REPRESENTATION ALIGNMENT OF GRAPH FOUNDATION MODELS FOR CROSS-DATASET CLASSIFICATION

Anonymous authors
Paper under double-blind review

ABSTRACT

CLIP-based graph foundation models tackle cross-dataset node classification by aligning graph and text embeddings via contrastive learning. However, their practical performance is fundamentally constrained by the distribution shift between source-domain pre-training texts and inference-time classification templates, which causes severe semantic mismatch. To achieve effective cross-domain representation alignment without any fine-tuning, we propose **Perturbation Alignment (PA)**—a training-free adaptation framework that performs directed perturbations on classification template embeddings guided by subgraph encoding. Specifically, PA applies a single-step perturbation along the composite objective function to enhance the classification margin of the original model, injecting target-domain subgraph information into textual prototypes while preserving pre-trained semantic knowledge. Extensive experiments on a total of seven benchmark datasets demonstrate that PA not only substantially outperforms the vanilla baseline, but also yields particularly significant improvements when source-domain data or template descriptions are missing. Moreover, PA matches or even surpasses lightweight fine-tuning on certain datasets, providing a practical, parameter-efficient, and theoretically principled solution for cross-dataset graph node classification.

1 INTRODUCTION

Large-scale pre-training has fundamentally transformed graph representation learning in recent years (Hu et al., 2019). Following the success of CLIP-style vision-language models (Radford et al., 2021), Graph Foundation Models (GFMs) have emerged as a promising paradigm that learns to align graph information with textual descriptions via contrastive pre-training (Zeng & Xie, 2021), learning transferable representations that generalize to unseen datasets. On the surface, this paradigm appears to provide a universal solution: a single pre-trained model can be directly applied to downstream tasks without any task-specific fine-tuning.

However, the practical efficacy of cross-dataset graph node classification is fundamentally constrained by the intrinsic capacity of the pre-trained model and the distribution discrepancy between training and test samples (Brannon et al., 2024). Despite possessing strong representation capabilities (Hu et al., 2020b), recent large graph and text encoders still fail to deliver satisfactory performance on domain-specific graph data, as the absence of downstream task adaptation prevents them from effectively activating task-relevant features (Xia et al., 2024). The pre-trained parameters are never explicitly optimized for the inference-time classification space, leaving the model underactivated for the target domain and limiting the realization of its intrinsic capabilities (Liu et al., 2023; Zhou et al., 2022b).

In CLIP-based graph foundation models, this distribution discrepancy appears as a shift between source-domain pre-training texts (e.g., graph summaries) and target-domain classification templates (e.g., "a graph of class X") (Parameswaran et al., 2026). Because the model predicts via cosine similarity between graph and template embeddings, the design of these templates critically determines cross-dataset accuracy. However, the brevity of test-time prompts contrasts sharply with the verbosity of pre-training texts, driving the source and target domains into mismatched representation subspaces and undermining classification performance. This naturally raises a critical question: *Without any fine-tuning, can we achieve effective cross-domain representation alignment between*

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.

the source domain (pre-trained network) and the target domain (inference-time classification templates)?

To address this question, we propose Perturbation Alignment (PA), a training-free adaptation framework that performs directed perturbations on classification template embeddings to achieve cross-domain representation alignment. Instead of updating the pre-trained network parameters, we introduce a lightweight inference-stage mechanism. PA leverages subgraph encodings as guidance to perturb the template embeddings along the gradient of a composite objective function. This objective simultaneously reduces subgraph-template similarity to prevent overconfident mispredictions and encourages inter-template orthogonality, with the classification margin improvement theoretically guaranteed. This directed perturbation injects target-domain information into textual prototypes while preserving their original semantic knowledge, effectively bridging the source–target representation gap without any neural network parameter updates. A visual illustration of this alignment process is provided in Figure 1. As can be intuitively observed, the perturbed text embeddings are pushed farther apart from each other, yet each undergoes a distinct shift magnitude. This asymmetric displacement ultimately positions the perturbed prototypes such that they achieve better class separation with respect to the subgraph embeddings to be classified. As a result, PA serves as a parameter-efficient, training-free solution for cross-dataset graph node classification. Beyond this, we further optimize the graph encoder architecture of the representative GraphCLIP framework (Zhu et al., 2025).

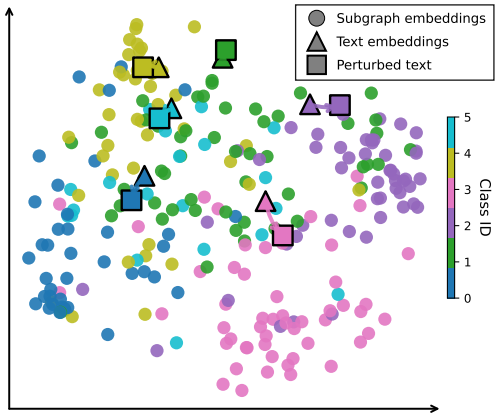


Figure 1: Schematic illustration of representation alignment under textual perturbation on CiteSeer. Arrows denote shifts from original to perturbed text embeddings in the shared embedding space.

Extensive experiments on cross-dataset graph node classification benchmarks demonstrate the effectiveness of our proposed framework. On the seven target datasets—Cora, CiteSeer, Photo, Computers, History, WikiCS, and Instagram, the experimental results show that our method substantially improves accuracy over the vanilla baseline. Furthermore, the accuracy gains are particularly pronounced in extreme scenarios where source-domain training data is unavailable or detailed class template descriptions are missing. Remarkably, even when the pre-trained model has no access to academic paper training data as the source domain, our perturbation alignment method achieves state-of-the-art performance on the Cora dataset (academic paper domain classification). Moreover, our training-free method matches or even surpasses the performance of lightweight fine-tuning with task-specific updates on certain datasets. These findings indicate that our framework provides a practical, parameter-efficient, and theoretically principled solution for cross-dataset graph node classification.

Our contributions are summarized as follows:

- We propose Perturbation Alignment (PA), a training-free adaptation framework for cross-dataset graph node classification. PA performs directed perturbations on classification template embeddings guided by subgraph encoding, injecting target-domain structural information into textual prototypes to align source and target domains without any fine-tuning.
- We provide a theoretical analysis of PA, proving that the perturbed classification margin improves under the proposed joint perturbation.
- Building upon GraphCLIP (Zhu et al., 2025), we introduce a targeted architectural improvement by replacing the average pooling operation with gated pooling, which enhances the representation capacity of the graph encoder.
- Extensive experiments on seven benchmark datasets demonstrate that PA outperforms existing baselines and matches or surpasses lightweight fine-tuning approaches on multiple datasets, with particularly significant gains under distribution shift where source-domain data or template descriptions are unavailable.

2 RELATED WORK

Node classification is one of the most fundamental and extensively studied tasks in graph representation learning. Early methods primarily focused on aggregating neighbor information over graph structures. Among them, GCN (Kipf & Welling, 2016) achieves semi-supervised classification via localized spectral filtering, GraphSAGE (Hamilton et al., 2017) introduces inductive sampling-based aggregation to handle large-scale graphs, and GAT (Veličković et al., 2017) adaptively captures heterogeneous neighbor importance through an attention mechanism. These works collectively laid the architectural foundation for graph neural networks.

In recent years, with the rise of pre-training techniques, the research focus has gradually shifted from designing specific architectures to building transferable graph foundation models. On the one hand, self-supervised pre-training methods—such as DGI (Veličković et al., 2018), GraphCL (You et al., 2020), GRACE (Zhu et al., 2020), and GPT-GNN (Hu et al., 2020b)—learn general structural priors from unlabeled graphs via mutual information maximization, contrastive learning, or generative objectives. Masked autoencoding frameworks such as GraphMAE (Hou et al., 2022) further reduce the reliance on labeled data. On the other hand, to break the single-graph limitation and achieve cross-domain generalization, multi-domain unified foundation models recast node classification as link prediction (e.g., OFA (Liu et al., 2024) and PRODIGY (Huang et al., 2023)) or perform feature alignment via coordinator nodes (e.g., GCOPE (Zhao et al., 2024a), MDGPT (Yu et al., 2024)), while AnyGraph (Xia & Huang, 2026) aims to learn domain-invariant graph encoders. Under the tuning-free inductive setting, GraphAny (Zhao et al., 2024b) enables in-context inference through closed-form solution fusion, offering a new perspective for zero-shot node classification. Furthermore, on text-attributed graphs, graph-text aligned models such as GraphCLIP (Zhu et al., 2025) extend the CLIP-style contrastive learning framework to the graph domain, achieving cross-dataset node classification by matching subgraph embeddings with class-template embeddings.

