COME: COMPLEMENTARY RELIABILITY MONITORING WITH EXPERT ALLOCATION FOR MULTI-SOURCE-FREE GRAPH DOMAIN ADAPTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

This paper studies the problem of multi-source-free graph domain adaptation, which aims to adapt multiple pretrained source models to target graphs without access to source graph data. Existing approaches usually combine pseudo-labeling techniques with domain alignment to learn from unlabeled target graphs. Despite the progress, their performance can degrade when source predictions are overconfident due to overconfident pseudo-labels from unaligned source models and thus potential error accumulation. To address this problem, we propose a novel approach named Complementary Reliability Monitoring with Expert Allocation (COME) for multi-source-free graph domain adaptation. The core of our COME extracts complementary structural and temporal stability signals for target supervision and expert allocation, i.e., *structural stability* and *temporal stability*. From the structural perspective, we feed perturbed views of the target graph into source models and compare their predictions to estimate structural stability. From the temporal perspective, we train a target-adaptive model and accumulate disagreement between target predictions and memory-based pseudo-labels that characterize temporal stability. These conflict scores are further incorporated into node states to stabilize pseudo-label supervision during target model adaptation. During inference, we leverage complementary stability signals as a routing function to dynamically allocate both frozen source and trainable target experts for reliable target predictions. Experiments on three benchmark families show that COME achieves the best mean accuracy in all six transfer settings.

1 INTRODUCTION

Unsupervised graph domain adaptation (UGDA) aims to transfer knowledge from labeled source graphs to an unlabeled target graph under distribution shifts (Liu et al., 2024a; Mao et al., 2024). Conventional approaches typically rely on access to source graph data during adaptation, which may be restricted by privacy and data-sharing requirements (Yuan et al., 2025). Moreover, a model trained on a single source domain may capture only a subset of the patterns present in the target graph (Liu et al., 2024b). These limitations motivate adaptation approaches that exploit knowledge from multiple source domains while retaining only the pretrained models.

Multi-source-free graph domain adaptation aims to adapt multiple pretrained source models to an unlabeled target graph without access to the original source graph data (Pei et al., 2024). Existing multi-source-free domain adaptation studies leverage information from multiple pretrained source models to support target adaptation (Han et al., 2023; Zhao et al., 2024). Unlike adaptation for independent samples, node prediction on graphs is also affected by neighborhood relations. Differences in node attributes and local structures therefore make different source models suitable for different target nodes (Zhang & He, 2025). In graph-based adaptation, multi-source knowledge is further exploited at the node level by adjusting source-model contributions according to local context, typically through pretrained parameter transfer, source-prediction aggregation, or source-guided target embedding optimization (Pei et al., 2025). However, without target labels, effectively using source-model knowledge and providing reliable supervision for target adaptation remain difficult.

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

These challenges make the reliability of source predictions and target pseudo-labels central to multi-source-free graph domain adaptation. Accordingly, two key problems still need to be addressed. ❶ ***How can source-model reliability be assessed for different target nodes without target labels?*** Source models trained on different graphs are not aligned with the target structure, so high-confidence predictions can still be unreliable. Since target labels are unavailable, prediction stability under local structural changes provides label-free evidence for differentiating source behavior on target nodes. ❷ ***How can reliable target supervision be maintained as predictions evolve during adaptation?*** Target adaptation relies on pseudo-labels generated from predictions, but their quality varies across nodes and iterations. Once incorrect pseudo-labels are reused as supervision, prediction errors can persist and accumulate during adaptation. Temporal consistency across iterations provides another basis for assessing target supervision. Reliable multi-source-free graph domain adaptation thus requires reliability assessment from both structural and temporal perspectives.

To address the aforementioned challenges, we propose Complementary Reliability Monitoring with Expert Allocation (named `COME`) for multi-source-free graph domain adaptation. `COME` examines target adaptation from both structural and temporal perspectives and uses the resulting stability information to coordinate source and target models at the node level. From the structural perspective, `COME` evaluates frozen source models on perturbed target graphs and compares their responses across graph views, providing node-specific evidence of source-model stability. From the temporal perspective, `COME` tracks target predictions throughout adaptation and accumulates prediction conflicts to characterize node states, which regulate pseudo-label supervision during target-model training. Finally, structural and temporal stability jointly guide node-wise expert allocation, assigning different contributions to frozen source models and the target-adaptive model for each target node. Experiments on three benchmark families across six transfer settings show that `COME` consistently outperforms existing baselines. The main contributions of this work are summarized as follows:

❶ ***New Perspective.*** We characterize reliability in multi-source-free graph domain adaptation from complementary structural and temporal perspectives, providing a more fine-grained node-level view of both source-model utilization and target adaptation across heterogeneous target nodes.

❷ ***Dynamic Framework.*** We propose `COME`, a framework that links complementary reliability monitoring with node-wise expert allocation for target adaptation. The allocation adapts expert participation to individual target nodes rather than applying a fixed combination across the graph.

❸ ***Comprehensive Validation.*** We conduct extensive experiments on multiple benchmark datasets and transfer settings, showing that `COME` achieves the best mean performance in all settings.

2 Related Work

Multi-Source-Free Graph Domain Adaptation. Multi-source-free domain adaptation uses several pretrained source models to support target adaptation without accessing the original source data (Li et al., 2024b; Pei et al., 2025). Existing methods combine source predictions, transfer pretrained parameters, or use source-model outputs to guide target training (Li et al., 2024a; Nehdi et al., 2026; Liu et al., 2025). These strategies aim to exploit transferable knowledge encoded in pretrained source models without relying on source-domain samples. For graph-structured data, this setting further uses multiple source models to support node-level adaptation on an unlabeled target graph with heterogeneous node characteristics (Pei et al., 2024). Different target nodes may exhibit distinct structural and feature patterns, making the usefulness of each pretrained source model vary across nodes (Zhang & He, 2025). However, source-model reliability varies across target nodes, while target predictions may evolve during adaptation, limiting the effective use of multiple source models. `COME` instead focuses on complementary reliability monitoring by modeling structural source stability and temporal target evolution to guide target supervision and node-specific expert allocation.

Mixture of Experts. Mixture of experts (MoE) coordinates multiple expert models through a gating mechanism for input-specific prediction (Xie et al., 2024). Two common expert-routing strategies are dense routing (Cai et al., 2025) and sparse routing (Zhou et al., 2022). Dense routing assigns a weight to every expert and combines all expert outputs for each input, capturing complementary knowledge from different experts to improve prediction (Ye & Xu, 2023). In contrast, sparse routing activates only a small subset of experts for each input, allowing the model to select suitable experts at a lower computational cost (Dai et al., 2024). Recently, MoE has been explored in graph learning for tasks such as molecular property prediction (Jiang et al., 2025) and fair graph representation learning (Liu et al., 2023). Graph-based MoE methods further enable different nodes to select or

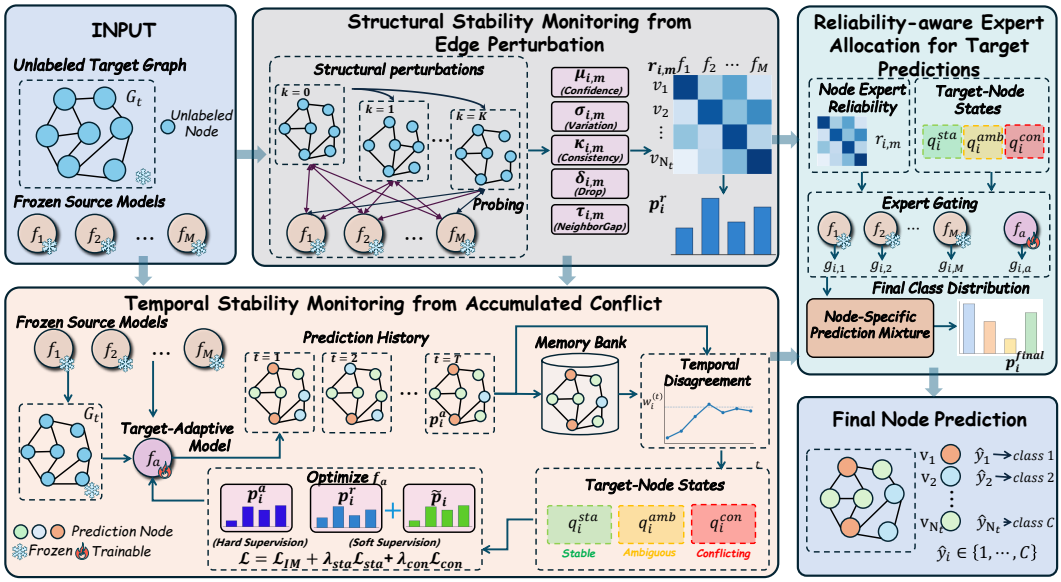


Figure 1: The overall framework of the proposed COME. Edge perturbations monitor structural stability for node-specific source reliability under structural shifts, while accumulated conflicts capture temporal stability and target-node states during adaptation. These complementary signals guide reliability-aware expert allocation at the node level for final target-node predictions.

combine experts according to their individual characteristics, improving adaptation to heterogeneous graph patterns (Wang et al., 2023). However, MoE has been less explored in multi-source-free graph domain adaptation, where source models perform differently across target nodes and target predictions evolve over time. Our method addresses this gap by jointly using structural source reliability and temporal target-node states to determine node-specific expert contributions.

3 The Proposed COME

Problem Definition. Multi-source-free graph domain adaptation transfers knowledge from multiple graph neural networks (GNNs) pretrained on source domains to an unlabeled target graph without access to source data. We consider C-class node classification on the target graph $\mathcal{G}_t = (\mathcal{V}_t, \mathcal{E}_t, \mathbf{X}_t)$, where $\mathcal{V}_t = \{v_i\}_{i=1}^{N_t}$ is the node set with N_t nodes, $\mathcal{E}_t$ is the edge set, and $\mathbf{X}_t \in \mathbb{R}^{N_t \times d}$ is the node-feature matrix. The graph structure is represented by $\mathbf{A}_t \in \{0,1\}^{N_t \times N_t}$, and the label space is $\mathcal{Y} = \{1, \ldots, C\}$. Each target node v_i has a label $y_i \in \mathcal{Y}$ that is unavailable during adaptation. We are given M pretrained source GNNs $\mathcal{F}_s = \{f_m\}_{m=1}^{M}$, each trained on a labeled source domain. During adaptation, source data are inaccessible and all source models remain frozen. For each target node v_i, the goal is to predict a class probability vector $\hat{\mathbf{p}}_i \in [0,1]^C$ and its label $\hat{y}_i$.

3.1 Framework Overview

For this adaptation setting, we propose COME, comprising structural stability monitoring from edge perturbation, temporal stability monitoring from accumulated conflict, and reliability-aware expert allocation for target predictions. Our COME captures structural source information, tracks temporal changes in target predictions, and integrates both signals for node-specific prediction under both structural variation and temporal evolution. The structural and temporal views provide complementary reliability evidence for coordinating source and target-adaptive models. First, we perturb the target graph and probe the frozen source models across structural views, using cross-view prediction changes to estimate node-specific source reliability. Second, we track the disagreement between target predictions and memory-based pseudo-label supervision over time, using the accumulated history to characterize target-node states and optimize the target-adaptive model. Finally, source-model reliability and temporal node states are jointly used to determine the contributions of the source and target-adaptive models for each node, producing the final target prediction. The overall framework is illustrated in Figure 1, with each part detailed in the following subsections.

