DAPrompt: Dual Alignment Prompt of Structure and Semantics for Few-shot Graph Learning

Lifan Jiang[1], Mengying Zhu[1]*, Yangyang Wu[1], Xuan Liu[2], Xiaolin Zheng[1], Shenglin Ben[1]
[1]Zhejiang University, China
[2]East China University of Science and Technology, China
{jianglf, mengyingzhu, zjuwuyy, xlzheng, benshenglin}@zju.edu.cn
xuanliu@ecust.edu.cn

Abstract

Few-shot graph learning remains a fundamental yet challenging problem, especially under heterophilic graph settings where connected nodes are likely to belong to different classes. In such scenarios, two key challenges arise: (1) unreliable or noisy graph structures that hinder effective message passing, and (2) semantic inconsistency: in heterophilic graphs, aggregating messages from neighbors of different classes entangles representations and introduces misleading semantics. These issues are further exacerbated by the limited labeled data inherent to few-shot learning, making it difficult to adaptively repair structure or disentangle semantics. To address these challenges, we propose DAPrompt, a Dual Alignment Prompt framework that jointly calibrates graph structure and semantic representations across the learning pipeline. In the pretraining stage, DAPrompt incorporates a graph structure learning module to denoise and repair the underlying topology, enhancing structural reliability. In the prompt tuning stage, we introduce two coordinated modules: a structure-aware prompt learner, which employs prompt tokens to repair unreliable graph structures and capture structure-level alignment, and a semantics-aligned prompt learner, which enhances the graph using target node semantics to mitigate representation noise caused by class-mismatched propagation. Extensive experiments on both node-level and graph-level few-shot benchmarks validate its effectiveness, achieving state-of-the-art performance and highlighting the value of structure-semantic dual alignment in heterophilic few-shot graph learning.

Code — https://github.com/JLifan/DAPrompt

1 Introduction

Graph Neural Networks (GNNs) (Wu et al. 2019; Gilmer et al. 2017) have emerged as powerful models for learning on graph-structured data due to their ability to capture complex node relationships (Han et al. 2025). However, their training usually demands substantial labeled data, which is often costly or infeasible in domains such as social computing or molecular chemistry. This challenge is particularly severe in few-shot graph learning (Zhang, Ding, and Li 2025), where labeled instances are extremely scarce. To address this, recent studies on graph pretraining (You et al. 2020; Hou et al. 2022; Xia et al. 2022a) learn transferable representations from unlabeled data before fine-tuning on downstream tasks, achieving notable gains under label-scarce settings.

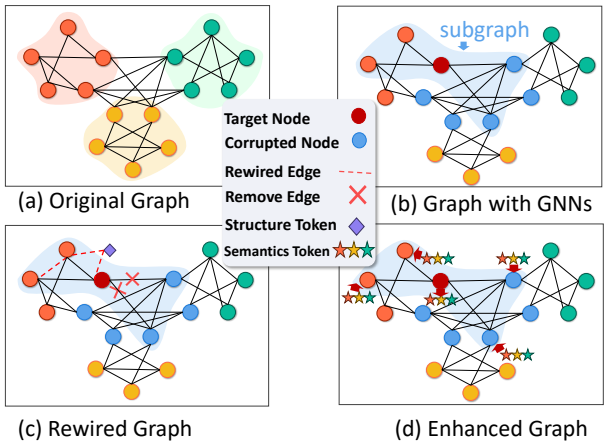

Figure 1: Motivation Example: Mitigating Representation Corruption via Structure Rewiring and Semantics Enhancing.

Despite the success of pretraining methods in graph learning, one significant limitation is the presence of the *objective gap* (Li et al. 2023; Fang et al. 2023). Pretraining tasks, e.g., node feature prediction (Hou et al. 2022) or graph structure reconstruction (You et al. 2020), are often poorly aligned with downstream objectives, e.g., node or graph classification. This misalignment hampers the effectiveness of knowledge transfer, as the learned representations may not fully capture the task-specific nuances for fine-tuning. This problem becomes even more pronounced in few-shot settings, where the model has to adapt to new tasks with limited labeled data.

To mitigate this gap, recent studies have introduced *prompt learning* into graph representation learning. Methods (Liu et al. 2023; Sun et al. 2023, 2022; Yan et al. 2025) employ learnable prompt tokens to guide model adaptation to downstream tasks. By injecting task-relevant prompts into the input space or model architecture, they help the model focus on salient semantic and structural cues and alleviate the misalignment between pretraining and fine-tuning. This paradigm shows strong potential in few-shot settings by making more effective use of limited supervision.

*Corresponding author

Although prompt learning has made notable progress in alleviating the objective gap, existing methods still struggle with the challenges of few-shot graph learning under **heterophilic** conditions, in which connected nodes often belong to different classes, violating the homophily assumption commonly adopted by GNNs and prompt-based methods.

As illustrated in Figure 1, when a heterophilic graph undergoes message passing and aggregation, semantic information from neighbors belonging to diverse classes may lead to **corrupted node representations** (highlighted in blue), ultimately resulting in unreliable predictions. This degradation, as seen in the transition from the original graph (Figure 1(a)) to its propagated form under a standard GNN (Figure 1(b)), primarily stems from two key factors:

(1) Unreliable graph structures: The original topology may contain noisy or missing edges, preventing the message-passing process from capturing reliable node representations and thus leading to misleading propagation that degrades prompt-based adaptation.

(2) Semantic inconsistency: In heterophilic graphs, neighboring nodes often belong to different classes, which makes them tend to reside in distinct semantic spaces. As shown in Figure 1(b), aggregating messages from such neighbors leads to entangled representations, where task-relevant semantics are diluted or overridden by misleading signals.

To address the challenges of few-shot graph learning under heterophilic conditions, we propose a unified framework, `DAPrompt`, which integrates structure-aware pretraining and prompt-based downstream tuning. In the pretraining stage, we devise a structure reconstruction strategy that jointly optimizes the encoder and graph topology, enabling the GNN backbone to learn robust structural priors from noisy graphs (*for addressing* **CH1**). During downstream tasks, `DAPrompt` employs two specialized prompt learners. First, the **structure-aware prompt learner** dynamically rewires the graph to improve connectivity and facilitate effective message propagation (see Figure 1(c)). Second, the **semantics-aligned prompt learner** injects learnable semantic tokens into each subgraph. These tokens interact bidirectionally with the target node to capture task-relevant semantics, and then broadcast this information unidirectionally to other nodes, promoting semantic consistency across the subgraph (*for addressing* **CH2**; see Figure 1(d)). Together, these components form a cohesive and robust framework for learning under both structural noise and scarce supervision.

To the best of our knowledge, this is the first work to simultaneously address both structural unreliability and semantic inconsistency in few-shot graph learning. Our main contributions are as follows:

(1) We identify two key challenges in few-shot graph learning: unreliable graph structures and semantic inconsistency, particularly under heterophilic conditions.

(2) We propose a unified framework that integrates structure-aware pretraining with two specialized prompt learners, i.e., a structure-aware module for topology refinement and a semantics-aware module for space alignment.

(3) We conduct extensive experiments on both node- and graph-level benchmarks, demonstrating the superiority of `DAPrompt` over existing pretrain and prompt methods.