Regarding CLIP-style classification models, recent research has also achieved notable progress. Prompt optimization methods have evolved from learnable contexts (CoOp (Zhou et al., 2022b), CoCoOp (Zhou et al., 2022a)) to constraining update directions to preserve semantic priors (KgCoOp (Yao et al., 2023), ProGrad (Zhu et al., 2023)), continuously improving zero-shot generalization. Building on this, diverse text prompting further enhances domain generalization (Wen et al., 2025), while test-time adversarial countermeasures (TTC (Xing et al., 2025)) and text-induced subspace alignment (COLA (Zhu et al., 2026)) effectively improve adversarial robustness.

3 PRELIMINARY

We adapt the CLIP framework for graph node classification, where original images are replaced by node-induced subgraphs and the image encoder is substituted with a graph encoder. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ denote the input graph. For each node $v_i \in \mathcal{V}$ with ground-truth label $y_i \in \{1, \dots, C\}$, we extract its localized subgraph $\mathcal{G}_i \subseteq \mathcal{G}$ (e.g., k-hop neighborhood or random walk). For any class c, we denote its class name as c_{name}. During training, we leverage summary-style textual descriptions (e.g., abstracts) as the class text t_c, which provides richer semantic supervision than bare class names. During inference, t_c is formed by concatenating the class name with a natural language description, i.e., $t_c = \text{Concat}(c_{\text{name}}, c_{\text{desc}})$. The set of all class texts is $\mathcal{T} = \{t_1, \dots, t_C\}$. The graph encoder f_θ and text encoder g_ϕ project inputs into a shared d-dimensional embedding space:

$$\boldsymbol{h}_i = f_\theta(\mathcal{G}_i), \quad \boldsymbol{z}_c = g_\phi(t_c), \quad |\boldsymbol{h}_i|_2 = |\boldsymbol{z}_c|_2 = 1. \tag{1}$$

During training, we sample batches of B subgraphs. For a subgraph $\mathcal{G}_i$ in a batch (indexed by i or j) and a class $c \in \{1, \dots, C\}$, the two encoders are aligned via a contrastive loss with a temperature coefficient τ:

$$\mathcal{L}_{\text{CLIP}} = -\frac{1}{2B} \sum_{i=1}^{B} \left[\log \frac{\exp(\boldsymbol{h}_i^\top \boldsymbol{z}_{y_i}/\tau)}{\sum_{c=1}^{C} \exp(\boldsymbol{h}_i^\top \boldsymbol{z}_c/\tau)} + \log \frac{\exp(\boldsymbol{h}_i^\top \boldsymbol{z}_{y_i}/\tau)}{\sum_{j=1\ j\neq i\ y_j \neq y_i}^{B} \exp(\boldsymbol{h}_j^\top \boldsymbol{z}_{y_i}/\tau)} \right]. \tag{2}$$

During inference, the predicted class for node y_i is:

$$\hat{y}_i = \arg\max_c \, (\boldsymbol{h}_i^\top \boldsymbol{z}_c). \tag{3}$$

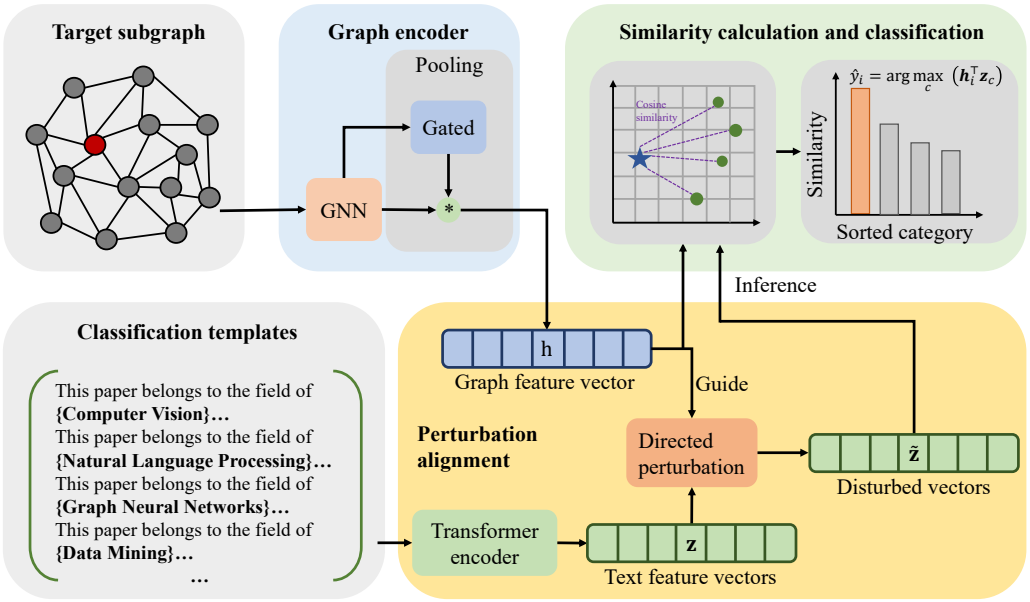


Figure 2: Framework of our method.

4 METHOD

4.1 CROSS-DOMAIN REPRESENTATION ALIGNMENT VIA DIRECTED PERTURBATION

In cross-dataset graph node classification, the source-domain graph encoder f_θ and text encoder are well aligned in the representation space after pre-training. However, at inference time, the graph encoder must match against target-domain classification templates $z_c = g_\phi(t_c)$ that were never encountered during training. This distributional shift induces a cross-modal semantic mismatch between the graph encoder and the target-domain text encoder, creating a representation gap that degrades cross-domain transfer performance.

To bridge this gap, we propose Perturbation Alignment (PA), which leverages subgraph encodings $\mathbf{h}_i = f_\theta(\mathcal{G}_i)$ to guide a directed perturbation of the classification template embeddings. As illustrated in Figure 2, PA is driven by two intuitive objectives for each subgraph: (1) reduce its similarity to all classification templates, thereby preventing overconfident but incorrect predictions; and (2) enlarge the separation between template embeddings themselves to mitigate inter-class confusion. Since no labeled data is available at inference time, we approximate these objectives through a composite loss function and execute a single gradient-descent step on the template embeddings. This injects target-domain graph structural information and the discriminative knowledge encoded in the pre-trained source-domain graph encoder into the target-domain text prototypes, achieving effective cross-domain representation alignment without compromising the original semantic structure of the text space.

Formally, the perturbation is guided by a composite objective function that integrates the inter-class decorrelation objective $\mathcal{L}_{\text{deco}}$ and the subgraph-text compression objective $\mathcal{L}_{\text{comp}}$:

$$\mathcal{L}(\mathbf{h}, \mathbf{Z}) = \frac{\lambda}{N} \sum_{j=1}^{N} \mathbf{h}^\top \mathbf{z}_j + \frac{1}{N(N-1)} \sum_{i=1}^{N} \sum_{j \neq i} \mathbf{z}_i^\top \mathbf{z}_j. \tag{4}$$

In implementation, we perform a single gradient-descent step on the embedding set $\mathbf{Z} = \{\mathbf{z}_k\}_{k=1}^K$ with respect to $\mathcal{L}$, thereby simultaneously promoting subgraph-template alignment and inter-template decorrelation. The single-step perturbation is computed as follows:

$$\boldsymbol{\delta} = \epsilon \nabla_{\mathbf{Z}} \mathcal{L}, \tag{5}$$

where ϵ controls the perturbation magnitude. The template embeddings are updated as $\tilde{\mathbf{z}}_k = \mathbf{z}_k - \boldsymbol{\delta}_k$ (followed by L_2 normalization to satisfy $\|\tilde{\mathbf{z}}_k\|_2 = 1$). The gradient-based perturbation encourages the template embeddings to align with the target-domain subgraph representations while promoting inter-template decorrelation. Notably, our goal is not to converge to a global optimum of this composite objective, but rather to execute a single gradient-descent step to minimize $\mathcal{L}$. This is because over-optimization would cause the template embeddings to deviate from their original semantics in the pre-trained language space, losing the categorical discriminative information inherently carried by the text prototypes. By controlling the perturbation magnitude ϵ, we strike a balance between injecting graph structural information and preserving the original semantics, thereby flexibly regulating the strength of cross-domain alignment. The effect of PA on the classification margin is discussed in detail in Section 4.3.

