HGMP:Heterogeneous Graph Multi-Task Prompt Learning

Pengfei Jiao[1] , **Jialong Ni**[1] , **Di Jin**[2] , **Xuan Guo**[2] and **Huan Liu**[1] , **Hongjiang Chen**[1] , **Yanxian Bi**[3*]

[1]School of Cyberspace Security,Hangzhou Dianzi University
[2]College of Intelligence and Computing,Tianjin University
[3]CETC Academy of Electronics and Information Technology Group,China Academy of Electronic and Information Technology
{nijialong, pjiao,huanliu,hchen}@hdu.edu.cn, {jindi,guoxuan}@tju.edu.cn, biyanxian@cetc.com.cn

Abstract

The pre-training and fine-tuning methods have gained widespread attention in the field of heterogeneous graph neural networks due to their ability to leverage large amounts of unlabeled data during the pre-training phase, allowing the model to learn rich structural features. However, these methods face the issue of a mismatch between the pre-trained model and downstream tasks, leading to suboptimal performance in certain application scenarios. Prompt learning methods have emerged as a new direction in heterogeneous graph tasks, as they allow flexible adaptation of task representations to address target inconsistency. Building on this idea, this paper proposes a novel multi-task prompt framework for the heterogeneous graph domain, named HGMP. First, to bridge the gap between the pre-trained model and downstream tasks, we reformulate all downstream tasks into a unified graph-level task format. Next, we address the limitations of existing graph prompt learning methods, which struggle to integrate contrastive pre-training strategies in the heterogeneous graph domain. We design a graph-level contrastive pre-training strategy to better leverage heterogeneous information and enhance performance in multi-task scenarios. Finally, we introduce heterogeneous feature prompts, which enhance model performance by refining the representation of input graph features. Experimental results on public datasets show that our proposed method adapts well to various tasks and significantly outperforms baseline methods.

1 Introduction

Heterogeneous graphs are powerful data structures that can simultaneously represent multiple types of nodes and edges, making them ideal for modeling complex relationships in various domains. For example, in social networks [Cao *et al.*, 2021], heterogeneous graphs can model different types of entities such as users, posts, and comments, as well as various types of relationships, including friendships, comments, and likes. This flexibility enables heterogeneous graphs to capture rich and multifaceted interactions between entities, making them highly suitable for applications that require integration and analysis of diverse information, such as knowledge graphs [Ji *et al.*, 2021; Hogan *et al.*, 2021], recommender systems [Gao *et al.*, 2023], and citation networks.

Graph Neural Networks (GNNs) [Yun *et al.*, 2019; Kipf and Welling, 2016; Veličković *et al.*, 2017] have demonstrated great performance in learning representations of graph-structured data by capturing local node relationships through message passing and aggregation. Heterogeneous Graph Neural Networks (HGNNs) [Mao *et al.*, 2023; Fu *et al.*, 2020; Yang *et al.*, 2020] are an extension of GNNs, designed to handle the complexities of heterogeneous graphs. HGNNs incorporate both node and edge type information, allowing message passing to be more specialized and more effectively capture heterogeneous relationships. These models show improvements over standard GNNs in heterogeneous environments, particularly for tasks where understanding the relationships between entities depends heavily on node and edge types. However, the performance of both standard GNNs and HGNNs is limited by the scarcity of labeled data and the generalization ability on out-of-distribution data [Hu *et al.*, 2019; Knyazev *et al.*, 2019; Yehudai *et al.*, 2021; Morris *et al.*, 2019]. Furthermore, in multi-task scenarios, each task requires separate training, which further increases computational complexity.

To address these limitations, pre-training and fine-tuning strategies [Dong *et al.*, 2019] have been widely adopted. During pre-training, models learn general feature representations from large-scale unlabeled data, reducing the dependence on labeled datasets and improving generalization. Fine-tuning is then applied to adapt the model to specific tasks. However, the mismatch between pre-training objectives and downstream task requirements can negatively impact performance [Liu *et al.*, 2023], especially in scenarios with label imbalance or few-shot learning, where models struggle to generalize to new tasks or unseen data [Zhang *et al.*, 2022].

Prompt learning, a rapidly emerging technique in the machine learning community, provides an attractive alternative. It embeds task-specific prompts into the input data, guiding the model to perform specific tasks. For example, a sentiment analysis task like "IJCAI2025 will bring together top

[*] Corresponding author.

researchers from around the world. I feel so [MASK]" can be reformulated using a preset prompt ("I feel so [MASK]"). It is anticipated that a language model trained on masked word prediction might predict "[MASK]" as "enthusiastic" rather than "disappointed," without requiring further parameter tuning for the new sentiment task, due to the model's pre-training that embeds relevant knowledge to address this type of question. This demonstrates that prompt learning can effectively leverage existing knowledge, enabling the model to generalize across tasks with minimal additional training.

The success of prompt learning in NLP [Li and Liang, 2021; Lester *et al.*, 2021] has inspired exploration into its application in graph learning. However, in graph learning, the use of prompt learning is still in its early stages. While some early works have explored the use of prompts in homogeneous graph settings [Fang *et al.*, 2024; Sun *et al.*, 2023a; Sun *et al.*, 2022; Liu *et al.*, 2023; Tan *et al.*, 2023; Huang *et al.*, 2024; Sun *et al.*, 2023b; Zi *et al.*, 2024], these approaches do not address the unique challenges posed by heterogeneous graphs. Even recent prompt learning methods specifically designed for heterogeneous graphs still face some challenges [Yu *et al.*, 2024; Ma *et al.*, 2024]. First, aligning the pre-training objectives with the requirements of downstream tasks remains complex, as many early methods focus on link prediction tasks that connect data, neglecting the richness of heterogeneous data. Additionally, methods designed for single-task settings limit their applicability in multi-task scenarios. The main challenges in heterogeneous graph prompt learning can be summarized as: (C1) *how to design unified graph templates for multi-task scenarios*, (C2) *how to develop pre-training methods that adapt to heterogeneous data*, and (C3) *how to design prompts that account for heterogeneity and feature diversity*.

