r_S(H_1) = \mathbb{E}_{P_T}[r_T(H_1, H_L) - r_S(H_1, H_L) \mid H_1] \\ + \int_{\mathcal{H}_L} r_S(H_1, H_L)\big(dP_T(H_L \mid H_1) - dP_S(H_L \mid H_1)\big). \end{aligned}$$

The first term measures possible changes in the labeling function after fixing the representation state (H_1, H_L). Under the standard covariate-shift assumption, the labeling rule is shared across domains after representation. That is, there exists a common labeling function g^* such that $f_S(X, A) = g^*(H_L) = f_T(X, A)$. Hence, $r_T(H_1, H_L) = r_S(H_1, H_L)$, and the first term vanishes. The remaining term is therefore entirely induced by the conditional propagation distribution $P(H_L \mid H_1)$. This is precisely the propagation-stage discrepancy.

Substituting the two discrepancy terms into (A.6) gives

$$\begin{aligned} \mathcal{R}_T(g \circ \phi) &\le \mathcal{R}_S(g \circ \phi) + \int_{\mathcal{H}_1} r_S(H_1)\big|dP_T(H_1) - dP_S(H_1)\big| \\ &\quad + \mathbb{E}_{H_1 \sim P_T} \int_{\mathcal{H}_L} r_S(H_1, H_L)\big|dP_T(H_L \mid H_1) - dP_S(H_L \mid H_1)\big| \\ &\le \mathcal{R}_S(g \circ \phi) + \int_{\mathcal{H}_1} \big|dP_T(H_1) - dP_S(H_1)\big| \\ &\quad + \mathbb{E}_{H_1 \sim P_T} \int_{\mathcal{H}_L} \big|dP_T(H_L \mid H_1) - dP_S(H_L \mid H_1)\big|, \end{aligned} \tag{A.7}$$

where the last inequality follows from $r_S(H_1) \le 1$ and $r_S(H_1, H_L) \le 1$.

This inequality already reveals the key structure of the theorem: target risk is controlled by the source risk, a initial representation discrepancy in $P(H_1)$, and a conditional propagation discrepancy in $P(H_L \mid H_1)$.

Step 3: Converting L^1 discrepancies to KL terms. For any two probability measures P and Q, Pinsker's inequality gives $\mathrm{TV}(P, Q) \le \sqrt{\frac{1}{2} D_{\mathrm{KL}}(P, Q)}$. Since $\int |dP - dQ| = 2\,\mathrm{TV}(P, Q)$, we have

$$\int |dP - dQ| \le \sqrt{2} \cdot D_{\mathrm{KL}}^{1/2}(P, Q). \tag{A.8}$$

Applying (A.8) to the distributions of H_1 yields

$$\int_{\mathcal{H}_1} \big|dP_S(H_1) - dP_T(H_1)\big| \le \sqrt{2} \cdot D_{\mathrm{KL}}^{1/2}\big(P_S(H_1), P_T(H_1)\big). \tag{A.9}$$

Applying the same inequality to the conditional distributions of H_L given H_1, and then taking expectation over $H_1 \sim P_T$, yields

$$\begin{aligned}\mathbb{E}_{H_1 \sim P_T} &\int_{\mathcal{H}_L} \big|dP_S(H_L \mid H_1) - dP_T(H_L \mid H_1)\big| \\ &\le \sqrt{2} \cdot \mathbb{E}_{H_1 \sim P_T} D_{\mathrm{KL}}^{1/2}\big(P_S(H_L \mid H_1), P_T(H_L \mid H_1)\big).\end{aligned} \tag{A.10}$$

Combining (A.7), (A.9), and (A.10) proves (12). □

Proof of Theorem 4.2. The proof proceeds by explicit constructions that isolate each component in the stage-wise decomposition of Theorem 4.1. The key idea is to show that each discrepancy term—representation-level, propagation-level, and their entangled marginal at the final layer—can independently induce maximal target error even when the others are perfectly aligned. This establishes that the corresponding alignment objectives are fundamentally non-interchangeable.

(i) Initial representation alignment alone is insufficient. We construct a setting where the initial representation distributions are perfectly aligned, yet the propagation mechanism differs.

Let $H_1 \in \{-1, +1\}$. Define identical initial representation: $P_S(H_1 = +1) = P_T(H_1 = +1) = 0.5$, $P_S(H_1 = -1) = P_T(H_1 = -1) = 0.5$. Thus, $P_S(H_1) = P_T(H_1)$.