2 Related Work

Graph Pretraining. Inspired by the success of pretraining in NLP (Beltagy, Lo, and Cohan 2019; Dong et al. 2019) and vision (Lu et al. 2019; Bao et al. 2021), graph pretraining has gained attention as a way to extract structural and semantic knowledge from unlabeled graphs (Xia et al. 2022b). Existing methods can be broadly categorized into edge-level modeling and contrastive learning. Edge-based methods typically predict missing or corrupted links (Kipf and Welling 2016; Long et al. 2022), or detect important structural patterns (Jin et al. 2020), thereby enhancing structural awareness. In contrast, contrastive learning techniques aim to maximize agreement between different graph views (You et al. 2020; Zhu et al. 2020; Zhang et al. 2024), or improve robustness via perturbation (Xia et al. 2022a; Zhao et al. 2024) and masked reconstruction (Hou et al. 2022).

Graph Prompt Tuning. Recently, prompt tuning has been widely explored in NLP as an efficient alternative to full fine-tuning (Brown et al. 2020; Lester, Al-Rfou, and Constant 2021; Liu et al. 2024). This idea has also been extended to graphs to enhance the adaptability of pretrained graph models to downstream tasks. GPPT (Sun et al. 2022) reformulates link prediction pretraining into node classification by injecting task-specific and structural cues through prompt tokens. ProG (Sun et al. 2023) appends learnable node-level prompts to each subgraph and connects them to the original graph via feature similarity, facilitating task adaptation. GraphPrompt (Liu et al. 2023) proposes a similarity-preserving pretraining objective and introduces a prompt-based transformation after the readout layer to support both node and graph classification. HeterGP (Yan et al. 2025) incorporates both homophilic and heterophilic views into the prompting process to better address heterophilic graph structures. Nevertheless, a key challenge remains in jointly aligning the structural and semantic spaces of graphs during prompting, ensuring consistency between the injected prompts and the underlying task-relevant information.

3 Preliminary and Problem Formulation

Unified Subgraph Classification. Here, we refer to subgraph classification as a unified abstraction that covers both node-level and graph-level. Node classification and graph classification are two fundamental tasks in graph-based learning. The former focuses on predicting labels for individual nodes within a single graph, while the latter assigns labels to entire graphs. According to the formulation in (Sun et al. 2023), both tasks are unified under a common perspective by treating them as subgraph classification problems. This abstraction provides a consistent basis for prompt-based modeling across different levels of granularity.

For node classification, the input is a subgraph $G_s = (V_s, E_s)$ centered at a target node v_i, where $V_s = N_K(v_i) \cup \{v_i\}$ includes the K-hop neighbors of v_i, and $E_s = \{(u, v) \mid u, v \in V_s\}$ denotes the edges among them. Although the original goal is to predict the label of v_i, this task can be equivalently reformulated as predicting the label of G_s, which reflects the class of the target node.

For graph classification, we treat the entire graph $G =$

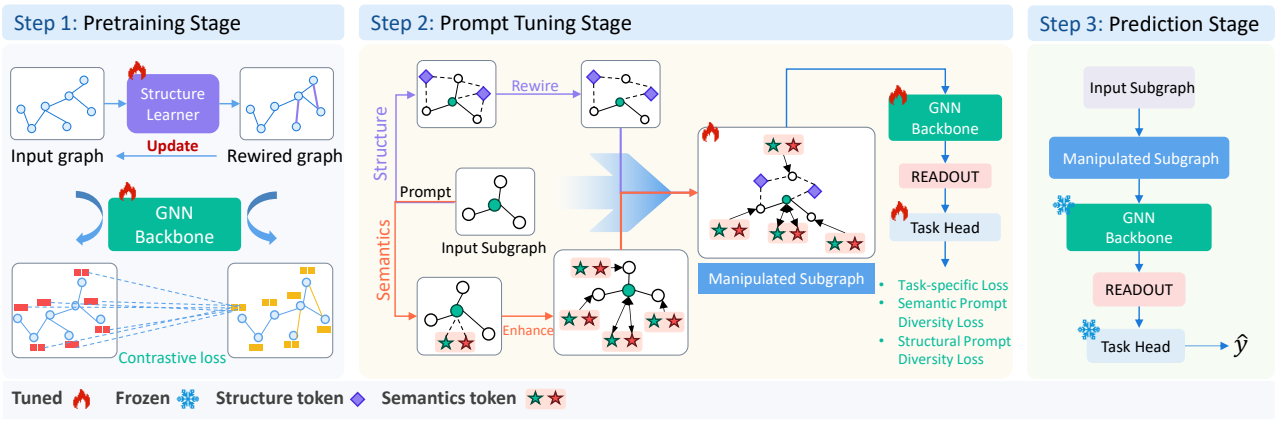

Figure 2: The pipeline of DAPrompt.

(V, E) as the subgraph input, i.e., $G_s = G$, and the task is to predict the label of the graph. Note that, in the case of graph classification, to maintain consistency with the unified subgraph formulation, we designate the node with the highest degree as the target node, denoted by $v_{\max} = \arg\max_{v \in V} \deg(v)$, where $\deg(v)$ is the number of edges incident to node v.

The goal in both cases is to learn a function f_θ that maps the subgraph G_s to its corresponding label. The final prediction $\hat{y}_i$ is obtained by applying a readout function over the node representations $h_u^{(K)} : u \in V_s$, followed by a classifier, i.e., $\hat{y}_i = \text{Classifier}(\text{Readout}(h_u^{(K)} : u \in V_s))$.

Few-shot Graph Learning. We now describe the problem formulation of few-shot graph learning. Given a collection of subgraphs $\{G_1, G_2, \ldots, G_N\}$, where each subgraph $G_i = (V_i, E_i)$ is associated with a node feature matrix $X_i \in \mathbb{R}^{|V_i| \times d}$ and a label $y_i \in Y$, the objective is to train a model that can generalize from a limited number of labeled subgraphs to correctly classify unseen ones.

Under the K**-shot** Q**-query** setting, a support set $D_S = \{(G_{s1}, y_1), \ldots, (G_{sm}, y_m)\}$ is provided, where $m = K \times C$ consists of K labeled subgraphs per class across C classes. Alongside, a query set $D_Q = \{(G_{q1}, y_1), \ldots, (G_{qn}, y_n)\}$ is given, where each query subgraph G_{qi} has an unknown label y_{qi}. The model must predict y_{qi} using the information from the support set.

4 Method

In this section, we introduce our proposed framework, DAPrompt. As shown in Figure 2, the framework consists of three sequential stages:

(1) *Pretraining Stage*: A GNN backbone is pretrained with a graph structure learning strategy to capture transferable structural representations in an unsupervised manner.

(2) *Prompt Tuning Stage*: Structure-aware and semantics-aligned prompt tokens are optimized on few-shot support sets to improve structural robustness and semantic consistency.

(3) *Prediction Stage*: The fine-tuned backbone and prompt tokens are jointly applied to classify unseen query subgraphs.

4.1 Pretraining Stage

The pretraining stage alleviates label scarcity via structural self-supervision. While prior methods (Jin et al. 2020; Xia et al. 2022a) assume a reliable graph topology, this often fails under heterophilic or noisy settings (Wang et al. 2021). To address this, we adopt a graph structure learning (GSL) strategy (Liu et al. 2022) that jointly optimizes the backbone and graph structure. As shown in Figure 2, a structure learner rewires the input graph based on semantic similarity, and both original and rewired graphs are encoded in parallel. A contrastive loss aligns their representations, enabling robust structure-aware embedding learning.

