
ProG: A Graph Prompt Learning Benchmark

Chenyi Zi^{1*}, Haihong Zhao^{1*}, Xiangguo Sun^{2,✉}, Yiqing Lin³, Hong Cheng², Jia Li¹

¹Hong Kong University of Science and Technology (Guangzhou)

²Department of Systems Engineering and Engineering Management,
and Shun Hing Institute of Advanced Engineering, The Chinese University of Hong Kong

³Tsinghua University

✉ corresponding author: xiangguosun@cuhk.edu.hk

Abstract

Artificial general intelligence on graphs has shown significant advancements across various applications, yet the traditional ‘Pre-train & Fine-tune’ paradigm faces inefficiencies and negative transfer issues, particularly in complex and few-shot settings. Graph prompt learning emerges as a promising alternative, leveraging lightweight prompts to manipulate data and fill the task gap by reformulating downstream tasks to the pretext. However, several critical challenges still remain: how to unify diverse graph prompt models, how to evaluate the quality of graph prompts, and to improve their usability for practical comparisons and selection. In response to these challenges, we introduce the first comprehensive benchmark for graph prompt learning. Our benchmark integrates **SIX** pre-training methods and **FIVE** state-of-the-art graph prompt techniques, evaluated across **FIFTEEN** diverse datasets to assess performance, flexibility, and efficiency. We also present ‘ProG’, an easy-to-use open-source library that streamlines the execution of various graph prompt models, facilitating objective evaluations. Additionally, we propose a unified framework that categorizes existing graph prompt methods into two main approaches: prompts as graphs and prompts as tokens. This framework enhances the applicability and comparison of graph prompt techniques. The code is available at: <https://github.com/sheldonresearch/ProG>.

1 Introduction

Recently, artificial general intelligence (AGI) on graphs has emerged as a new trend in various applications like drug design [59, 38], protein prediction [10], social analysis [49, 64, 52], etc. To achieve this vision, one key question is how to learn useful knowledge from non-linear data (like graphs) and how to apply it to various downstream tasks or domains. Classical approaches mostly leverage the ‘Pre-train & Fine-tune’ paradigm, which first designs some pretext with easily access data as the pre-training task for the graph neural network model, and then adjusts partial or entire model parameters to fit new downstream tasks. Although much progress has been achieved, they are still not effective and efficient enough. For example, adjusting the pre-trained model will become very time-consuming with the increase in model complexity [25, 24, 48, 23]. In addition, a natural gap between these pretexts and downstream tasks makes the task transferring very hard, and sometimes may even cause negative transfer [57, 30, 29]. These problems are particularly serious in few-shot settings.

To further alleviate the problems above, Graph Prompt Learning [50, 28] has attracted more attention recently. As shown in Figure 1, graph prompts seek to manipulate downstream data by inserting an additional small prompt graph and then reformulating the downstream task to the pre-training task without changing the pre-trained Graph Neural Network (GNN) model. Since a graph prompt is usually lightweight, tuning this prompt is more efficient than the large backbone model. Compared

*The first two authors contributed equally to this work. Listing order is random.

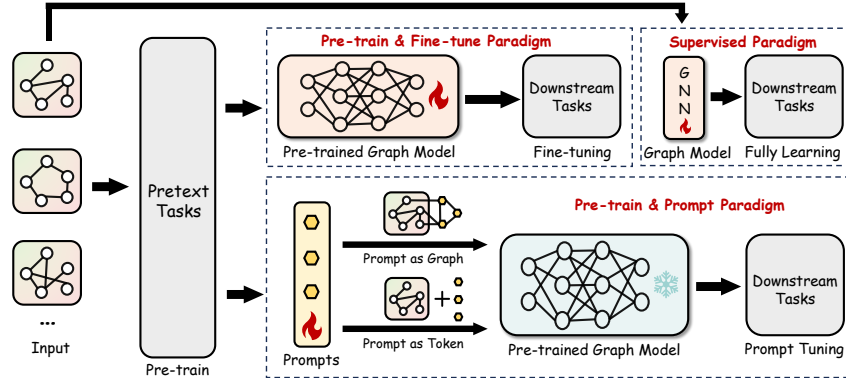


Figure 1: An overview of our benchmark.

with graph ‘pre-training & fine-tuning’, which can be seen as a model-level retraining skill, graph ‘pre-training & prompting’ is more customized for data-level operations, which means its potential applications might go beyond task transferring in the graph intelligence area like data quality evaluation, multi-domain alignment, learnable data augmentation, etc. Recently, many graph prompt models have been proposed and presented significant performance in graph learning areas [47, 46, 33, 8, 66, 5], indicating their huge potential towards more general graph intelligence. Despite the enthusiasm around graph prompts in research communities, three significant challenges impede further exploration:

- ❶ **How can we unify diverse graph prompt models given their varying methodologies?** The fragmented landscape of graph prompts hinders systematic research advancement. A unified framework is essential for integrating these diverse methods, creating a cohesive taxonomy to better understand current research and support future studies.
- ❷ **How do we evaluate the quality of graph prompts?** Assessing the efficiency, power, and flexibility of graph prompts is crucial for understanding their impact and limitations. Currently, there is no standardized benchmark for fair and comprehensive comparison, as existing works have inconsistent experimental setups, varying tasks, metrics, pre-training strategies, and data-splitting techniques, which obstruct a comprehensive understanding of the current state of research.
- ❸ **How can we make graph prompt approaches more user-friendly for practical comparison and selection?** Despite numerous proposed methods, the lack of an easy-to-use toolkit for creating graph prompts limits their potential applications. The implementation details of existing works vary significantly in programming frameworks, tricks, and running requirements. Developing a standardized library for various graph prompt learning approaches is urgently needed to facilitate broader exploration and application.

In this work, we wish to push forward graph prompt research to be more standardized, provide suggestions for practical choices, and uncover both their advantages and limitations with a comprehensive evaluation. To the best of our knowledge, this is the first benchmark for graph prompt learning. **To solve the first challenge**, our benchmark treats existing graph prompts as three basic components: prompt tokens, token structure, and insert patterns. According to their detailed implementation, we categorize the current graph prompts into two branches: graph prompt as an additional graph, and graph prompt as tokens. This taxonomy makes our benchmark approach compatible with nearly all existing graph prompting approaches. **To solve the second challenge**, our benchmark encompasses 6 different classical pre-training methods covering node-level, edge-level, and graph-level tasks, and 5 most representative and state-of-the-art graph prompt methods. Additionally, we include 15 diverse datasets in our benchmark with different scales, covering both homophilic and heterophilic graphs from various domains. We systematically investigate the effectiveness, flexibility, and efficiency of graph prompts under few-shot settings. Compared with traditional supervised methods and ‘pre-training & fine-tuning’ methods, graph prompting approaches present significant improvements. We even observe some negative transfer cases caused by ‘pre-training & fine-tuning’ methods but effectively reversed by graph prompting methods. **To solve the third challenge**, we develop a unified library for creating various graph prompts. In this way, users can evaluate their models or datasets with less effort. The contributions are summarized as follows:

- ❶ We propose the first comprehensive benchmark for graph prompt learning. The benchmark integrates 6 most used pre-training methods and 5 state-of-the-art graph prompting methods with

15 diverse graph datasets. We systematically evaluate the effectiveness, flexibility, and efficiency of current graph prompt models to assist future research efforts. (Detailed in Section 5)

- ② We offer ‘ProG’², an easy-to-use open-source library to conduct various graph prompt models (Section 3). We can get rid of the distraction from different implementation skills and reveal the potential benefits/shortages of these graph prompts in a more objective and fair setting.
- ③ We propose a unified view for current graph prompt methods (Section 2). We decompose graph prompts into three components: prompt tokens, token structure, and insert patterns. With this framework, we can easily group most of the existing work into two categories (prompt as graph, and prompt as tokens). This view serves as the infrastructure of our developed library and makes our benchmark applicable to most of the existing graph prompt works.