Now define domain-specific propagation: $H_L = H_1$ (source), $H_L = -H_1$ (target). Equivalently, $P_S(H_L = H_1 \mid H_1) = 1$, $P_T(H_L = -H_1 \mid H_1) = 1$, so $P_S(H_L \mid H_1) \neq P_T(H_L \mid H_1)$.

Let the labeling rule be $Y = H_1$. Then in the source domain, $g_S(H_L) = H_L$ achieves $\mathcal{R}_S(g_S) = 0$. However, in the target domain, $H_L = -H_1$, $\Rightarrow g_S(H_L) = H_L = -H_1 = -Y$, and therefore $\mathcal{R}_T(g_S) = 1$.

This construction shows that even perfect alignment of $P(H_1)$ does not control target risk if the conditional propagation distribution $P(H_L \mid H_1)$ is mismatched. Hence, representation alignment alone cannot eliminate propagation-induced error.

(ii) Propagation alignment alone is insufficient. We now construct the complementary case where propagation is identical across domains, but representation distributions differ.

Let $H_1 \in \{-1, +1\}$. Define $P_S(H_1 = +1) = 1$, $P_T(H_1 = -1) = 1$. Thus, $P_S(H_1) \neq P_T(H_1)$.

Use the same propagation rule: $H_L = H_1$ in both domains. Hence, $P_S(H_L \mid H_1) = P_T(H_L \mid H_1)$.

Let $Y = H_1$. In the source domain, $H_1 = +1$, $H_L = +1$, $Y = +1$, so the constant classifier $g_S(H_L) = +1$ is source-optimal and satisfies $\mathcal{R}_S(g_S) = 0$. In the target domain, $H_1 = -1$, $H_L = -1$, $Y = -1$, but $g_S(H_L) = +1 \neq -1 = Y$, so $\mathcal{R}_T(g_S) = 1$.

This shows that even perfect alignment of the conditional propagation distribution $P(H_L \mid H_1)$ does not guarantee generalization if the initial representation distribution $P(H_1)$ shifts. Thus, propagation alignment alone cannot compensate for representation bias.

(iii) Final-layer alignment alone is insufficient. Finally, we construct a case where the distribution of the final representation is identical across domains, but both underlying discrepancies persist.

Let $H_1 \in \{-1, +1\}$, $H_L \in \{-1, +1\}$. Define the source domain: $P_S(H_1 = +1) = 1$, $H_L = +1$, and the target domain: $P_T(H_1 = -1) = 1$, $H_L = +1$.

Then $P_S(H_L = +1) = P_T(H_L = +1) = 1$, so $P_S(H_L) = P_T(H_L)$.

However, $P_S(H_1 = +1) = 1$, $P_T(H_1 = -1) = 1$, so $P_S(H_1) \neq P_T(H_1)$.

The propagation mechanisms differ: $H_L = H_1$ (source), $H_L = -H_1$ (target), so $P_S(H_L \mid H_1) \neq P_T(H_L \mid H_1)$.

Let $Y = H_1$. In the source domain, $H_1 = +1$, $H_L = +1$, $Y = +1$, so $g_S(H_L) = +1$ achieves $\mathcal{R}_S(g_S) = 0$. In the target domain, $H_1 = -1$, $H_L = +1$, $Y = -1$, hence $g_S(H_L) = +1 \neq -1 = Y$, and $\mathcal{R}_T(g_S) = 1$.

This construction demonstrates that aligning only the marginal distribution $P(H_L)$ is insufficient, as it entangles representation and propagation discrepancies and therefore fails to control their underlying causes. □

Proof of Proposition 4.3. Setup and intuition. The key observation is that in a GNN, the randomness of the final-layer representation H_L given the initial representation H_1 is entirely induced by the randomness of the graph structure A. Formally, a GNN computes $H_L = \phi(H_1, A)$, where ϕ is a deterministic propagation operator shared across domains. Therefore, once H_1 is fixed, the conditional distribution $P(H_L \mid H_1)$ is fully determined by the conditional distribution of A given H_1. This establishes that any discrepancy in $P(H_L \mid H_1)$ must originate from a mismatch in $P(A \mid H_1)$.