To address these challenges, we propose the HGMP (Heterogeneous Graph Multi-task Prompt Learning) framework. Our approach rephrases node and edge-level tasks as graph-level tasks, aligning both downstream tasks and pre-training tasks in a common graph-level task space. This redefinition of tasks allows the model to undergo unified and effective training on graph-level objectives, capturing global graph structural features and enhancing the model's generalization ability. We introduce a heterogeneous pre-training strategy that combines subgraph extraction and augmentation for contrastive pre-training, fully leveraging the rich and diverse information in heterogeneous graphs. Furthermore, we design node-type-specific prompts using shared learnable vectors to modify node features, thereby enhancing the representation of input data and making it more suitable for the pre-training objectives.

In summary, the contributions of this paper are as follows:

- We propose a method that transforms node-level and edge-level tasks into graph-level tasks and introduces a flexible heterogeneous graph contrastive pre-training strategy to better capture the complexities of heterogeneous graph data.
- We design the HGMP framework, specifically tailored for multi-task heterogeneous graph scenarios, which effectively bridges the gap between pre-training and downstream task objectives, enhancing the model's generalization ability.
- We conduct comprehensive experiments on multiple public datasets, demonstrating the effectiveness, robustness, and adaptability of the HGMP framework across various task settings.

2 Related Work

Heterogeneous Graph Neural Networks. GNNs were initially designed for homogeneous graphs, focusing on node representation learning in graphs with consistent structures and attributes. To address the complexity of heterogeneous graphs, HGNNs emerged, enabling the modeling of diverse node and edge types. Representative works include HAN [Wang *et al.*, 2019], which employs hierarchical attention mechanisms, HGT [Hu *et al.*, 2020], a Transformer-based architecture for large-scale graphs, and Simple-HGN [Lv *et al.*, 2021], which enhances GAT [Veličković *et al.*, 2017] with learnable edge-type embeddings. Despite these advancements, HGNNs heavily rely on high-quality labeled data, which is often costly or infeasible to obtain, driving interest in pre-training techniques as alternatives to supervised learning.

Graph Pre-training. Graph pre-training aims to learn general knowledge from readily available data, reducing annotation costs. For homogeneous graphs, methods like GraphCL [You *et al.*, 2020] utilize contrastive learning to exploit graph structures. In heterogeneous graphs, pre-training methods must handle diverse semantic relationships. Generative approaches reconstruct masked structures, while contrastive learning maximizes mutual information between positive sample pairs [Jing *et al.*, 2021; Park *et al.*, 2020], often outperforming generative methods. However, misalignment between pre-training objectives and downstream tasks can hinder the effective utilization of pre-trained knowledge.

Prompt-based Learning On Graphs. Prompt learning bridges the gap between pre-training and downstream tasks by introducing task-specific prompts, reducing fine-tuning demands and enhancing adaptability. Methods like GPPT [Sun *et al.*, 2022], GPF [Fang *et al.*, 2024], and ProG [Sun *et al.*, 2023a] leverage prompts for task alignment and adaptation. In heterogeneous graphs, approaches such as HGPrompt [Yu *et al.*, 2024] and HetGPT [Ma *et al.*, 2024] use templates to support few-shot learning and align pre-training with downstream tasks. Nonetheless, current graph prompt learning methods often struggle with multi-task demands or rely on rigid templates, limiting flexibility and effectiveness.

3 Preliminaries

Heterogeneous Graph. In graph data structures, a Heterogeneous Graph is an extended form designed to model complex networks with different types of nodes and edges. Compared to a homogeneous graph, a heterogeneous graph can more accurately represent diverse entities and their relationships in the real world. Formally, a heterogeneous graph can be defined as $G = (V, E, T_V, T_E)$, where V represents the set of nodes, E represents the set of edges, and T_V and T_E denote the sets of node types and edge types, respectively. In this definition, each node $v \in V$ is identified by a node type mapping $\phi(v) \in T_V$, and each edge $e \in E$ is associated with an edge type mapping $\psi(e) \in T_E$.

In addition to the structural aspects, nodes in a heterogeneous graph also have associated features. Different types of nodes often have different feature dimensions. Specifically, if a node v belongs to type t_1, its feature vector has a dimension d_{t_1}, while if a node u belongs to type t_2, its feature vector has a dimension d_{t_2}, where d_{t_1} and d_{t_2} are generally not equal.

4 Method

4.1 Overview Framework

In this paper, we aim to develop a graph-level pre-training task on a heterogeneous graph and learn a heterogeneous prompt vector capable of adjusting the original node features of the graph. By employing this heterogeneous prompt vector, we seek to bridge the gap between graph pre-training strategies and multiple downstream tasks, as well as to mitigate the challenges associated with transferring prior knowledge across different domains.To address this issue, we adopt a three-step approach. First, we unify all downstream tasks into a graph-level task format to minimize the gap between the pre-trained model and downstream task objectives. Next, recognizing that existing graph prompt learning methods struggle to effectively integrate contrastive pre-training strategies in the heterogeneous graph domain, we design a specialized graph-level contrastive pre-training strategy. This approach fully leverages heterogeneous information to enhance model performance in multi-task scenarios. Finally, we introduce heterogeneous feature prompts, which further improve model performance by optimizing the feature representation of input graphs. The overall framework of HGMF is shown in Fig 2.