Formally, given an input graph $G = (V, E)$ with node features $\mathbf{X} \in \mathbb{R}^{N \times d}$ and adjacency matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$, our goal is to learn a refined adjacency matrix $\tilde{\mathbf{A}}$ that better captures semantic similarity between nodes. To this end, we employ a structure learner $\mathcal{S}_\phi$, which takes $(\mathbf{X}, \mathbf{A})$ as input and produces node embeddings $\mathbf{E} = \mathcal{S}_\phi(\mathbf{X}, \mathbf{A})$. In our implementation, $\mathcal{S}_\phi$ is instantiated as a stack of GCN layers (Kipf 2016).

Based on these embeddings, we compute a similarity matrix $\mathbf{S} \in \mathbb{R}^{N \times N}$ using cosine similarity:

$$S_{ij} = \frac{\mathbf{E}_i \cdot \mathbf{E}_j}{\|\mathbf{E}_i\|_2 \cdot \|\mathbf{E}_j\|_2}, \tag{1}$$

where $S_{ij} \in [-1, 1]$ reflects the semantic closeness between nodes i and j. A binarization step with a fixed threshold is applied to $\mathbf{S}$ to obtain the refined adjacency matrix $\tilde{\mathbf{A}}$, which preserves sparsity and enhances structural quality.

To gradually refine the graph topology, we update the original adjacency matrix $\mathbf{A}$ using the similarity matrix $\mathbf{S}$ via a momentum-based rule:

$$\mathbf{A} \leftarrow \alpha \mathbf{A} + (1 - \alpha) \cdot \mathbf{S}, \tag{2}$$

where $\alpha \in [0, 1]$ controls the update strength. A binarization step is then applied to preserve sparsity and ensure stable structural refinement.

We pass both the original graph $(\mathbf{X}, \mathbf{A})$ and the rewired graph $(\mathbf{X}, \tilde{\mathbf{A}})$ into a shared GNN backbone f_θ, yielding two sets of node embeddings: $\mathbf{Z} = f_\theta(\mathbf{X}, \mathbf{A})$ and $\mathbf{Z}' =$

$f_\theta(\mathbf{X}, \tilde{\mathbf{A}})$. To learn structure-invariant and noise-robust representations, we adopt a symmetric contrastive objective:

$$\mathcal{L}_{\text{con}} = -\frac{1}{2N}\sum_{i=1}^{N}\left(\ell(\mathbf{Z}_i, \mathbf{Z}'_i) + \ell(\mathbf{Z}'_i, \mathbf{Z}_i)\right), \tag{3}$$

where

$$\ell(\mathbf{Z}_i, \mathbf{Z}'_i) = \log \frac{\exp\left(\text{sim}(\mathbf{Z}_i, \mathbf{Z}'_i)/\tau\right)}{\sum_{j=1}^{N}\exp\left(\text{sim}(\mathbf{Z}_i, \mathbf{Z}'_j)/\tau\right)}, \tag{4}$$

and $\text{sim}(\mathbf{Z}_i, \mathbf{Z}'_j) = \frac{\mathbf{Z}_i \cdot \mathbf{Z}'_j}{|\mathbf{Z}_i|_2 \cdot |\mathbf{Z}'_j|_2}$ is the cosine similarity, with τ being a temperature parameter. The term $\ell(\mathbf{Z}'_i, \mathbf{Z}_i)$ is computed analogously by reversing the roles of inputs.

The pretraining process, detailed in Appendix A, enables the GNN backbone to capture intrinsic semantics while reducing the influence of noisy or heterophilic edges through iterative structure refinement and contrastive learning.

4.2 Prompt Tuning Stage

While the pretraining stage improves the GNN backbone's robustness to noisy structures, such imperfections still hinder downstream tasks, especially under heterophilic conditions. To further enhance task adaptation in the few-shot setting, we design a prompt-based fine-tuning framework from two perspectives: a *structure-aware prompt learner*, which rewires the subgraph to repair noisy connectivity, and a *semantics-aligned prompt learner*, which injects tokens to propagate the target node's semantics across the subgraph. Together, these modules guide the model toward more structure-aware and task-relevant representations under limited supervision.

Structure-aware Prompt Learner. In real-world graphs, noisy and heterophilic edges often disrupt effective message passing, degrading performance in downstream tasks such as node and graph classification. To mitigate this, we devise a *structure-aware prompt learner* that dynamically rewires subgraph topology during fine-tuning. As shown in Figure 2, a set of learnable structure prompt tokens is injected into each subgraph, acting as topological anchors that adaptively influence neighborhood connectivity. This refinement enhances robustness to unreliable edges and complements the GNN backbone through end-to-end supervision.

Formally, given an input subgraph $G_s = (V_s, E_s)$ with node features $\mathbf{X}_s \in \mathbb{R}^{N\times d}$, we introduce a set of learnable structure prompt tokens $\mathbf{P}_\mathbf{r} \in \mathbb{R}^{K_{\text{str}}\times d}$, which are appended to $\mathbf{X}_s$ to form an extended representation. We then compute pairwise cosine similarities among all nodes, including prompt tokens, to construct a similarity matrix $\mathbf{S}_{\text{str}} \in \mathbb{R}^{(N+K_{\text{str}})\times(N+K_{\text{str}})}$.

We threshold the similarity matrix $\mathbf{S}_{\text{str}}$ to obtain a sparse auxiliary adjacency matrix, which is added to the original adjacency $\mathbf{A}_s$. The result is passed through a sigmoid function and binarized to yield the final rewired adjacency $\tilde{\mathbf{A}}_{\text{str}} \in \{0,1\}^{(N+K_{\text{str}})\times(N+K_{\text{str}})}$, defining a new subgraph $G'_s = (V'_s, E'_s)$ with $V'_s = V_s \cup \mathcal{P}_r$. This process enables the model to refine subgraph connectivity by integrating structural priors with learned prompt-based relations.

Semantics-aligned Prompt Learner. To improve semantic consistency within subgraphs, we propose the *semantics-aligned prompt learner*, which propagates the target node's semantic information to its neighborhood. Unlike standard GNN aggregation that treats all nodes uniformly or relies solely on topological proximity, this module injects a set of learnable semantics prompt tokens to encode and broadcast the target node's latent representation to surrounding nodes. This design is particularly beneficial in heterophilic or noisy graphs, where semantic distortion often occurs during message passing. In both node and graph classification tasks reformulated as subgraph classification, the target node plays a central role in prediction, yet its representation may be diluted by global aggregation operations such as `Readout`. By introducing semantics-aligned prompt tokens, we explicitly align the subgraph representation with the semantics of the target node.