Step 1: Expanding $P(H_L \mid H_1)$. By the law of total probability over all possible adjacency matrices $\mathcal{A}$,

$$P(H_L \mid H_1) = P(\phi(H_1, A) \mid H_1) = \sum_{A \in \mathcal{A}} P(\phi(H_1, A) \mid A, H_1)\, P(A \mid H_1). \tag{A.11}$$

Step 2: Domain invariance of the GNN mapping. The propagation function $\phi(H_1, A)$ is deterministic and shared across domains. Consequently, for any fixed (H_1, A),

$$P_S(\phi(H_1, A) \mid A, H_1) = P_T(\phi(H_1, A) \mid A, H_1). \tag{A.12}$$

Step 3: Aligning $P(H_L \mid H_1)$. Substituting $P_S(A \mid H_1) = P_T(A \mid H_1)$ and (A.12) into (A.11) yields

$$\begin{aligned} P_S(H_L \mid H_1) &= \sum_{A \in \mathcal{A}} P_S(\phi(H_1, A) \mid A, H_1)\, P_S(A \mid H_1) \\ &= \sum_{A \in \mathcal{A}} P_T(\phi(H_1, A) \mid A, H_1)\, P_T(A \mid H_1) = P_T(H_L \mid H_1). \end{aligned}$$

This establishes the desired alignment of the final-layer representations and completes the proof. □

Proof of Theorem 4.4. Under CS, the dominant source of target-domain error beyond representation mismatch arises from the propagation stage, where message passing is governed by domain-dependent graph structure. This phenomenon is captured by FCSS, which induces a discrepancy in $P(A \mid H_l, H_1)$ and consequently biases the aggregation operator AH_l.

This theorem formalizes a key design principle of DACS: propagation bias can be corrected by reweighting source-domain message passing so that its conditional expectation matches that of the target domain. In particular, it shows that once intermediate representations are aligned in expectation, a properly constructed importance weighting recovers the correct aggregation statistics. This establishes layer-wise reweighting as a principled mechanism for approximating conditional propagation alignment.

We aim to show that reweighting with the feature-conditioned likelihood ratio

$$w_l := \frac{P_T(A \mid H_l^w, H_1)}{P_S(A \mid H_l^w, H_1)}$$

aligns the conditional expectation of aggregated messages:

$$\mathbb{E}_{P_T}(AH_l \mid H_1) = \mathbb{E}_{P_S}(w_l A H_l^w \mid H_1).$$

Step 1: Decomposition via the tower rule. We first expand the target-domain aggregation using the law of total expectation:

$$\mathbb{E}_{P_T}(AH_l \mid H_1) = \mathbb{E}_{P_T}\big(\mathbb{E}_{P_T}(AH_l \mid H_l, H_1) \,\big|\, H_1\big). \tag{A.13}$$

Conditioned on (H_l, H_1), the only remaining source of randomness is the adjacency matrix A, so the inner expectation isolates the propagation mechanism.

Step 2: Importance reweighting of the propagation operator. The inner expectation depends on the target-domain distribution $P_T(A \mid H_l, H_1)$. Using importance sampling, we rewrite it under the source distribution:

$$\mathbb{E}_{P_T}(AH_l \mid H_l, H_1) = \mathbb{E}_{P_S}\left(\frac{P_T(A \mid H_l, H_1)}{P_S(A \mid H_l, H_1)} \cdot A \;\middle|\; H_l, H_1\right) \cdot H_l. \tag{A.14}$$

The likelihood ratio $P_T(A \mid H_l, H_1)/P_S(A \mid H_l, H_1)$ serves as a correction factor that rescales source-domain edges according to their relative likelihood under the target propagation mechanism.

Substituting (A.14) into (A.13), we obtain

$$\mathbb{E}_{P_T}(AH_l \mid H_1) = \mathbb{E}_{P_T}\left[\mathbb{E}_{P_S}\left(\frac{P_T(A \mid H_l, H_1)}{P_S(A \mid H_l, H_1)} \cdot A \;\middle|\; H_l, H_1\right) \cdot H_l \;\middle|\; H_1\right]. \tag{A.15}$$

Step 3: Source-domain reweighted aggregation. By definition of the reweighted propagation,

$$w_l AH_l^w = \frac{P_T(A \mid H_l^w, H_1)}{P_S(A \mid H_l^w, H_1)} \cdot AH_l^w.$$