4.2 GATED POOLING

GraphCLIP (Zhu et al., 2025) adopts average pooling over subgraph nodes, implicitly assuming uniform node importance—an unreasonable assumption since nodes play asymmetric roles in real-world graphs. We replace this with a learnable gated pooling mechanism (which is independent of our main contribution PA and serves merely as a minor architectural refinement). A lightweight MLP assigns a score to each node, and the subgraph representation is computed as a weighted sum:

$$\alpha_i = \frac{\exp(\text{MLP}_{\text{gate}}(x_i))}{\sum_{j \in \mathcal{V}(b_i)} \exp(\text{MLP}_{\text{gate}}(x_j)) + \eta}, \quad h_{b_i} = \sum_{j \in \mathcal{V}(b_i)} \alpha_j \cdot x_j, \tag{6}$$

where x_i is the node feature, b_i the subgraph index, $\mathcal{V}(b_i)$ the node set of subgraph b_i, and $\eta = 10^{-8}$. As a drop-in replacement for the original average pooling, it introduces minimal computational overhead while delivering substantial gains in cross-dataset classification accuracy.

4.3 THEORETICAL INSIGHTS

Theorem 1 (Effect of joint perturbation on classification margin)**.** *Given unit-norm pretrained templates $\mathbf{z}_k$, a unit-norm subgraph encoding $\mathbf{h}$, and the combined objective $\mathcal{L} = \lambda \mathcal{L}_{comp} + \mathcal{L}_{deco}$, where $\mathcal{L}_{comp} = \frac{1}{N}\sum_{k=1}^{N} \mathbf{h}^\top \tilde{\mathbf{z}}_k$ and $\mathcal{L}_{deco} = \frac{1}{N(N-1)}\sum_{k \neq j} \mathbf{z}_k^\top \mathbf{z}_j$. Let $\tilde{\mathbf{z}}_k = \mathbf{z}_k - \epsilon \nabla_{\mathbf{z}_k} \mathcal{L}$ be the perturbed templates from a single gradient step. Then the perturbed classification margin $m = \mathbf{h}^\top \tilde{\mathbf{z}}_y - \max_k \mathbf{h}^\top \tilde{\mathbf{z}}_k$ satisfies*

$$m = m_0 + \epsilon(\max_k \mathbf{h}^\top \nabla_{\mathbf{z}_k} \mathcal{L}_{deco} - \mathbf{h}^\top \nabla_{\mathbf{z}_y} \mathcal{L}_{deco}), \tag{7}$$

where $m_0 = \mathbf{h}^\top \mathbf{z}_y - \max_k \mathbf{h}^\top \mathbf{z}_k$ is the original margin.

Proof of Theorem. The gradient of $\mathcal{L}_{\text{comp}}$ w.r.t. $\mathbf{z}_k$ is $\nabla_{\mathbf{z}_k} \mathcal{L}_{\text{comp}} = \frac{1}{N}\mathbf{h}$. The gradient of $\mathcal{L}_{\text{deco}}$ w.r.t. $\mathbf{z}_k$ is $\nabla_{\mathbf{z}_k} \mathcal{L}_{\text{deco}} = \frac{1}{N(N-1)}\sum_{j \neq k} \mathbf{z}_j$, which directly reduces the contribution of each $\mathbf{z}_j$ ($j \neq k$) to the template $\mathbf{z}_k$, shrinking cross-template similarity. The combined gradient is $\nabla_{\mathbf{z}_k} \mathcal{L} = \frac{\lambda}{N}\mathbf{h} + \nabla_{\mathbf{z}_k} \mathcal{L}_{\text{deco}}$. The perturbation is $\delta_k = \epsilon \nabla_{\mathbf{z}_k} \mathcal{L}$, giving

$$\mathbf{h}^\top \delta_k = \frac{\epsilon\lambda}{N} + \epsilon \mathbf{h}^\top \nabla_{\mathbf{z}_k} \mathcal{L}_{\text{deco}}. \tag{8}$$

Thus $\mathbf{h}^\top \tilde{\mathbf{z}}_k = \mathbf{h}^\top \mathbf{z}_k - \mathbf{h}^\top \delta_k$. Substituting into the margin:

$$\begin{aligned} m &= \mathbf{h}^\top \tilde{\mathbf{z}}_y - \max_k \mathbf{h}^\top \tilde{\mathbf{z}}_k \\ &= \mathbf{h}^\top \mathbf{z}_y - \frac{\epsilon\lambda}{N} - \epsilon \mathbf{h}^\top \nabla_{\mathbf{z}_y} \mathcal{L}_{\text{deco}} - \max_k \left[\mathbf{h}^\top \mathbf{z}_k - \frac{\epsilon\lambda}{N} - \epsilon \mathbf{h}^\top \nabla_{\mathbf{z}_k} \mathcal{L}_{\text{deco}} \right] \\ &= m_0 + \epsilon(\max_k \mathbf{h}^\top \nabla_{\mathbf{z}_k} \mathcal{L}_{\text{deco}} - \mathbf{h}^\top \nabla_{\mathbf{z}_y} \mathcal{L}_{\text{deco}}). \end{aligned} \tag{9}$$

□

Remark 1. *Theorem 1 reveals that the perturbed classification margin m is directly governed by the decoupling gradient term $\mathcal{L}_{deco}$. In the infinitesimal limit, since $\max_k \mathbf{h}^\top \nabla_{\mathbf{z}_k} \mathcal{L}_{deco} - \mathbf{h}^\top \nabla_{\mathbf{z}_y} \mathcal{L}_{deco} \geq 0$ holds trivially, a sufficiently small perturbation along the proposed direction is* guaranteed *to enlarge the classification margin ($m \geq m_0$) whenever the decoupling gradient at the correct template $\mathbf{z}_y$ is not the global maximum. Intuitively, this means our directional perturbation actively pushes the perturbed templates $\tilde{\mathbf{z}}_k$ to reduce cross-template similarity via $\mathcal{L}_{deco}$, which amplifies the geometric gap between the correct template and the most competitive incorrect template in the direction of $\mathbf{h}$. In practice, however, the actual perturbation is applied with a finite step size ϵ; while this preserves the margin-improving trend for the vast majority of cases (as shown in Table 3), a small fraction of datasets may exhibit a slight decrease in Δm due to local curvature effects or discretization of the template embeddings. Nevertheless, the dominant positive Δm observed across datasets empirically corroborates the theoretical prediction that the proposed joint perturbation mechanism improves class separability from a geometric perspective.*

5 EXPERIMENT

5.1 DATASET OVERVIEW

We evaluate our method on multiple Text-Attributed Graph (TAG) datasets, which are partitioned into the source domain (G_{source}) and the target domain (G_{target}), detailed in Table 1. The source datasets include ogbn-ArXiv (Wang et al., 2020), ArXiv_2023 (He et al., 2023), PubMed (Yang et al., 2016), ogbn-Products (Hu et al., 2020a) and Reddit (Huang et al., 2024), covering academic, e-commerce and social network scenarios. The target datasets contain Cora (Sen et al., 2008), CiteSeer (Giles et al., 1998), Photo, Computers, History (Yan et al., 2023), WikiCS (Mernyei & Cangea, 2020) and Instagram (Huang et al., 2024), spanning academic, e-commerce, Wikipedia and social domains. We train the model on source domains and directly conduct inference on target domains. With substantial differences in graph scales and semantics, this diverse collection of datasets can well evaluate the generalization capability of models on cross-dataset node classification tasks.

Metric	*ogbn-ArXiv*	*ArXiv_2023*	*PubMed*	*ogbn-Products*	*Reddit*	*Cora*	*CiteSeer*	*Photo*	*Computers*	*History*	*WikiCS*	*Instagram*
Nodes	169,343	46,198	19,717	2,449,029	33,434	2,708	3,186	48,362	87,229	41,551	11,701	11,339
Edges	1,166,243	78,543	44,338	61,859,140	198,448	5,429	4,277	500,928	721,081	358,574	215,863	144,010
Classes	40	40	3	47	2	7	6	12	10	12	10	2
Domain	Academic			E-commerce	Social	Academic		E-commerce			Wikipedia	Social
Split	G_{source}					G_{target}						

Table 1: Summary statistics of the employed text-attributed graph datasets. Datasets are split into source domain G_{source} and target domain G_{target}.