2 Preliminaries

Understand Graph Prompt Nature in A Unified View. Graph prompt learning aims to learn suitable transformation operations for graphs or graph representations to reformulate the downstream task to the pre-training task. Let $t(\cdot)$ be any graph-level transformation (e.g., "modifying node features", "augmenting original graphs", etc.), and $\psi^*(\cdot)$ be a frozen pre-trained graph model. For any graph \mathcal{G} with adjacency matrix $\mathbf{A} \in \{0, 1\}^{N \times N}$ and node feature matrix $\mathbf{X} \in \mathbb{R}^{N \times d}$ where N denotes the node number and d is feature number. It has been proven [47, 8] that we can always learn an appropriate prompt module \mathcal{P} , making them can be formulated as the following formula:

$$\psi^*(\mathcal{P}(\mathbf{X}, \mathbf{A}, \mathbf{P}, \mathbf{A}_{inner}, \mathbf{A}_{cross})) = \psi^*(t(\mathbf{X}, \mathbf{A})) + O_{\mathcal{P}\psi} \quad (1)$$

Here $\mathbf{P} \in \mathbb{R}^{K \times d}$ is the learnable representations of K prompt tokens, $\mathbf{A}_{inner} \in \{0, 1\}^{K \times K}$ indicate token structures and $\mathbf{A}_{cross} \in \{0, 1\}^{K \times N}$ is the inserting patterns, which indicates how each prompt token connects to each node in the original graph. Prompt module \mathcal{P} takes the input graph and then outputs an integrated graph with a graph prompt. This equation indicates that we can learn an appropriate prompt module \mathcal{P} applied to the original graph to imitate any graph manipulation. $O_{\mathcal{P}\psi}$ denotes the *error bound* between the manipulated graph and the prompting graph w.r.t. their representations from the pre-trained graph model, which can be seen as a measurement of graph prompt flexibility.

Task Definition (Few-shot Graph Prompt Learning). Given a graph \mathcal{G} , a frozen pre-trained graph model ψ^* , and labels \mathcal{Y} , the loss function is to optimize the task loss \mathcal{L}_{Task} as follows:

$$\mathcal{L}_{\mathcal{P}} = \mathcal{L}_{Task}(\psi^*(\mathcal{P}(\mathbf{X}, \mathbf{A}, \mathbf{P}, \mathbf{A}_{inner}, \mathbf{A}_{cross})), \mathcal{Y})$$

During the few-shot training phase, the labeled samples are very scarce. Specifically, given a set of classes C , each class C_i has only k labeled examples. Here, k is a small number (e.g., $k \leq 10$), and the total number of labeled samples is given by $k \times |C| = |\mathcal{Y}|$, where $|C|$ is the number of classes. This setup is commonly referred to as k -shot classification [33]. In this paper, our evaluation includes both node and graph classification tasks.

Evaluation Questions. We are particular interested in the **effectiveness**, **flexibility**, and **efficiency** of graph prompt learning methods. Specifically, **in effectiveness**, we wish to figure out how effective are different graph prompt methods on various tasks (Section 5.1); and how well graph prompting methods overcome the negative transferring caused by pre-training strategies (Section 5.2). **In flexibility**, we wish to uncover how powerful different graph prompt methods are to simulate data operations (Section 5.3). **In efficiency**, we wish to know how efficient are these graph prompt methods in terms of time and space cost (Section 5.4).

3 Benchmark Methods

In general, graph prompt learning aims to transfer pre-trained knowledge with the expectation of achieving better positive transfer effects in downstream tasks compared to classical ‘Pre-train & Fine-tune’ methods. To conduct a comprehensive comparison, we first consider using a supervised method as the baseline for judging positive transfer (For example, if a ‘pre-training and fine-tuning’ approach can not beat supervised results, one negative transfer case is observed). We introduce various pre-training methods to generate pre-trained knowledge, upon which we perform fine-tuning

²Our code is available at <https://github.com/sheldonresearch/ProG>

as the comparison for graph prompts. Finally, we apply current popular graph prompt learning methods with various pre-training methods to investigate their knowledge transferability. The details of these baselines are as follows (Further introduction can be referred to Appendix A):

- **Supervised Method:** In this paper, we utilize Graph Convolutional Network (GCN) [21] as the baseline, which is one of the classical and effective graph models based on graph convolutional operations [50, 67, 32, 69] and also the backbone for both ‘Pre-train & Fine-tune’ and graph prompt learning methods. We also explore more kinds of graph models like GraphSAGE [13], GAT [54], and Graph Transformer [43], and put the results in Appendix E.
- **‘Pre-train & Fine-tune’ Methods:** We select 6 mostly used pre-training methods covering node-level, edge-level, and graph-level strategies. For **node-level**, we consider **DGI** [55] and **GraphMAE** [14], where DGI maximizes the mutual information between node and graph representations for informative embeddings and GraphMAE learns deep node representations by reconstructing masked features. For **edge-level**, we introduce **EdgePreGPPT** [46] and **EdgePreGprompt** [33], where EdgePreGPPT calculates the dot product as the link probability of node pairs and EdgePreGprompt samples triplets from label-free graphs to increase the similarity between the contextual subgraphs of linked pairs while decreasing the similarity of unlinked pairs. For **graph-level**, we involve **GCL** [65], **SimGRACE** [60], where GCL maximizes agreement between different graph augmentations to leverage structural information and SimGRACE tries to perturb the graph model parameter spaces and narrow down the gap between different perturbations for the same graph.
- **Graph Prompt Learning Methods:** With our proposed unified view, current popular graph prompt learning methods can be easily classified into two types, ‘Prompt as graph’ and ‘Prompt as token’ types. Specifically, the **‘Prompt as graph’** type means a graph prompt has multiple prompt tokens with inner structure and insert patterns, and we introduce **All-in-one** [47], which aims to learn a set of insert patterns for original graphs via learnable graph prompt modules. For the **‘Prompt as token’** type, we consider involving **GPPT** [46], **Gprompt** [33], **GPF** and **GPF-plus** [8]. Concretely, GPPT defines graph prompts as additional tokens that contain task tokens and structure tokens that align downstream node tasks and link prediction pretext. Gprompt focuses on inserting the prompt vector into the graph pooling by element-wise multiplication. GPF and GPF-plus mainly add soft prompts to all node features of the input graph.