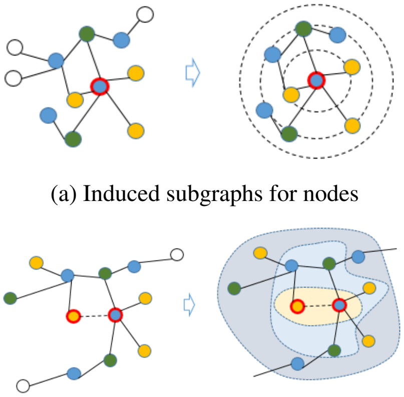
4.2 Reformulating Downstream Tasks

Challenges in Heterogeneous Graphs and Task Reformulation

In homogeneous graphs, reformulating downstream tasks into a unified graph-level representation is a widely used approach to bridge the gap between pre-training and downstream tasks. By constructing τ-ego networks as induced subgraphs and mapping node or edge labels to these subgraphs, node and edge classification problems can be reformulated as graph classification tasks. However, in heterogeneous graphs, this method faces new challenges due to the diverse types of nodes and edges, each carrying different semantics. Additionally, the semantic dependencies between these types are complex and cannot be captured by simple τ-hop neighborhoods. The heterogeneity of these graphs makes it difficult to construct a consistent graph-level representation that can accommodate the full range of tasks.

Adapting Induced Subgraphs and Label Mapping

To address these challenges, we adapt the task reformulation method to heterogeneous graphs by modifying the construction of induced subgraphs. For node-level tasks, the induced subgraph includes the target node, its τ-hop neighbors, and all connecting edges along with their types, as show in Fig 1a. This subgraph captures both the structural context of the target node and the type-specific semantics of its neighboring nodes and edges. For edge-level tasks, the induced subgraph includes the two endpoints of the target edge and their respective τ-hop neighbors, retaining all nodes, edges, and their types within this subgraph, as depicted in Fig 1b. This ensures that the semantic context surrounding the target edge is preserved, which is critical for accurate edge-level task performance. After constructing the induced subgraphs, we map the labels of the downstream tasks to these subgraphs. For node-level tasks, the label of the target node is assigned to its induced subgraph, transforming node classification into a graph classification task. Similarly, for edge-level tasks, the label of the target edge (e.g., relationship type) is assigned to its induced subgraph, turning edge classification into a graph classification problem. This mapping unifies node-level and edge-level tasks under a consistent graph-level framework, enabling knowledge transfer across tasks.


(a) Induced subgraphs for nodes

(b) Induced subgraphs for edge

Figure 1: Induced subgraphs for nodes and edges

Summary

By reformulating both node-level and edge-level tasks into graph classification tasks, we provide a unified framework suitable for heterogeneous graphs. This method preserves the important type-specific semantics of heterogeneous graphs and lays a solid foundation for pre-training and downstream prompt-based learning in such graphs.

4.3 Heterogeneous Graph Pre-training

Why Design Heterogeneous Graph Pre-training Tasks

To unify pre-training and downstream tasks at the graph level, thereby maximizing the task subspace and fully exploring the potential relationships in heterogeneous graphs, we propose to adopt graph-level pre-training strategies during the pre-training phase. However, there are currently no pre-training methods specifically designed for graph-level tasks in heterogeneous graphs. This limitation prevents existing strategies from effectively adapting to graph-level downstream tasks, thereby restricting their performance and generalization. Moreover, contrastive learning has demonstrated superior performance in representation learning across various domains by constructing positive and negative sample pairs to enhance feature discrimination. Therefore, we propose a novel graph-level contrastive pre-training strategy for heterogeneous graphs to address these gaps and leverage the advantages of contrastive learning.

How to Design Heterogeneous Graph Pre-training Tasks

Heterogeneous graphs contain diverse node and edge types that play a critical role in preserving their structural integrity. For instance, in the ACM dataset, "subject" nodes, despite comprising only 56 entities, connect to others through 3025 edges. To enhance model robustness without disrupting the graph structure, we propose two augmentation strategies: **Heterogeneous Node Masking Augmentation** and **Heterogeneous Edge Permutation Augmentation**, both guided by an augmentation ratio r. This ratio determines how much of the node or edge set is altered, ensuring variability while maintaining structural integrity.

For **Heterogeneous Node Masking**, the adjusted augmentation ratio for each node type is computed as:

$$\text{adjusted\_ratio}(i) = \frac{\text{count}(i)^2}{\sum_{j \in T_V} \text{count}(j)^2}$$

where count(i) is the number of nodes of type i, and T_V denotes all node types. Using this ratio, the final number of masked nodes is:

$$\text{num\_to\_mask}(i) = r \cdot a(i) \cdot |V|$$

where $a(i)$ is the adjusted ratio for node type i, r is the augmentation ratio, and $|V|$ is the total number of nodes.

Similarly, for **Heterogeneous Edge Permutation**, the adjusted ratio for each edge type is:

$$\text{adjusted\_ratio}(i) = \frac{\text{count}(i)^2}{\sum_{j \in T_E} \text{count}(j)^2}$$

where count(i) is the number of edges of type i, and T_E represents all edge types. The number of permuted edges is given by:

$$\text{num\_to\_permute}(i) = r \cdot b(i) \cdot |E|$$

where $b(i)$ is the adjusted ratio for edge type i, and $|E|$ is the total number of edges.