3.2 Structural Stability Monitoring from Edge Perturbation

In multi-source-free graph domain adaptation, source graphs are unavailable and pretrained source models are retained. Existing methods typically combine multiple source models through prediction aggregation or related integration strategies to support adaptation on the target graph. In this work, we revisit this problem from two complementary perspectives: *structural stability* and *temporal stability*. We first examine structural stability by perturbing edges in the target graph (Zhang et al., 2026) to characterize source-model reliability for individual target nodes.

To begin with, we retain the target graph as the original view $k = 0$ and construct K edge-perturbed views by adding or removing a small number of edges, while keeping the node features unchanged. For each frozen source model f_m, we obtain its C-class predictions for every target node across the original and perturbed views. These predictions are summarized by five statistics to monitor structural stability for each node-model pair. Specifically, $\mu_{i,m}$ and $\sigma_{i,m}$ denote the mean and standard deviation of prediction confidence across views, where confidence is the maximum class probability. The term $\kappa_{i,m}$ measures the proportion of views predicting the modal class, $\delta_{i,m}$ measures the confidence drop from the original view to the cross-view mean, and $\tau_{i,m}$ measures the confidence difference between the node and its one-hop neighborhood. For isolated nodes, we set $\tau_{i,m} = 0$. The node-specific source reliability under structural perturbations is then computed as:

$$r_{i,m} = \exp(-\left[(1-\mu_{i,m}) + \sigma_{i,m} + (1-\kappa_{i,m}) + [\delta_{i,m}]_+ + [\tau_{i,m}]_+\right]), \tag{1}$$

where $r_{i,m} \in (0, 1]$ denotes the reliability of source model m for node v_i, and $[x]_+$ retains the positive part of x. A larger $r_{i,m}$ indicates stronger prediction stability across structural views. We further aggregate the source predictions for subsequent target adaptation. Let $\bar{\mathbf{p}}_{i,m}$ denote the view-averaged prediction of source model m, and let $\tilde{r}_{i,m}$ denote its reliability normalized across all source models. The resulting reliability-weighted source prediction is:

$$\mathbf{p}_i^r = \sum_{m=1}^{M} \tilde{r}_{i,m} \bar{\mathbf{p}}_{i,m}, \tag{2}$$

where $\tilde{r}_{i,m} \in (0, 1]$ is the normalized reliability satisfying $\sum_{m=1}^{M} \tilde{r}_{i,m} = 1$, and $\mathbf{p}_i^r \in [0, 1]^C$ is the resulting reliability-weighted source prediction for node v_i. The cross-view statistics characterize source-model prediction stability under edge perturbations across different target graph views. We next formalize the structural change between the original and perturbed target views.

Definition 3.1. *Let $\mathbf{A}^{(0)} = \mathbf{A}_t$ be the target adjacency matrix and let $\{\mathbf{A}^{(k)}\}_{k=1}^{K}$ be its perturbed views. The views form an ϵ_A-structural neighborhood of $\mathbf{A}^{(0)}$ if $\max_{1\le k\le K} \|\mathbf{P}(\mathbf{A}^{(k)}) - \mathbf{P}(\mathbf{A}^{(0)})\|_2 \le \epsilon_A$, where $\mathbf{P}(\cdot)$ denotes the normalized propagation operator and $\epsilon_A \ge 0$ bounds the normalized structural change across the perturbed views. The spectral and Frobenius norms used throughout this analysis are denoted by $\|\cdot\|_2$ and $\|\cdot\|_F$ for all structural views considered.*

Theorem 3.1. *Consider source model $f_m = \psi_m \circ \phi_m$. For each structural view $k \in \{0, \dots, K\}$, the L-layer encoder ϕ_m starts from $\mathbf{H}_{m,k}^{(0)} = \mathbf{X}_t$ and updates $\mathbf{H}_{m,k}^{(\ell+1)} = \varphi\Big(\mathbf{P}(\mathbf{A}^{(k)})\mathbf{H}_{m,k}^{(\ell)}\mathbf{W}_m^{(\ell)}\Big)$ for $\ell = 0, \dots, L-1$. Assume that φ is 1-Lipschitz under the Frobenius norm with $\varphi(0) = 0$, $\|\mathbf{P}(\mathbf{A}^{(k)})\|_2 \le 1$ for every view k, and $\|\mathbf{W}_m^{(\ell)}\|_2 \le B_{m,\ell}$ for each encoder layer. The probability map ψ_m is $B_{m,\psi}$-Lipschitz under the Frobenius norm over all structural views considered during adaptation. If the structural views satisfy Definition 3.1, then every perturbed view obeys:*

$$\left\| f_m(\mathbf{X}_t, \mathbf{A}^{(k)}) - f_m(\mathbf{X}_t, \mathbf{A}^{(0)}) \right\|_F \le C_m \epsilon_A, \qquad C_m = B_{m,\psi} L \|\mathbf{X}_t\|_F \prod_{\ell=0}^{L-1} B_{m,\ell}. \tag{3}$$

For node v_i, let $\hat{y}_{i,m}^{(0)} = \arg\max_{c\in\mathcal{Y}} p_{i,m,c}^{(0)}$ and define the original-view margin as $\Gamma_{i,m} = p_{i,m,\hat{y}_{i,m}^{(0)}}^{(0)} - \max_{c\neq\hat{y}_{i,m}^{(0)}} p_{i,m,c}^{(0)}$, capturing class separation in the original view. If $\Gamma_{i,m} > 2C_m\epsilon_A$, every perturbed view preserves the predicted class and $\kappa_{i,m} = 1$ under the considered structural perturbations.

Theorem 3.1 shows that bounded edge perturbations lead to bounded changes in source-model predictions, while a large class margin preserves prediction consistency across views. The resulting upper bounds on $\sigma_{i,m}$ and $\delta_{i,m}$, together with the complete proof, are provided in Appendix B.1.

3.3 Temporal Stability Monitoring from Accumulated Conflict

After assessing structural stability, we further examine temporal stability during target adaptation. The goal of temporal stability monitoring is to characterize the evolution of pseudo-label supervision during target adaptation (Zhang et al., 2025). Therefore, we monitor target predictions across adaptation steps and accumulate their conflicts to characterize temporal stability.

During target adaptation, we construct a target-adaptive model f_a by combining frozen source-model parameters with trainable target-side parameters. At each graph layer, the representation of node v_i is updated by combining source-specific messages with a trainable target-specific message:

$$\mathbf{h}_i^{(\ell+1)} = \varphi\left(\sum_{m=1}^{M} \alpha_{i,m}^{(\ell)} \mathbf{z}_{i,m}^{(\ell)} + \lambda_g \mathbf{z}_{i,g}^{(\ell)}\right), \qquad \ell = 0, \ldots, L-1, \tag{4}$$

where $\mathbf{z}_{i,m}^{(\ell)}$ and $\mathbf{z}_{i,g}^{(\ell)}$ denote the source-specific and target-specific messages computed from the local representation of v_i using frozen source transformations and trainable target-side transformations. The local representation is obtained by averaging the features of v_i and its one-hop neighbors. Each source message is assigned a node-specific weight $\alpha_{i,m}^{(\ell)}$ according to its compatibility with the local representation, with Sparsemax used to suppress source responses with low compatibility. The coefficient λ_g controls the contribution of the target-specific message, and $\varphi(\cdot)$ denotes the activation function. After the final layer, the frozen source classifiers produce class-probability distributions from $\mathbf{h}_i^{(L)}$. Their outputs are assigned node-specific Sparsemax weights $\omega_{i,m}$ and combined as:

$$\mathbf{p}_i^a = \sum_{m=1}^{M} \omega_{i,m} \psi_m(\mathbf{h}_i^{(L)}) = f_a(\mathbf{X}_t, \mathbf{A}_t; \theta_a)_i, \tag{5}$$

where the weights are non-negative and sum to one, and θ_a denotes the trainable parameters. All source-model parameters remain frozen. To monitor the temporal stability of target predictions, we use a memory bank to record historical target features and predictions. For each node, its K_{nn} nearest neighbors in the stored feature space are retrieved, and their predictions are averaged to form the memory-based pseudo-label distribution $\tilde{\mathbf{p}}_i^{(t)}$. The class with the highest probability is taken as the pseudo-label $\tilde{y}_i^{(t)}$. At adaptation step t, we measure the disagreement between the current target-adaptive prediction and this pseudo-label as $d_i^{(t)} = 1 - p_{i,\tilde{y}_i^{(t)}}^{(t)}$. Rather than using this instantaneous disagreement directly, we accumulate it over time as a conflict score:

$$w_i^{(t)} = \rho w_i^{(t-1)} + (1-\rho) d_i^{(t)}, \qquad 0 \le \rho < 1, \tag{6}$$

where $w_i^{(t)} \in [0,1]$ summarizes the prediction history of node v_i, and ρ controls the contribution of previous observations. The first observation initializes the accumulated disagreement, and we write w_i for its current value below to simplify subsequent node-state characterization during target adaptation. We convert the accumulated disagreement into three continuous node-state scores:

$$q_i^{\mathrm{sta}} = 1 - w_i, \qquad q_i^{\mathrm{con}} = w_i, \qquad q_i^{\mathrm{amb}} = 4 w_i (1 - w_i), \tag{7}$$

where q_i^{sta}, q_i^{con}, and q_i^{amb} continuous, non-exclusive state weights. A smaller w_i indicates temporally stable supervision, a larger value indicates accumulated conflict, and the ambiguity score reaches its maximum at $w_i = 0.5$. To optimize the target-adaptive model, we employ information maximization to encourage confident predictions while maintaining class diversity:

$$\mathcal{L}_{\mathrm{IM}} = \frac{1}{N_t} \sum_{i=1}^{N_t} H(\mathbf{p}_i^a) - H\left(\frac{1}{N_t} \sum_{i=1}^{N_t} \mathbf{p}_i^a\right), \tag{8}$$

where $H(\cdot)$ denotes the entropy of a class-probability distribution. We construct a soft target $\mathbf{t}_i = \frac{1}{2}(\tilde{\mathbf{p}}_i + \mathbf{p}_i^r)$ by combining the memory-based prediction with the reliability-weighted source prediction. The stable and conflicting state scores provide node-specific supervision weights:

$$\mathcal{L}_{\mathrm{sta}} = \frac{1}{N_t} \sum_{i=1}^{N_t} q_i^{\mathrm{sta}} \, \mathrm{CE}(\mathbf{p}_i^a, \tilde{y}_i), \quad \mathcal{L}_{\mathrm{con}} = \frac{1}{N_t} \sum_{i=1}^{N_t} q_i^{\mathrm{con}} \max_{c \in \mathcal{Y}} t_{i,c} H(\mathbf{t}_i, \mathbf{p}_i^a). \tag{9}$$

The term $\mathrm{CE}(\mathbf{p}_i^a, \tilde{y}_i)$ denotes hard-label cross-entropy, and $H(\mathbf{t}_i, \mathbf{p}_i^a) = -\sum_{c=1}^{C} t_{i,c} \log p_{i,c}^a$ denotes soft-label cross-entropy. The factor $\max_{c\in\mathcal{Y}} t_{i,c}$ represents the confidence of the soft target. The ambiguous score is used in dynamic expert allocation. Accumulating conflict over time reduces the influence of isolated prediction fluctuations on node-state estimation across successive adaptation steps. We analyze this temporal stability within a locally stationary adaptation interval.