ProG: A Unified Library for Graph Prompts. We offer our developed graph prompt library ProG as shown in Figure 2. In detail, the library first offers a **Model Backbone** module, including many widely-used graph models, which can be used for supervised learning independently. Then, it designs a **Pre-training** module involving many pre-training methods from various level types, which can pre-train the graph model initialized from the model backbone to preserve pre-trained knowledge and further fine-tune the pre-trained graph model. Finally, it seamlessly incorporates different graph prompt learning methods together into a unified **Prompting** module, which can freeze and execute prompt tuning for the pre-trained graph model from the pre-training module. For a fair comparison, it defines an **Evaluation** module, including comprehensive metrics, batch evaluator, and dynamic dispatcher, to ensure the same settings across different selected methods. Many essential operations are fused into the **Utils** module for repeated use, and all the data-related operations are included in the **Data** module for user-friendly operations to use the library. Additionally, the **Tutorial** module introduces the key functions that ProG offers.

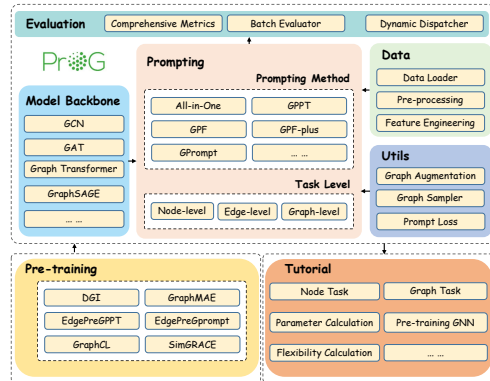


Figure 2: An overview of ProG

4 Benchmark Settings

4.1 Datasets

To figure out how adaptive existing graph prompts on different graphs, we conduct our experiments across 15 datasets on node-level or graph-level tasks, providing a broad and rigorous testing ground

Table 1: Statistics of all datasets. N indicates node classification, and G indicates graph classification.

| Dataset | Graphs | Avg.nodes | Avg.edges | Features | Node classes | Task (N / G) | Category |
|------------|--------|-----------|-----------|----------|--------------|--------------|--------------------------|
| Cora | 1 | 2,708 | 5,429 | 1,433 | 7 | N | Homophilic |
| Pubmed | 1 | 19,717 | 88,648 | 500 | 3 | N | Homophilic |
| CiteSeer | 1 | 3,327 | 9,104 | 3,703 | 6 | N | Homophilic |
| Actor | 1 | 7600 | 30019 | 932 | 5 | N | Heterophilic |
| Wisconsin | 1 | 251 | 515 | 1703 | 5 | N | Heterophilic |
| Texas | 1 | 183 | 325 | 1703 | 5 | N | Heterophilic |
| ogbn-arxiv | 1 | 169,343 | 1,166,243 | 128 | 40 | N | Homophilic & Large scale |

| Dataset | Graphs | Avg.nodes | Avg.edges | Features | Graph classes | Task (N / G) | Domain |
|-------------|--------|-----------|-----------|----------|---------------|--------------|----------------|
| MUTAG | 188 | 17.9 | 19.8 | 7 | 2 | G | Small Molecule |
| IMDB-BINARY | 1000 | 19.8 | 96.53 | 0 | 2 | G | Social Network |
| COLLAB | 5000 | 74.5 | 2457.8 | 0 | 3 | G | Social Network |
| PROTEINS | 1,113 | 39.1 | 72.8 | 3 | 2 | G | Proteins |
| ENZYMES | 600 | 32.6 | 62.1 | 3 | 6 | G | Proteins |
| DD | 1,178 | 284.1 | 715.7 | 89 | 2 | G | Proteins |
| COX2 | 467 | 41.2 | 43.5 | 3 | 2 | G | Small Molecule |
| BZR | 405 | 35.8 | 38.4 | 3 | 2 | G | Small Molecule |

for algorithmic comparisons. As shown in Table 1, we include 7 node classification datasets, covering homophilic datasets (Cora, Citeseer, PubMed, ogbn-arxiv) [42, 35, 15], heterophilic datasets (Wisconsin, Texas, Actor) [37, 53], and large-scale datasets (ogbn-arxiv). Additionally, we consider 8 graph classification datasets from different domains, including social networks (IMDB-B, COLLAB) [63], biological datasets (ENZYMES, PROTEINS, DD) [7, 2, 58], and small molecule datasets (MUTAG, COX2, BZR) [22, 41]. More details on these datasets can be found in Appendix C.

4.2 Implementation Details

Data Split. For the node task, we use 90% data as the test set. For the graph task, we use 80% data as the test set. To ensure the robustness of our findings, we repeat the sampling process five times to construct k-shot tasks for both node and graph tasks. We then report the average and standard deviation over these five results.

Metrics. Following existing multi-class classification benchmarks, we select Accuracy, Macro F1 Score, and AUROC (Area Under the Receiver Operating Characteristic Curve) as our main performance metrics [47, 68]. The Accuracy results are used to provide a straightforward indication of the model’s overall performance in Section 5. F1 Score and AUROC results are provided in Appendix D.

Prompt Adaptation to Specific Tasks. We observe that certain prompt methods, such as GPPT, are only applicable to node classification, while GPF and GPF-plus are only suitable for graph classification. To this end, we design graph task tokens to replace the original node task tokens to make GPPT also applicable in graph-level tasks. Inspired by All-in-one and Gprompt, we utilize induced graphs in our benchmark to adapt GPF and GPF-plus for the node task. The concrete adaptation designs are shown in Appendix A.

Hyperparameter Optimization. To manage the influence of hyperparameter selection and maintain fairness, the evaluation process is standardized both with and without hyperparameter adjustments. Initially, default hyperparameters as specified in the original publications are used. To ensure fairness during hyperparameter tuning, **random search** is then employed to optimize the settings. In each trial across different datasets, a set of hyperparameters is randomly selected from a predefined search space for each model. For additional details on metrics, default hyperparameters, search spaces, and other implementation aspects, please see Appendix D.

5 Experimental Results

5.1 Overall Performance of Graph Prompt Learning

In Table 2 and Table 3, we present the optimal experimental results of the supervised methods, classical ‘Pre-train & Fine-tune’ methods, and various graph prompt methods on 1-shot node/graph

classification tasks across 15 datasets, where the evaluation metric is accuracy (Complete results can be referred to Appendix E). Here the 'optimal' results mean that the best result is shown from six classical 'Pre-train & Fine-tune' methods or from six pretraining-based variants of each graph prompt method. This presentation style can help reflect the upper bounds of classical 'Pre-train & Fine-tune' methods and graph prompt methods, better highlighting the advantages of graph prompts.

From these results, we can observe consistent advantages of various graph prompt learning methods. Almost all graph prompt methods consistently outperform the Supervised method on both node-level and graph-level tasks, demonstrating positive transfer and superior performance. Specifically, graph prompt methods generally surpass classical 'Pre-train & Fine-tune' methods across most datasets, showcasing enhanced knowledge transferability. In node-level tasks, GPF-plus, which focuses on modifying node features via learnable prompt features, achieves the best results on 4 out of 7 datasets with significant effects, attributed to its inherent ability to easily adapt to different pre-training methods [8]. In contrast, All-in-one, which concentrates on modifying graphs via learnable subgraphs, achieves the best results on 7 out of 8 datasets on graph-level tasks.