After applying the augmentation strategies, the processing of augmented graphs involves three key steps. First, a GNN-based Encoder is used, incorporating a feature transformation layer to preprocess input features and extract graph-level representations $\mathbf{Z}_{G_i}, \mathbf{Z}_{G_j}$. This flexible design ensures compatibility with various tasks. Second, a Projection Head maps high-dimensional representations into a smaller latent space using a two-layer MLP, which facilitates contrastive learning. Finally, the Contrastive Loss Function maximizes the similarity between positive pairs of augmented graphs while minimizing it for negative pairs. Given N graphs, $2N$ augmented graphs are generated per batch, and the loss is computed as:

$$\mathcal{L}_{\text{con}} = -\sum_{i=1}^{N} \log \frac{\exp(\text{sim}(\mathbf{z}_i^+, \mathbf{z}_i^-)/\tau)}{\sum_{j=1}^{2N} \mathbb{1}_{[i \neq j]} \exp(\text{sim}(\mathbf{z}_i^+, \mathbf{z}_j)/\tau)}$$

where $\mathbf{z}_i^+$ and $\mathbf{z}_i^-$ are the positive and negative representations, and τ is the temperature coefficient. By optimizing this loss, the model learns robust and discriminative graph-level representations that effectively support downstream tasks.

Through this integrated approach, our augmentation process efficiently combines the unique characteristics of heterogeneous graphs with contrastive pre-training, offering a unified and powerful graph representation learning framework.

4.4 Heterogeneous Graph Prompt Feature (HGPF)

Why Design Heterogeneous Graph Prompt Feature (HGPF)

In heterogeneous graphs, due to the diversity of node types, the feature dimensions of each node type are typically different. Even when the feature dimensions are consistent across nodes, the meanings of these dimensions often vary between different node types. For example, in a scholarly paper graph, author nodes might represent their research domains, while paper nodes contain publication information, making the meaning of the same feature dimension vary across different node types.

To effectively adjust the input graph for better adaptation to pre-trained models during downstream tasks, a mechanism must be designed to modify the feature representations of nodes according to their types. Therefore, we introduce the Heterogeneous Graph Prompt Feature (HGPF) to provide distinct feature prompts for each node type. This allows the adjustment of features of different node types, improving their alignment with pre-training tasks.

How To Design Heterogeneous Graph Prompt Feature (HGPF)

For each node type, we design a specific feature prompt associated with that type. Within each node type, the same feature prompt is shared. The dimensionality of the prompt matches the dimensionality of the corresponding node type's features. The role of this prompt is to adjust the node's feature representation by adding or multiplying it with the node's original feature.

Let $\mathbf{x}_i$ denote the feature vector of node i, where $\mathbf{x}_i \in \mathbb{R}^{d_i}$ represents the feature vector of node i, and T_i is the node type. The corresponding feature prompt is $\mathbf{p}_{T_i} \in \mathbb{R}^{d_i}$. The adjustment process for the node feature is as follows:

$$\hat{\mathbf{x}}_i = \mathbf{x}_i \times \mathbf{p}_{T_i}$$

Through this process, the node features are optimized by the adjustment based on their original features and the feature prompt, making them more suitable for the pre-training task.

Why It Works?

The effectiveness of HGPF is based on the prompt mechanism, which works by adjusting the node features of the input graph to better align with the pre-training task. In graph-based tasks, the flexibility of feature space operations is crucial to the effectiveness of prompting. By designing feature prompts, we make node features more task-relevant during pre-training, facilitating effective knowledge transfer.

Fang et al [Fang *et al.*, 2024]. demonstrated that for any graph G (with adjacency matrix A and node feature matrix

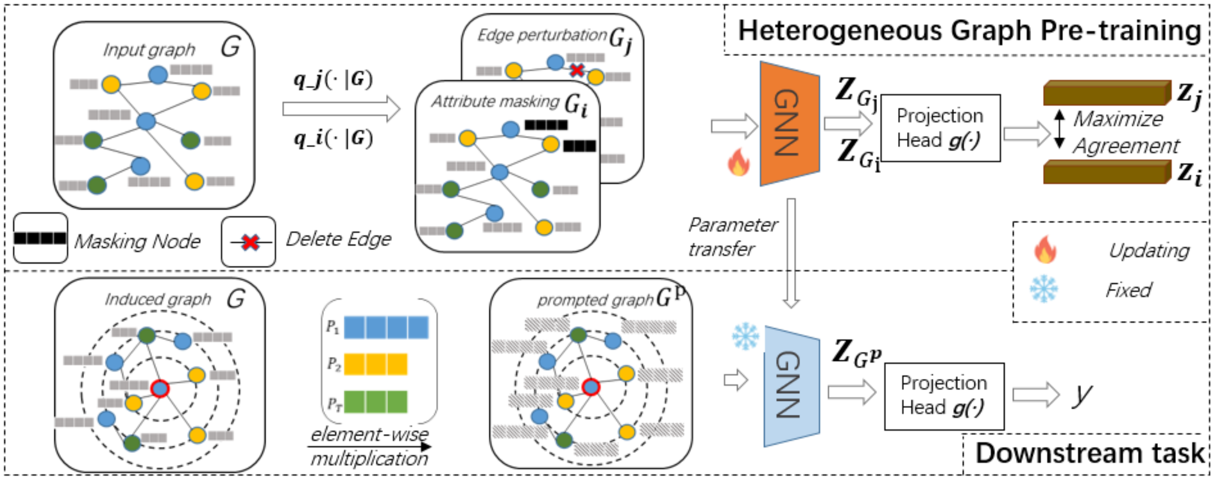


Figure 2: Overview of the HGMP architecture: The input graph is first subjected to heterogeneous graph augmentation, resulting in an augmented graph. Contrastive learning is then employed for pre-training, after which the GNN parameters are frozen. During downstream tasks, heterogeneous graph prompt features are multiplied with the node features of the input induced graph to perform graph augmentation, enabling alignment with the pre-training objective.