Let $\mathbf{x}_t \in \mathbb{R}^d$ denote the feature of the target node in a subgraph G_s. We initialize K_{sem} semantics tokens $\{\mathbf{s}_1, \ldots, \mathbf{s}_{K_{\text{sem}}}\} \in \mathbb{R}^{K_{\text{sem}}\times d}$, where K_{sem} defaults to the number of classes. These tokens are learnable parameters that are optimized during fine-tuning. To absorb semantic information, each token interacts with the target node via bidirectional message passing:

$$\mathbf{s}'_k = \mathcal{F}(\mathbf{s}_k, \mathbf{x}_t), \quad \mathbf{x}'_t = \mathcal{F}(\mathbf{x}_t, \mathbf{s}_k), \quad \forall k \in \{1, \ldots, K_{\text{sem}}\}, \tag{5}$$

where $\mathcal{F}(\cdot,\cdot)$ denotes a message passing function.

To propagate semantics, we then apply unidirectional message passing from each updated semantics token to the rest of the subgraph nodes $\mathbf{x}_i$ with $i \neq t$:

$$\mathbf{x}'_i = \mathcal{F}(\mathbf{x}_i, \mathbf{s}'_k), \quad \forall i \neq t, \quad \forall k \in \{1, \ldots, K_{\text{sem}}\}. \tag{6}$$

This mechanism allows the semantics of the target node to guide the representations of all nodes within the subgraph.

This directional design ensures that semantics tokens act as focused information broadcasters, encoding and disseminating target-specific information throughout the subgraph while avoiding contamination from noisy edges. As illustrated in Figure 2, the semantics tokens first interact with the target node to absorb task-relevant semantics and then propagate this information outward to other subgraph nodes. This mechanism facilitates the learning of more discriminative and semantically aligned subgraph representations, particularly under structurally unreliable conditions.

Prompt Tuning Objective. To optimize the prompt learners effectively for downstream subgraph classification, we design a composite loss function consisting of three parts: (1) the task-specific loss, (2) the semantic prompt diversity loss, and (3) the structural prompt diversity loss.

(1) Task-specific Loss: For each subgraph, we apply a classification loss based on the representation $\mathbf{h}_G$ obtained via graph-level readout. Specifically, the classification loss is defined as the standard cross-entropy loss:

$$\mathcal{L}_{\text{cls}} = \text{CE}(f_{\text{cls}}(\mathbf{h}_G), y), \tag{7}$$

where f_{cls} is a task-specific classifier, y is the ground-truth label of the subgraph, and $\mathbf{h}_G$ is the global pooled representation of the subgraph.

(2) Semantic Prompt Diversity Loss: To encourage the semantics tokens to encode distinct semantic concepts and avoid redundancy, we regularize them to be as orthogonal as possible. Let $\mathbf{P_s} \in \mathbb{R}^{K_{sem} \times d}$ be the semantic prompt matrix containing K_{sem} tokens of dimension d. We first normalize the prompt embeddings and then minimize the mean of pairwise cosine similarities excluding self-similarity:

$$\mathcal{L}_{\text{div}}^{\text{sem}} = \frac{1}{K_{sem}(K_{sem}-1)} \sum_{i \neq j} \left(\frac{\mathbf{P_s}_i \cdot \mathbf{P_s}_j}{\|\mathbf{P_s}_i\|_2 \cdot \|\mathbf{P_s}_j\|_2} \right), \quad (8)$$

which penalizes semantic overlap and promotes the diversity of prompt representations.

(3) Structural Prompt Diversity Loss: Similarly, for the structure-aware prompt tokens $\mathbf{P_r} \in \mathbb{R}^{K_{str} \times d}$, we impose the same regularization to avoid mode collapse and encourage token-level structural specialization:

$$\mathcal{L}_{\text{div}}^{\text{str}} = \frac{1}{K_{str}(K_{str}-1)} \sum_{i \neq j} \left(\frac{\mathbf{P_r}_i \cdot \mathbf{P_r}_j}{\|\mathbf{P_r}_i\|_2 \cdot \|\mathbf{P_r}_j\|_2} \right). \quad (9)$$

Overall Objective: The total optimization objective combines the three terms with balancing coefficients λ_1 and λ_2:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{cls}} + \lambda_1 \mathcal{L}_{\text{div}}^{\text{sem}} + \lambda_2 \mathcal{L}_{\text{div}}^{\text{str}}. \quad (10)$$

This overall loss ensures that the learned prompts are not only task-effective but also diverse and semantically disentangled. The full pipeline of the prompt-based fine-tuning stage, which incorporates both structure-aware and semantics-aligned prompt learners, is detailed in Appendix A.

4.3 Prediction Stage

As illustrated in Figure 2, during the prediction stage, we augment the test-time subgraph $\mathcal{G}_{\text{input}}$ with the learned structure and semantics prompts to obtain $\mathcal{G}_{\text{manip}}$, which is then processed by the fine-tuned backbone f_θ and a classification head f_{cls} to produce the final prediction:

$$\hat{y} = f_{\text{cls}}(\text{READOUT}(f_\theta(\mathcal{G}_{\text{manip}}))). \quad (11)$$

This enables robust inference even under heterophilic or noisy graph conditions.

4.4 Discussion

We provide an in-depth analysis of the proposed `DAPrompt` framework by examining the design principles and observed behaviors of its two core components: the *structure-aware prompt learner* and the *semantics-aligned prompt learner*.

Discussion on Graph Rewiring Strategies. Ideally, GNNs achieve optimal performance on an oracle graph with adjacency $\mathbf{A}^*$ and features $\mathbf{X}^*$:

$$f(\mathbf{A}^*, \mathbf{X}^*) \text{ yields optimal performance.} \quad (12)$$

However, in heterophilic graphs, the observed $\mathbf{A}$ and $\mathbf{X}$ are often noisy or misaligned.

As shown in (Alon and Yahav 2020), for any transformation $g(\mathbf{A}, \mathbf{X})$, there exists a prompt $\hat{p}$ such that:

$$f(\mathbf{A}, \mathbf{X} + \hat{p}) = f(g(\mathbf{A}, \mathbf{X})) + O_{pf}, \quad (13)$$

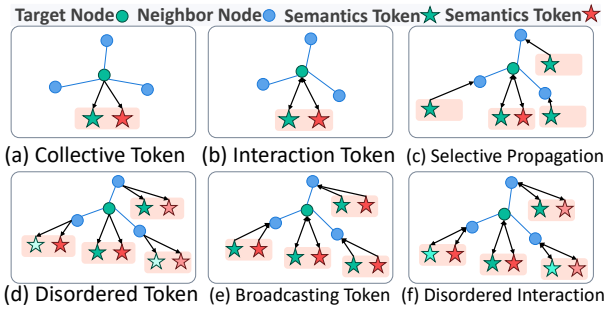

Figure 3: Connection strategies for semantic enhancing.

indicating that one can approximate the effect of graph transformation through learned prompts with bounded error.

Works like ProG exploit this by inserting a prompt graph $\mathcal{G}_p$ into the original graph, effectively pushing the model's input toward a better feature distribution $\mathbf{X}^*$. This is achieved via:

$$f(\psi(\mathcal{G} + \mathcal{G}_p)) = f(\mathbf{A}, \mathbf{X}^*) + O_{pf}^{ProG}. \quad (14)$$

However, this method leaves the original structure $\mathbf{A}$ untouched, meaning that the model still suffers from message-passing over noisy or irrelevant edges. In other words, it only moves the representation toward $\mathbf{X}^*$ but not $\mathbf{A}^*$.