Definition 3.2. *For target node v_i, let $\{d_i^{(t)}\}_{t\geq 1}$ denote its instantaneous disagreement across adaptation steps. Over a locally stationary interval, we assume $0 \leq d_i^{(t)} \leq 1$ and that the conditional first two moments remain constant: $\mathbb{E}[d_i^{(t)} \mid w_i^{(0)}] = \pi_i$ and $\mathrm{Var}(d_i^{(t)} \mid w_i^{(0)}) = \nu_i^2$. We further assume $\mathrm{Cov}(d_i^{(r)}, d_i^{(s)} \mid w_i^{(0)}) = 0$ for $r \neq s$ within this interval. Here, $\pi_i \in [0,1]$ and ν_i^2 denote the local mean and temporal variance of the disagreement during the considered adaptation interval.*

Theorem 3.2. *Under Definition 3.2, the aggregated disagreement $w_i^{(t)} = \rho w_i^{(t-1)} + (1-\rho) d_i^{(t)}$ with $0 \leq \rho < 1$ and $w_i^{(0)} \in [0,1]$ satisfies $\mathbb{E}[w_i^{(t)} \mid w_i^{(0)}] = \rho^t w_i^{(0)} + (1-\rho^t)\pi_i$, so its conditional mean converges to π_i as t increases. This convergence characterizes the long-term temporal behavior, and its conditional variance satisfies under the local stationarity assumption:*

$$\mathrm{Var}\left(w_i^{(t)} \mid w_i^{(0)}\right) = \frac{1-\rho}{1+\rho}\left(1-\rho^{2t}\right)\nu_i^2 \leq \frac{1-\rho}{1+\rho}\nu_i^2. \tag{10}$$

As t increases, the conditional variance converges to $\frac{1-\rho}{1+\rho}\nu_i^2$. For $0 < \rho < 1$ and $\nu_i^2 > 0$, this limiting variance is strictly smaller than the instantaneous disagreement variance.

Theorem 3.2 shows that temporal aggregation reduces disagreement variance and provides a more stable signal for characterizing target-node states. The proof is provided in Appendix B.2.

3.4 Reliability-aware Expert Allocation for Target Predictions

Structural stability monitoring captures node-specific source-model reliability, whereas temporal stability monitoring characterizes target-node states. The two signals vary across target nodes, making a fixed combination of source and target-adaptive models unsuitable for node-specific prediction. Therefore, we use these complementary stability signals for reliability-aware expert allocation, assigning node-specific weights to the source and target-adaptive models for final target prediction.

For each source model m, we track its historical agreement with the memory pseudo-label of node v_i. At adaptation step t, the current agreement $\eta_{i,m}^{(t)} \in [0,1]$ is measured from the view-averaged source prediction $\bar{\mathbf{p}}_{i,m}$ with respect to the current pseudo-label. We accumulate this agreement over time as $e_{i,m}^{(t)} = \rho_e e_{i,m}^{(t-1)} + (1-\rho_e)\eta_{i,m}^{(t)}$, where $\rho_e \in [0,1)$ controls the contribution of previous observations, and the first observation initializes the historical agreement. We write $e_{i,m}$ for its current value below. The quantity $u_i = \max(q_i^{\mathrm{con}}, q_i^{\mathrm{amb}})$ captures the degree of conflict or ambiguity in the current target supervision. We also min–max normalize the negative log-reliability $-\log r_{i,m}$ across source models to obtain $\widehat{D}_{i,m}^r \in [0,1]$, with $\varepsilon > 0$ used for numerical stability. We integrate the structural reliability and temporal state information for reliability-aware expert allocation as:

$$R_{i,m} = r_{i,m}\left[e_{i,m} + (1-e_{i,m})u_i\right], \quad D_{i,m} = q_i^{\mathrm{amb}}\left(1 + \widehat{D}_{i,m}^r\right), \tag{11}$$

where $R_{i,m} \in [0,1]$ combines source reliability with historical agreement, while $D_{i,m} \geq 0$ is an ambiguity-dependent compensation term that counterbalances reliability-induced source separation when target supervision is uncertain. For stable nodes, source ranking is governed primarily by reliability and source–pseudo agreement; as ambiguity increases, $D_{i,m}$ flattens this preference and retains alternative source hypotheses in the routing pool. We then compute reliability-aware scores for the source and target-adaptive models and normalize them with Sparsemax:

$$\mathbf{g}_i = \mathrm{Sparsemax}([s_{i,1}, \ldots, s_{i,M}, s_{i,a}]), \tag{12}$$

where $s_{i,m} = \log(R_{i,m}+\varepsilon) + D_{i,m} - H(\bar{\mathbf{p}}_{i,m})$ combines state-aware reliability, ambiguity compensation, and prediction uncertainty. The target-adaptive score is $s_{i,a} = \log(1 - \max_{1\leq m\leq M} R_{i,m} + \varepsilon) - \bar{D}_i + \bar{H}_i$, where $\bar{D}_i$ and $\bar{H}_i$ denote the mean ambiguity-compensation term and mean source entropy across the source models. The resulting $\mathbf{g}_i$ contains the node-specific allocation weights of

the M source models and the target-adaptive model. These weights combine their corresponding predictions to produce the final target-node prediction:

$$\mathbf{p}_i^{\text{final}} = \sum_{m=1}^{M} g_{i,m}\bar{\mathbf{p}}_{i,m} + g_{i,a}\mathbf{p}_i^a, \tag{13}$$

where $\mathbf{p}_i^{\text{final}} \in [0,1]^C$ is the final class distribution of node v_i. The expert allocation determines the contribution of each prediction without introducing an additional training loss.

3.5 Training Objective and Adaptation Procedure

We optimize the target-adaptive model through three complementary loss terms:

$$\mathcal{L} = \mathcal{L}_{\text{IM}} + \lambda_{\text{sta}}\mathcal{L}_{\text{sta}} + \lambda_{\text{con}}\mathcal{L}_{\text{con}}, \tag{14}$$

where $\mathcal{L}$ is the overall training objective, $\mathcal{L}_{\text{IM}}$ is the information-maximization loss, and $\mathcal{L}_{\text{sta}}$ and $\mathcal{L}_{\text{con}}$ provide stable-state and conflicting-state supervision. The coefficients $\lambda_{\text{sta}}, \lambda_{\text{con}} \geq 0$ control the two supervision terms. The complete adaptation procedure of `COME` is provided in Appendix A.

4 Experiments

4.1 Experimental Settings

Datasets. We evaluate `COME` on CSBM (Deshpande et al., 2018), Twitch (Rozemberczki et al., 2021), and Citation (Tang et al., 2008), covering synthetic, social-network, and citation graphs under 6 transfer settings. CSBM consists of four synthetic graphs generated by a four-class contextual stochastic block model with different structural distributions. Twitch contains six regional social networks, where nodes represent users, edges denote friendships, and node labels indicate whether users employ explicit language. Citation contains three research-paper citation networks collected from different platforms and time periods, with papers categorized into five research areas.

Baselines. We compare `COME` with 22 baselines, including MDAN (Zhao et al., 2018), M^{3}SDA (Peng et al., 2019), UDAGCN (Wu et al., 2020), GRADE (Wu et al., 2023), SpecReg (You et al., 2023), GCN (Kipf & Welling, 2017), SAGE (Hamilton et al., 2017), GAT (Veličković et al., 2017), GIN (Xu et al., 2018), SHOT (Liang et al., 2020), BNM (Cui et al., 2020), ATDOC (Liang et al., 2021), NRC (Yang et al., 2021), JMDS (Lee et al., 2022), GTRANS (Jin et al., 2022), SOGA (Mao et al., 2024), GraphCTA (Zhang et al., 2024), TPDS (Tang et al., 2024), DECISION (Ahmed et al., 2021), CAiDA (Dong et al., 2021), MSFDA (Shen et al., 2023), and GraphATA (Zhang & He, 2025). Detailed descriptions of the baselines are provided in Appendix C.2.

Implementation Settings. We use GCN as the default architecture with 128-dimensional node representations and optimize the model using Adam. All source models remain frozen and only target-side parameters are optimized; target labels are not used by the adaptation procedure. All results are reported as mean $\pm$ standard deviation over 5 runs under the same experimental protocol. Detailed implementation settings and configurations are provided in Appendix C.3.

4.2 Experimental Results

Performance Comparison. We compare `COME` with 22 baselines across 6 transfer settings on CSBM, Twitch, and Citation, with results reported in Table 1. Our method ranks first in all 6 settings and achieves the highest average accuracy of 76.21%. Based on these results, we make the following observations (Obs.): ***Obs.❶ Source-needed and no-adaptation methods remain behind `COME` across datasets.*** Among methods with complete results, the best average accuracies in these two groups are 67.20% and 63.84%, compared with 76.21% for our method. The gaps are clear on CSBM and Twitch. ***Obs.❷ The advantage persists over the single-source-free baselines.*** GraphCTA gives the highest average accuracy in this group at 71.37%, while our method reaches 76.21%. Across settings, our method exceeds the best single-source-free results by 2.11-9.08 percentage points, showing the benefit of coordinated multi-source adaptation over independently adapting and averaging source models. ***Obs.❸ Multi-source-free methods provide the closest comparison, with GraphATA as the strongest baseline.*** GraphATA achieves an average accuracy of 75.06%, compared with 76.21% for our method. Our method improves accuracy by 3.36 percentage

Table 1: Comparison of node classification accuracy (%) across six transfer settings on CSBM, Twitch, and Citation for four method groups (mean ± std over five runs, Avg. denotes the mean across six settings). Best and second-best results are bolded and underlined.