Table 2: Performance on 1-shot node classification. The best results for each dataset are highlighted in bold with a dark red background. The second-best are underlined with a light red background.

| Methods/Datasets | Cora | Citeseer | Pubmed | Wisconsin | Texas | Actor | Ogbn-arxiv |
|-----------------------|-----------------------------------|-----------------------------------|-----------------------------------|----------------------------------|----------------------------------|----------------------------------|----------------------------------|
| Supervised | 26.56 \pm 5.55 | 21.78 \pm 7.32 | 39.37 \pm 16.34 | 41.60 \pm 3.10 | 37.97 \pm 5.80 | 20.57 \pm 4.47 | 10.99 \pm 3.19 |
| Pre-train & Fine-tune | 52.61 \pm 1.73 | 35.05 \pm 4.37 | 46.74 \pm 14.89 | 40.69 \pm 4.13 | 46.88 \pm 4.69 | 20.74 \pm 4.12 | 16.21 \pm 3.82 |
| GPPT | 43.15 \pm 9.44 | 37.26 \pm 6.17 | 48.31\pm17.72 | 30.40 \pm 6.81 | 31.81 \pm 15.33 | 22.58 \pm 1.97 | 14.65 \pm 3.07 |
| All-in-one | 52.39 \pm 10.17 | 40.41 \pm 2.80 | 45.17 \pm 6.45 | 78.24 \pm 16.68 | 65.49 \pm 7.06 | 24.61 \pm 2.80 | 13.16 \pm 5.98 |
| Gprompt | 56.66\pm11.22 | <u>53.21\pm10.94</u> | 39.74 \pm 15.35 | 83.80 \pm 2.44 | 33.25 \pm 40.11 | <u>25.26\pm1.10</u> | 75.72\pm4.95 |
| GPF | 38.57 \pm 5.41 | 31.16 \pm 8.05 | 49.99\pm8.86 | <u>88.67\pm5.78</u> | 87.40\pm3.40 | 28.70 \pm 3.35 | 78.37\pm5.47 |
| GPF-plus | <u>55.77\pm10.30</u> | 59.67\pm11.87 | 46.64 \pm 18.97 | 91.03\pm4.11 | 95.83\pm4.19 | 29.32\pm8.56 | 71.98 \pm 12.23 |

Table 3: Performance on 1-shot graph classification. The best results for each dataset are highlighted in bold with a dark red background. The second-best are underlined with a light red background.

| Methods/Datasets | IMDB-B | COLLAB | PROTEINS | MUTAG | ENZYMES | COX2 | BZR | DD |
|-----------------------|----------------------------------|-----------------------------------|----------------------------------|----------------------------------|----------------------------------|----------------------------------|-----------------------------------|----------------------------------|
| Supervised | 57.30 \pm 0.98 | 47.23 \pm 0.61 | 56.36 \pm 7.97 | 65.20 \pm 6.70 | 20.58 \pm 2.00 | 27.08 \pm 1.95 | 25.80 \pm 6.53 | 55.33 \pm 6.22 |
| Pre-train & Fine-tune | 57.75 \pm 1.22 | 48.10 \pm 0.23 | 63.44 \pm 3.64 | 65.47 \pm 5.89 | 22.21 \pm 2.79 | <u>76.19\pm5.41</u> | 34.69 \pm 8.50 | 57.15 \pm 4.32 |
| GPPT | 50.15 \pm 0.75 | 47.18 \pm 5.93 | 60.92 \pm 2.47 | 60.40 \pm 15.43 | 21.29 \pm 3.79 | 78.23\pm1.38 | 59.32 \pm 11.22 | 57.69 \pm 6.89 |
| All-in-one | 60.07\pm4.81 | <u>51.66\pm0.26</u> | 66.49\pm6.26 | 79.87\pm5.34 | 23.96\pm1.45 | 76.14 \pm 5.51 | 79.20\pm1.65 | 59.72\pm1.52 |
| Gprompt | 54.75 \pm 12.43 | <u>48.25\pm13.64</u> | 59.17 \pm 11.26 | 73.60\pm4.76 | 22.29 \pm 3.50 | 54.64 \pm 9.94 | 55.43 \pm 13.69 | 57.81 \pm 2.68 |
| GPF | 59.65\pm5.06 | 47.42 \pm 11.22 | 63.91\pm3.26 | 68.40 \pm 5.09 | 22.00 \pm 1.25 | 65.79 \pm 17.72 | <u>71.67\pm14.71</u> | <u>59.36\pm1.18</u> |
| GPF-plus | 57.93 \pm 1.62 | 47.24 \pm 0.29 | 62.92 \pm 2.78 | 65.20 \pm 6.04 | 22.92\pm1.64 | 33.78 \pm 1.52 | 71.17 \pm 14.92 | 57.62 \pm 2.42 |

Similar observations can also be found in 3-shot and 5-shot results in the optimal and complete versions, which can be checked in Appendix E. Additionally, for comprehensive, we also offer the curve of the shot number from 1 to 10 and the accuracy in Appendix E.

5.2 Impact of Pre-training Methods

From Table 2, we further find that on the Wisconsin dataset, the classical Pre-train & Fine-tune method consistently exhibited negative transfer, whereas most graph prompt methods effectively transferred upstream pre-training knowledge, achieving significant results and positive transfer. Therefore, it is crucial to explore the effectiveness of various pre-training methods, especially for graph prompt learning. Table 4 and Table 5 show the experimental results of six variants of GPF-plus/All-in-one and the corresponding 'Pre-train & Fine-tune' methods. Concretely, we have the following key observations:

Consistency in Pretext and Downstream Tasks. In node classification tasks, the pre-trained knowledge learned by the node-level pre-training method (e.g., GraphMAE) shows superior transferability to downstream tasks using GPF-plus, achieving better results compared to traditional fine-tuning methods. Conversely, in graph classification tasks, the graph-level pre-training method (e.g., SimGRACE) demonstrates better transferability with All-in-one, resulting in more significant outcomes.

Table 4: 1-shot node classification performance comparison across different datasets of GPF-plus. \uparrow/\downarrow indicates the positive/negative transfer compared to the supervised baseline. $\downarrow(\%)$ means the negative transfer rate, computed by dividing the number of \downarrow by the number of datasets.