In contrast, our method explicitly manipulates the graph structure itself through a combination of edge addition and edge deletion:

$$f(\psi(\mathcal{G} + \mathcal{G}_p^+ - \mathcal{G}_p^-)) = f(\mathbf{A}^*, \mathbf{X}^*) + O_{pf}^{DAPrompt}. \quad (15)$$

This structure-aware prompting not only enriches the features but also rewires the graph topology to better approximate $\mathbf{A}^*$.

Empirically, we observe that this rewiring mechanism yields tighter alignment with the ideal decision boundary and results in a smaller approximation error, i.e., $O_{pf}^{DAPrompt} < O_{pf}^{ProG}$, demonstrating the superiority of rewire-based structure correction over simple prompt graph insertion.

Discussion on Semantic Alignment Strategies. To clarify the role of semantic tokens, we compare six design variants.

`Collective Token`, the simplest variant (Figure 3(a)), enhances global representation at the readout stage, leveraging learned semantics. `Interaction Token` (Figure 3(b)) introduces bidirectional interaction with the target node, but shows limited improvement, suggesting its marginal utility.

For propagation, `Disordered Token` (Figure 3(d)) aggregates noisy neighbor messages and harms performance. In contrast, `Broadcasting Token` (Figure 3(e)) uses one-way propagation to pass target-node semantics outward, improving consistency and yielding the best results.

Then, we evaluate the `Disordered Interaction` variant (Figure 3(f)), which allows bidirectional exchange with neighbors, thereby polluting the semantic space. Our method, in contrast, avoids this by maintaining a unidirectional influence.

Finally, `Selective Propagation` (Figure 3(c)) limits its propagation to tokens most similar to the target, delivering strong empirical performance as an effective alternative.

Task	Dataset	#G	Avg. #N	Avg. #E	#F	#C	L.H.R
NC	Cora	1	2,708	10,556	1,433	7	0.81
	CiteSeer	1	3,327	9,104	3,703	6	0.74
	PubMed	1	19,717	88,648	500	3	0.80
	Texas	1	183	325	1,703	5	0.09
	Wisconsin	1	251	515	1,703	5	0.19
	Cornell	1	183	298	1,703	5	0.13
	Actor	1	7,600	30,019	932	5	0.22
	Chameleon	1	2,277	36,101	2,325	5	0.23
	Squirrel	1	5,201	217073	2,089	5	0.22
GC	ENZYMES	600	32.6	124.3	3	6	0.67
	PROTEINS	1,113	39.1	145.6	3	2	0.66
	MUTAG	188	17.9	39.6	7	2	0.72
	COX2	467	41.22	43.45	35	2	0.41
	BZR	405	35.75	38.36	53	2	0.41

Table 1: Summary of node classification (NC) and graph classification (GC) datasets

5 Experiment

In this section, we present extensive experiments to answer the following questions: **RQ1:** How does DAPrompt perform on few-shot node classification tasks compared to existing baselines? **RQ2:** How does DAPrompt perform on few-shot graph classification tasks compared to existing baselines? **RQ3:** How does each component of DAPrompt contribute to the overall performance?

5.1 Experimental Settings

Dataset. We evaluate DAPrompt on both node-level and graph-level few-shot classification tasks. For node classification (NC), we utilize 9 datasets covering homophilic (**Cora** (Yang, Cohen, and Salakhudinov 2016), **CiteSeer** (Yang, Cohen, and Salakhudinov 2016), and **PubMed**) and heterophilic (**Texas**, **Wisconsin**, **Cornell**, **Actor** (Pei et al. 2020), **Chameleon** (Pei et al. 2020), and **Squirrel** (Pei et al. 2020)) settings, with varying structures, label homophily ratios, and feature dimensions. For graph classification (GC), we evaluate on 5 standard benchmarks: **ENZYMES** (Wang et al. 2022), **PROTEINS**, **MUTAG**, **COX2**, and **BZR**. These datasets differ in graph sizes, class distributions, and domain semantics (e.g., bioinformatics and chemistry), making them suitable for testing the generalization ability of our model on few-shot graph-level tasks.

The detailed statistics of all datasets are summarized in Table 1, including the number of graphs (#G), average number of nodes (Avg. #N), edges (Avg. #E), features (#F), classes (#C), and the label homophily ratio (L.H.R).

Baselines. We compare DAPrompt with three categories of baselines: (1) supervised models, including GCN, GAT, GT, GraphSage, and GeomGCN; (2) pre-training + fine-tuning methods, including EdgeMask, GraphCL, GraphMAE, SimGRACE; and (3) pre-training + prompt-tuning methods, including GraphPrompt, ProG, GPPT, HeterGP, and our full model. Further descriptions of baselines and implementation details are provided in Appendix B.

Metrics. We adopt accuracy as the primary evaluation metric for both few-shot node and graph classification tasks. Experiments are conducted under 1-shot and 5-shot settings, and all results are averaged over 10 independent runs with different sampled episodes.

5.2 Overall Comparison (RQ1 & RQ2)

Node Classification Results. As shown in Table 2, under the 5-shot setting, our method consistently outperforms prior state-of-the-art methods on all benchmark datasets, demonstrating strong generalization ability across diverse graph structures. Although the datasets differ in structure and semantics, our method maintains consistently strong performance throughout. In the more challenging 1-shot setting, where supervision is extremely limited, our method remains effective, achieving superior or comparable results on most datasets. Notably, the performance gains are more pronounced on datasets with lower L.H.R, highlighting the method's particular strength in heterophilic scenarios characterized by structural noise and semantic inconsistency.

Moreover, in terms of best-performing counts across all datasets, our method achieves the highest number of top-ranking results in both 5-shot and 1-shot settings, outperforming all baselines. These results validate the robustness and adaptability of our method under both homophilic and heterophilic conditions.

Graph Classification Results. We evaluate our method on five benchmark datasets: *ENZYMES* (L.H.R = 0.67), *PROTEINS* (0.66), *MUTAG* (0.72), *COX2* (0.41), and *BZR* (0.41). As shown in Table 3, under the 5-shot setting, our method achieves the best performance on *PROTEINS*, *MUTAG*, and *COX2*. In the more challenging 1-shot setting, it achieves state-of-the-art results on four out of five datasets, with the exception of *ENZYMES*. These results further validate the effectiveness of our method in few-shot graph classification. In addition, our method consistently achieves the highest number of wins compared to any single baseline across both settings, demonstrating superior generalization ability. Notably, the improvements are especially pronounced on low-homophily datasets such as *COX2* and *BZR*, confirming the robustness of our design under heterophilic conditions.

5.3 Ablation Experiment (RQ3)

To better understand the effectiveness of each design choice in our framework, we conduct ablation studies from two perspectives: the contribution of individual components and the impact of different GNN backbones.

Ablation on Model Components. Figure 4 presents the ablation results by removing each core component of DAPrompt. We observe that all components make meaningful contributions to the overall performance. Notably, the pretraining module and the structure-aware prompt learner yield substantial improvements on homophilic datasets, indicating their effectiveness in repairing unreliable graph structures. In contrast, the semantics-aligned prompt learner shows greater impact on heterophilic datasets, highlighting its critical role in aligning semantics across structurally inconsistent neighborhoods. These findings validate the necessity of each module and demonstrate the complementary strengths of structural and semantic alignment in our framework.