Method	CSBM	Twitch		Citation			Avg.
	C1,C2,C3→C4	R,P,F,ES→DE	R,P,F,ES→EN	A,C→D	C,D→A	A,D→C	
`Source-needed`							
MDAN	76.49 ± 0.39	61.78 ± 0.33	49.52 ± 0.35	70.38 ± 0.17	65.31 ± 0.09	73.92 ± 0.13	66.23
M^{3}SDA	OOM	59.75 ± 0.64	53.71 ± 0.65	70.22 ± 0.16	64.71 ± 0.33	71.30 ± 0.16	–
UDAGCN	72.18 ± 0.12	43.34 ± 0.30	48.88 ± 0.22	70.54 ± 0.04	61.67 ± 0.01	69.72 ± 0.03	61.06
GRADE	78.12 ± 0.21	53.52 ± 0.07	46.80 ± 0.04	74.37 ± 0.49	70.92 ± 0.04	79.48 ± 0.05	67.20
SpecReg	79.14 ± 0.38	42.85 ± 2.72	47.21 ± 1.34	76.42 ± 0.66	72.69 ± 0.51	78.67 ± 2.46	66.16
`No-adaptation`							
GCN	73.18 ± 0.45	53.57 ± 0.49	47.76 ± 0.11	70.92 ± 0.15	64.64 ± 0.12	72.98 ± 0.18	63.84
SAGE	76.91 ± 0.78	42.24 ± 0.43	45.89 ± 0.15	68.92 ± 0.28	61.25 ± 0.38	67.77 ± 0.17	60.50
GAT	71.59 ± 0.36	39.56 ± 0.32	45.44 ± 0.93	63.85 ± 1.08	56.12 ± 1.51	62.73 ± 1.09	56.55
GIN	76.92 ± 0.25	40.75 ± 0.61	45.43 ± 0.26	67.02 ± 0.36	59.50 ± 0.25	70.34 ± 0.23	59.99
`Single-source-free`							
SHOT	83.82 ± 0.25	64.01 ± 0.09	57.97 ± 0.07	75.34 ± 0.13	68.71 ± 0.41	77.92 ± 0.05	71.30
BNM	81.79 ± 0.47	64.11 ± 0.23	57.79 ± 0.20	75.17 ± 0.16	68.81 ± 0.12	77.38 ± 0.11	70.84
ATDOC	64.72 ± 0.33	57.92 ± 0.97	50.94 ± 2.80	72.67 ± 0.63	65.52 ± 1.16	76.12 ± 0.29	64.65
NRC	84.42 ± 0.16	63.76 ± 0.09	57.66 ± 0.05	73.13 ± 0.08	69.52 ± 0.25	77.51 ± 0.22	71.00
JMDS	63.74 ± 0.39	64.00 ± 0.05	46.78 ± 0.17	72.21 ± 0.37	65.26 ± 0.18	74.34 ± 0.29	64.39
GTRANS	64.81 ± 0.89	62.54 ± 0.02	57.29 ± 0.07	73.67 ± 0.65	69.59 ± 2.06	78.74 ± 0.43	67.77
SOGA	65.50 ± 0.55	58.24 ± 0.85	46.98 ± 0.48	73.38 ± 0.17	66.96 ± 0.33	78.06 ± 0.23	64.85
GraphCTA	83.21 ± 0.43	62.70 ± 0.24	56.28 ± 0.16	75.53 ± 0.30	70.91 ± 0.63	79.59 ± 0.22	71.37
TPDS	84.03 ± 0.23	63.14 ± 0.10	57.83 ± 0.14	71.63 ± 0.12	68.45 ± 0.23	75.87 ± 0.21	70.16
`Multi-source-free`							
DECISION	88.02 ± 0.28	63.92 ± 0.26	57.92 ± 0.12	76.02 ± 0.67	70.40 ± 0.31	79.39 ± 0.35	72.61
CAiDA	87.96 ± 0.94	63.73 ± 0.49	57.73 ± 0.45	75.75 ± 0.42	70.25 ± 0.54	79.70 ± 0.74	72.52
MSFDA	88.94 ± 0.62	54.42 ± 5.50	55.77 ± 0.58	76.03 ± 1.27	68.60 ± 2.94	78.63 ± 0.92	70.40
GraphATA	90.14 ± 0.36	66.71 ± 0.39	59.56 ± 0.14	78.45 ± 0.87	73.17 ± 0.52	82.33 ± 0.75	75.06
`COME`	**93.50** ± 0.80	**68.24** ± 0.07	**60.08** ± 0.18	**78.77** ± 0.19	**73.86** ± 0.12	**82.82** ± 0.27	**76.21**

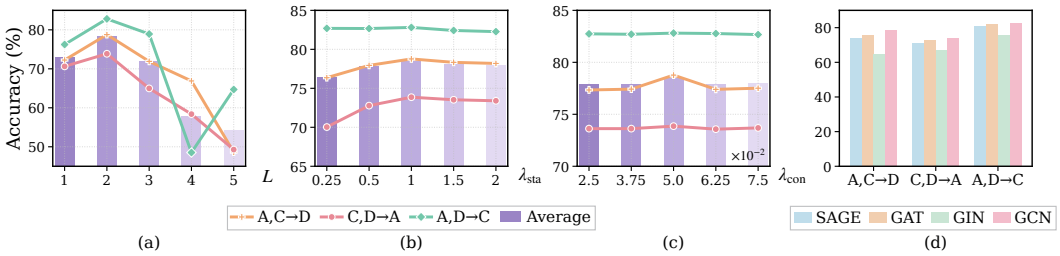


Figure 2: (a)-(c) Sensitivity analysis of L, λ_{sta}, and λ_{con} on Citation transfer settings using node classification accuracy (%). (d) Comparison of different GNN backbones using the same metric.

points on CSBM, 1.53 percentage points and 0.52 percentage points on two Twitch settings, and 0.32 percentage points, 0.69 percentage points, and 0.49 percentage points on three Citation settings.

Ablation Study. We conduct ablation experiments on SMEP, TMAC, and RETP across all three benchmark datasets, corresponding to structural stability monitoring from edge perturbation, temporal stability monitoring from accumulated conflict, and reliability-aware expert allocation for target predictions, as shown in Table 2. w/o All removes all three designs, while the other variants remove one at a time. `COME` achieves the best performance across all 6 settings, with an average accuracy of 76.21%. From the results, we observe that: ***Obs.❹ SMEP provides node-specific source reliability from structural responses.*** Removing SMEP reduces the average accuracy to 73.88%, with the largest drop on CSBM. This result shows that the structural-monitoring module provides useful node-wise information for source weighting. ***Obs.❺ TMAC yields the largest module-level contribution.*** Without TMAC, the average accuracy falls to 61.43%, with substantial drops on CSBM and both Twitch settings. The large decrease indicates that temporal prediction information is important for characterizing target-node states during adaptation. ***Obs.❻ RETP integrates reliability information for expert allocation.*** Removing RETP lowers the average accuracy to 73.57%, as source

Table 2: Ablation results (%) across six transfer settings on CSBM, Twitch, and Citation (mean ± std over five runs, Avg. denotes the mean across six settings). Best results are bolded.

Method	CSBM	Twitch		Citation			Avg.
	C1,C2,C3→C4	R,P,F,ES→DE	R,P,F,ES→EN	A,C→D	C,D→A	A,D→C	
COME w/o All	73.23 ± 0.98	48.16 ± 0.46	45.59 ± 0.02	70.78 ± 0.26	64.45 ± 0.28	72.94 ± 0.19	62.53
COME w/o SMEP	85.29 ± 14.82	68.10 ± 0.13	59.15 ± 0.08	77.20 ± 0.70	70.82 ± 0.16	82.72 ± 0.18	73.88
COME w/o TMAC	74.03 ± 0.16	45.06 ± 0.07	45.77 ± 0.02	69.64 ± 0.06	62.99 ± 0.07	71.06 ± 0.17	61.43
COME w/o RETP	81.06 ± 18.00	68.02 ± 0.15	59.12 ± 0.12	77.12 ± 0.61	73.37 ± 0.12	82.71 ± 0.26	73.57
COME	**93.50** ± 0.80	**68.24** ± 0.07	**60.08** ± 0.18	**78.77** ± 0.19	**73.86** ± 0.12	**82.82** ± 0.27	**76.21**

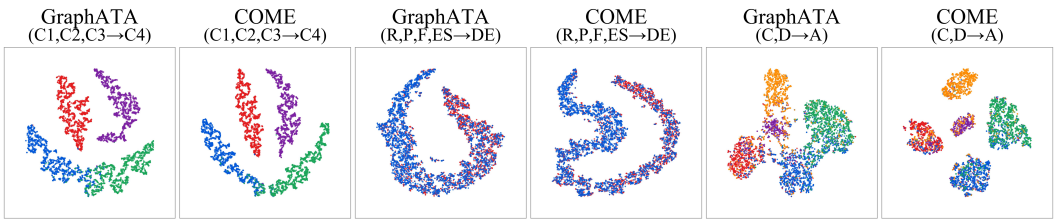


Figure 3: Visualization results of target-node "representations learned by GraphATA and COME on three transfer settings. Different colors denote different node classes.

reliability and temporal node states are no longer jointly used to determine node-specific expert contributions. Removing all three designs yields 62.53%, versus 76.21% for full COME.

Sensitivity Study. We then study the sensitivity to the number of graph layers L, stable-state loss weight λ_{sta}, and conflicting-state loss weight λ_{con} on the three Citation transfers, as shown in Figures 2(a)-(c). ***Obs.❼ COME is robust to the loss-weight choices, with stable performance around the selected values.*** For L, all three transfers achieve their best results at $L = 2$, with the average accuracy increasing from 73.04% at $L = 1$ to 78.48%. Further increasing L reduces the average accuracy to 71.94%, 57.93%, and 54.20% at $L = 3$, 4, and 5. For λ_{sta}, the average accuracy rises from 76.36% at 0.25 to 78.48% at 1 and then decreases with larger values. For λ_{con}, the average accuracy remains within 77.90%–78.48% and reaches its maximum at 0.05. We therefore use $L = 2$, $\lambda_{\text{sta}} = 1$, and $\lambda_{\text{con}} = 0.05$. Detailed sensitivity results are provided in Appendix D.1.

Architecture Study. We further compare SAGE, GAT, GIN, and GCN on the same three Citation transfers in Figure 2(d). ***Obs.❽ COME adapts well to different GNN architectures, with GCN consistently achieving the best performance.*** GCN achieves the highest accuracy on all three transfers, reaching 78.77%, 73.86%, and 82.82%, with an average accuracy of 78.48%. GAT and SAGE remain relatively close to GCN on Citation, especially on A,D→C, where they reach 81.84% and 80.81%. In contrast, GIN shows a larger drop across all three transfers and obtains an average accuracy of 69.18%. The average accuracies of GAT and SAGE are 76.63% and 75.36%, respectively, both below GCN. Therefore, we use GCN as the default architecture in COME for all main experiments. Detailed architecture results across all 6 transfer settings are provided in Appendix D.2.

Representation Visualization. We further compare target-node representations from GraphATA and COME on representative transfers from CSBM, Twitch, and Citation in Figure 3. The results suggest the following observation: ***Obs.❾ COME generally produces more compact and better-separated target representations than GraphATA.*** On C1,C2,C3→C4, our method forms more compact class clusters with wider gaps between classes, whereas GraphATA produces more elongated and closely connected regions. On R,P,F,ES→DE, the two classes remain overlapped under both methods, but COME produces fewer scattered regions and a more coherent local structure. On C,D→A, our method reduces cross-class mixing and forms localized class regions with clearer boundaries. Across the three datasets, the improvement is more evident on CSBM and Citation, while Twitch remains more challenging. More visualization results are provided in Appendix D.3.

5 Conclusion

This paper presents COME, a framework for multi-source-free graph domain adaptation. COME monitors structural stability through edge perturbations and temporal stability through accumulated prediction conflict to characterize source reliability and target-node states. It integrates complementary reliability signals for node-specific expert allocation between source and target-adaptive models for target prediction. Experiments on graph benchmarks demonstrate improvements over existing methods, while ablation and visualization results validate the effectiveness of the proposed designs.

AI USE STATEMENT

In this work, we used generative AI tools to assist with code development and language editing, including grammar correction, limited rephrasing, and improvements to readability. All AI-assisted code was manually reviewed and tested, and all language edits were reviewed by the authors. We take full responsibility for the final content of this work.

ETHICS STATEMENT

This study uses only publicly available benchmark datasets and does not involve new data collection or direct interaction with human subjects. We are not aware of any specific ethical concerns arising from the methodology or experiments presented in this work. The authors declare no conflicts of interest.