| Methods/Datasets | Cora | Citeseer | Pubmed | Wisconsin | Texas | Actor | Ogbn-arxiv | $\downarrow(\%)$ |
|------------------|---|---|---|--|--|--|---|------------------|
| Supervised | 26.56 \pm 5.55 (-) | 21.78 \pm 7.32 (-) | 39.37 \pm 16.34 (-) | 41.60 \pm 3.10 (-) | 37.97 \pm 5.80 (-) | 20.57 \pm 4.47 (-) | 10.99 \pm 3.19 (-) | 0.0% |
| DGI | 33.15 \pm 7.84 (\uparrow) | 21.64 \pm 3.92 (\downarrow) | 42.01 \pm 12.54 (\uparrow) | 37.49 \pm 7.56 (\downarrow) | 45.31 \pm 5.01 (\uparrow) | 19.76 \pm 3.53 (\downarrow) | 7.21 \pm 2.91 (\downarrow) | 57.0% |
| + GPF_plus | 17.29 \pm 6.18 (\downarrow) | 26.60 \pm 13.24 (\uparrow) | 34.02 \pm 11.94 (\downarrow) | 76.34 \pm 6.26 (\uparrow) | 79.54 \pm 1.89 (\uparrow) | 22.42 \pm 9.66 (\uparrow) | 16.83 \pm 10.02 (\uparrow) | 29% |
| GraphMAE | 32.93 \pm 3.17 (\uparrow) | 21.26 \pm 3.57 (\downarrow) | 42.99 \pm 14.25 (\uparrow) | 36.80 \pm 7.17 (\downarrow) | 37.81 \pm 8.62 (\downarrow) | 19.86 \pm 2.70 (\downarrow) | 12.35 \pm 3.60 (\uparrow) | 43.0% |
| + GPF_plus | 54.26 \pm 7.48 (\uparrow) | 59.67 \pm 11.87 (\uparrow) | 46.64 \pm 18.57 (\uparrow) | 89.17 \pm 9.79 (\uparrow) | 94.09 \pm 4.40 (\uparrow) | 26.58 \pm 7.84 (\uparrow) | 49.81 \pm 2.62 (\uparrow) | 0.0% |
| EdgePreGPPT | 38.12 \pm 5.29 (\uparrow) | 18.09 \pm 5.39 (\downarrow) | 46.74 \pm 14.09 (\uparrow) | 35.31 \pm 9.31 (\downarrow) | 47.66 \pm 2.37 (\uparrow) | 19.17 \pm 2.53 (\downarrow) | 16.21 \pm 3.82 (\uparrow) | 43.0% |
| + GPF_plus | 28.49 \pm 18.73 (\downarrow) | 28.04 \pm 14.31 (\uparrow) | 46.51 \pm 15.84 (\uparrow) | 91.03 \pm 4.11 (\uparrow) | 92.14 \pm 5.46 (\uparrow) | 29.32 \pm 8.56 (\uparrow) | 71.98 \pm 12.23 (\uparrow) | 14.0% |
| EdgePreGprompt | 35.57 \pm 5.83 (\uparrow) | 22.28 \pm 3.80 (\uparrow) | 41.50 \pm 7.54 (\uparrow) | 40.69 \pm 4.13 (\downarrow) | 40.62 \pm 7.95 (\uparrow) | 20.74 \pm 4.16 (\uparrow) | 14.83 \pm 2.38 (\uparrow) | 14.0% |
| + GPF_plus | 55.77 \pm 10.30 (\uparrow) | 49.43 \pm 8.21 (\uparrow) | 42.79 \pm 18.18 (\uparrow) | 85.44 \pm 2.23 (\uparrow) | 87.70 \pm 3.77 (\uparrow) | 22.68 \pm 3.64 (\uparrow) | 57.44 \pm 6.95 (\uparrow) | 0.0% |
| GCL | 52.61 \pm 1.73 (\uparrow) | 27.02 \pm 4.31 (\uparrow) | 42.49 \pm 11.29 (\uparrow) | 33.94 \pm 7.74 (\downarrow) | 40.31 \pm 13.68 (\uparrow) | 20.19 \pm 1.98 (\downarrow) | 4.65 \pm 1.19 (\downarrow) | 43.0% |
| + GPF_plus | 34.18 \pm 17.71 (\downarrow) | 28.86 \pm 22.88 (\uparrow) | 37.02 \pm 11.29 (\downarrow) | 68.57 \pm 8.77 (\uparrow) | 95.83 \pm 4.19 (\uparrow) | 22.82 \pm 4.99 (\uparrow) | 32.11 \pm 4.86 (\uparrow) | 29.0% |
| SimGRACE | 40.40 \pm 4.66 (\uparrow) | 35.05 \pm 4.37 (\uparrow) | 37.59 \pm 8.17 (\downarrow) | 37.37 \pm 3.68 (\downarrow) | 46.88 \pm 4.64 (\uparrow) | 19.78 \pm 1.89 (\downarrow) | 8.13 \pm 3.26 (\downarrow) | 57.0% |
| + GPF_plus | 21.33 \pm 14.86 (\downarrow) | 24.61 \pm 21.21 (\downarrow) | 35.90 \pm 9.06 (\downarrow) | 73.64 \pm 5.25 (\uparrow) | 90.74 \pm 3.12 (\uparrow) | 20.51 \pm 4.24 (\uparrow) | 46.71 \pm 3.17 (\uparrow) | 43.0% |

Table 5: 1-shot graph classification performance comparison across different datasets of All-in-one.

| Methods/Datasets | IMDB-B | COLLAB | PROTEINS | MUTAG | ENZYMES | COX2 | BZR | DD | $\downarrow(\%)$ |
|------------------|--|--|--|--|--|---|---|--|------------------|
| Supervised | 57.30 \pm 0.98 (-) | 47.23 \pm 0.61 (-) | 56.36 \pm 7.97 (-) | 65.20 \pm 6.70 (-) | 20.58 \pm 2.00 (-) | 27.08 \pm 11.94 (-) | 25.80 \pm 6.53 (-) | 55.33 \pm 6.22 (-) | 0.0% |
| DGI | 57.32 \pm 0.90 (\uparrow) | 42.22 \pm 0.73 (\downarrow) | 64.65 \pm 2.10 (\uparrow) | 64.13 \pm 7.90 (\downarrow) | 17.83 \pm 1.88 (\downarrow) | 29.44 \pm 9.68 (\uparrow) | 26.48 \pm 7.61 (\uparrow) | 57.15 \pm 4.32 (\uparrow) | 38.0% |
| + All-in-one | 60.07 \pm 4.81 (\uparrow) | 39.56 \pm 5.00 (\downarrow) | 62.58 \pm 7.07 (\uparrow) | 75.73 \pm 7.75 (\uparrow) | 23.96 \pm 1.45 (\uparrow) | 50.72 \pm 9.93 (\uparrow) | 79.20 \pm 1.65 (\uparrow) | 55.97 \pm 6.52 (\uparrow) | 13.0% |
| GraphMAE | 57.70 \pm 1.13 (\uparrow) | 48.10 \pm 0.23 (\uparrow) | 63.57 \pm 3.57 (\uparrow) | 65.20 \pm 5.00 (-) | 22.21 \pm 2.79 (\uparrow) | 28.47 \pm 14.72 (\uparrow) | 25.80 \pm 6.53 (-) | 57.54 \pm 4.41 (\uparrow) | 0.0% |
| + All-in-one | 52.62 \pm 3.04 (\downarrow) | 40.82 \pm 14.63 (\downarrow) | 66.49 \pm 6.26 (\uparrow) | 72.27 \pm 8.87 (\uparrow) | 23.21 \pm 1.72 (\uparrow) | 56.68 \pm 7.38 (\uparrow) | 75.37 \pm 5.99 (\uparrow) | 58.77 \pm 1.05 (\uparrow) | 25.0% |
| EdgePreGPPT | 57.20 \pm 0.85 (\downarrow) | 47.14 \pm 0.55 (\downarrow) | 58.27 \pm 10.66 (\uparrow) | 64.27 \pm 4.73 (\downarrow) | 19.79 \pm 2.17 (\downarrow) | 27.83 \pm 13.44 (\uparrow) | 72.10 \pm 14.30 (\uparrow) | 52.82 \pm 9.38 (\downarrow) | 63.0% |
| + All-in-one | 59.12 \pm 0.77 (\uparrow) | 42.74 \pm 4.65 (\downarrow) | 65.71 \pm 5.49 (\uparrow) | 79.87 \pm 5.34 (\uparrow) | 20.92 \pm 2.04 (\uparrow) | 60.27 \pm 16.97 (\uparrow) | 64.20 \pm 19.74 (\uparrow) | 56.24 \pm 2.46 (\uparrow) | 13.0% |
| EdgePreGprompt | 57.35 \pm 0.92 (\uparrow) | 47.20 \pm 0.53 (\downarrow) | 61.84 \pm 2.59 (\uparrow) | 62.67 \pm 2.67 (\downarrow) | 19.75 \pm 2.33 (\downarrow) | 27.13 \pm 12.05 (\uparrow) | 29.44 \pm 11.20 (\uparrow) | 56.16 \pm 5.10 (\uparrow) | 38.0% |
| + All-in-one | 53.78 \pm 2.82 (\downarrow) | 42.87 \pm 6.19 (\downarrow) | 61.82 \pm 7.53 (\uparrow) | 75.47 \pm 5.00 (\uparrow) | 21.88 \pm 0.56 (\uparrow) | 49.06 \pm 5.53 (\uparrow) | 66.60 \pm 17.36 (\uparrow) | 57.60 \pm 4.37 (\uparrow) | 25.0% |
| GCL | 57.75 \pm 1.02 (\uparrow) | 39.62 \pm 0.63 (\downarrow) | 63.44 \pm 3.64 (\uparrow) | 65.07 \pm 8.38 (\downarrow) | 23.96 \pm 1.99 (\uparrow) | 53.14 \pm 21.32 (\uparrow) | 29.07 \pm 7.00 (\uparrow) | 60.62 \pm 1.56 (\uparrow) | 25.0% |
| + All-in-one | 58.75 \pm 0.80 (\uparrow) | 51.66 \pm 2.60 (\uparrow) | 64.36 \pm 7.30 (\uparrow) | 70.53 \pm 8.58 (\uparrow) | 19.46 \pm 2.85 (\downarrow) | 52.55 \pm 13.51 (\uparrow) | 50.93 \pm 16.83 (\uparrow) | 59.72 \pm 1.52 (\uparrow) | 13.0% |
| SimGRACE | 57.33 \pm 0.96 (\uparrow) | 46.89 \pm 0.42 (\downarrow) | 60.07 \pm 3.21 (\uparrow) | 65.47 \pm 5.89 (\uparrow) | 19.71 \pm 1.76 (\downarrow) | 76.19 \pm 5.41 (\uparrow) | 28.48 \pm 6.49 (\uparrow) | 53.23 \pm 9.71 (\downarrow) | 38.0% |
| + All-in-one | 58.83 \pm 0.85 (\uparrow) | 47.60 \pm 3.90 (\uparrow) | 61.17 \pm 1.73 (\uparrow) | 65.47 \pm 6.91 (\uparrow) | 22.50 \pm 1.56 (\uparrow) | 76.14 \pm 5.51 (\uparrow) | 69.88 \pm 18.06 (\uparrow) | 58.26 \pm 1.18 (\uparrow) | 0.0% |