Ablation on Backbones. Figure 5 illustrates the performance of DAPrompt when integrated with different GNN backbones. We observe that GCN consistently delivers the best overall performance across datasets, while GAT also performs strongly on most benchmarks. In comparison, GT,

Dataset	Supervised					Pre-train + Finetune				Pre-train + Prompt				
	GCN	GAT	GT	GraphSage	GeomGCN	EdgeMask	GraphCL	GraphMAE	SimGRACE	GraphPrompt	ProG	GPPT	HeterGP	Ours
5-shot														
Cora	67.29 ± 0.96	65.11 ± 0.73	64.97 ± 1.15	66.54 ± 0.88	67.52 ± 0.96	67.81 ± 0.64	66.47 ± 0.82	63.73 ± 0.58	64.33 ± 1.00	60.69 ± 0.84	68.70 ± 0.94	62.45 ± 0.92	69.87 ± 1.06	**70.08 ± 0.72**
CiteSeer	46.76 ± 1.18	47.99 ± 0.68	45.13 ± 1.03	46.95 ± 0.64	45.40 ± 0.76	48.79 ± 0.48	41.34 ± 0.74	44.04 ± 0.88	41.16 ± 0.90	36.12 ± 0.71	49.18 ± 0.81	41.49 ± 1.23	49.60 ± 1.93	**52.30 ± 0.73**
PubMed	67.88 ± 0.98	66.35 ± 1.17	65.85 ± 1.44	65.85 ± 1.49	67.79 ± 1.09	67.53 ± 0.99	65.98 ± 1.02	66.60 ± 1.30	66.73 ± 1.12	65.99 ± 0.72	60.27 ± 1.07	43.45 ± 1.50	65.49 ± 0.85	**67.99 ± 1.23**
Chameleon	28.32 ± 1.01	27.93 ± 0.46	28.70 ± 0.61	28.44 ± 0.66	24.92 ± 0.94	28.22 ± 0.84	27.72 ± 0.67	28.29 ± 0.97	27.22 ± 0.87	26.14 ± 0.86	26.20 ± 0.91	28.33 ± 0.67	28.51 ± 0.78	**29.72 ± 0.77**
Actor	20.73 ± 0.68	20.89 ± 1.20	20.83 ± 0.88	20.66 ± 0.84	20.42 ± 0.63	20.52 ± 0.70	20.36 ± 0.86	20.31 ± 0.77	20.28 ± 0.82	20.66 ± 1.00	20.53 ± 0.90	20.32 ± 0.62	19.62 ± 0.58	**21.56 ± 0.55**
Squirrel	21.53 ± 0.68	21.42 ± 0.65	21.52 ± 0.75	21.75 ± 0.89	23.78 ± 0.66	22.08 ± 0.59	21.81 ± 0.60	21.70 ± 0.87	21.49 ± 0.91	21.36 ± 0.55	21.44 ± 0.59	22.35 ± 0.83	21.38 ± 0.76	**24.63 ± 0.68**
Wisconsin	36.79 ± 0.93	37.34 ± 0.86	40.14 ± 1.10	40.61 ± 0.90	34.51 ± 0.87	37.20 ± 0.91	36.94 ± 0.88	36.88 ± 0.56	36.95 ± 1.16	34.90 ± 1.08	36.66 ± 0.99	33.72 ± 0.88	34.65 ± 0.97	**59.31 ± 1.26**
Cornell	30.40 ± 1.22	30.12 ± 0.80	35.79 ± 1.31	35.44 ± 1.00	26.99 ± 0.72	30.38 ± 1.40	29.71 ± 1.11	30.42 ± 0.83	29.40 ± 1.22	29.04 ± 0.98	30.67 ± 1.01	28.05 ± 0.97	26.58 ± 0.74	**51.23 ± 0.71**
Texas	30.03 ± 0.99	29.64 ± 0.89	33.90 ± 1.22	34.25 ± 1.18	29.13 ± 0.60	29.85 ± 0.96	29.42 ± 0.95	30.05 ± 0.94	29.72 ± 0.99	28.66 ± 0.81	31.72 ± 0.83	29.83 ± 0.67	31.43 ± 0.77	**54.54 ± 0.83**
1-shot														
Cora	33.46 ± 1.08	35.75 ± 0.96	35.87 ± 1.12	36.63 ± 0.87	32.63 ± 1.03	37.10 ± 0.87	28.03 ± 0.77	33.72 ± 0.54	27.66 ± 0.80	28.71 ± 0.67	38.77 ± 1.18	34.67 ± 0.92	**46.57 ± 1.21**	36.79 ± 0.73
CiteSeer	26.74 ± 1.15	27.00 ± 1.28	26.02 ± 0.90	27.39 ± 1.02	25.55 ± 1.31	30.16 ± 1.80	22.79 ± 1.04	24.81 ± 1.02	22.97 ± 0.86	20.89 ± 0.65	30.28 ± 1.31	25.19 ± 1.02	**32.05 ± 0.91**	29.63 ± 1.23
PubMed	49.65 ± 2.25	49.80 ± 1.84	47.95 ± 1.87	48.57 ± 2.07	47.05 ± 2.03	**53.63 ± 1.72**	48.87 ± 1.88	49.37 ± 1.74	48.56 ± 2.05	48.36 ± 1.71	45.36 ± 1.15	34.20 ± 0.72	52.61 ± 1.84	49.05 ± 1.86
Chameleon	24.68 ± 0.92	24.88 ± 0.75	24.73 ± 0.76	24.52 ± 0.48	21.87 ± 0.39	25.57 ± 0.98	24.36 ± 0.74	24.31 ± 1.01	24.02 ± 0.81	23.61 ± 0.70	22.00 ± 0.68	24.86 ± 0.80	24.36 ± 1.07	**25.65 ± 0.92**
Actor	20.30 ± 0.63	20.40 ± 0.55	20.31 ± 0.68	20.21 ± 0.56	20.14 ± 0.28	20.13 ± 0.30	20.45 ± 0.48	20.00 ± 0.48	**20.56 ± 0.53**	20.47 ± 0.50	20.28 ± 0.51	20.02 ± 0.38	20.00 ± 0.51	20.27 ± 0.41
Squirrel	20.32 ± 0.47	20.31 ± 0.56	20.97 ± 0.49	20.25 ± 0.39	22.57 ± 0.68	20.27 ± 0.71	20.37 ± 0.48	20.03 ± 0.44	20.43 ± 0.50	20.66 ± 0.67	21.45 ± 0.48	20.86 ± 0.80	20.43 ± 1.07	**22.77 ± 0.76**
Wisconsin	30.14 ± 1.16	29.67 ± 1.00	30.87 ± 0.89	30.90 ± 1.06	28.54 ± 1.00	29.72 ± 1.18	29.46 ± 1.13	29.43 ± 1.10	29.70 ± 0.74	28.26 ± 0.59	30.63 ± 0.95	27.82 ± 1.41	28.75 ± 0.75	**38.40 ± 1.11**
Cornell	23.64 ± 0.60	24.36 ± 0.46	26.28 ± 0.58	26.32 ± 0.62	22.65 ± 0.69	24.47 ± 0.91	24.31 ± 0.51	24.18 ± 0.48	23.76 ± 0.59	23.47 ± 0.86	24.88 ± 0.80	22.94 ± 0.82	22.97 ± 0.75	**27.62 ± 1.09**
Texas	27.31 ± 0.70	27.36 ± 1.11	29.16 ± 1.21	29.06 ± 0.95	27.84 ± 0.63	27.65 ± 1.09	26.74 ± 0.86	27.87 ± 0.61	27.97 ± 0.97	25.96 ± 0.75	28.23 ± 0.86	27.03 ± 0.90	27.69 ± 0.91	**33.22 ± 1.11**

Results are reported in percent. The best method is bolded and the runner-up is underlined.