REPRODUCIBILITY STATEMENT

To support reproducibility, we provide the download links for all publicly available datasets used in our experiments, together with detailed data preprocessing instructions. The model architecture, training settings, hyperparameters, and experimental protocols are described in the main paper and appendix. The complete implementation is publicly available at https://anonymous.4open.science/r/iclr2027-COME-code-8F4A/.

REFERENCES

Sk Miraj Ahmed, Dripta S Raychaudhuri, Sujoy Paul, Samet Oymak, and Amit K Roy-Chowdhury. Unsupervised multi-source domain adaptation without access to source data. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10098–10107. IEEE, 2021.

Weilin Cai, Juyong Jiang, Fan Wang, Jing Tang, Sunghun Kim, and Jiayi Huang. A survey on mixture of experts in large language models. *IEEE Transactions on Knowledge and Data Engineering*, 37:3896–3915, 2025.

Shuhao Cui, Shuhui Wang, Junbao Zhuo, Liang Li, Qingming Huang, and Qi Tian. Towards discriminability and diversity: Batch nuclear-norm maximization under label insufficient situations. In *2020 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 3940–3949. IEEE, 2020.

Damai Dai, Chengqi Deng, Chenggang Zhao, RX Xu, Huazuo Gao, Deli Chen, Jiashi Li, Wangding Zeng, Xingkai Yu, Yu Wu, et al. Deepseekmoe: Towards ultimate expert specialization in mixture-of-experts language models. In *Proceedings of the 62nd annual meeting of the association for computational linguistics (volume 1: Long papers)*, pp. 1280–1297, 2024.

Yash Deshpande, Subhabrata Sen, Andrea Montanari, and Elchanan Mossel. Contextual stochastic block models. *Advances in neural information processing systems*, 31, 2018.

Jiahua Dong, Zhen Fang, Anjin Liu, Gan Sun, and Tongliang Liu. Confident anchor-induced multi-source free domain adaptation. *Advances in neural information processing systems*, 34:2848–2860, 2021.

Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017.

Zhongyi Han, Zhiyan Zhang, Fan Wang, Rundong He, Wan Su, Xiaoming Xi, and Yilong Yin. Discriminability and transferability estimation: a bayesian source importance estimation approach for multi-source-free domain adaptation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pp. 7811–7820, 2023.

Tianyi Jiang, Qiang Yao, Zeyu Wang, Xiaoze Bao, Shanqing Yu, and Qi Xuan. Expert-guided substructure information bottleneck for molecular property prediction. *Journal of Chemical Information and Modeling*, 65:7887–7900, 2025.

Wei Jin, Tong Zhao, Jiayuan Ding, Yozen Liu, Jiliang Tang, and Neil Shah. Empowering graph representation learning with test-time graph transformation. *arXiv preprint arXiv:2210.03561*, 2022.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017.

Jonghyun Lee, Dahuin Jung, Junho Yim, and Sungroh Yoon. Confidence score for source-free unsupervised domain adaptation. In *International conference on machine learning*, pp. 12365–12377. PMLR, 2022.

Bingxin Li, Ying Li, and Shihui Ying. Cross-evaluation and re-weighting for multi-source-free domain adaptation. In *2024 IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1–6. IEEE, 2024a.

Xinyao Li, Jingjing Li, Fengling Li, Lei Zhu, and Ke Lu. Agile multi-source-free domain adaptation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 13673–13681, 2024b.

Jian Liang, Dapeng Hu, and Jiashi Feng. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In *International conference on machine learning*, pp. 6028–6039. PMLR, 2020.

Jian Liang, Dapeng Hu, and Jiashi Feng. Domain adaptation with auxiliary target domain-oriented classifier. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16627–16637. IEEE, 2021.

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

Shikun Liu, Deyu Zou, Han Zhao, and Pan Li. Pairwise alignment improves graph domain adaptation. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 32552–32575, 2024b.

Zheyuan Liu, Chunhui Zhang, Yijun Tian, Erchi Zhang, Chao Huang, Yanfang Ye, and Chuxu Zhang. Fair graph representation learning via diverse mixture-of-experts. In *Proceedings of the ACM web conference 2023*, pp. 28–38, 2023.

Ziyi Liu, Chaoran Cui, Chunyun Zhang, Fan'an Meng, Shuai Gong, Muzhi Xi, and Lei Li. Consistency-guided multi-source-free domain adaptation. *Engineering Applications of Artificial Intelligence*, 139:109497, 2025.

Haitao Mao, Lun Du, Yujia Zheng, Qiang Fu, Zelin Li, Xu Chen, Shi Han, and Dongmei Zhang. Source free graph unsupervised domain adaptation. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, pp. 520–528, 2024.

Taha Mustapha Nehdi, Nairouz Mrabah, Atif Belal, Marco Pedersoli, and Eric Granger. Low-rank expert merging for multi-source domain adaptation in person re-identification. In *2026 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 1809–1819. IEEE, 2026.

Jiangbo Pei, Aidong Men, Yang Liu, Xiahai Zhuang, and Qingchao Chen. Evidential multi-source-free unsupervised domain adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46:5288–5305, 2024.

Jiangbo Pei, Ruizhe Li, Aidong Men, Yang Liu, Xiahai Zhuang, and Qingchao Chen. Selection, ensemble, and adaptation: Advancing multi-source-free domain adaptation via architecture zoo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1406–1415. IEEE, 2019.

Benedek Rozemberczki, Carl Allen, and Rik Sarkar. Multi-scale attributed node embedding. *Journal of Complex Networks*, 9:cnab014, 2021.

Maohao Shen, Yuheng Bu, and Gregory W Wornell. On balancing bias and variance in unsupervised multi-source-free domain adaptation. In *International conference on machine learning*, pp. 30976–30991. PMLR, 2023.

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*, pp. 990–998, 2008.

Song Tang, An Chang, Fabian Zhang, Xiatian Zhu, Mao Ye, and Changshui Zhang. Source-free domain adaptation via target prediction distribution searching. *International journal of computer vision*, 132:654–672, 2024.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

Haotao Wang, Ziyu Jiang, Yuning You, Yan Han, Gaowen Liu, Jayanth Srinivasa, Ramana Kompella, Zhangyang Wang, et al. Graph mixture of experts: Learning on large-scale graphs with explicit diversity modeling. *Advances in Neural Information Processing Systems*, 36:50825–50837, 2023.

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

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

Zhitian Xie, Yinger Zhang, Chenyi Zhuang, Qitao Shi, Zhining Liu, Jinjie Gu, and Guannan Zhang. Mode: A mixture-of-experts model with mutual distillation among the experts. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 16067–16075, 2024.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*, 2018.

Shiqi Yang, Joost Van de Weijer, Luis Herranz, Shangling Jui, et al. Exploiting the intrinsic neighborhood structure for source-free domain adaptation. *Advances in neural information processing systems*, 34:29393–29405, 2021.

Hanrong Ye and Dan Xu. Taskexpert: Dynamically assembling multi-task representations with memorial mixture-of-experts. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 21771–21780. IEEE, 2023.

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

Qi Yuan, Yang Liu, Yateng Tang, Xinhuan Chen, Xuehao Zheng, Qing He, and Xiang Ao. Dynamic graph learning with static relations for credit risk assessment. In *Proceedings of the AAAI conference on artificial intelligence*, volume 39, pp. 13133–13141, 2025.

Jiahao Zhang, Yilong Wang, Zhiwei Zhang, Xiaorui Liu, and Suhang Wang. Unlearning inversion attacks for graph neural networks. In *Proceedings of the Nineteenth ACM International Conference on Web Search and Data Mining*, pp. 934–945, 2026.

Kuan Zhang, Chengliang Chai, Jingzhe Xu, Chi Zhang, Han Han, Ye Yuan, Guoren Wang, and Lei Cao. Handling label noise via instance-level difficulty modeling and dynamic optimization. In *Advances in Neural Information Processing Systems*, volume 38, pp. 46667–46696, 2025.

Zhen Zhang and Bingsheng He. Aggregate to adapt: Node-centric aggregation for multi-source-free graph domain adaptation. In *Proceedings of the ACM on Web Conference 2025*, pp. 4420–4431, 2025.

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

Han Zhao, Shanghang Zhang, Guanhang Wu, José MF Moura, Joao P Costeira, and Geoffrey J Gordon. Adversarial multiple source domain adaptation. *Advances in neural information processing systems*, 31, 2018.

Sicheng Zhao, Huizai Yao, Chuang Lin, Yue Gao, and Guiguang Ding. Multi-source-free domain adaptive object detection. *International Journal of Computer Vision*, 132:5950–5982, 2024.

Yanqi Zhou, Tao Lei, Hanxiao Liu, Nan Du, Yanping Huang, Vincent Zhao, Andrew M Dai, Quoc V Le, James Laudon, et al. Mixture-of-experts with expert choice routing. *Advances in Neural Information Processing Systems*, 35:7103–7114, 2022.

A ALGORITHM

The complete adaptation procedure of COME is summarized in Algorithm 1. It consists of structural stability monitoring from edge perturbation, temporal stability monitoring from accumulated conflict, and reliability-aware expert allocation for target predictions.

Algorithm 1 Adaptation Procedure of COME

Require: Unlabeled target graph $\mathcal{G}_t = (\mathcal{V}_t, \mathcal{E}_t, \mathbf{X}_t)$; frozen source models $\mathcal{F}_s = \{f_m\}_{m=1}^{M}$; number of perturbed views K; number of memory neighbors K_{nn}; number of graph layers L; adaptation epochs T; coefficients ρ, ρ_e, λ_g, λ_{sta}, and λ_{con}

Ensure: Final node predictions $\{\hat{y}_i\}_{i=1}^{N_t}$

Initialize the target-adaptive model f_a and memory bank $\mathcal{B}$
Part I: Structural Stability Monitoring from Edge Perturbation
Construct the original target view $\mathbf{A}^{(0)} = \mathbf{A}_t$ and K edge-perturbed views $\{\mathbf{A}^{(k)}\}_{k=1}^{K}$
for $m = 1, \ldots, M$ **do**
    Obtain source predictions $\{\mathbf{p}_{i,m}^{(k)}\}_{k=0}^{K}$ across all structural views
    Compute node-specific source reliability $r_{i,m}$ from cross-view prediction stability (Eq. 1)
    Compute the view-averaged source prediction $\bar{\mathbf{p}}_{i,m}$
end for
Compute the reliability-weighted source prediction $\mathbf{p}_i^r$ (Eq. 2)
Part II: Temporal Stability Monitoring from Accumulated Conflict
for $t = 1, \ldots, T$ **do**
    Obtain the current target-adaptive prediction $\mathbf{p}_i^{(t)} = \mathbf{p}_i^a$ (Eqs. 4 and 5)
    Retrieve K_{nn} neighbors from $\mathcal{B}$ to obtain $\tilde{\mathbf{p}}_i^{(t)}$ and $\tilde{y}_i^{(t)}$
    Compute the current disagreement $d_i^{(t)}$ and update the accumulated conflict score $w_i^{(t)}$ (Eq. 6)
    Compute stable, conflicting, and ambiguous node-state scores (Eq. 7)
    Update the historical source–pseudo agreement $e_{i,m}^{(t)}$ for all source models
    Compute $\mathcal{L}_{\text{IM}}$, $\mathcal{L}_{\text{sta}}$, and $\mathcal{L}_{\text{con}}$ (Eqs. 8 and 9)
    Compute the overall objective $\mathcal{L}$ (Eq. 14)
    Update the trainable parameters θ_a by minimizing $\mathcal{L}$
    Update memory bank $\mathcal{B}$ with the current target features and predictions
end for
Part III: Reliability-aware Expert Allocation for Target Predictions
Integrate source reliability and temporal node states to compute $R_{i,m}$ and ambiguity-compensation term $D_{i,m}$ (Eq. 11)
Obtain node-specific expert allocation weights $\mathbf{g}_i$ (Eq. 12)
Compute the final target-node distribution $\mathbf{p}_i^{\text{final}}$ (Eq. 13)
Predict $\hat{y}_i = \arg\max_{c \in \mathcal{Y}} p_{i,c}^{\text{final}}$
return $\{\hat{y}_i\}_{i=1}^{N_t}$