Applying the tower rule yields

$$\mathbb{E}_{P_S}(w_l AH_l^w \mid H_1) = \mathbb{E}_{P_S}\left[\mathbb{E}_{P_S}\left(\frac{P_T(A \mid H_l^w, H_1)}{P_S(A \mid H_l^w, H_1)} \cdot A \;\middle|\; H_l^w, H_1\right) \cdot H_l^w \;\middle|\; H_1\right]. \tag{A.16}$$

Step 4: Matching expectations via representation alignment. Define

$$f(h, H_1) := \mathbb{E}_{P_S}\left(\frac{P_T(A \mid h, H_1)}{P_S(A \mid h, H_1)} \cdot A \;\middle|\; h, H_1\right) \cdot h.$$

Then (A.15) and (A.16) become

$$\mathbb{E}_{P_T}(AH_l \mid H_1) = \mathbb{E}_{P_T}\big[f(H_l, H_1) \mid H_1\big],$$
$$\mathbb{E}_{P_S}(w_l AH_l^w \mid H_1) = \mathbb{E}_{P_S}\big[f(H_l^w, H_1) \mid H_1\big].$$

By the assumption that the conditional expectation of H_ℓ and H_ℓ^w given H_1 are identical for any encoder $\phi^{\ell-1}$ across domains, we have

$$P_T(H_\ell \mid H_1) = P_S(H_\ell^w \mid H_1).$$

It suffices to ensure the expectation alignment

$$\mathbb{E}_{P_T}\big[f(H_l, H_1) \mid H_1\big] = \mathbb{E}_{P_S}\big[f(H_l^w, H_1) \mid H_1\big],$$

which proves that layer-wise reweighting aligns the conditional expectation of aggregated representations across domains and completes the proof. □

Proof of Theorem 4.5. Using conditional expectation, the source-domain risk can be rewritten as

$$\mathbb{E}_{P_S(H_L^w, Y)}\,\mathcal{L}(g(H_L^w), Y) = \mathbb{E}_{H_L^w \sim P_S}\big[\mathbb{E}_{Y \sim P_S(Y \mid H_L^w)}\,\mathcal{L}(g(H_L^w), Y)\big].$$

Define $\mathcal{L}_S(h) := \mathbb{E}_{Y \sim P_S(Y \mid h)}\,\mathcal{L}(g(h), Y)$, and similarly $\mathcal{L}_T(h) := \mathbb{E}_{Y \sim P_T(Y \mid h)}\,\mathcal{L}(g(h), Y)$. Then

$$\mathbb{E}_{P_S(H_L^w, Y)}\mathcal{L}(g(H_L^w), Y) = \mathbb{E}_{P_S(H_L^w)}\mathcal{L}_S(H_L^w), \quad \mathbb{E}_{P_T(H_L, Y)}\mathcal{L}(g(H_L), Y) = \mathbb{E}_{P_T(H_L)}\mathcal{L}_T(H_L).$$

We now decompose the risk difference by adding and subtracting $\mathbb{E}_{P_T(H_L)}\mathcal{L}_S(H_L)$:

$$\begin{aligned}
&\big|\mathbb{E}_{P_S(H_L^w)}\mathcal{L}_S(H_L^w) - \mathbb{E}_{P_T(H_L)}\mathcal{L}_T(H_L)\big| \\
&\le \underbrace{\big|\mathbb{E}_{P_S(H_L^w)}\mathcal{L}_S(H_L^w) - \mathbb{E}_{P_T(H_L)}\mathcal{L}_S(H_L)\big|}_{\text{representation discrepancy}} + \underbrace{\big|\mathbb{E}_{P_T(H_L)}\mathcal{L}_S(H_L) - \mathbb{E}_{P_T(H_L)}\mathcal{L}_T(H_L)\big|}_{\text{conditional label discrepancy}}.
\end{aligned}$$

By the assumption that $\mathcal{L}_S \in \mathcal{F}_H$, the first term is bounded by the IPM over the representation space:

$$\big|\mathbb{E}_{P_S(H_L^w)}\mathcal{L}_S(H_L^w) - \mathbb{E}_{P_T(H_L)}\mathcal{L}_S(H_L)\big| \le d_{\mathcal{F}_H}\big(P_S(H_L^w), P_T(H_L)\big).$$