X), an appropriate prompt token p^* can be learned such that the following equation holds:

$$\varphi^*(A, X + p^*) = \varphi^*(g(A, X)) + O_\varphi$$

Here, $g(A, X)$ represents a transformation applied to the original graph (e.g., changing node features, adding/removing edges), and O_φ denotes the error between the graph augmented by the prompt and the graph after transformation. This error depends on the model's nonlinear layers and the quality of the learned prompt. With further optimization, the error can be significantly reduced.

In our approach, HGPF adjusts node features by customizing task-specific prompts for each node type. Unlike traditional methods, our feature prompts are combined with the original features of the nodes through multiplication. This multiplication method enables a more rapid adaptation to few-shot scenarios, particularly when dealing with attributed graphs, thereby enhancing the node representations for task alignment. This adaptive adjustment mechanism facilitates a more effective transfer of knowledge from the pre-trained model to the downstream task, significantly improving the performance of downstream tasks.

Dataset	#Nodes	#Node Types	#Edges	#Edge Types	Target	#Classes
ACM	10,942	4	547,872	8	paper	3
IMDB	21,420	4	86,642	6	movie	5

Table 1: Summary of datasets.

5 Experiments

This section presents experimental results and analyses to demonstrate the superiority of our approach.

5.1 Experimental Setup

Datasets. We use two real-world HIN datasets, summarized in Table 1. (1) *ACM* [Lv *et al.*, 2021] dataset is an academic network containing relationships among papers (P), authors (A), subjects (S), and terms (T), making it suitable for studying academic influence and knowledge discovery. (2) *IMDB* [Lv *et al.*, 2021] dataset serves as a benchmark for graph neural networks, involving movies (M), actors (A), directors (D), and keywords (K), with a wide range of relationships reflecting its suitability for tasks in the entertainment domain.

Baselines. We evaluate our method against baselines spanning five categories: (1)Supervised GNNs: GCN [Kipf and Welling, 2016], GAT [Veličković *et al.*, 2017]; (2) GNNs with "pre-train, fine-tune": GraphCL [You *et al.*, 2020], SimGRACE [Xia *et al.*, 2022]; (3) Supervised HGNNs: HGT [Hu *et al.*, 2020], HAN [Wang *et al.*, 2019]; (4) HGNNs with "pre-train, fine-tune": DMGI [Park *et al.*, 2020], HeCo [Wang *et al.*, 2021], HDMI [Jing *et al.*, 2021]; (5) HGNNs with "pre-train, prompt": HGPROMPT [Yu *et al.*, 2024].

Settings and parameters. For the few-shot setting, we evaluate tasks under a 10-shot configuration across three categories: node classification, edge classification, and graph classification. The task construction mainly follows the approach of ProG [Sun *et al.*, 2023a] works. However, in heterogeneous graphs, target nodes are often connected to non-target nodes, whose learning value is relatively lower. This makes directly applying the ProG [Sun *et al.*, 2023a] approach, which combines both endpoints of edges as labels, less suitable in the heterogeneous domain.

To address this, we make specific adjustments: only the labels of nodes with endpoint types corresponding to target nodes are used as the induced graph's labels. The evaluation metrics include micro-F1 and macro-F1, ensuring a comprehensive assessment of the model's performance.For repro-

Methods	ACM		IMDB	
	Micro F1	Macro F1	Micro F1	Macro F1
GAT	68.96±7.83	68.77±8.26	64.04±0.72	55.55±0.88
GCN	81.50±1.62	81.46±1.70	68.66±1.71	61.22±2.12
GCL	66.73±1.89	66.61±1.94	65.08±0.92	56.81±1.15
SimGRACE	55.00±2.79	54.71±2.86	64.63±0.36	56.27±0.45
HGT	61.78±7.81	57.12±11.19	64.64±1.30	56.26±1.61
HAN	80.42±1.73	80.31±1.83	67.09±1.34	59.29±1.67
HeCo	83.99±1.74	83.98±1.67	66.18±1.08	58.16±1.32
DMGI	65.42±2.56	64.07±2.81	59.41±1.60	49.80±1.97
HGPROMPT	77.29±1.81	75.40±2.80	61.76±0.78	52.71±0.95
HGMP	**84.60±3.18**	**84.54±3.19**	**68.76±0.82**	**61.34±1.00**

Table 2: Evaluation of node classification(%). The best method is bolded and the runner-up is underlined.

ducibility, we provide the specific construction methods for each task and the detailed hyperparameter values in the supplement.

5.2 Performance Evaluation on Multi-Tasks

Few-shot node classification

The classification results of the 10-shot nodes are shown in Table 2. Based on the experimental results, we summarize our observations as follows:(1) HGMP outperforms all baseline methods across all datasets, demonstrating its effectiveness in node classification tasks under the few-shot setting within heterogeneous graph domains.(2) On the ACM dataset, HeCo, which also adopts a pre-training and fine-tuning strategy, achieves performance second only to HGMP and significantly surpasses homogeneous pre-training methods. This result highlights the necessity of fully leveraging heterogeneity information in pre-training methods for heterogeneous graphs. Furthermore, HeCo's performance, closely following that of HGMP, further validates the effectiveness of heterogeneous graph prompting mechanisms and the rationality of incorporating graph contrastive pre-training strategies that integrate heterogeneous information. Additionally, HAN, which utilizes meta-path strategies, also demonstrates competitive performance, showcasing its advantages in capturing structural and semantic information.(3) In few-shot scenarios, GCN significantly outperforms other homogeneous methods, indicating that models with smaller parameter sizes are more effective at avoiding overfitting when training data is limited, thereby achieving better generalization performance.