These observations indicate that although graph prompts may perform better, they are also selective to specific pre-training tasks. In general, node-level pre-training is more suitable for node-level tasks, while graph-level pre-training is more effective for graph-level tasks.

Mitigation of Negative Transfer. Classical pre-trained models often experience **significant negative transfer issues** during fine-tuning. In contrast, **prompt tuning can effectively handle these problems**. For example, in node classification, GPF-plus can reduce the negative transfer rate from 43% to 0%. Similarly, in graph classification, All-in-one successfully resolves negative transfer issues, reducing the rate from 38% to 0%. This demonstrates the efficacy of graph prompt methods, such as GPF-plus and All-in-one, in mitigating negative transfer, ensuring more reliable and accurate knowledge transfer in both node-level and graph-level tasks.

Figure 3a and Figure 3b respectively show the heat maps of Table 4 and Table 5, respectively, for help better understanding the impact of pre-training methods (Other heat maps of other graph prompt learning methods can be referred to Appendix E).

5.3 Flexibility Analysis

We leverage *error bound*, defined by Equation 1, to measure the flexibility of graph prompt methods [47]. The *error bound* quantifies the difference in graph representations between the original

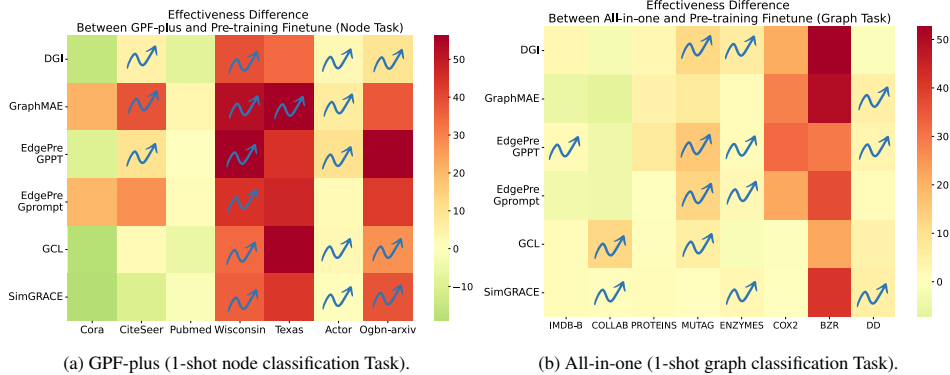


Figure 3: Head map of GPF-plus and All-in-one on node-level and graph-level tasks across various pre-training methods and datasets. The green interval shows that the graph prompt learning method performs worse than ‘Pre-train & Fine-tune’. The yellow-to-red interval shows that the graph prompt learning method performs better. **An upward arrow** indicates that the ‘Pre-train & Fine-tune’ method experiences **negative transfer**, while the graph **prompt** learning method can **alleviate it significantly**.

graph and the manually manipulated graph restored by graph prompt methods. These manipulations include dropping nodes, dropping edges, and masking features. Table 6 presents the specific results. The experimental results indicate that **all graph prompt methods significantly reduced the error compared to the original error across all types of manipulations**. Notably, All-in-one and Gprompt achieved the lowest error bounds, demonstrating their flexibility in adapting to graph transformations. Specifically, All-in-one achieved a RED% of 95.06, indicating a 95.06% reduction in original error, underscoring its comprehensive approach to capturing graph transformations. These findings highlight the importance of flexibility in prompt design for effective knowledge transfer. Flexible prompts that can adapt to various graph transformations enable more powerful graph data operation capability, thereby improving the performance of downstream tasks. Moreover, this flexibility is inspiring for more customized graph data augmentation, indicating more potential for other applications.