B THEORETICAL PROOFS

B.1 PROOF OF THEOREM 3.1

Proof. Fix a source model $m \in \{1, \ldots, M\}$ and a perturbed view $k \in \{1, \ldots, K\}$. For brevity, let $\mathbf{P}_0 = \mathbf{P}(\mathbf{A}^{(0)})$ and $\mathbf{P}_k = \mathbf{P}(\mathbf{A}^{(k)})$. Definition 3.1 gives $\|\mathbf{P}_k - \mathbf{P}_0\|_2 \le \epsilon_A$, while the theorem assumptions give $\|\mathbf{P}_0\|_2 \le 1$ and $\|\mathbf{P}_k\|_2 \le 1$. Let $\mathbf{H}_{m,0}^{(\ell)}$ and $\mathbf{H}_{m,k}^{(\ell)}$ denote the hidden representations after ℓ message-passing layers on the original and perturbed views. Since both views use the same node features, $\mathbf{H}_{m,0}^{(0)} = \mathbf{H}_{m,k}^{(0)} = \mathbf{X}_t$. Define $\Delta_{m,\ell}^{(k)} = \|\mathbf{H}_{m,k}^{(\ell)} - \mathbf{H}_{m,0}^{(\ell)}\|_F$, so $\Delta_{m,0}^{(k)} = 0$.

We first bound the hidden representations on the original view. Since φ is 1-Lipschitz with respect to the Frobenius norm and satisfies $\varphi(0) = 0$, $\|\mathbf{P}_0\|_2 \le 1$, and $\|\mathbf{W}_m^{(\ell)}\|_2 \le B_{m,\ell}$, the inequality $\|\mathbf{UVW}\|_F \le \|\mathbf{U}\|_2 \|\mathbf{V}\|_F \|\mathbf{W}\|_2$ gives $\|\mathbf{H}_{m,0}^{(\ell+1)}\|_F \le B_{m,\ell} \|\mathbf{H}_{m,0}^{(\ell)}\|_F$. Starting from $\mathbf{H}_{m,0}^{(0)} = \mathbf{X}_t$

and applying this bound recursively yields

$$\left\|\mathbf{H}_{m,0}^{(\ell)}\right\|_F \le \|\mathbf{X}_t\|_F \prod_{j=0}^{\ell-1} B_{m,j}, \qquad \ell = 0, \dots, L, \tag{15}$$

where the product is defined as one for $\ell = 0$.

We next consider the difference between the two forward passes. Using the encoder update and the 1-Lipschitz property of φ, then adding and subtracting $\mathbf{P}_k \mathbf{H}_{m,0}^{(\ell)} \mathbf{W}_m^{(\ell)}$, gives

$$\begin{aligned} \Delta_{m,\ell+1}^{(k)} &\le \left\|\mathbf{P}_k\big(\mathbf{H}_{m,k}^{(\ell)} - \mathbf{H}_{m,0}^{(\ell)}\big)\mathbf{W}_m^{(\ell)}\right\|_F + \left\|(\mathbf{P}_k - \mathbf{P}_0)\mathbf{H}_{m,0}^{(\ell)}\mathbf{W}_m^{(\ell)}\right\|_F \\ &\le B_{m,\ell}\Delta_{m,\ell}^{(k)} + \epsilon_A B_{m,\ell}\left\|\mathbf{H}_{m,0}^{(\ell)}\right\|_F. \end{aligned} \tag{16}$$

Combining this recursion with Eq. 15 and $\Delta_{m,0}^{(k)} = 0$ gives, by induction,

$$\Delta_{m,\ell}^{(k)} \le \ell\epsilon_A \|\mathbf{X}_t\|_F \prod_{j=0}^{\ell-1} B_{m,j}, \qquad \ell = 0, \dots, L. \tag{17}$$

The claim is immediate for $\ell = 0$. If it holds at layer ℓ, substituting the induction hypothesis and Eq. 15 into the recursion gives $\Delta_{m,\ell+1}^{(k)} \le (\ell+1)\epsilon_A \|\mathbf{X}_t\|_F \prod_{j=0}^{\ell} B_{m,j}$, which completes the induction. In particular, $\Delta_{m,L}^{(k)} \le L\epsilon_A \|\mathbf{X}_t\|_F \prod_{\ell=0}^{L-1} B_{m,\ell}$.

For brevity, let $\zeta_m = C_m \epsilon_A$ denote the prediction-deviation bound. Since $f_m = \psi_m \circ \phi_m$ and ψ_m is $B_{m,\psi}$-Lipschitz, the encoder bound gives

$$\begin{aligned} \left\|f_m(\mathbf{X}_t, \mathbf{A}^{(k)}) - f_m(\mathbf{X}_t, \mathbf{A}^{(0)})\right\|_F &\le B_{m,\psi}\Delta_{m,L}^{(k)} \\ &\le B_{m,\psi} L\epsilon_A \|\mathbf{X}_t\|_F \prod_{\ell=0}^{L-1} B_{m,\ell} = C_m\epsilon_A = \zeta_m. \end{aligned} \tag{18}$$

For node v_i, $\mathbf{p}_{i,m}^{(k)}$ is one row of the prediction matrix, so $\|\mathbf{p}_{i,m}^{(k)} - \mathbf{p}_{i,m}^{(0)}\|_2 \le \zeta_m$. Consequently, each class probability satisfies $|p_{i,m,c}^{(k)} - p_{i,m,c}^{(0)}| \le \zeta_m$. For the confidence $c_{i,m}^{(k)} = \max_{c\in\mathcal{Y}} p_{i,m,c}^{(k)}$, the inequality $|\max_c x_c - \max_c y_c| \le \|\mathbf{x} - \mathbf{y}\|_\infty$ gives $|c_{i,m}^{(k)} - c_{i,m}^{(0)}| \le \zeta_m$.

Recall that $\mu_{i,m} = (K+1)^{-1}\sum_{k=0}^{K} c_{i,m}^{(k)}$ and $\delta_{i,m} = c_{i,m}^{(0)} - \mu_{i,m}$. It follows that $|\delta_{i,m}| \le (K+1)^{-1}\sum_{k=1}^{K} |c_{i,m}^{(0)} - c_{i,m}^{(k)}| \le \zeta_m$. Since $\mu_{i,m}$ minimizes the sum of squared distances among the $K+1$ confidence values, $\sigma_{i,m}^2 \le (K+1)^{-1}\sum_{k=0}^{K}(c_{i,m}^{(k)} - c_{i,m}^{(0)})^2 \le \zeta_m^2$. Therefore, $\sigma_{i,m} \le \zeta_m$ and $|\delta_{i,m}| \le \zeta_m$.

Finally, let $\hat{y}_{i,m}^{(0)}$ be the original-view predicted class and let $\Gamma_{i,m}$ be its probability margin defined in Theorem 3.1. For any competing class $c \ne \hat{y}_{i,m}^{(0)}$, each of the two class probabilities can change by at most ζ_m, which gives

$$p_{i,m,\hat{y}_{i,m}^{(0)}}^{(k)} - p_{i,m,c}^{(k)} \ge \Gamma_{i,m} - 2\zeta_m. \tag{19}$$

Since $\zeta_m = C_m\epsilon_A$, the condition $\Gamma_{i,m} > 2C_m\epsilon_A$ is equivalent to $\Gamma_{i,m} > 2\zeta_m$. Under this condition, the right-hand side is positive for every competing class and every perturbed view. Hence every view predicts $\hat{y}_{i,m}^{(0)}$, and the class-consistency score satisfies $\kappa_{i,m} = 1$. This completes the proof. □

B.2 Proof of Theorem 3.2

Proof. Fix a target node v_i and condition on $w_i^{(0)}$. Repeatedly substituting $w_i^{(t)} = \rho w_i^{(t-1)} + (1-\rho)d_i^{(t)}$ gives

$$w_i^{(t)} = \rho^t w_i^{(0)} + (1-\rho)\sum_{s=1}^{t} \rho^{t-s} d_i^{(s)}. \tag{20}$$

The expansion follows recursively from the temporal update. Its coefficients are non-negative and satisfy $\rho^t + (1-\rho)\sum_{s=1}^{t}\rho^{t-s} = 1$. Since $w_i^{(0)} \in [0,1]$ and $d_i^{(s)} \in [0,1]$, the aggregated disagreement $w_i^{(t)}$ also remains in $[0,1]$.

By Definition 3.2, $\mathbb{E}[d_i^{(s)} \mid w_i^{(0)}] = \pi_i$ for every s. Taking the conditional expectation of Eq. 20 and using $\sum_{j=0}^{t-1}\rho^j = (1-\rho^t)/(1-\rho)$ yields

$$\mathbb{E}[w_i^{(t)} \mid w_i^{(0)}] = \rho^t w_i^{(0)} + (1-\rho^t)\pi_i.$$

Since $0 \le \rho < 1$, the conditional mean converges to π_i as $t \to \infty$.

For the conditional variance, $w_i^{(0)}$ is fixed after conditioning. Definition 3.2 assumes $\mathrm{Cov}(d_i^{(r)}, d_i^{(s)} \mid w_i^{(0)}) = 0$ for $r \neq s$, so the cross-covariance terms vanish. Using $\mathrm{Var}(d_i^{(s)} \mid w_i^{(0)}) = \nu_i^2$ and $\sum_{j=0}^{t-1}\rho^{2j} = (1-\rho^{2t})/(1-\rho^2)$, we obtain

$$\begin{aligned}
\mathrm{Var}\left(w_i^{(t)} \mid w_i^{(0)}\right) &= (1-\rho)^2 \sum_{s=1}^{t} \rho^{2(t-s)}\,\mathrm{Var}\left(d_i^{(s)} \mid w_i^{(0)}\right) \\
&= (1-\rho)^2\nu_i^2 \sum_{j=0}^{t-1}\rho^{2j} \\
&= \frac{1-\rho}{1+\rho}\left(1-\rho^{2t}\right)\nu_i^2 \le \frac{1-\rho}{1+\rho}\nu_i^2.
\end{aligned} \tag{21}$$

The last inequality follows from $0 \le 1-\rho^{2t} \le 1$. As $t \to \infty$, the conditional variance converges to $\frac{1-\rho}{1+\rho}\nu_i^2$. For $0 < \rho < 1$, $\frac{1-\rho}{1+\rho} < 1$, so the limiting variance is strictly smaller than ν_i^2 whenever $\nu_i^2 > 0$.