Table 2: Few-shot node classification performance comparison across methods (mean ± std.).

Dataset	Supervised					Pre-train + Finetune				Pre-train + Prompt				
	GCN	GAT	GT	GraphSage	GeomGCN	EdgeMask	GraphCL	GraphMAE	SimGRACE	GraphPrompt	ProG	GPPT	HeterGP	Ours
5-shot														
ENZYMES	20.25 ± 0.77	20.05 ± 0.64	22.04 ± 0.59	20.43 ± 0.62	20.42 ± 0.54	20.86 ± 0.64	21.89 ± 0.48	20.56 ± 0.68	21.82 ± 0.68	**22.64 ± 0.36**	20.82 ± 0.84	17.80 ± 0.95	21.79 ± 1.03	20.43 ± 0.81
PROTEINS	54.79 ± 1.26	55.30 ± 0.93	56.38 ± 2.16	55.29 ± 1.37	54.39 ± 1.37	56.68 ± 1.63	56.79 ± 1.69	54.66 ± 1.69	56.01 ± 1.43	57.22 ± 1.98	55.33 ± 1.40	54.59 ± 0.93	61.27 ± 1.52	**64.30 ± 1.72**
MUTAG	63.48 ± 1.17	63.36 ± 1.72	65.33 ± 1.78	63.80 ± 1.10	61.93 ± 1.61	65.02 ± 1.02	62.54 ± 1.50	61.88 ± 1.36	64.71 ± 1.46	64.61 ± 1.18	65.99 ± 1.50	65.63 ± 1.01	68.80 ± 1.30	**70.62 ± 1.17**
COX2	53.30 ± 1.22	52.94 ± 1.30	54.88 ± 1.41	52.85 ± 1.88	51.60 ± 1.42	51.75 ± 1.36	55.02 ± 1.42	50.94 ± 1.13	55.36 ± 2.18	54.86 ± 1.28	51.10 ± 1.36	52.79 ± 1.31	51.86 ± 1.52	**55.85 ± 1.46**
BZR	57.31 ± 1.27	57.04 ± 1.51	57.96 ± 1.87	55.48 ± 1.51	54.02 ± 0.95	51.97 ± 1.10	**60.26 ± 1.36**	54.25 ± 1.00	58.65 ± 1.62	57.78 ± 1.20	52.03 ± 0.66	50.42 ± 1.12	56.48 ± 1.25	53.87 ± 0.96
1-shot														
ENZYMES	18.78 ± 0.63	19.00 ± 0.49	19.76 ± 0.84	19.16 ± 0.83	19.08 ± 0.61	18.59 ± 0.45	19.71 ± 0.61	18.81 ± 0.35	19.64 ± 0.66	**20.36 ± 0.50**	19.33 ± 0.54	17.04 ± 0.55	18.11 ± 0.61	19.22 ± 0.59
PROTEINS	54.27 ± 2.21	53.66 ± 1.94	54.06 ± 1.95	53.95 ± 1.96	53.79 ± 1.79	53.39 ± 1.85	53.01 ± 1.85	53.77 ± 1.65	53.40 ± 1.66	53.59 ± 1.61	53.90 ± 2.24	50.44 ± 1.16	54.58 ± 1.51	**55.07 ± 1.79**
MUTAG	58.32 ± 1.42	58.34 ± 1.71	58.42 ± 1.50	58.36 ± 1.38	58.34 ± 1.35	58.96 ± 1.48	57.17 ± 2.10	58.04 ± 1.42	59.26 ± 1.60	59.85 ± 1.49	59.60 ± 1.59	63.10 ± 1.60	59.81 ± 1.77	**65.57 ± 1.28**
COX2	51.56 ± 1.05	51.62 ± 1.60	52.11 ± 1.59	51.39 ± 1.49	51.41 ± 0.88	52.72 ± 1.11	51.84 ± 1.85	51.73 ± 1.69	52.65 ± 1.56	51.45 ± 1.91	50.79 ± 0.92	50.28 ± 0.34	51.35 ± 1.09	**53.64 ± 1.84**
BZR	52.81 ± 1.86	52.27 ± 1.74	51.86 ± 1.89	51.64 ± 1.77	52.08 ± 1.44	51.18 ± 1.64	53.70 ± 1.66	53.07 ± 2.26	53.35 ± 2.17	52.70 ± 1.81	50.46 ± 1.06	50.82 ± 0.54	52.21 ± 1.15	**53.74 ± 1.47**

Results are reported in percent. The best method is bolded and the runner-up is underlined.

Table 3: Few-shot graph classification performance comparison across methods (mean ± std.).

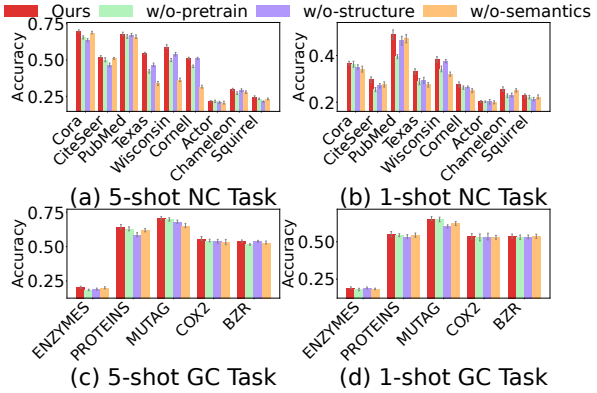

Figure 4: Comparison of Model Variants.

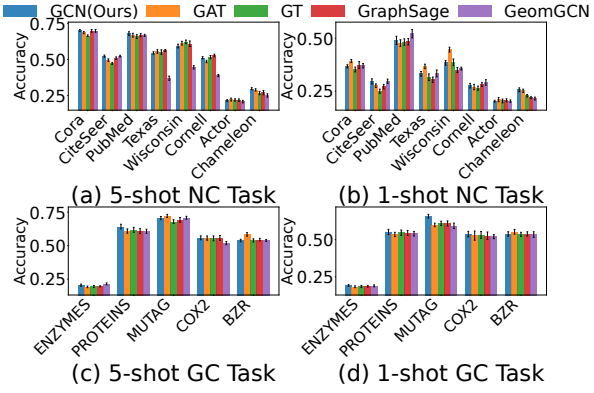

Figure 5: Ablation on Different Backbones.

GraphSAGE, and GeomGCN achieve competitive results on specific datasets, demonstrating the adaptability of our method to diverse architectures. These results highlight the flexibility and generality of DAPrompt in accommodating various GNN backbones, with GCN and GAT offering particularly robust and stable performance under both homophilic and heterophilic conditions.