Table 6: Error bound between different prompt methods, RED(%): average reduction of each method to the original error.

| Prompt Model | Drop Nodes | Drop Edges | Mask Feature | RED (%) |
|--------------|------------|------------|--------------|---------|
| ori_error | 0.4862 | 0.0713 | 0.6186 | |
| All-in-one | 0.0200 | 0.0173 | 0.0207 | 95.06 ↓ |
| Gprompt | 0.0218 | 0.0141 | 0.0243 | 94.88 ↓ |
| GPF-plus | 0.0748 | 0.0167 | 0.1690 | 77.85 ↓ |
| GPF | 0.0789 | 0.0146 | 0.1858 | 76.25 ↓ |

5.4 Efficiency Analysis

We evaluate the efficiency and training duration of various graph prompt methods across tasks, focusing on node classification in Cora and Wisconsin and graph classification in BZR and DD. **GPF-plus** achieves **high accuracy** with **relatively low training times** despite its **larger parameter size** on Cora and Wisconsin. In contrast, **All-in-One**, with a **smaller parameter size**, has **slightly larger training times** than GPF-plus due to similarity calculations. Among all the graph prompt methods, GPPT shows lower performance and the longest training duration because of its k-means clustering update in every epoch. Besides, we **compare the tunable parameters of graph prompt methods** during the training phase with **classical ‘Pre-train & Fine-tune’ methods** with various backbones for further analyzing the advantages of graph prompt learning. Consider an input graph has N nodes, M edges, d features, along with a graph model which has L layers with maximum layer dimension D . **Classical backbones (e.g., GCN) have $O(dD + D^2(L - 1))$ learnable parameter complexity**. Other more complex backbones (e.g., graph transformer) may have larger parameters. In the **graph prompt learning framework**, tuning the prompt parameters with a frozen pre-trained backbone makes training converge faster. For **All-in-One**, with K tokens and m edges, **learnable parameter complexity is $O(Kd)$** . Other graph prompt methods also have similar tunable parameter advantages, with specific sizes depending on their designs (e.g., Gprompt only includes a learnable vector inserted into the graph pooling by element-wise multiplication).

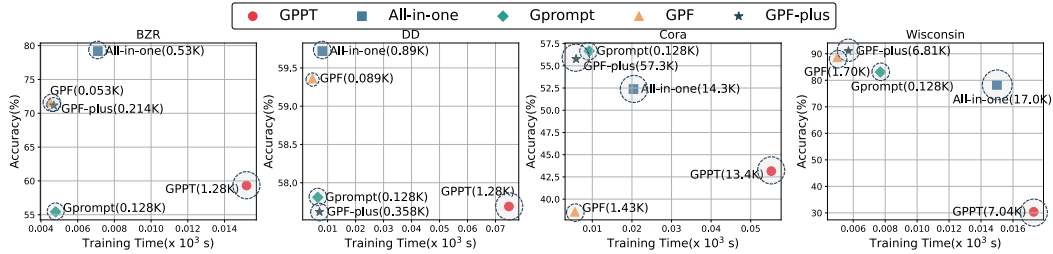


Figure 4: Analysis of accuracy, training time and tunable parameters of various graph prompt methods. Note that the gray area enclosed by the dashed line represents the scale of the tunable parameters.

5.5 Adaptability of Graph Prompts on Different Graphs.

Besides the main observations mentioned before, we here further discuss the adaptability of graph prompts on various kinds of graphs. Specifically, considering graph prompts performance on **various domains** as shown in Table 3, we can find that the adaptability to various domains of graph prompt methods, particularly those of the Prompt as Token type, is not always good, where graph prompt methods even may encounter negative transfer issues. GPPT, in particular, faces significant challenges, likely because first-order subgraphs fail to comprehensively capture graph patterns, introducing noise and affecting transferability. For **Large-scale Datasets** like node-level dataset ogbn-arxiv in Table 2, the transferability of graph prompt methods involving subgraph operations, such as GPPT and All-in-one, requires further enhancement. Furthermore, regarding the relatively large graph-level dataset DD in Table 3, all graph prompt methods achieve positive transfer and outperform Pre-train & Fine-tune. Despite this success, the magnitude of improvement indicates that the transferability on graph-level large-scale datasets also requires further enhancement. For **Homophilic & Heterophilic Datasets** in Table 2, most graph prompt methods can handle both homophilic and heterophilic graph datasets effectively, with the exception of GPPT, possibly due to instability caused by the clustering algorithms of GPPT. For **Pure Structural Dataset** like IMDB-BINARY and COLLAB in Table 3, the ability to handle pure structural datasets of graph prompt methods needs further refinement. In detail, some ‘Pre-train & Fine-tune’ methods can achieve positive transfer on these datasets, while some variants of the All-in-one method even exhibit negative transfer. Further detailed analysis can be referred to Appendix E.

6 Conclusion and Future Plan

In this paper, we introduce ProG, a comprehensive benchmark for comparing various graph prompt learning methods with supervised and ‘Pre-train & Fine-tune’ methods. Our evaluation includes 6 Pre-training methods and 5 graph prompt learning methods across 15 real-world datasets, demonstrating that graph prompt methods generally outperform classical methods. This performance is attributed to the high flexibility and low complexity of graph prompt methods. Our findings highlight the promising future of graph prompt learning research and its challenges. By making ProG open-source, we aim to promote further research and better evaluations of graph prompt learning methods.

As the empirical analysis in this paper, graph prompt learning shows its significant potential beyond task transferring, though our current evaluation focuses on node/graph-level tasks. We view ProG as a long-term project and are committed to its continuous development. Our future plans include expanding to a wider range of graph tasks, adding new datasets, etc. Ultimately, we aim to develop ProG into a robust graph prompt toolbox with advanced features like automated prompt selection.

Acknowledgments and Disclosure of Funding

This work was supported by National Key Research and Development Program of China Grant No. 2023YFF0725100 and Guangzhou-HKUST(GZ) Joint Funding Scheme 2023A03J0673. The research of Cheng was supported in part by project #MMT-p2-23 of the Shun Hing Institute of Advanced Engineering, The Chinese University of Hong Kong and by grants from the Research Grant Council of the Hong Kong Special Administrative Region, China (No. CUHK 14217622).