As an additional consequence, consider two disagreement sequences with the same initial value and identical observations except at step $s \le t$. Let the corresponding aggregated disagreements be $w_i^{(t)}$ and $w_i'^{(t)}$. From Eq. 20,

$$\left|w_i^{(t)} - w_i'^{(t)}\right| = (1-\rho)\rho^{t-s}\left|d_i^{(s)} - d_i'^{(s)}\right| \le (1-\rho)\rho^{t-s}. \tag{22}$$

Thus, the effect of an isolated disagreement decreases geometrically over subsequent adaptation steps.

The stable and conflicting state scores, $q_i^{\mathrm{sta}} = 1 - w_i$ and $q_i^{\mathrm{con}} = w_i$, are 1-Lipschitz functions of w_i. Therefore,

$$|q_i^{\mathrm{sta}} - q_i'^{\mathrm{sta}}| \le (1-\rho)\rho^{t-s}, \qquad |q_i^{\mathrm{con}} - q_i'^{\mathrm{con}}| \le (1-\rho)\rho^{t-s}.$$

For the ambiguous score, $q_i^{\mathrm{amb}} = 4w_i(1-w_i)$. Its derivative satisfies $\max_{w\in[0,1]}|4-8w| = 4$, so

$$|q_i^{\mathrm{amb}} - q_i'^{\mathrm{amb}}| \le 4(1-\rho)\rho^{t-s}.$$

Hence, the influence of an isolated disagreement also decreases over time in all three node-state scores. This completes the proof. □

Together, the two results formalize the monitoring properties used by `COME`: bounded cross-view variation under structural perturbation and variance reduction through temporal accumulation.

Table 3: Statistics of the datasets used in our experiments.

Statistics	CSBM	Twitch	Citation
Nodes	8,000	1,912–9,498	5,484–9,360
Edges	607,699–752,776	31,299–153,138	8,117–15,556
Features	128	3,170	6,775
Classes	4	2	5

C Detailed Experimental Settings

C.1 Dataset Details

We use three node-classification datasets, including synthetic graphs, social networks, and citation networks. The details of each dataset are provided below.

CSBM (Deshpande et al., 2018). CSBM contains four synthetic graphs, denoted as C1–C4, generated from a four-class contextual stochastic block model. Each synthetic graph contains 2,000 nodes per class. The intra-class connection probability is fixed at $p = 0.04$, while the inter-class connection probability is set to $q \in \{0.012, 0.014, 0.016, 0.018\}$ for C1–C4, respectively. Each node is represented by a 128-dimensional feature vector sampled from a class-dependent multivariate Gaussian distribution with identity covariance. The class-wise means are set to -2, $-2/3$, $2/3$, and 2, respectively.

Twitch (Rozemberczki et al., 2021). Twitch consists of six regional social networks from Germany (DE), England (EN), Spain (ES), France (F), Portugal (P), and Russia (R). Nodes represent users and edges represent friendships. Each node is described by a 3,170-dimensional binary feature vector containing information about gaming activities, user preferences, geographic information, and streaming habits. The node labels indicate whether users employ explicit language, resulting in a binary node classification task.

Citation (Tang et al., 2008). The citation dataset contains three paper citation networks: DBLPv7 (D), ACMv9 (A), and Citationv1 (C). Nodes represent research papers and edges represent citation relations. DBLPv7 contains papers from DBLP published between 2004 and 2008, while ACMv9 contains papers from ACM published between 2000 and 2010. Citationv1 is derived from the Microsoft Academic Graph (MAG) and contains papers published before 2008. Each paper is represented by a 6,775-dimensional sparse bag-of-words feature vector derived from its title. The papers are classified into five research areas: Databases (DB), Artificial Intelligence (AI), Computer Vision (CV), Information Security (IS), and Networking.

C.2 Baseline Details

We compare `COME` with twenty-two baselines from four groups, including source-needed, no-adaptation, single-source-free, and multi-source-free methods. Each baseline is briefly described below. For baselines evaluated under the identical benchmark protocol and data splits, we report their published results, COME is evaluated under the same protocol.

Source-needed. Methods in this group access labeled source data during target adaptation. For single-source graph adaptation methods, we combine the labeled source domains as the source side.

- **MDAN** (Zhao et al., 2018). MDAN performs multi-source domain adaptation through adversarial learning to obtain representations that are invariant to multiple domain shifts. It jointly preserves task-discriminative information and reduces discrepancies between the source and target domains.
- **M^{3}SDA** (Peng et al., 2019). M^{3}SDA transfers knowledge from multiple labeled source domains by aligning feature-distribution moments across domains. It reduces source–target discrepancies through moment matching while exploiting information from multiple sources.
- **UDAGCN** (Wu et al., 2020). UDAGCN combines graph convolution with domain adaptation to learn transferable node representations across graphs. It captures local and global graph information and aligns source and target representations through domain-adaptive training.
- **GRADE** (Wu et al., 2023). GRADE introduces graph subtree discrepancy to characterize distribution shifts between source and target graphs. It minimizes this discrepancy during graph representation learning to improve cross-network knowledge transfer.
- **SpecReg** (You et al., 2023). SpecReg studies graph domain adaptation from the spectral properties of graph neural networks. It regularizes spectral smoothness and frequency responses to improve the transferability of graph representations.

No-adaptation. Methods in this group are trained separately on each labeled source domain and directly applied to the target graph without target adaptation. We average their source-model predictions to obtain the final prediction.

- **GCN** (Kipf & Welling, 2017). GCN performs graph convolution using a localized approximation of spectral graph filtering. It aggregates node features from local neighborhoods to learn representations for node classification.
- **SAGE** (Hamilton et al., 2017). SAGE learns node representations by sampling and aggregating features from local neighborhoods. Its inductive formulation enables the learned aggregation functions to generalize to previously unseen nodes.
- **GAT** (Veličković et al., 2017). GAT introduces self-attention into neighborhood aggregation to assign different importance to neighboring nodes. It uses learned attention coefficients to combine neighborhood information for node representation learning.
- **GIN** (Xu et al., 2018). GIN uses an expressive neighborhood aggregation scheme motivated by the Weisfeiler–Lehman graph isomorphism test. It combines sum aggregation with nonlinear transformations to distinguish different neighborhood structures.

Single-source-free. Methods in this group adapt a pretrained source model to the unlabeled target graph without accessing its source data. For multiple source models, we independently perform adaptation and average their resulting class-probability vectors.

- **SHOT** (Liang et al., 2020). SHOT freezes the source classifier and adapts the feature extractor using unlabeled target data. It combines information maximization with self-supervised pseudo-labeling to align target representations with the source hypothesis.
- **BNM** (Cui et al., 2020). BNM maximizes the nuclear norm of target prediction matrices to improve both prediction discriminability and diversity. It provides an unsupervised objective for adapting the pretrained model using only target predictions.
- **ATDOC** (Liang et al., 2021). ATDOC introduces an auxiliary target-oriented classifier to reduce the source bias in pseudo-label generation. It uses a memory mechanism and neighborhood aggregation to obtain more reliable target pseudo-labels.
- **NRC** (Yang et al., 2021). NRC exploits the intrinsic neighborhood structure of target representations for source-free adaptation. It encourages prediction consistency among reciprocal and expanded target neighbors while reducing the effect of noisy neighborhoods.
- **JMDS** (Lee et al., 2022). JMDS estimates sample-wise confidence by jointly considering source-model knowledge and target-data structure. It uses the resulting confidence scores to weight target samples and reduce the influence of unreliable pseudo-label supervision.
- **GTRANS** (Jin et al., 2022). GTRANS adapts graph data at test time rather than directly retraining the pretrained graph model. It refines target graph information to improve predictions under distribution shifts and other graph perturbations.
- **SOGA** (Mao et al., 2024). SOGA performs source-free graph domain adaptation using only a pretrained source model and an unlabeled target graph. It preserves structural proximity on the target graph while exploiting the discriminative information learned by the source model.
- **GraphCTA** (Zhang et al., 2024). GraphCTA performs source-free graph adaptation by jointly adapting the pretrained model and the target graph. It alternates neighborhood-based model adaptation with graph structure and attribute adaptation to progressively improve target predictions.
- **TPDS** (Tang et al., 2024). TPDS reformulates source-free adaptation as searching for an appropriate target prediction distribution. It progressively constructs and aligns proxy distributions to bridge the distribution shift from source predictions to the target domain.

Multi-source-free. Methods in this group directly use multiple pretrained source models for target adaptation without accessing their corresponding source data. For non-graph methods, we use a graph neural network architecture for node classification.

- **DECISION** (Ahmed et al., 2021). DECISION performs multi-source adaptation by automatically combining pretrained source models without accessing source data. It estimates suitable source-model contributions from unlabeled target data to obtain an adapted target predictor.
- **CAiDA** (Dong et al., 2021). CAiDA estimates the transferable contribution of different source models and constructs confident anchors from target predictions. It assigns pseudo-labels using these anchors and introduces class-relationship consistency to support multi-source-free adaptation.
- **MSFDA** (Shen et al., 2023). MSFDA studies the bias–variance trade-off in multi-source-free domain adaptation and develops an adaptation strategy based on domain aggregation, selective pseudo-labeling, and joint feature alignment. For graph node classification, we instantiate its feature learner with the corresponding GNN backbone.
- **GraphATA** (Zhang & He, 2025). GraphATA performs multi-source-free graph adaptation at the node level rather than aggregating source predictions globally. It constructs node-specific graph convolution parameters by combining pretrained source-model weights according to local target context. Relation to GraphATA. GraphATA uses local target context to construct node-specific combinations of pretrained source parameters. `COME` builds on node-wise source utilization from a different axis: it explicitly probes each frozen source under target-graph perturbations, tracks target-side conflict throughout adaptation, and injects the resulting structural and temporal signals into both target supervision and final source–target routing. Thus, the distinguishing component of `COME` is reliability-conditioned coordination rather than node-wise parameter aggregation alone.

C.3 Implementation Settings

We use GCN as the default architecture with a node representation dimension of 128. The model is optimized using Adam, with the learning rate and weight decay selected from $\{0.1, 0.01, 0.001, 10^{-4}\}$. We use a single shared configuration $L = 2$, $K = 5$, $\lambda_{\text{sta}} = 1$, and $\lambda_{\text{con}} = 0.05$, Appendix D.1 reports sensitivity around this configuration. Unless otherwise specified, we use $K_{\text{nn}} = 40$, $T = 5$, $\rho = 0.9$, $\rho_e = 0.9$, $\lambda_g = 0.2$, numerical constant $\epsilon = 10^{-12}$, and edge perturbation ratios of 0.10 for removal and 0.05 for addition. The memory bank is initialized by target features and the mean predictions of the frozen source models and updated once per adaptation step. The memory bank is initialized with target features from the target-adaptive model and the mean predictions of the frozen source models; its features and predictions are updated once per adaptation step with momentum 0.9. During target adaptation, all source models remain fixed, and only the target-side parameters are optimized. Each experiment is repeated 5 times, and we report the mean node classification accuracy and standard deviation.

Ablation settings. In Table 2, each w/o variant removes the corresponding complete module. SMEP removes structural probing and reliability-weighted source prediction, TMAC removes accumulated-conflict-based state modeling and its state-conditioned supervision, and RETP removes final reliability-aware source-target routing.