Few-shot edge classification

The results of the 10-shot edge classification task are shown in the Table 3. Based on the experimental results, we summarize the following observations: (1) HGMP outperforms all baseline methods in edge-type tasks, demonstrating its broad adaptability to few-shot settings in tasks within heterogeneous graph domains. (2) Since HeCo and DMGI only model the features of the target nodes, they fail to capture the complete information required for the edge classification task, thus losing their previous second-best performance in node classification tasks. GCN, once again, benefits from its smaller parameter size, achieving success in the few-shot setting due to its ability to avoid overfitting.

Methods	ACM		IMDB	
	Micro F1	Macro F1	Micro F1	Macro F1
GAT	69.72±2.91	68.64±3.28	64.71±1.76	56.29±2.2
GCN	73.42±6.80	73.24±6.95	66.36±1.05	58.32±1.30
GCL	61.82±3.07	61.34±3.60	64.68±1.05	56.56±1.28
SimGRACE	50.85±2.05	50.19±2.30	63.23±1.23	54.45±1.53
HGT	66.61±9.85	61.21±13.68	63.49±0.75	54.77±0.92
HAN	70.88±4.16	69.71±5.40	63.39±0.74	54.65±0.92
HeCo	70.13±1.45	70.30±1.43	66.50±1.31	58.80±1.62
DMGI	57.09±2.08	55.47±3.50	63.05±1.58	54.55±1.95
HGPROMPT	61.74±5.77	58.33±7.35	60.43±2.46	50.98±3.02
HGMP	**77.33±2.53**	**77.44±2.53**	**67.65±0.83**	**60.19±1.01**

Table 3: Evaluation of edge classification(%). The best method is bolded and the runner-up is underlined.

Methods	ACM		IMDB	
	Micro F1	Macro F1	Micro F1	Macro F1
GAT	98.07±0.67	98.06±0.67	51.39±4.34	50.8±5.02
GCN	98.16±0.58	98.16±0.58	49.42±3.32	47.38±3.28
GCL	86.92±2.09	87.00±2.00	42.21±2.22	40.50±2.53
SimGRACE	70.44±3.05	70.31±2.93	34.78±2.15	33.36±2.73
HGT	89.18±8.76	88.21±11.19	33.99±6.69	25.85±8.22
HAN	96.92±0.96	96.91±0.97	27.97±4.96	22.77±8.71
HeCo	97.55±0.42	97.55±0.42	46.36±1.78	45.30±1.76
DMGI	89.02±2.68	88.96±2.67	41.25±3.68	40.53±3.57
HGPROMPT	91.13±1.54	90.94±1.72	32.32±1.93	31.01±1.76
HGMP	**98.30±0.45**	**98.30±0.45**	**53.73±3.38**	**52.88±3.60**

Table 4: Evaluation of graph classification(%). The best method is bolded and the runner-up is underlined.

Few-shot graph classification

The results of the 10-shot graph classification task are shown in Table 4. Based on the experimental results, we summarize the following observations: (1) On the ACM dataset, all baseline methods improved significantly in graph-type tasks, due to the high concentration of same-label target nodes during subgraph construction. HGMP outperformed all baselines, demonstrating its superiority in multi-task few-shot settings within heterogeneous graphs. (2) The IMDB dataset, being a multi-label multi-class classification task, performed well in node and edge classification. However, in the graph classification task, to adhere to the previous task setup, we could only select one label, converting the problem into a single-label multi-class classification task. This conversion resulted in significant information loss, leading to a noticeable drop in performance across all baselines on the IMDB dataset in the graph classification task. (3) The strong performance of all baselines in the graph classification task highlights that the knowledge obtained from pretraining on graph-level tasks is better suited for downstream graph classification tasks. This further underscores the necessity of unifying task designs at the graph level. (4) GAT demonstrated a performance recovery, indicating that as the graph size increases slightly, attention-based models can regain competitive performance.

Methods	Heterogeneous	Heterogeneous	Node classification	Edge classification	Graph classification
	graph augmentation	graph feature	ACM IMDB	ACM IMDB	ACM IMDB
VARIANT 1	×	×	66.73 65.08	61.82 64.68	86.92 42.21
VARIANT 2	×	✓	81.71 67.94	74.93 65.86	97.80 52.46
VARIANT 3	✓	×	68.83 65.38	66.04 65.33	89.89 44.86
HGMP	✓	✓	**84.60 68.76**	**77.33 67.65**	**98.30 53.73**

Table 5: Variants used in ablation study, and corresponding results in Micro F1 (%) on node,edge and graph classification.

Backbone	method	Node Classification	Edge Classification	Graph Classification
		Micro F1 Macro F1	Micro F1 Macro F1	Micro F1 Macro F1
GCN	SUPERVISED	81.50 81.46	73.42 73.24	98.16 98.16
	HGMP	**84.60 84.54**	**77.33 77.44**	**98.30 98.30**
GAT	SUPERVISED	68.96 68.77	69.72 68.64	**98.07 98.06**
	HGMP	**73.18 73.37**	**70.18 70.20**	95.78 95.78
HGT	SUPERVISED	61.78 57.12	66.61 61.21	89.18 88.21
	HGMP	**68.25 67.62**	**72.47 70.23**	**89.49 88.35**

Table 6: Evaluation of different backbones for NC,EC and GC tasks(%) on ACM. SUPERVISED means end-to-end training.