5.2 BASELINE METHODS

To enable comprehensive comparisons, we divide existing baseline methods into four categories. (1) *Text-only classification models*: BERT (Devlin et al., 2019), SBERT (Reimers & Gurevych, 2019), DeBERTa (He et al., 2020), and E5 (Wang et al., 2022) classify nodes solely based on raw textual features, without leveraging graph structure. (2) *Models integrating node text and graph structure*: GraphGPT (Tang et al., 2024) and LLaGA (Chen et al., 2024) are representative graph foundation models that jointly model node text and graph topology. In addition, Qwen2 (7B/72B) (Yang et al., 2024) and LLaMA3.1 (8B/70B) (Sam, 2024) encode graph topology through textually serialized graph prompts, enabling large language models to capture structural context. (3) *Graph self-supervised learning methods*: OFA (Liu et al., 2024), ZeroG (Li et al., 2024), DGI (Veličković et al., 2018), GRACE (Zhu et al., 2020), BGRL (Thakoor et al., 2021), GraphMAE (Hou et al., 2022), G2P2 (Wen & Fang, 2023), and GraphCLIP (Zhu et al., 2025). These methods learn high-quality node representations from unlabeled data. (4) *CLIP prompt optimization methods*: MPE (Dong et al., 2026) adopts uniform aggregation over multiple prompt embeddings. WPE (Dong et al., 2026) estimates global class-agnostic weights from unlabeled data to reweight prompt embeddings. CARPRT (Dong et al., 2026) further introduces class-specific prompt weights in a training-free manner, capturing the varying relevance between prompts and different categories. The selected baselines cover mainstream technical routes, and we will conduct comprehensive comparisons with our proposed method.

5.3 EXPERIMENTS RESULTS

Main results and comparison with state-of-the-art methods:

Table 2 presents the cross-dataset node classification accuracy of different methods across seven benchmark datasets, with implementation details provided in Appendix A.2. Our method achieves the best average accuracy of 63.44%, outperforming the second-best baseline GraphCLIP (62.01%) and all other competing models. CARPRT is also a training-free method that optimizes classification prompt templates; compared with it, our method demonstrates significant performance advantages. This fully validates that our method can effectively unleash the cross-dataset classification potential of the CLIP model. Furthermore, although large language models such as Qwen2 and LLaMA3.1 have parameter scales far exceeding ours, our method still outperforms them in most cases. On the vast majority of datasets, our model achieves leading results, demonstrating strong cross-dataset generalization capability.

Methods	Params	Cora	CiteSeer	WikiCS	Instagram	Photo	Computers	History	Avg
BERT	110M	$19.56_{\pm 0.98}$	$33.26_{\pm 2.35}$	$29.37_{\pm 0.00}$	$57.02_{\pm 0.57}$	$21.80_{\pm 0.14}$	$13.88_{\pm 0.29}$	$9.95_{\pm 0.42}$	26.30
SBERT	66M	$54.35_{\pm 1.26}$	$50.47_{\pm 0.90}$	$48.16_{\pm 0.00}$	$48.34_{\pm 1.23}$	$35.96_{\pm 0.44}$	$41.82_{\pm 0.22}$	$30.45_{\pm 0.19}$	44.22
DeBERTa	184M	$18.21_{\pm 1.34}$	$16.42_{\pm 1.26}$	$15.29_{\pm 0.00}$	$39.81_{\pm 0.58}$	$12.38_{\pm 0.26}$	$10.62_{\pm 0.15}$	$8.70_{\pm 0.26}$	17.35
E5	110M	$44.65_{\pm 0.82}$	$42.57_{\pm 0.54}$	$31.49_{\pm 0.00}$	$61.28_{\pm 0.97}$	$35.14_{\pm 0.28}$	$16.54_{\pm 0.14}$	$12.92_{\pm 0.48}$	34.94
Qwen2	7B	$61.44_{\pm 1.29}$	$60.57_{\pm 0.86}$	$60.72_{\pm 0.00}$	$49.13_{\pm 0.78}$	$45.55_{\pm 0.12}$	$59.18_{\pm 0.20}$	$23.79_{\pm 0.34}$	51.48
Qwen2	72B	$62.18_{\pm 0.98}$	$53.97_{\pm 0.87}$	$58.91_{\pm 0.08}$	$37.70_{\pm 0.31}$	$\mathbf{52.41}_{\pm 0.39}$	$60.88_{\pm 0.30}$	$\underline{53.56}_{\pm 0.64}$	54.23
LLaMA3.1	8B	$57.75_{\pm 1.21}$	$53.54_{\pm 1.71}$	$58.32_{\pm 0.21}$	$39.37_{\pm 1.14}$	$34.38_{\pm 0.25}$	$46.98_{\pm 0.21}$	$22.28_{\pm 0.18}$	44.66
LLaMA3.1	70B	$65.72_{\pm 1.24}$	$62.79_{\pm 1.24}$	$62.82_{\pm 0.04}$	$43.68_{\pm 0.52}$	$51.26_{\pm 0.53}$	$61.62_{\pm 0.42}$	$53.33_{\pm 0.55}$	57.32
GraphGPT	7B	$23.25_{\pm 1.45}$	$18.04_{\pm 1.45}$	$6.30_{\pm 0.26}$	$45.12_{\pm 1.16}$	$7.62_{\pm 0.22}$	$29.71_{\pm 0.83}$	$15.92_{\pm 0.14}$	20.85
LLaGA	7B	$21.44_{\pm 0.65}$	$16.07_{\pm 1.15}$	$2.65_{\pm 0.72}$	$41.12_{\pm 0.94}$	$6.50_{\pm 0.53}$	$23.10_{\pm 0.33}$	$11.17_{\pm 0.58}$	17.44
OFA	30M	$37.25_{\pm 1.38}$	$29.64_{\pm 0.19}$	$45.52_{\pm 1.06}$	$32.71_{\pm 0.16}$	$33.03_{\pm 0.64}$	$22.09_{\pm 0.39}$	$16.87_{\pm 0.93}$	31.02
ZeroG	66M	$62.32_{\pm 1.91}$	$52.55_{\pm 1.23}$	$54.93_{\pm 0.06}$	$48.97_{\pm 0.78}$	$45.12_{\pm 0.65}$	$56.20_{\pm 0.35}$	$40.74_{\pm 0.65}$	51.55
DGI	128M	$24.03_{\pm 1.40}$	$18.71_{\pm 1.22}$	$18.86_{\pm 0.25}$	$61.42_{\pm 1.12}$	$13.96_{\pm 0.17}$	$27.12_{\pm 0.03}$	$15.77_{\pm 0.02}$	25.70
GRACE	128M	$13.69_{\pm 1.27}$	$22.88_{\pm 1.49}$	$16.07_{\pm 0.32}$	$62.23_{\pm 0.93}$	$10.16_{\pm 0.13}$	$10.94_{\pm 0.12}$	$32.39_{\pm 0.11}$	24.04
BGRL	128M	$20.80_{\pm 1.06}$	$26.50_{\pm 1.22}$	$18.35_{\pm 0.22}$	$61.45_{\pm 0.82}$	$5.21_{\pm 0.22}$	$24.12_{\pm 0.22}$	$16.28_{\pm 0.35}$	24.67
GraphMAE	128M	$23.25_{\pm 1.07}$	$20.75_{\pm 0.88}$	$12.14_{\pm 0.20}$	$62.39_{\pm 0.84}$	$12.53_{\pm 0.08}$	$8.36_{\pm 0.06}$	$21.76_{\pm 0.17}$	23.03
G2P2	63M	$41.51_{\pm 0.78}$	$51.02_{\pm 0.62}$	$31.92_{\pm 0.15}$	$52.87_{\pm 0.78}$	$22.21_{\pm 0.12}$	$32.52_{\pm 0.13}$	$26.18_{\pm 0.25}$	36.89
GraphCLIP	150M	$\underline{66.42}_{\pm 1.89}$	$\underline{66.61}_{\pm 1.48}$	$\underline{71.76}_{\pm 0.30}$	$\underline{63.67}_{\pm 0.31}$	$51.36_{\pm 0.05}$	$\underline{62.91}_{\pm 0.62}$	$51.35_{\pm 0.26}$	$\underline{62.01}$
MPE	150M	$63.84_{\pm 1.92}$	$65.99_{\pm 1.09}$	$67.39_{\pm 0.25}$	$61.90_{\pm 0.66}$	$49.73_{\pm 0.17}$	$47.41_{\pm 0.10}$	$24.75_{\pm 0.13}$	54.43
WPE	150M	$63.68_{\pm 1.91}$	$65.32_{\pm 1.17}$	$67.47_{\pm 0.24}$	$61.85_{\pm 0.57}$	$49.05_{\pm 0.17}$	$46.21_{\pm 0.16}$	$24.39_{\pm 0.13}$	54.00
CARPRT	150M	$63.71_{\pm 1.91}$	$65.88_{\pm 1.22}$	$67.42_{\pm 0.24}$	$61.90_{\pm 0.66}$	$49.72_{\pm 0.17}$	$47.34_{\pm 0.10}$	$24.77_{\pm 0.14}$	54.39
Ours	152M	$\mathbf{68.02}_{\pm 1.99}$	$\mathbf{68.81}_{\pm 1.55}$	$\mathbf{72.13}_{\pm 0.22}$	$\mathbf{64.20}_{\pm 0.38}$	$\underline{51.81}_{\pm 0.33}$	$\mathbf{63.94}_{\pm 0.57}$	$\mathbf{55.18}_{\pm 0.16}$	**63.44**

Table 2: Node classification accuracy (%) on seven datasets.