For the second term, since $0 \le \mathcal{L}(g(h), y) \le 1$, the definition of total variation distance gives

$$\begin{aligned}
\big|\mathcal{L}_S(h) - \mathcal{L}_T(h)\big| &= \big|\mathbb{E}_{P_S(Y \mid h)}\mathcal{L}(g(h), Y) - \mathbb{E}_{P_T(Y \mid h)}\mathcal{L}(g(h), Y)\big| \\
&\le 2\, d_{\mathrm{TV}}\big(P_S(Y \mid h), P_T(Y \mid h)\big) \le 2\Delta_Y.
\end{aligned}$$

Taking expectation over $H_L \sim P_T(H_L)$ preserves the bound:

$$\big|\mathbb{E}_{P_T(H_L)}\mathcal{L}_S(H_L) - \mathbb{E}_{P_T(H_L)}\mathcal{L}_T(H_L)\big| \le 2\Delta_Y.$$

Combining the two bounds proves the theorem. □

C Additional Experimental Details and Analysis

This section provides additional experimental details and analyses supporting the empirical results in Section 5. We focus on four aspects: dataset construction, implementation details, model behavior, and parameter sensitivity. Beyond reproducibility, our goal is to further examine whether DACS behaves consistently with the stage-aware decomposition developed in the main text: FS should be mitigated before message passing, FCSS should be corrected during propagation, and residual mismatch should be reduced after propagation.

C.1 Synthetic Datasets

We construct synthetic graph datasets in which FS and FCSS can be independently controlled. This allows us to isolate the two sources of shift and evaluate whether each component of DACS addresses its intended failure mode, rather than only improving performance under entangled shifts.

Source-domain construction. In the source domain, node features are sampled from three Gaussian components $P_0 = \mathcal{N}([-1, 0], I)$, $P_1 = \mathcal{N}([1, 0], I)$, and $P_2 = \mathcal{N}([3, 2], I)$, with 1,000 samples drawn from each component. We apply k-means clustering with three clusters to all features and use the resulting cluster assignments as node labels, ensuring that labels are entirely determined by feature geometry. Edges are generated using a *Stochastic Block Model* (SBM) [1], where node pairs with the same label are connected with probability p, and those with different labels with probability q. In the source domain, we set $(p, q) = (0.02, 0.002)$. This construction separates the label-generating mechanism from the edge-generating mechanism, making it possible to perturb features and feature-conditioned connectivity independently.

Inducing FS. To construct FS-only scenarios, we generate target-domain features and labels in the same manner as in the source domain and then apply a global rotation of 30° or 60° to all target features. This operation changes the feature distribution, yielding $P_S(X) \neq P_T(X)$, while preserving pairwise distances and cluster-based labels. The SBM parameters are kept identical across domains, implying $P_S(A \mid X) = P_T(A \mid X)$. Accordingly, no rotation, 30° rotation, and 60° rotation correspond to no FS, mild FS, and severe FS, respectively. This setting isolates representation bias before message passing.

Inducing FCSS. To construct FCSS-only scenarios, we generate target-domain features and labels identically to the source domain and modify only the SBM parameters. Specifically, the target-domain parameters are set to $(p, q) = (0.02, 0.002)$, $(0.02, 0.010)$, and $(0.015, 0.016)$, corresponding to no FCSS, mild FCSS, and severe FCSS. In this setting, the feature distribution remains unchanged ($P_S(X) = P_T(X)$), while the feature-conditioned adjacency distribution differs across domains ($P_S(A \mid X) \neq P_T(A \mid X)$). This isolates propagation bias while keeping the input representation distribution fixed.

Combining FS & FCSS. Finally, we construct scenarios in which FS and FCSS co-occur by jointly rotating target-domain features and modifying the SBM parameters. This setting tests whether methods can handle coupled representation distortion and propagation bias, rather than correcting only one source of shift.

C.2 Real-world Datasets

We evaluate DACS on three real-world benchmarks that naturally exhibit cross-domain CS: citation networks, airport traffic networks, and large-scale temporal citation graphs. These benchmarks complement the synthetic setting by testing whether the same stage-wise mechanisms remain useful when FS and FCSS are no longer manually separated. Dataset statistics are summarized in Table 5.

Citation networks. `DBLP` and `ACM` are citation networks collected from different sources [24]. We use the processed `DBLPv8` and `ACMv9` datasets from [30], where `DBLPv8` contains papers published after 2010 and `ACMv9` covers papers from 2000 to 2010. Nodes represent papers, edges correspond to citation links, and node features are bag-of-words vectors extracted from paper titles. Each paper is classified into one of six research areas.