6 Conclusion and Future Work

We propose DAPrompt, a dual-alignment prompt framework that mitigates structural unreliability and semantic inconsistency in few-shot graph learning. It integrates structure-aware pretraining with two prompt learners to enhance representations under heterophily settings. Experiments show that DAPrompt surpasses state-of-the-art methods. Future work will extend it to heterogeneous graphs.

Acknowledgements

This work was supported by the National Key R&D Program of China (2024YFC3307700) and the cooperation project of ZJU-ZTCB Financial Technology Joint Research Center.

References

Alon, U.; and Yahav, E. 2020. On the bottleneck of graph neural networks and its practical implications. *arXiv preprint arXiv:2006.05205*.

Bao, H.; Dong, L.; Piao, S.; and Wei, F. 2021. Beit: Bert pre-training of image transformers. *arXiv preprint arXiv:2106.08254*.

Beltagy, I.; Lo, K.; and Cohan, A. 2019. SciBERT: A pre-trained language model for scientific text. *arXiv preprint arXiv:1903.10676*.

Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901.

Dong, L.; Yang, N.; Wang, W.; Wei, F.; Liu, X.; Wang, Y.; Gao, J.; Zhou, M.; and Hon, H.-W. 2019. Unified language model pre-training for natural language understanding and generation. *Advances in neural information processing systems*, 32.

Fang, T.; Zhang, Y.; Yang, Y.; Wang, C.; and Chen, L. 2023. Universal prompt tuning for graph neural networks. *Advances in Neural Information Processing Systems*, 36: 52464–52489.

Gilmer, J.; Schoenholz, S. S.; Riley, P. F.; Vinyals, O.; and Dahl, G. E. 2017. Neural message passing for quantum chemistry. In *International conference on machine learning*, 1263–1272. Pmlr.

Han, S.; Zhou, Z.; Chen, J.; Hao, Z.; Zhou, S.; Wang, G.; Feng, Y.; Chen, C.; and Wang, C. 2025. Uncertainty-aware graph structure learning. In *Proceedings of the ACM on Web Conference 2025*, 4863–4874.

Hou, Z.; Liu, X.; Cen, Y.; Dong, Y.; Yang, H.; Wang, C.; and Tang, J. 2022. Graphmae: Self-supervised masked graph autoencoders. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*, 594–604.

Jin, W.; Derr, T.; Liu, H.; Wang, Y.; Wang, S.; Liu, Z.; and Tang, J. 2020. Self-supervised learning on graphs: Deep insights and new direction. *arXiv preprint arXiv:2006.10141*.

Kipf, T. 2016. Semi-Supervised Classification with Graph Convolutional Networks. *arXiv preprint arXiv:1609.02907*.

Kipf, T. N.; and Welling, M. 2016. Variational graph auto-encoders, CoRR abs/1611.07308. *arXiv preprint arXiv:1611.07308*.

Lester, B.; Al-Rfou, R.; and Constant, N. 2021. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*.

Li, W.-Z.; Wang, C.-D.; Xiong, H.; and Lai, J.-H. 2023. Homogcl: Rethinking homophily in graph contrastive learning. In *Proceedings of the 29th ACM SIGKDD conference on knowledge discovery and data mining*, 1341–1352.

Liu, X.; Zheng, Y.; Du, Z.; Ding, M.; Qian, Y.; Yang, Z.; and Tang, J. 2024. GPT understands, too. *AI Open*, 5: 208–215.

Liu, Y.; Zheng, Y.; Zhang, D.; Chen, H.; Peng, H.; and Pan, S. 2022. Towards unsupervised deep graph structure learning. In *Proceedings of the ACM Web Conference 2022*, 1392–1403.

Liu, Z.; Yu, X.; Fang, Y.; and Zhang, X. 2023. Graphprompt: Unifying pre-training and downstream tasks for graph neural networks. In *Proceedings of the ACM web conference 2023*, 417–428.

Long, Y.; Wu, M.; Liu, Y.; Fang, Y.; Kwoh, C. K.; Chen, J.; Luo, J.; and Li, X. 2022. Pre-training graph neural networks for link prediction in biomedical networks. *Bioinformatics*, 38(8): 2254–2262.

Lu, J.; Batra, D.; Parikh, D.; and Lee, S. 2019. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *Advances in neural information processing systems*, 32.

Pei, H.; Wei, B.; Chang, K. C. C.; Lei, Y.; and Yang, B. 2020. GEOM-GCN: GEOMETRIC GRAPH CONVOLUTIONAL NETWORKS. In *8th International Conference on Learning Representations, ICLR 2020*.

Sun, M.; Zhou, K.; He, X.; Wang, Y.; and Wang, X. 2022. Gppt: Graph pre-training and prompt tuning to generalize graph neural networks. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 1717–1727.

Sun, X.; Cheng, H.; Li, J.; Liu, B.; and Guan, J. 2023. All in one: Multi-task prompting for graph neural networks. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2120–2131.

Wang, R.; Mou, S.; Wang, X.; Xiao, W.; Ju, Q.; Shi, C.; and Xie, X. 2021. Graph structure estimation neural networks. In *Proceedings of the web conference 2021*, 342–353.

Wang, S.; Dong, Y.; Huang, X.; Chen, C.; and Li, J. 2022. Faith: Few-shot graph classification with hierarchical task graphs. *arXiv preprint arXiv:2205.02435*.

Wu, F.; Souza, A.; Zhang, T.; Fifty, C.; Yu, T.; and Weinberger, K. 2019. Simplifying graph convolutional networks. In *International conference on machine learning*, 6861–6871. Pmlr.

Xia, J.; Wu, L.; Chen, J.; Hu, B.; and Li, S. Z. 2022a. Simgrace: A simple framework for graph contrastive learning without data augmentation. In *Proceedings of the ACM web conference 2022*, 1070–1079.

Xia, J.; Zhu, Y.; Du, Y.; and Li, S. Z. 2022b. A survey of pretraining on graphs: Taxonomy, methods, and applications. *arXiv preprint arXiv:2202.07893*.

Yan, F.; Wang, X.; He, D.; Wang, L.; Dang, J.; and Jin, D. 2025. HeterGP: Bridging Heterogeneity in Graph Neural Networks with Multi-View Prompting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 21895–21903.

Yang, Z.; Cohen, W.; and Salakhudinov, R. 2016. Revisiting semi-supervised learning with graph embeddings. In *International conference on machine learning*, 40–48. PMLR.

You, Y.; Chen, T.; Sui, Y.; Chen, T.; Wang, Z.; and Shen, Y. 2020. Graph contrastive learning with augmentations. *Advances in neural information processing systems*, 33: 5812–5823.

Zhang, C.; Ding, K.; and Li, J. 2025. Few-shot learning on graphs. In *Handbook on Neurosymbolic AI and Knowledge Graphs*, 96–117. IOS Press.

Zhang, T.; Hou, C.; Jiang, R.; Zhang, X.; Zhou, C.; Tang, K.; and Lv, H. 2024. Label informed contrastive pretraining for node importance estimation on knowledge graphs. *IEEE Transactions on Neural Networks and Learning Systems*.

Zhao, H.; Chen, A.; Sun, X.; Cheng, H.; and Li, J. 2024. 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*, 4443–4454.

Zhu, Y.; Xu, Y.; Yu, F.; Liu, Q.; Wu, S.; and Wang, L. 2020. Deep graph contrastive representation learning. *arXiv preprint arXiv:2006.04131*.