Ablation study:

Table 3 presents the ablation results under two settings: complete classification templates with descriptive text (Desc ✓) and category-only templates without descriptions (Desc ×). When templates lack detailed descriptions (Desc ×), the overall accuracy drops sharply; for example, the base variant decreases from 62.01% to 53.76%. However, introducing Perturbation Alignment (PA) substantially mitigates this performance degradation.

Quantitatively, across multiple benchmark datasets, under the Desc ✓ setting, replacing average pooling with gated pooling alone (w/o. PA) yields an average gain of +0.48% (62.01% → 62.49%), while PA alone (w/o. gated) contributes +0.93% (62.01% → 62.94%) and increases the average classification margin by 0.78. Combining both achieves 63.44%, confirming their complementary effect (+1.43% over the base variant on average). Under the more challenging Desc × setting, gated pooling alone yields an average gain of +0.74% (53.76% → 54.50%), while PA alone contributes +3.41% (53.76% → 57.17%) and increases the average classification margin by 2.80—a substantially larger gain than the +0.78 observed under Desc ✓. The full model achieves 58.18% on average, recovering most of the performance drop (+4.42% over the base variant). This demonstrates that perturbation alignment is particularly crucial for compensating the information loss caused by unclear or incomplete templates, while the gated mechanism helps stabilize node interactions, bringing slight improvements across datasets.

Table 4 presents ablation results across different training data domains to analyze how source domain selection affects cross-dataset classification performance. We categorize the training datasets into three groups: Academic, E-commerce, and Social. The full model trained on all source domains with perturbation alignment achieves the best average accuracy of 63.44%. Removing partial

Methods	Desc	Cora	CiteSeer	WikiCS	Instagram	Photo	Computers	History	Avg
Ours	✓	$68.02_{\pm 1.99}$	$68.81_{\pm 1.55}$	$72.13_{\pm 0.22}$	$64.20_{\pm 0.38}$	$51.81_{\pm 0.33}$	$63.94_{\pm 0.57}$	$55.18_{\pm 0.16}$	63.44
w/o. PA	✓	$67.04_{\pm 2.31}$	$67.71_{\pm 1.05}$	$71.44_{\pm 0.16}$	$63.04_{\pm 0.32}$	$51.15_{\pm 0.03}$	$63.94_{\pm 0.56}$	$53.13_{\pm 0.26}$	62.49
w/o. gated	✓	$68.51_{\pm 2.29}$	$67.45_{\pm 1.62}$	$72.64_{\pm 0.21}$	$63.67_{\pm 0.31}$	$52.23_{\pm 0.09}$	$63.02_{\pm 0.64}$	$53.09_{\pm 0.26}$	62.94
base	✓	$66.42_{\pm 1.89}$	$66.61_{\pm 1.48}$	$71.76_{\pm 0.30}$	$63.67_{\pm 0.31}$	$51.36_{\pm 0.05}$	$62.91_{\pm 0.62}$	$51.35_{\pm 0.26}$	62.01
$\Delta m \uparrow (\times 10^{-2})$	✓	$0.28_{\pm 0.04}$	$1.98_{\pm 0.06}$	$0.01_{\pm 0.01}$	$0.01_{\pm 0.01}$	$2.99_{\pm 0.03}$	$0.26_{\pm 0.00}$	$-0.05_{\pm 0.02}$	0.78
Ours	×	$67.40_{\pm 2.90}$	$67.08_{\pm 0.56}$	$69.54_{\pm 0.26}$	$49.19_{\pm 0.46}$	$35.46_{\pm 0.18}$	$61.13_{\pm 0.31}$	$57.48_{\pm 0.20}$	58.18
w/o. PA	×	$66.05_{\pm 2.23}$	$66.72_{\pm 0.73}$	$67.44_{\pm 0.31}$	$43.37_{\pm 0.62}$	$28.00_{\pm 0.20}$	$57.09_{\pm 0.50}$	$52.80_{\pm 0.29}$	54.50
w/o. gated	×	$67.53_{\pm 3.30}$	$67.50_{\pm 1.21}$	$67.73_{\pm 0.42}$	$49.07_{\pm 0.62}$	$34.14_{\pm 0.27}$	$58.42_{\pm 0.40}$	$55.78_{\pm 0.12}$	57.17
base	×	$66.11_{\pm 2.26}$	$66.30_{\pm 0.90}$	$67.27_{\pm 0.17}$	$42.90_{\pm 0.54}$	$27.82_{\pm 0.34}$	$54.98_{\pm 0.50}$	$50.93_{\pm 0.35}$	53.76
$\Delta m \uparrow (\times 10^{-2})$	×	$3.36_{\pm 0.46}$	$1.55_{\pm 0.01}$	$-0.13_{\pm 0.02}$	$1.59_{\pm 0.12}$	$11.79_{\pm 0.15}$	$0.87_{\pm 0.04}$	$0.57_{\pm 0.02}$	2.80

Table 3: Ablation study under two template settings: complete templates (Desc ✓) and category-only templates (Desc ×). For each setting, all values represent classification accuracy (%) unless otherwise specified. Rows annotated with $\Delta m \uparrow$ report the theoretical classification margin improvement.

training data degrades target-domain accuracy to varying degrees under the vanilla setting without perturbation alignment: excluding academic datasets notably reduces Cora accuracy (from 67.04% to 55.35%); removing e-commerce datasets severely impairs Computers (from 63.94% to 47.05%); removing social datasets similarly degrades Instagram accuracy (from 63.04% to 47.65%).

The perturbation alignment strategy effectively mitigates the negative impact caused by missing training domains. When e-commerce datasets are removed, enabling perturbation alignment raises Computers from 47.05% to 52.59%. After removing social datasets, perturbation alignment boosts Instagram accuracy from 47.65% to 53.95%. A particularly representative phenomenon is that when no academic datasets are used for training, the classification accuracy on Cora is only 55.35% without perturbation alignment; with perturbation alignment, the accuracy on Cora surges to 71.16%, reaching the highest level among all experimental settings and even surpassing the 68.02% obtained with full training data. This case will be discussed in detail in the hyperparameter search paragraph. This indicates that even without any training on in-domain data, the model can still achieve competitive performance on the target classification task. The general representations learned from out-of-domain data merely require activation via PA—analogous to few-shot fine-tuning, but without parameter updates or labeled data.

Source-domain	PA	Cora	CiteSeer	WikiCS	Instagram	Photo	Computers	History	Avg
Full	✓	$68.02_{\pm 1.99}$	$68.81_{\pm 1.55}$	$72.13_{\pm 0.22}$	$64.20_{\pm 0.38}$	$51.81_{\pm 0.33}$	$63.94_{\pm 0.57}$	$55.18_{\pm 0.16}$	63.44
Full	×	$67.04_{\pm 2.31}$	$67.71_{\pm 1.05}$	$71.44_{\pm 0.16}$	$63.04_{\pm 0.32}$	$51.15_{\pm 0.03}$	$63.94_{\pm 0.56}$	$53.13_{\pm 0.26}$	62.49
w/o. Academic	✓	$71.16_{\pm 0.97}$	$66.61_{\pm 1.51}$	$68.33_{\pm 0.34}$	$63.36_{\pm 0.41}$	$52.81_{\pm 0.17}$	$65.41_{\pm 0.42}$	$54.56_{\pm 0.20}$	63.18
w/o. Academic	×	$55.35_{\pm 0.84}$	$64.05_{\pm 1.54}$	$65.04_{\pm 0.23}$	$63.24_{\pm 0.37}$	$51.50_{\pm 0.21}$	$64.94_{\pm 0.50}$	$51.86_{\pm 0.26}$	59.43
w/o. E-commerce	✓	$64.94_{\pm 2.49}$	$67.40_{\pm 1.09}$	$68.98_{\pm 0.28}$	$65.01_{\pm 0.31}$	$46.22_{\pm 0.29}$	$52.59_{\pm 0.38}$	$52.76_{\pm 0.14}$	59.70
w/o. E-commerce	×	$64.64_{\pm 1.81}$	$67.35_{\pm 1.26}$	$68.19_{\pm 0.34}$	$64.89_{\pm 0.45}$	$43.75_{\pm 0.18}$	$47.05_{\pm 0.68}$	$51.20_{\pm 0.34}$	58.15
w/o. Social	✓	$68.03_{\pm 1.83}$	$66.56_{\pm 0.63}$	$71.07_{\pm 0.31}$	$53.95_{\pm 0.38}$	$52.43_{\pm 0.27}$	$63.72_{\pm 0.56}$	$53.90_{\pm 0.19}$	61.38
w/o. Social	×	$67.34_{\pm 2.03}$	$66.30_{\pm 0.77}$	$70.87_{\pm 0.21}$	$47.65_{\pm 0.99}$	$51.23_{\pm 0.09}$	$63.57_{\pm 0.55}$	$51.74_{\pm 0.35}$	59.81

Table 4: Ablation study on seven datasets under different source domain training settings.