Table 5: Statistics of datasets used in our experiments.

Dataset	Domains	# Nodes	# Edges	# Labels
DBLP/ACM	ACMv9	7,410	11,135	6
	DBLPv8	5,578	7,341	
Airport	USA	1,190	13,599	4
	Brazil	131	1,038	
	Europe	399	5,995	
Arxiv	1950–2007	4,980	5,849	40
	1950–2009	9,410	13,179	
	1950–2011	17,401	30,486	
	2014–2016	69,499	232,419	
	2016–2018	120,740	615,415	

The domain shift between ACMv9 and DBLPv8 arises from differences in vocabulary usage, publication venues, and topical emphasis, which can induce both FS and FCSS. This benchmark therefore provides a natural testbed for dual alignment across heterogeneous citation sources.

Airport traffic networks. Brazil, Europe, and USA are airport traffic networks from the Airport dataset [18]. Nodes represent airports, edges denote flight connections, and node labels correspond to airport activity levels. Cross-domain shifts arise from region-specific traffic patterns, geographical constraints, and infrastructure differences, leading to both feature and structure discrepancies across domains. These shifts are particularly relevant for evaluating whether propagation correction can handle region-dependent connectivity patterns.

Temporal citation networks. ArXiv [11] is a large-scale citation network for studying temporal distribution shift. We treat papers published in earlier periods (1950–2007/2009/2011) as source domains and papers from later periods (2014–2016 and 2016–2018) as target domains. This split induces progressive CS driven by topic evolution, terminology changes, and evolving citation behavior, making it a challenging setting for evaluating FCSS correction at scale.

For notation, we denote ACMv9 and DBLPv8 as A and D, and Brazil, Europe, and USA as B, E, and U, respectively. Two cross-domain tasks are constructed for citation networks, and six tasks are constructed for the airport dataset.

C.3 Implementation Details

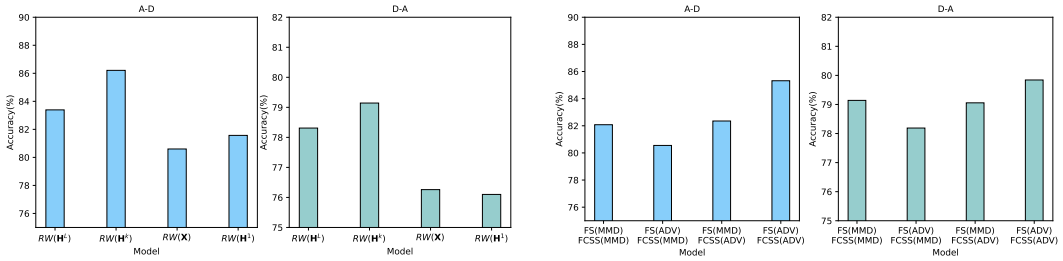
All experiments are implemented in PyTorch on a workstation equipped with an Intel(R) Core(TM) i7-14700K CPU @ 3.40GHz and an NVIDIA GeForce RTX 4080 GPU with 16GB memory.

Models for addressing FS and FCSS. To mitigate FS, we employ an MLP-based feature extractor together with an MLP domain discriminator. The number of layers in the feature extractor is selected from $\{1, 2, 3\}$, with hidden dimensions chosen from $\{32, 64, 128, 256\}$ for real-world datasets and $\{8, 16, 32, 64\}$ for synthetic datasets. The domain discriminator is implemented as a two-layer MLP with hidden dimensions selected from $\{8, 16, 32, 64\}$ across all datasets.

To address FCSS, we use a GNN to learn node representations. The number of GNN layers is also selected from $\{1, 2, 3\}$, with hidden dimensions chosen from $\{32, 64, 128, 256\}$ for real-world datasets and $\{8, 16, 32, 64\}$ for synthetic datasets. The domain discriminator used for final-layer alignment shares the same architecture as that used for FS. This keeps the representation-side and propagation-side alignment modules comparable, so that differences in performance can be attributed to the stage-wise intervention rather than architecture imbalance.

Optimization and training protocol. For the synthetic and Airport datasets, the learning rate is set to 0.003. For the DBLP and ACM datasets, the learning rate is fixed to 0.007, as suggested in [15]. All models are trained for 300 epochs using the source-domain graph.