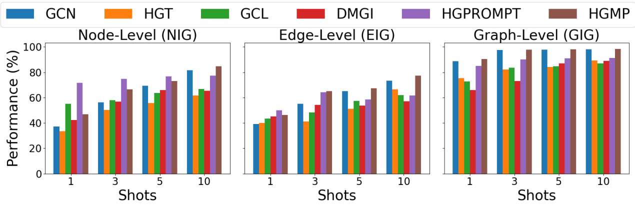


Figure 3: Impact of shots on ACM.

However, HeCo and DMGI, as they only model the features of target nodes, remain unable to capture complete graph-level information and consequently fell out of the top three in this task.

5.3 Flexibility on backbones

To validate the adaptability and performance of the proposed HGMP model across different backbones, we conducted experiments using GCN, GAT, and HGT on node classification, edge classification, and graph classification tasks. HGMP consistently outperformed traditional supervised methods in node and edge classification. For GCN, HGMP achieved improvements of approximately 3% and 4% in node and edge classification, respectively, while maintaining comparable performance in graph classification. With GAT, HGMP excelled in edge classification, outperforming the baseline by nearly 1%, though it showed a slight performance decrease in graph classification. For HGT, HGMP provided noticeable gains in node and edge classification, albeit with modest improvements in graph classification. These results confirm HGMP's adaptability and effectiveness in few-shot heterogeneous graph tasks, leveraging graph augmentation and prompt features to enhance performance.

5.4 Performance with different shots

To analyze the impact of shot numbers, we conducted experiments on the ACM dataset for node classification (NC), edge classification (EC), and graph classification (GC). The results, shown in Figs 3, indicate that HGMP generally outperforms baselines, especially in EC and GC tasks. For NC, HGPROMPT excels in few-shot scenarios (fewer than 5 shots) due to its optimized class prototype but loses its edge as sample size increases, with HGMP showing stronger robustness and generalization. In EC and GC tasks, HGMP consistently achieves superior performance, attributed to its graph-level training objectives and stable adaptability across settings.

5.5 Ablation study

To evaluate the contribution of each component in HGMP, we conducted an ablation study comparing the full model with its variants, as shown in Table 5. All variants were trained using the reconstruction task template based on induced subgraphs to ensure fair comparisons. The results demonstrate that each component significantly enhances performance.

Comparing Variant 1 with Variant 3 highlights that the absence of heterogeneous graph augmentation causes performance degradation, validating the importance of optimizing augmentation strategies to capture complex structural information. Similarly, the comparison between Variant 1 and Variant 2 shows that heterogeneous prompt features provide substantial performance gains, effectively constructing task-relevant representations and improving classification accuracy.

Overall, HGMP outperforms all variants, demonstrating the synergistic benefits of combining heterogeneous graph augmentation and prompt features. This design aligns induced subgraphs in downstream tasks with pretraining objectives, enhancing generalization and overall performance.

6 Conclusions

This paper proposes HGMP, a novel multi-task prompt framework that bridges pre-trained models and downstream tasks in heterogeneous graphs by unifying downstream tasks into a graph-level format and incorporating a graph-level contrastive pre-training strategy leveraging heterogeneous information. Experimental results demonstrate its significant advantages over existing methods, particularly in multi-task scenarios. While effective for attributed graphs, HGMP performs suboptimally on unattributed graphs, highlighting a need for improved adaptability. Future work will focus on addressing this limitation, offering new perspectives for heterogeneous graph learning and advancing multi-task learning and practical graph applications.

Acknowledgements

This work was supported in part by the Zhejiang Provincial Natural Science Foundation of China under Grant LDT23F01015F01, in part by the Key Technology Research and Development Program of the Zhejiang Province under Grant No. 2025C1023 and in part by the National Natural Science Foundation of China under Grants 62372146.

References

[Cao *et al.*, 2021] Yuwei Cao, Hao Peng, Jia Wu, Yingtong Dou, Jianxin Li, and Philip S Yu. Knowledge-preserving incremental social event detection via heterogeneous gnns. In *Proceedings of the Web Conference 2021*, pages 3383–3395, 2021.

[Dong *et al.*, 2019] Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. Unified language model pre-training for natural language understanding and generation. *Advances in neural information processing systems*, 32, 2019.

[Fang *et al.*, 2024] Taoran Fang, Yunchao Zhang, Yang Yang, Chunping Wang, and Lei Chen. Universal prompt tuning for graph neural networks. *Advances in Neural Information Processing Systems*, 36, 2024.

[Fu *et al.*, 2020] Xinyu Fu, Jiani Zhang, Ziqiao Meng, and Irwin King. Magnn: Metapath aggregated graph neural network for heterogeneous graph embedding. In *Proceedings of the web conference 2020*, pages 2331–2341, 2020.

[Gao *et al.*, 2023] Chen Gao, Yu Zheng, Nian Li, Yinfeng Li, Yingrong Qin, Jinghua Piao, Yuhan Quan, Jianxin Chang, Depeng Jin, Xiangnan He, et al. A survey of graph neural networks for recommender systems: Challenges, methods, and directions. *ACM Transactions on Recommender Systems*, 1(1):1–51, 2023.

[Hogan *et al.*, 2021] Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia d'Amato, Gerard De Melo, Claudio Gutierrez, Sabrina Kirrane, José Emilio Labra Gayo, Roberto Navigli, Sebastian Neumaier, et al. Knowledge graphs. *ACM Computing Surveys (Csur)*, 54(4):1–37, 2021.

[Hu *et al.*, 2019] 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.

[Hu *et al.*, 2020] Ziniu Hu, Yuxiao Dong, Kuansan Wang, and Yizhou Sun. Heterogeneous graph transformer. In *Proceedings of the web conference 2020*, pages 2704–2710, 2020.

[Huang *et al.*, 2024] 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, 2024.