Few-Shot performance comparison:

To highlight the advantage of the perturbation alignment strategy, we conduct comparison experiments between our training-free perturbation alignment and few-shot fine-tuning methods. For a relatively fair comparison, few-shot fine-tuning only optimizes the text embeddings of classification templates, which is consistent with the optimization target of perturbation alignment. Detailed implementation is provided in Appendix A.3. As shown in Figure 3, the accuracy curves across varying shots demonstrate that perturbation alignment can achieve classification performance comparable to few-shot fine-tuning without updating any model parameters. On the Cora dataset, the perturbation-based model yields 68.02, outperforming GraphCLIP under Shot-0 (66.42) and Shot-1 (68.63), and approaching the optimal Shot-10 result (69.31). On CiteSeer and WikiCS, the perturbation-based model achieves performance comparable to few-shot fine-tuning, though it falls slightly short of the Shot-10 configuration. Furthermore, on datasets such as Instagram and Computers, few-shot fine-tuning fails to deliver stable improvements. Although perturbation alignment also cannot achieve obvious gains on these datasets, it effectively prevents performance degradation owing to controllable perturbation intensity, and consistently outperforms various baseline models. The consistent

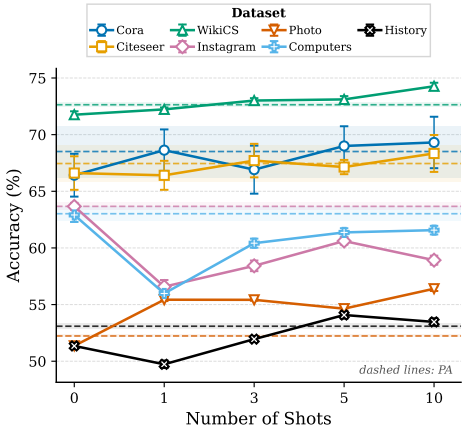


Figure 3: Accuracy comparison between few-shot fine-tuned classification and the directed perturbation strategy.

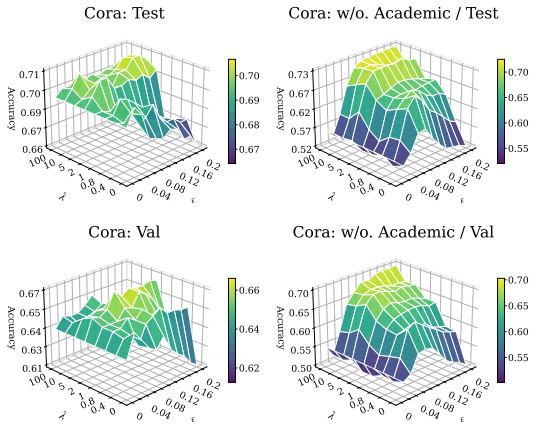


Figure 4: Visualization of hyperparameter (λ and ϵ) grid search on the Cora validation and test sets.

trends across multiple datasets demonstrate that graph foundation models inherently possess abundant cross-domain general representations. Simple parameter-free perturbation alignment is sufficient to activate the model potential.

Hyperparameter search:

To further investigate hyperparameter selection under different training domain settings, we conduct grid search experiments on the Cora dataset. Figure 4 presents 3D accuracy surfaces with respect to the composite objective weight λ and perturbation step size ϵ. The first column corresponds to full-data training, and the second to w/o. Academic (excluding academic source datasets). Upper sub-figures show validation results; lower sub-figures show test results. The highly consistent accuracy trends between validation and test sets indicate that validation performance reliably reflects test performance, justifying hyperparameter selection via grid search on the validation set.

A critical observation is that at $\epsilon = 0$, the gap between Full and w/o. Academic is substantial; after introducing PA, the gap narrows significantly, demonstrating that PA effectively activates cross-domain general representations. However, we must clarify that the higher test accuracy of w/o. Academic (71.16%) over Full (68.02%) under validation-selected hyperparameters is not indicative of superior model capacity. Rather, it reflects the limitation of grid search on the validation set, leading to a suboptimal solution. Comparing the test-set hyperparameter surfaces, Full consistently outperforms w/o. Academic. This reversed accuracy is a result of hyperparameter selection bias, not a genuine inversion of model capability. It corroborates that PA effectively bridges the cross-domain gap, while also highlighting the need for better hyperparameter selection to fully unlock the potential of the graph foundation model. All hyperparameter search results across all datasets are provided in Appendix A.4.

6 Conclusion

In this work, we proposed Perturbation Alignment (PA), a training-free adaptation framework that performs directed perturbations on classification template embeddings guided by subgraph encoding. We further introduced a dedicated gated pooling optimization tailored for the GraphCLIP graph encoder. Without requiring any parameter updates, PA effectively bridges the source–target domain gap and consistently outperforms existing baselines. Theoretically, we proved that the perturbation alignment mechanism guarantees an improvement in the classification margin. Together, these contributions offer a practical, parameter-efficient, and theoretically principled solution for cross-dataset graph node classification.

AI USE STATEMENT

In this work, we used generative AI tools for writing polishing and grammar correction. We have not used generative AI tools for experimental design, data collection, or result analysis, and image generation, video synthesis, and audio generation are not applicable to this work. Additionally, we used generative AI tools for brainstorming research ideas and drafting section outlines. We have reviewed all AI-assisted work. Specifically, we manually checked all LLM-polished text for factual accuracy, verified LLM-generated code through unit tests and manual debugging, and conducted a literature survey to ensure no plagiarism in AI-suggested ideas. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

The authors declare that this work does not raise any ethical concerns. No human subjects, personal data, or sensitive information were involved in this research. All datasets used in this work are publicly available benchmark datasets that do not contain personally identifiable information. There are no potential conflicts of interest or harmful applications associated with this work.

REPRODUCIBILITY STATEMENT

We provide comprehensive details to ensure reproducibility. All implementation details, including hyperparameters, and training configurations, are described in Appendix A.2 to A.4. Anonymous source code with training and evaluation scripts is submitted as supplementary material.

REFERENCES

William Brannon, Wonjune Kang, Suyash Fulay, Hang Jiang, Brandon Roy, Deb Roy, and Jad Kabbara. Congrat: Self-supervised contrastive pretraining for joint graph and text embeddings. In *Proceedings of TextGraphs-17: Graph-based Methods for Natural Language Processing*, pp. 19–39, 2024.

Runjin Chen, Tong Zhao, Ajay Jaiswal, Neil Shah, and Zhangyang Wang. Llaga: Large language and graph assistant. *arXiv preprint arXiv:2402.08170*, 2024.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pp. 4171–4186, 2019.

Ruijiang Dong, Zesheng Ye, Jianzhong Qi, Lei Feng, Feng Liu, Gang Niu, and Masashi Sugiyama. Carprt: Class-aware zero-shot prompt reweighting for vision-language model. In *International Conference on Learning Representations*, volume 2026, pp. 7859–7889, 2026.

C Lee Giles, Kurt D Bollacker, and Steve Lawrence. Citeseer: An automatic citation indexing system. In *Proceedings of the third ACM conference on Digital libraries*, pp. 89–98, 1998.

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

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*, 2020.

Xiaoxin He, Xavier Bresson, Thomas Laurent, Bryan Hooi, et al. Explanations as features: Llm-based features for text-attributed graphs. *arXiv preprint arXiv:2305.19523*, 2(4):8, 2023.

Zhenyu Hou, Xiao Liu, Yukuo Cen, Yuxiao Dong, Hongxia Yang, Chunjie Wang, and Jie Tang. Graphmae: Self-supervised masked graph autoencoders. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*, pp. 594–604, 2022.

Weihua Hu, Bowen Liu, Joseph Gomes, Marinka Zitnik, Percy Liang, Vijay Pande, and Jure Leskovec. Strategies for pre-training graph neural networks. *arXiv preprint arXiv:1905.12265*, 2019.

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. *Advances in neural information processing systems*, 33:22118–22133, 2020a.

Ziniu Hu, Yuxiao Dong, Kuansan Wang, Kai-Wei Chang, and Yizhou Sun. Gpt-gnn: Generative pre-training of graph neural networks. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 1857–1867, 2020b.

Qian Huang, Hongyu Ren, Peng Chen, Gregor Kržmanc, Daniel Zeng, Percy S Liang, and Jure Leskovec. Prodigy: Enabling in-context learning over graphs. *Advances in Neural Information Processing Systems*, 36:16302–16317, 2023.