Edge reweighting and hyperparameters. The reweighting coefficient λ in (A.5) controls the contribution of reweighted edges during message passing, where $\lambda = 1$ fully adopts the reweighted graph and $\lambda = 0$ recovers the original graph. In the edge probability estimator (A.4), all node pairs



(a) Different strategies of weight estimation.

(b) Different strategies of domain adaptation.

Figure 5: Results of model analysis on reweighting and domain adaptation strategies.

are used for DBLP, ACM, and Airport, while for ArXiv we retain only the k nearest neighbors per node when forming node pairs to reduce computational cost. Unless otherwise specified, k is fixed to 100.

The alignment coefficients γ_{IA} and γ_{FA} in (11) follow the standard scheduling strategy $\min\{2/(1+e^{-10p})-1,\ 0.1\}$, where p increases linearly from 0 to 1 over training [10].

Following [15], we introduce three reweighting-related hyperparameters: the starting epoch e, the reweighting frequency t, and the reweighting ratio λ. The search spaces are $e \in \{100, 150, 200, 250\}$, $t \in \{1, 5, 10, 15\}$, and $\lambda \in \{0.1, 0.2, \ldots, 0.9\}$. The number of distance bins in (A.4) is fixed to $J = 100$ for all experiments.

Optimal hyperparameters. On real-world datasets, the optimal hyperparameters are as follows: the number of GNN layers and the GNN hidden dimension are set to 3 and 64, respectively, while the number of MLP layers (for initial alignment) and the MLP hidden dimension are set to 1 and 128, respectively. For the two domain discriminators, the number of layers and the hidden dimension are set to 2 and 64, respectively.

On the synthetic dataset, the optimal numbers of GNN and MLP layers remain the same as those on the real-world datasets, while the corresponding hidden dimensions are set to 16 and 64, respectively. For the two domain discriminators, the number of layers and the hidden dimension are set to 2 and 16, respectively.

The optimal values of e, t, and λ are 200, 5, and 0.5, respectively.

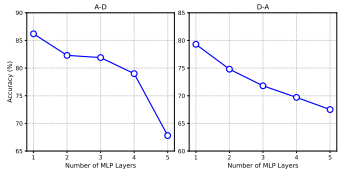
C.4 Model Analysis

This subsection analyzes how design choices in DACS affect feature-conditioned edge reweighting and domain alignment. The analysis focuses on two questions: *(i)* which representation level should be used to estimate reweighting coefficients for FCSS correction, and *(ii)* how alignment objectives should be coordinated across FS and FCSS. Both questions directly test whether DACS should operate at the stage where the corresponding bias is introduced.

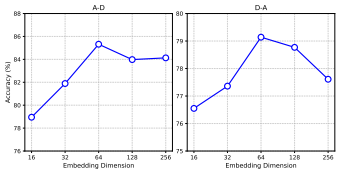
Which representation level should be used for weight estimation? We compare four reweighting strategies based on different representation levels:

- RW(X): reweighting based on the original node features X,
- RW(H_1): reweighting based on the input representation of the GNN,
- RW(H_k): reweighting based on the k-th layer representation, as described in (8),
- RW(H_L): reweighting based on the final-layer representation.

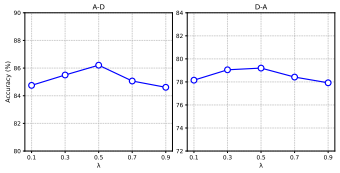
The results are shown in Figure 5a. RW(X) and RW(H_1) consistently underperform, suggesting that early representations do not contain sufficient structural abstraction for estimating feature-conditioned edge probabilities. RW(H_L) is also suboptimal, likely because final representations have already absorbed biased propagation effects. In contrast, RW(H_k) achieves the best and most stable performance, supporting the design choice of estimating edge weights at the same representation



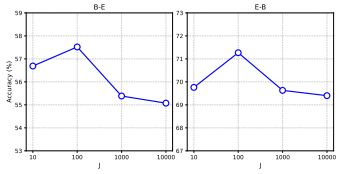
(a) Effect of the number of MLP layers.



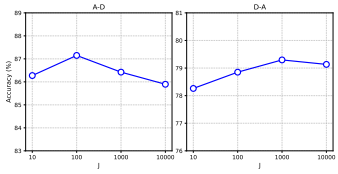
(b) Effect of embedding dimension.