D More Experiments

D.1 More Sensitivity Study

Table 4 examines the number of perturbed views K from 1 to 7. The effect of K remains limited on most transfer settings, although $K = 5$ produces a clear improvement on CSBM and R,P,F,ES→EN. On the two Twitch transfers, the accuracy varies within 0.17 percentage-point ranges and 0.98 percentage-point ranges, respectively. The corresponding ranges on the three Citation transfers are 0.23 percentage-point ranges, 0.20 percentage-point ranges, and 0.07 percentage-point ranges. CSBM shows a larger variation of 2.09%, mainly due to the increase from around 91.4–91.7% at other values to 93.50% at $K = 5$. Across the 6 settings, the average accuracy ranges from 75.66% to 76.21%, with $K = 5$ achieving the highest average accuracy of 76.21%. Increasing K beyond 5 does not bring further gains, so we use $K = 5$ throughout the main experiments.

Table 4: Sensitivity analysis (%) of the number of perturbed views K across six transfer settings (mean ± std over five runs).

K	CSBM	Twitch		Citation		
	C1,C2,C3→C4	R,P,F,ES→DE	R,P,F,ES→EN	A,C→D	C,D→A	A,D→C
1	91.74 ± 1.16	68.09 ± 0.10	59.10 ± 0.11	78.62 ± 0.16	73.88 ± 0.08	82.75 ± 0.22
2	91.41 ± 1.32	68.08 ± 0.08	59.10 ± 0.11	78.66 ± 0.17	**73.91** ± 0.07	82.77 ± 0.23
3	91.74 ± 1.08	68.07 ± 0.08	59.11 ± 0.13	78.55 ± 0.41	73.74 ± 0.15	82.82 ± 0.20
4	91.73 ± 1.10	68.08 ± 0.09	59.11 ± 0.13	78.54 ± 0.38	73.71 ± 0.12	82.80 ± 0.19
5	**93.50** ± 0.80	**68.24** ± 0.07	**60.08** ± 0.18	**78.77** ± 0.19	73.86 ± 0.12	**82.82** ± 0.27
6	91.42 ± 1.34	68.08 ± 0.04	59.13 ± 0.10	78.75 ± 0.20	73.89 ± 0.13	82.79 ± 0.28
7	91.73 ± 1.10	68.08 ± 0.05	59.12 ± 0.10	78.57 ± 0.38	73.78 ± 0.11	82.80 ± 0.23

Table 5: Sensitivity analysis (%) of the number of graph layers L across six transfer settings (mean ± std over five runs).

L	CSBM	Twitch		Citation		
	C1,C2,C3→C4	R,P,F,ES→DE	R,P,F,ES→EN	A,C→D	C,D→A	A,D→C
1	68.26 ± 16.38	54.76 ± 13.87	56.28 ± 1.42	72.25 ± 2.95	70.63 ± 2.95	76.24 ± 1.58
2	**93.50** ± 0.80	**68.24** ± 0.07	**60.08** ± 0.18	**78.77** ± 0.19	**73.86** ± 0.12	**82.82** ± 0.27
3	37.13 ± 16.39	65.50 ± 1.12	57.14 ± 0.71	71.91 ± 3.17	64.98 ± 3.17	78.93 ± 0.49
4	42.17 ± 4.76	44.59 ± 11.28	45.44 ± 0.00	66.89 ± 2.87	58.38 ± 2.87	48.53 ± 17.06
5	34.74 ± 3.92	59.96 ± 6.49	47.32 ± 4.07	48.62 ± 14.68	49.28 ± 14.68	64.70 ± 21.92

Table 5 shows a much stronger effect of the graph propagation depth L. $L = 2$ achieves the best accuracy on all 6 transfer settings, with an average accuracy of 76.21%. In comparison, the average accuracy is 66.40% at $L = 1$ and decreases to 62.60%, 51.00%, and 50.77% for $L = 3$, 4, and 5, respectively. The degradation is particularly pronounced on CSBM, where the accuracy drops from 93.50% at $L = 2$ to 37.13% at $L = 3$ and remains below 55% for deeper propagation. Similar declines appear on Twitch and Citation as L increases. These results favor shallow propagation for target adaptation, and we set $L = 2$ in all experiments.

Table 6 studies λ_{sta} over the range from 0.25 to 2.00. Increasing λ_{sta} from 0.25 to 1.00 raises the average accuracy from 73.12% to 76.21%. The change is most pronounced on CSBM, where the accuracy increases from 82.41% to 93.50%. Clear improvements also appear on Citation, with A,C→D increasing from 76.36% to 78.77% and C,D→A from 70.03% to 73.86%. In contrast, the two Twitch settings vary only slightly over the tested range. Further increasing λ_{sta} to 1.50 and 2.00 lowers the average accuracy to 74.49% and 74.20%, respectively. We therefore use $\lambda_{\text{sta}} = 1$ as the default setting.

Table 7 presents the results for λ_{con} from 0.025 to 0.075. The average accuracy remains between 75.38% and 76.21% across the tested range, with the best result obtained at $\lambda_{\text{con}} = 0.05$. The change is more evident on CSBM and A,C→D, where the accuracy ranges from 91.36% to 93.50% and from 77.34% to 78.77%, respectively. R,P,F,ES→EN also improves from around 59.1% to 60.08% at $\lambda_{\text{con}} = 0.05$. In comparison, R,P,F,ES→DE and A,D→C remain relatively stable across different values. We therefore use $\lambda_{\text{con}} = 0.05$ in the main experiments.

Table 6: Node classification accuracy (%) for different values of λ_{sta} across six transfer settings (mean ± std over five runs).

λ_{sta}	CSBM	Twitch		Citation		
	C1,C2,C3→C4	R,P,F,ES→DE	R,P,F,ES→EN	A,C→D	C,D→A	A,D→C
0.25	82.41 ± 19.06	68.02 ± 0.05	59.22 ± 0.16	76.36 ± 2.66	70.03 ± 1.80	82.70 ± 0.03
0.50	90.07 ± 2.93	68.05 ± 0.03	59.27 ± 0.12	77.94 ± 0.68	72.78 ± 1.96	82.67 ± 0.44
1.00	93.50 ± 0.80	68.24 ± 0.07	60.08 ± 0.18	78.77 ± 0.19	73.86 ± 0.12	82.82 ± 0.27
1.50	85.41 ± 13.63	68.10 ± 0.09	59.16 ± 0.19	78.33 ± 0.29	73.53 ± 0.15	82.42 ± 0.16
2.00	84.08 ± 11.09	68.10 ± 0.10	59.13 ± 0.25	78.19 ± 0.38	73.41 ± 0.14	82.28 ± 0.11

Table 7: Sensitivity analysis (%) of λ_{con} across six transfer settings (mean ± std over five runs).

λ_{con}	CSBM	Twitch		Citation		
	C1,C2,C3→C4	R,P,F,ES→DE	R,P,F,ES→EN	A,C→D	C,D→A	A,D→C
0.025	91.49 ± 1.32	68.08 ± 0.07	59.14 ± 0.08	77.34 ± 0.76	73.62 ± 0.17	82.74 ± 0.24
0.0375	91.36 ± 1.44	68.07 ± 0.06	59.12 ± 0.09	77.42 ± 0.97	73.62 ± 0.14	82.70 ± 0.27
0.05	**93.50** ± 0.80	**68.24** ± 0.07	**60.08** ± 0.18	**78.77** ± 0.19	**73.86** ± 0.12	**82.82** ± 0.27
0.0625	91.37 ± 1.55	68.08 ± 0.06	59.13 ± 0.12	77.40 ± 0.89	73.57 ± 0.14	82.77 ± 0.25
0.075	91.64 ± 0.97	68.08 ± 0.06	59.15 ± 0.14	77.51 ± 0.91	73.69 ± 0.08	82.67 ± 0.21

D.2 More architecture Study

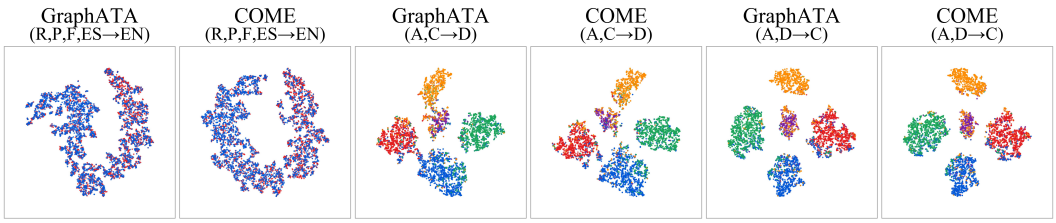


Figure 4: Additional target-domain representations of GraphATA and COME on representative Twitch and Citation transfer settings. Colors denote node classes.

Table 8 compares COME with four GNN architectures across all 6 transfer settings. The GCN-based variant achieves the highest accuracy in every setting and obtains an average accuracy of 76.21%. The differences are particularly clear on CSBM and Twitch. On CSBM, GCN reaches 93.50%, compared with 77.48%, 75.71%, and 38.40% for GIN, GAT, and SAGE, respectively. On Twitch, GCN also performs best on both transfer directions, while GIN remains competitive on R,P,F,ES→DE and GAT gives the second-best result on R,P,F,ES→EN. The gaps become smaller on Citation, where GAT and SAGE retain relatively strong performance; for example, they reach 81.84% and 80.81% on A,D→C, compared with 82.82% for GCN. Averaged over the 6 settings, GAT, GIN, and SAGE obtain 67.22%, 67.09%, and 58.36%, respectively, compared with 76.21% for GCN. Accordingly, we fix GCN throughout the main evaluation as the reference message-passing backbone, while Table 8 separately characterizes the sensitivity of COME to alternative backbone operators. We therefore use GCN as the default architecture in the main experiments.

D.3 More Representation Visualization

Figure 4 presents additional target-domain representation visualizations on Twitch and Citation. On R,P,F,ES→EN, GraphATA produces several fragmented regions with substantial mixing between the two classes. The overlap is not fully removed by COME, but the resulting representation is more concentrated and contains fewer isolated regions. On Citation, the difference is clearer. For A,C→D, COME produces more localized class regions with less mixing near the cluster boundaries. A similar pattern appears on A,D→C, where the class clusters become more compact and the separation between neighboring classes is clearer than that of GraphATA. Together with the visualizations in the main text, these results show a consistent change toward more compact within-class representations and clearer class separation, especially on the Citation transfers.

Table 8: Architecture comparison results (%) across six transfer settings (mean ± std over five runs).

Architectures	CSBM	Twitch		Citation		
	C1,C2,C3→C4	R,P,F,ES→DE	R,P,F,ES→EN	A,C→D	C,D→A	A,D→C
COME_{GCN}	**93.50** ± 0.80	**68.24** ± 0.07	**60.08** ± 0.18	**78.77** ± 0.19	**73.86** ± 0.12	**82.82** ± 0.27
$\text{COME}_{\text{SAGE}}$	38.40 ± 0.85	39.55 ± 0.00	46.11 ± 0.26	74.14 ± 0.17	71.13 ± 0.90	80.81 ± 0.99
COME_{GAT}	75.71 ± 0.35	39.55 ± 0.00	58.18 ± 0.24	75.50 ± 0.12	72.56 ± 0.31	81.84 ± 0.10
COME_{GIN}	77.48 ± 0.08	62.36 ± 0.02	55.16 ± 0.21	64.96 ± 3.21	66.98 ± 0.36	75.59 ± 0.20