Xuanwen Huang, Kaiqiao Han, Yang Yang, Dezheng Bao, Quanjin Tao, Ziwei Chai, and Qi Zhu. Can gnn be good adapter for llms? In *Proceedings of the ACM web conference 2024*, pp. 893–904, 2024.

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

Yuhan Li, Peisong Wang, Zhixun Li, Jeffrey Xu Yu, and Jia Li. Zerog: Investigating cross-dataset zero-shot transferability in graphs. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 1725–1735, 2024.

Hao Liu, Jiarui Feng, Lecheng Kong, Ningyue Liang, Dacheng Tao, Yixin Chen, and Muhan Zhang. One for all: Towards training one graph model for all classification tasks. In *International conference on learning representations*, volume 2024, pp. 20188–20210, 2024.

Zemin Liu, Xingtong Yu, Yuan Fang, and Xinming Zhang. Graphprompt: Unifying pre-training and downstream tasks for graph neural networks. In *Proceedings of the ACM web conference 2023*, pp. 417–428, 2023.

Péter Mernyei and Cătălina Cangea. Wiki-cs: A wikipedia-based benchmark for graph neural networks. *arXiv preprint arXiv:2007.02901*, 2020.

Sethupathy Parameswaran, Suresh Sundaram, and Yuan Fang. Prompt tuning without labeled samples for zero-shot node classification in text-attributed graphs. In *Proceedings of the Nineteenth ACM International Conference on Web Search and Data Mining*, pp. 520–530, 2026.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pp. 3982–3992, 2019.

Kira Sam. Llama 3.1: An in-depth analysis of the next-generation large language model. *Available at SSRN 6139407*, 2024.

Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Gallagher, and Tina Eliassi-Rad. Collective classification in network data. *AI magazine*, 29(3):93–106, 2008.

Jiabin Tang, Yuhao Yang, Wei Wei, Lei Shi, Lixin Su, Suqi Cheng, Dawei Yin, and Chao Huang. Graphgpt: Graph instruction tuning for large language models. In *Proceedings of the 47th international ACM SIGIR conference on research and development in information retrieval*, pp. 491–500, 2024.

Shantanu Thakoor, Corentin Tallec, Mohammad Gheshlaghi Azar, Rémi Munos, Petar Veličković, and Michal Valko. Bootstrapped representation learning on graphs. In *ICLR 2021 workshop on geometrical and topological representation learning*, 2021.

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

Petar Veličković, William Fedus, William L Hamilton, Pietro Liò, Yoshua Bengio, and R Devon Hjelm. Deep graph infomax. *arXiv preprint arXiv:1809.10341*, 2018.

Kuansan Wang, Zhihong Shen, Chiyuan Huang, Chieh-Han Wu, Yuxiao Dong, and Anshul Kanakia. Microsoft academic graph: When experts are not enough. *Quantitative Science Studies*, 1(1):396–413, 2020.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533*, 2022.

Changsong Wen, Zelin Peng, Yu Huang, Xiaokang Yang, and Wei Shen. Domain generalization in clip via learning with diverse text prompts. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9559–9569. IEEE, 2025.

Zhihao Wen and Yuan Fang. Augmenting low-resource text classification with graph-grounded pre-training and prompting. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 506–516, 2023.

Lianghao Xia and Chao Huang. Anygraph: Graph foundation model in the wild. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 882–896, 2026.

Lianghao Xia, Ben Kao, and Chao Huang. Opengraph: Towards open graph foundation models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 2365–2379, 2024.

Songlong Xing, Zhengyu Zhao, and Nicu Sebe. Clip is strong enough to fight back: Test-time counterattacks towards zero-shot adversarial robustness of clip. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15172–15182. IEEE, 2025.

Hao Yan, Chaozhuo Li, Ruosong Long, Chao Yan, Jianan Zhao, Wenwen Zhuang, Jun Yin, Peiyan Zhang, Weihao Han, Hao Sun, et al. A comprehensive study on text-attributed graphs: Benchmarking and rethinking. *Advances in Neural Information Processing Systems*, 36:17238–17264, 2023.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, et al. Qwen2. 5-math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*, 2024.

Zhilin Yang, William Cohen, and Ruslan Salakhudinov. Revisiting semi-supervised learning with graph embeddings. In *International conference on machine learning*, pp. 40–48. PMLR, 2016.

Hantao Yao, Rui Zhang, and Changsheng Xu. Visual-language prompt tuning with knowledge-guided context optimization. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6757–6767. IEEE, 2023.

Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. Graph contrastive learning with augmentations. *Advances in neural information processing systems*, 33: 5812–5823, 2020.

Xingtong Yu, Chang Zhou, Yuan Fang, and Xinming Zhang. Text-free multi-domain graph pre-training: Toward graph foundation models. *arXiv preprint arXiv:2405.13934*, 2024.

Jiaqi Zeng and Pengtao Xie. Contrastive self-supervised learning for graph classification. In *Proceedings of the AAAI conference on Artificial Intelligence*, volume 35, pp. 10824–10832, 2021.

Haihong Zhao, Aochuan Chen, Xiangguo Sun, Hong Cheng, and Jia Li. All in one and one for all: A simple yet effective method towards cross-domain graph pretraining. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 4443–4454, 2024a.

Jianan Zhao, Hesham Mostafa, Michael Galkin, Michael Bronstein, Zhaocheng Zhu, and Jian Tang. Graphany: A foundation model for node classification on any graph. *arXiv preprint arXiv:2405.20445*, 29, 2024b.

Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *2022 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 16795–16804. IEEE, 2022a.

Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International journal of computer vision*, 130(9):2337–2348, 2022b.

Beier Zhu, Yulei Niu, Yucheng Han, Yue Wu, and Hanwang Zhang. Prompt-aligned gradient for prompt tuning. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 15613–15623. IEEE, 2023.

Xingyu Zhu, Beier Zhu, Shuo Wang, Kesen Zhao, and Hanwang Zhang. Enhancing clip robustness via cross-modality alignment. *Advances in Neural Information Processing Systems*, 38:17553–17576, 2026.

Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. Deep graph contrastive representation learning. *arXiv preprint arXiv:2006.04131*, 2020.

Yun Zhu, Haizhou Shi, Xiaotang Wang, Yongchao Liu, Yaoke Wang, Boci Peng, Chuntao Hong, and Siliang Tang. Graphclip: Enhancing transferability in graph foundation models for text-attributed graphs. In *Proceedings of the ACM on Web Conference 2025*, pp. 2183–2197, 2025.

A APPENDIX

A.1 LIMITATIONS

Although PA requires no gradient-based optimization of model parameters, achieving optimal perturbation performance still relies on a held-out validation set to select the hyperparameters λ and ϵ via grid search. Thus, while parameter updates are entirely avoided, PA is not strictly free from validation data dependence. Future work may explore hyperparameter selection strategies that eliminate the need for grid search on a validation set, such as adaptive step-size rules or meta-learned perturbation schedules.

A.2 IMPLEMENTATION DETAILS AND COMPUTATIONAL OVERHEAD

Scale	#L	#H	Params	Cora	CiteSeer	WikiCS	Instagram	Photo	Computers	History	Avg
Small	4	512	35M	65.56±2.14	67.55±1.02	70.14±0.39	63.46±0.17	49.69±0.01	57.92±0.50	53.79±0.20	61.16
Medium	8	768	72M	67.84±2.93	68.23±1.28	71.35±0.29	62.57±0.44	51.80±0.25	61.32±0.60	54.55±0.37	62.52
Base	12	1024	152M	68.02±1.99	68.81±1.55	72.13±0.22	64.20±0.38	51.81±0.33	63.94±0.57	55.18±0.16	63.44
Large	16	1024	194M	69.00±2.03	68.29±1.12	70.76±0.19	60.77±0.36	52.38±0.11	64.65±0.49	53.28±0.43	62.73

Table 5: Cross-dataset classification accuracy (%) under different model scales. #L: number of layers; #H: hidden dimension; Params: total parameter count.

Hardware and inference overhead. All experiments are conducted on three NVIDIA A800 GPUs. The reported results are averaged over three independent runs with different random seeds for test set splitting, and the standard deviations are computed accordingly. To measure the computational overhead, we take the Cora dataset as an example and report the per-batch inference time: the base model requires 2.05s, while the base+PA, base+gated, and base+gated+PA variants all take 2.06s. This indicates that both the gating mechanism and perturbation alignment introduce negligible additional inference overhead.