[Ji *et al.*, 2021] Shaoxiong Ji, Shirui Pan, Erik Cambria, Pekka Marttinen, and S Yu Philip. A survey on knowledge graphs: Representation, acquisition, and applications. *IEEE transactions on neural networks and learning systems*, 33(2):494–514, 2021.

[Jing *et al.*, 2021] Baoyu Jing, Chanyoung Park, and Hanghang Tong. Hdmi: High-order deep multiplex infomax. In *Proceedings of the Web Conference 2021*, pages 2414–2424, 2021.

[Kipf and Welling, 2016] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *International Conference on Learning Representations*, 2016.

[Knyazev *et al.*, 2019] Boris Knyazev, Graham W Taylor, and Mohamed Amer. Understanding attention and generalization in graph neural networks. *Advances in neural information processing systems*, 32, 2019.

[Lester *et al.*, 2021] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. *Empirical Methods in Natural Language Processing*, 2021.

[Li and Liang, 2021] Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, 2021.

[Liu *et al.*, 2023] 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*, pages 417–428, 2023.

[Lv *et al.*, 2021] Qingsong Lv, Ming Ding, Qiang Liu, Yuxiang Chen, Wenzheng Feng, Siming He, Chang Zhou, Jianguo Jiang, Yuxiao Dong, and Jie Tang. Are we really making much progress? revisiting, benchmarking and refining heterogeneous graph neural networks. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, pages 1150–1160, 2021.

[Ma *et al.*, 2024] Yihong Ma, Ning Yan, Jiayu Li, Masood Mortazavi, and Nitesh V Chawla. Hetgpt: Harnessing the power of prompt tuning in pre-trained heterogeneous graph neural networks. In *Proceedings of the ACM on Web Conference 2024*, pages 1015–1023, 2024.

[Mao *et al.*, 2023] Qiheng Mao, Zemin Liu, Chenghao Liu, and Jianling Sun. Hinormer: Representation learning on heterogeneous information networks with graph transformer. In *Proceedings of the ACM Web Conference 2023*, pages 599–610, 2023.

[Morris *et al.*, 2019] Christopher Morris, Martin Ritzert, Matthias Fey, William L Hamilton, Jan Eric Lenssen, Gaurav Rattan, and Martin Grohe. Weisfeiler and leman go neural: Higher-order graph neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 4602–4609, 2019.

[Park *et al.*, 2020] Chanyoung Park, Donghyun Kim, Jiawei Han, and Hwanjo Yu. Unsupervised attributed multiplex network embedding. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 5371–5378, 2020.

[Sun *et al.*, 2022] Mingchen Sun, Kaixiong Zhou, Xin He, Ying Wang, and Xin Wang. 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*, pages 1717–1727, 2022.

[Sun *et al.*, 2023a] Xiangguo Sun, Hong Cheng, Jia Li, Bo Liu, and Jihong Guan. All in one: Multi-task prompting for graph neural networks. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 2120–2131, 2023.

[Sun *et al.*, 2023b] Xiangguo Sun, Jiawen Zhang, Xixi Wu, Hong Cheng, Yun Xiong, and Jia Li. Graph prompt learning: A comprehensive survey and beyond. *arXiv preprint arXiv:2311.16534*, 2023.

[Tan *et al.*, 2023] Zhen Tan, Ruocheng Guo, Kaize Ding, and Huan Liu. Virtual node tuning for few-shot node classification. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 2177–2188, 2023.

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

[Wang *et al.*, 2019] Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S Yu. Heterogeneous graph attention network. In *The world wide web conference*, pages 2022–2032, 2019.

[Wang *et al.*, 2021] Xiao Wang, Nian Liu, Hui Han, and Chuan Shi. Self-supervised heterogeneous graph neural network with co-contrastive learning. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, pages 1726–1736, 2021.

[Xia *et al.*, 2022] Jun Xia, Lirong Wu, Jintao Chen, Bozhen Hu, and Stan Z Li. Simgrace: A simple framework for graph contrastive learning without data augmentation. In *Proceedings of the ACM Web Conference 2022*, pages 1070–1079, 2022.

[Yang *et al.*, 2020] Carl Yang, Yuxin Xiao, Yu Zhang, Yizhou Sun, and Jiawei Han. Heterogeneous network representation learning: A unified framework with survey and benchmark. *IEEE Transactions on Knowledge and Data Engineering*, 34(10):4854–4873, 2020.

[Yehudai *et al.*, 2021] Gilad Yehudai, Ethan Fetaya, Eli Meirom, Gal Chechik, and Haggai Maron. From local structures to size generalization in graph neural networks. In *International Conference on Machine Learning*, pages 11975–11986. PMLR, 2021.

[You *et al.*, 2020] 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.

[Yu *et al.*, 2024] Xingtong Yu, Yuan Fang, Zemin Liu, and Xinming Zhang. Hgprompt: Bridging homogeneous and heterogeneous graphs for few-shot prompt learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 16578–16586, 2024.

[Yun *et al.*, 2019] Seongjun Yun, Minbyul Jeong, Raehyun Kim, Jaewoo Kang, and Hyunwoo J Kim. Graph transformer networks. *Advances in neural information processing systems*, 32, 2019.

[Zhang *et al.*, 2022] Chuxu Zhang, Kaize Ding, Jundong Li, Xiangliang Zhang, Yanfang Ye, Nitesh V Chawla, and Huan Liu. Few-shot learning on graphs. *International Joint Conference on Artificial Intelligence*, 2022.

[Zi *et al.*, 2024] Chenyi Zi, Haihong Zhao, Xiangguo Sun, Yiqing Lin, Hong Cheng, and Jia Li. Prog: A graph prompt learning benchmark. *arXiv preprint arXiv:2406.05346*, 2024.