(c) Sensitivity to the reweight. coeff. λ.



(d) Sensitivity to the J on `Airport`.



(e) Sensitivity to J on `DBLP` and `ACM`.

Figure 6: Results of parameter analysis.

level where reweighting is applied. This is consistent with the theoretical view that FCSS should be corrected during propagation, not solely before or after it.

How should FS and FCSS be aligned jointly? We further study how alignment objectives should be coordinated across the two stages. In the main model, DACS uses adversarial objectives (ADV) for both initial representation alignment and final-layer residual alignment, as defined in Eqs. (7) and (9). Here, we compare pairwise combinations of maximum mean discrepancy (MMD) and adversarial alignment (ADV).

The results in Figure 5b show that using a consistent alignment strategy across stages—either MMD+MMD or ADV+ADV—achieves better or comparable performance than mixing different objectives. This suggests that although FS and FCSS are distinct sources of shift, their effects are coupled along the GNN pipeline; inconsistent alignment objectives may introduce conflicting optimization signals. The results also highlight the modularity of DACS: adversarial alignment is not essential to the framework, provided that the chosen alignment tools are applied coherently across stages.

C.5 Parameter Analysis

We analyze the sensitivity of DACS to key hyperparameters associated with the dual-alignment framework: *(i)* the depth of the MLP used for feature alignment, *(ii)* the embedding dimension, *(iii)* the reweighting coefficient λ, and *(iv)* the number of distance bins J used in estimating feature-conditioned edge probabilities. The goal is to verify that the proposed components are stable and do not depend on delicate tuning.

Effect of feature-alignment depth. Figure 6a examines the number of MLP layers used for initial feature alignment. Excessively deep MLPs lead to performance degradation, suggesting that overly aggressive early-stage alignment may distort task-relevant feature semantics before structural information is incorporated. Moderate-depth MLPs provide a better balance between domain invariance and feature preservation.

Effect of embedding dimension. Figure 6b shows the influence of embedding dimension. An embedding dimension of 64 consistently yields strong performance, providing sufficient capacity to encode semantic and structural information while avoiding instability or overfitting.

Effect of the reweighting coefficient λ. The reweighting coefficient λ controls the strength of FCSS correction. As shown in Figure 6c, DACS remains stable across the tested range, with maximum accuracy variation around 1%. This indicates that the proposed reweighting mechanism does not rely on precise calibration and acts as a robust propagation-bias correction procedure.

Effect of the number of distance bins J. The number of distance intervals J determines the resolution of the empirical estimator for feature-conditioned edge probabilities. Figures 6d and 6e show that DACS is robust to a wide range of J values. On smaller Airport graphs, accuracy varies within 2%, while on larger citation networks, the variation is within approximately 1%. We therefore fix $J = 100$ in all experiments, avoiding dataset-specific tuning.

D Societal Impacts

This work studies graph domain adaptation under covariate shift and may improve the reliability of graph learning systems deployed across heterogeneous domains. Potential applications include recommendation systems, social network analysis, and biological graph modeling. However, like other graph learning methods, the proposed approach may inherit biases present in the training data and could potentially amplify unfair or spurious correlations across domains. Care should therefore be taken when deploying such systems in high-stakes settings.

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 make sure all the claims are accurate.

   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 discuss the limitations and future directions for improvement in the Conclusion and Future Work 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: [Yes]

   Justification: We provide the full set of assumptions for all theoretical results. We provide complete and correct proofs for all theoretical results in Appendix B.

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: We provide implementation details in Appendix C, and our code is publicly available at `https://github.com/hongweiaustinwen/DualAlign`.

   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: We make our code publicly available at `https://github.com/hongweiaustinwen/DualAlign` to facilitate the reproduction of our experimental results.

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: We specify all the training and test details in Appendix C to help understand the results.

   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 the standard error in all tables to illustrate the variability in performance across methods.

   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 introduce the required computer resources in Appendix C.

   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: We follow 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: [Yes]

    Justification: We discuss the societal impacts in Appendix D.

    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: The paper poses no such 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: We cite the original paper that produced the code package or dataset.

    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: The assets we submitted have detailed documentation.

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: The paper does not involve crowdsourcing nor research with human subjects.

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 paper does not involve crowdsourcing nor research with 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: The core method development in this research does not involve LLMs as any important, original, or non-standard 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.