Model scaling analysis. Table 5 summarizes the cross-dataset classification accuracy (%) across seven benchmark datasets under four model scales. We vary the model capacity by adjusting the hidden dimension and the number of transformer layers while keeping the input embedding dimension fixed at 384. Specifically, the Small scale uses 4 layers with a hidden dimension of 512 (35M parameters), the Medium scale uses 8 layers with a hidden dimension of 768 (72M parameters), the Base scale uses 12 layers with a hidden dimension of 1024 (152M parameters), and the Large scale uses 16 layers with a hidden dimension of 1024 (194M parameters). The Small, Medium, Base, and Large models achieve average accuracies of 61.16%, 62.52%, 63.44%, and 62.73%, respectively. Performance improves consistently from Small to Base, with the Base model reaching the best overall average. Further scaling to Large yields a slight decline in average accuracy, though Computers and Photo continue to benefit from the increased model capacity. These results demonstrate that the Base scale achieves the optimal trade-off between model capacity and generalization.

A.3 TRAINING AND FEW-SHOT EXPERIMENTAL DETAILS

Training Details. All models are trained for 30 epochs with a batch size of 2048. We use the Adam optimizer with a learning rate of 1×10^{-5} and a weight decay of 1×10^{-5}. For node attribute encoding, we employ the pre-trained all-MiniLM-L6-v2 model as the text encoder. During training, subgraphs are sampled via random walk with restart (probability 0.5, 64 steps); during inference, k-hop neighborhoods are used instead. The neighbor hop k is set to 2, and the subgraph sampler is configured as random walk.

Few-Shot Experimental Details. For few-shot evaluation, we adopt a prompt tuning strategy that keeps all pre-trained model parameters frozen and only optimizes a set of class-specific prompt vectors. Specifically, we define a learnable prompt vector for each class, which is initialized by uniform sampling from $[-0.1, 0.1]$ and added to the corresponding text embedding during training and inference. The prompt tuner is trained for 200 epochs using the AdamW optimizer with a learning rate of 1×10^{-4} and a weight decay of 1×10^{-5}. During prompt tuning, subgraph embeddings

and text embeddings are L_2-normalized, and the contrastive loss is computed via logits scaled by the learnable temperature parameter. For each dataset, we construct few-shot training batches by sampling a fixed number of labeled examples per class (determined by the `num_shots` setting) from the training split, while the text encoder processes class-specific templates filled with dataset-appropriate prompts (e.g., "Given a scientific paper, the main topic is $\{c\}$." for Cora). The tokenizer maximum length is set to 256 for training and 512 for testing. All few-shot results are reported as mean accuracy over three independently sampled test splits.

A.4 Hyperparameter selection for perturbation alignment in main experiments

In all main experiments, the perturbation step size ϵ and the composite objective weight λ are determined via grid search on a validation set. The search ranges are $\epsilon \in \{0.01, 0.02, \ldots, 0.20\}$ and $\lambda \in \{0, 0.2, 0.4, 0.6, 0.8, 1, 2, 5, 10, 100\}$. The selected hyperparameters for each dataset under all experimental settings are summarized in Table 6. As shown in the table, the optimal values vary substantially across datasets and settings, confirming that both ϵ and λ are dataset-dependent and that grid search is essential for achieving the best performance.

Table 6: Selected perturbation step size ϵ and composite objective weight λ for each dataset under all experimental settings.

Dataset	Base	Gated (w/o. desc)	Gated	w/o. Social	w/o. E-com	w/o. Academic	Large	Medium	Small
	Each cell shows (ϵ, λ)								
Cora	(0.19, 1)	(0.16, 10)	(0.07, 2)	(0.08, 2)	(0.08, 0)	(0.15, 100)	(0.07, 0.4)	(0.14, 0.6)	(0.09, 10)
CiteSeer	(0.10, 100)	(0.08, 1.5)	(0.13, 5)	(0.04, 0)	(0.08, 100)	(0.05, 1)	(0.10, 10)	(0.20, 100)	(0.05, 0.8)
WikiCS	(0.06, 0.4)	(0.06, 1)	(0.06, 0.2)	(0.06, 0.2)	(0.08, 100)	(0.11, 5)	(0.07, 10)	(0.05, 0.4)	(0.07, 0.2)
Instagram	(0, 0)	(0.19, 5)	(0, 100)	(0.20, 1)	(0.12, 0)	(0.01, 100)	(0.01, 100)	(0, 0)	(0.13, 0.2)
Photo	(0.15, 100)	(0.20, 100)	(0.14, 100)	(0.09, 0.4)	(0.17, 100)	(0.08, 0.4)	(0.11, 0.2)	(0.08, 0.4)	(0.09, 100)
Computers	(0.01, 100)	(0.07, 2.5)	(0.04, 2)	(0.03, 5)	(0.08, 2)	(0.05, 2.5)	(0.07, 0.2)	(0.04, 2.5)	(0.01, 100)
History	(0.06, 1.5)	(0.03, 0.8)	(0.06, 0.6)	(0.05, 0.6)	(0.05, 1)	(0.06, 0.8)	(0.11, 0.6)	(0.05, 0.8)	(0.02, 100)

A.5 Pseudocode

We summarize the Perturbation Alignment (PA) procedure in Algorithm 1.

Algorithm 1 Perturbation Alignment (PA) for cross-dataset node classification

Require: Target graph $\mathcal{G}$; graph encoder f_G; text encoder f_T; class templates $\{\tau_1, \ldots, \tau_N\}$; step size ϵ; weight λ

Ensure: Predicted class labels for nodes in $\mathcal{G}$

$\mathbf{G} \leftarrow f_G(\mathcal{G})$ {encode subgraphs into graph embeddings}

$\mathbf{T} \leftarrow f_T(\{\tau_1, \ldots, \tau_N\})$ {encode class templates into text embeddings}

$\mathbf{G} \leftarrow \mathbf{G}/\|\mathbf{G}\|_2,\ \mathbf{T} \leftarrow \mathbf{T}/\|\mathbf{T}\|_2$

$\mathcal{L}_{\text{comp}} \leftarrow \frac{1}{N}\sum_{i=1}^{N} \mathbf{G}_i^\top \mathbf{T}^\top$

$\mathcal{L}_{\text{deco}} \leftarrow \frac{1}{N(N-1)}\sum_{i \neq j} \mathbf{T}_i^\top \mathbf{T}_j$

$\mathcal{L} \leftarrow \lambda \mathcal{L}_{\text{comp}} + \mathcal{L}_{\text{deco}}$

$\mathbf{T}^* \leftarrow \mathbf{T} - \epsilon \cdot \nabla_{\mathbf{T}} \mathcal{L}$ {perturb text embeddings}

$\mathbf{T}^* \leftarrow \mathrm{UnitNorm}(\mathbf{T}^*)$

$\mathbf{S} \leftarrow \mathbf{G} \cdot (\mathbf{T}^*)^\top$ {cosine similarity matrix}

$\hat{\mathbf{y}} \leftarrow \arg\max(\mathbf{S}, \mathrm{dim} = 1)$ {predict class by highest similarity}

return $\hat{\mathbf{y}}$

A.6 Batch perturbation

In practice, our PA strategy operates in a batch-wise manner: all subgraphs within a batch jointly perturb a shared set of classification template embeddings. This is a natural consequence of batch inference and brings a practical benefit—aggregating signals from multiple subgraphs stabilizes the perturbation direction.

Table 7: Classification accuracy under varying batch sizes on Cora.

Batch Size	1	2	4	8	16	32	64	128	256	512
Accuracy (%)	67.41	67.35	68.08	66.79	65.99	68.08	67.34	67.28	68.02	68.04

As shown in Table 7, on the Cora dataset, PA yields accuracy gains across major batch sizes, with values ranging from 65.99% to 68.08%. Notably, even at a batch size of 1, the accuracy reaches 67.41%, confirming that the improvement stems from the perturbation mechanism itself rather than inter-sample aggregation. Performance stabilizes around 68.02%–68.04% for larger batches (256 and 512), while smaller batches exhibit higher variance—e.g., batch sizes 4 and 32 both peak at 68.08%, and 16 drops to 65.99%. This reveals a clear stabilization effect: the improvement from PA does not solely originate from the interactions among samples within a batch, but rather from the effective adjustment of template embeddings by the perturbation mechanism itself. As the batch size increases, signals from multiple subgraphs are aggregated, resulting in more stable and reliable perturbation directions and reduced performance fluctuation. The non-monotonic fluctuations observed at very small batch sizes (e.g., both 4 and 32 reaching 68.08%) reflect occasional "lucky batches" that happen to align well with the target classes but lack general reliability. When the batch size increases to 256, the performance plateaus, striking a favorable trade-off between perturbation stability and computational efficiency. Therefore, we uniformly adopt a batch size of 256 in all main experiments.