References

- [1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.
- [2] Karsten M Borgwardt, Cheng Soon Ong, Stefan Schönauer, SVN Vishwanathan, Alex J Smola, and Hans-Peter Kriegel. Protein function prediction via graph kernels. *Bioinformatics*, 21(suppl_1):i47–i56, 2005.
- [3] Mouxiang Chen, Zemin Liu, Chenghao Liu, Jundong Li, Qiheng Mao, and Jianling Sun. ULTRA-DP: Unifying Graph Pre-training with Multi-task Graph Dual Prompt. *arXiv preprint*, 2023.
- [4] Nuo Chen, Yuhan Li, Jianheng Tang, and Jia Li. Graphwiz: An instruction-following language model for graph computational problems. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 353–364, 2024.
- [5] Xi Chen, Siwei Zhang, Yun Xiong, Xixi Wu, Jiawei Zhang, Xiangguo Sun, Yao Zhang, Yinglong Zhao, and Yulin Kang. Prompt learning on temporal interaction graphs. *arXiv:2402.06326*, 2024.
- [6] Jiashun Cheng, Man Li, Jia Li, and Fugee Tsung. Wiener graph deconvolutional network improves graph self-supervised learning. In *AAAI*, pages 7131–7139, 2023.
- [7] Paul D Dobson and Andrew J Doig. Distinguishing enzyme structures from non-enzymes without alignments. *Journal of molecular biology*, 330(4):771–783, 2003.
- [8] Taoran Fang, Yunchao Zhang, YANG YANG, Chunping Wang, and Lei Chen. Universal prompt tuning for graph neural networks. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 52464–52489. Curran Associates, Inc., 2023.
- [9] Matthias Fey and Jan Eric Lenssen. Fast graph representation learning with pytorch geometric. *arXiv preprint arXiv:1903.02428*, 2019.
- [10] Ziqi Gao, Xiangguo Sun, Zijing Liu, Yu Li, Hong Cheng, and Jia Li. Protein multimer structure prediction via prompt learning. In *The Twelfth International Conference on Learning Representations*, 2024.
- [11] Qingqing Ge, Zeyuan Zhao, Yiding Liu, Anfeng Cheng, Xiang Li, Shuaiqiang Wang, and Dawei Yin. Enhancing Graph Neural Networks with Structure-Based Prompt. *arXiv preprint*, 2023.
- [12] Chenghua Gong, Xiang Li, Jianxiang Yu, Cheng Yao, Jiaqi Tan, Chengcheng Yu, and Dawei Yin. Prompt Tuning for Multi-View Graph Contrastive Learning. *arXiv preprint*, 2023.
- [13] William L Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *NeurIPS*, 2017.
- [14] Zhenyu Hou, Xiao Liu, Yukuo Cen, Yuxiao Dong, Hongxia Yang, Chunjie Wang, and Jie Tang. Graphmae: Self-supervised masked graph autoencoders. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 594–604, 2022.
- [15] Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. *Advances in neural information processing systems*, 33:22118–22133, 2020.
- [16] Weihua Hu, Bowen Liu, Joseph Gomes, Marinka Zitnik, Percy Liang, Vijay S. Pande, and Jure Leskovec. Strategies for Pre-training Graph Neural Networks. In *ICLR*, 2020.
- [17] Qian Huang, Hongyu Ren, Peng Chen, Gregor Kržmanc, Daniel Zeng, Percy Liang, and Jure Leskovec. PRODIGY: Enabling In-context Learning Over Graphs. In *NeurIPS*, 2023.
- [18] Xunqiang Jiang, Yuanfu Lu, Yuan Fang, and Chuan Shi. Contrastive Pre-Training of GNNs on Heterogeneous Graphs. In *CIKM*, pages 803–812, 2021.
- [19] Ming Jin, Yizhen Zheng, Yuan-Fang Li, Chen Gong, Chuan Zhou, and Shirui Pan. Multi-Scale Contrastive Siamese Networks for Self-Supervised Graph Representation Learning. In *IJCAI*, pages 1477–1483, 2021.
- [20] Wei Jin, Tyler Derr, Yiqi Wang, Yao Ma, Zitao Liu, and Jiliang Tang. Node Similarity Preserving Graph Convolutional Networks. In *WSDM*, pages 148–156, 2021.

- [21] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017.
- [22] Nils Kriege and Petra Mutzel. Subgraph matching kernels for attributed graphs. In *Proceedings of the 29th International Conference on Machine Learning*, ICML’12, page 291–298, Madison, WI, USA, 2012. Omnipress.
- [23] Guohao Li, Matthias Müller, Bernard Ghanem, and Vladlen Koltun. Training graph neural networks with 1000 layers. In *International conference on machine learning*, pages 6437–6449. PMLR, 2021.
- [24] Guohao Li, Matthias Müller, Guocheng Qian, Itzel C Delgadillo, Abdulullah Abualshour, Ali Thabet, and Bernard Ghanem. Deepgcns: Making gcns go as deep as cnns. *IEEE transactions on pattern analysis and machine intelligence (TPAMI)*, 45(6):6923–6939, 2021.
- [25] Guohao Li, Matthias Muller, Ali Thabet, and Bernard Ghanem. Deepgcns: Can gcns go as deep as cnns? In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9267–9276, 2019.
- [26] Shengrui Li, Xueting Han, and Jing Bai. AdapterGNN: Efficient Delta Tuning Improves Generalization Ability in Graph Neural Networks. *arXiv preprint*, 2023.
- [27] Yuening Li, Zhengzhang Chen, Daochen Zha, Kaixiong Zhou, Haifeng Jin, Haifeng Chen, and Xia Hu. Automated anomaly detection via curiosity-guided search and self-imitation learning. *IEEE Transactions on Neural Networks and Learning Systems*, 2021.
- [28] Yuhan Li, Zhixun Li, Peisong Wang, Jia Li, Xiangguo Sun, Hong Cheng, and Jeffrey Xu Yu. A survey of graph meets large language model: Progress and future directions. *arXiv preprint arXiv:2311.12399*, 2023.
- [29] Yuhan Li, Peisong Wang, Zhixun Li, Jeffrey Xu Yu, and Jia Li. Zerog: Investigating cross-dataset zero-shot transferability in graphs. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1725–1735, 2024.
- [30] Yuhan Li, Peisong Wang, Xiao Zhu, Aochuan Chen, Haiyun Jiang, Deng Cai, Victor Wai Kin Chan, and Jia Li. Glbench: A comprehensive benchmark for graph with large language models. *arXiv preprint arXiv:2407.07457*, 2024.
- [31] Zhixun Li, Xin Sun, Yifan Luo, Yanqiao Zhu, Dingshuo Chen, Yingtao Luo, Xiangxin Zhou, Qiang Liu, Shu Wu, Liang Wang, et al. Gslb: The graph structure learning benchmark. *Advances in Neural Information Processing Systems*, 36, 2024.
- [32] Yang Liu, Jiashun Cheng, Haihong Zhao, Tingyang Xu, Peilin Zhao, Fugee Tsung, Jia Li, and Yu Rong. Segno: Generalizing equivariant graph neural networks with physical inductive biases. In *The Twelfth International Conference on Learning Representations*.
- [33] Zemin Liu, Xingtong Yu, Yuan Fang, and Xinming Zhang. Graphprompt: Unifying Pre-Training and Downstream Tasks for Graph Neural Networks. In *The Web Conference*, pages 417–428, 2023.
- [34] Yahui Long, Min Wu, Yong Liu, Yuan Fang, Chee Keong Kwoh, Jinmiao Chen, Jiawei Luo, and Xiaoli Li. Pre-training graph neural networks for link prediction in biomedical networks. *Bioinformatics*, 38(8):2254–2262, 2022.
- [35] Galileo Namata, Ben London, Lise Getoor, Bert Huang, and U Edu. Query-driven active surveying for collective classification. In *10th international workshop on mining and learning with graphs*, volume 8, page 1, 2012.
- [36] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *NeurIPS*, 32:8026–8037, 2019.
- [37] Hongbin Pei, Bingzhe Wei, Kevin Chen-Chuan Chang, Yu Lei, and Bo Yang. Geom-gcn: Geometric graph convolutional networks. In *International Conference on Learning Representations*, 2019.
- [38] Chen Qian, Huayi Tang, Zhirui Yang, Hong Liang, and Yong Liu. Can Large Language Models Empower Molecular Property Prediction? *arXiv preprint*, 2023.
- [39] Yu Rong, Yatao Bian, Tingyang Xu, Weiyang Xie, Ying WEI, Wenbing Huang, and Junzhou Huang. Self-Supervised Graph Transformer on Large-Scale Molecular Data. In *NeurIPS*, volume 33, pages 12559–12571, 2020.

- [40] Michael T Rosenstein, Zvika Marx, Leslie Pack Kaelbling, and Thomas G Dietterich. To transfer or not to transfer. In *NeurIPS*, volume 898, 2005.
- [41] Ryan Rossi and Nesreen Ahmed. The network data repository with interactive graph analytics and visualization. In *Proceedings of the AAAI conference on artificial intelligence*, volume 29, 2015.
- [42] Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. Collective classification in network data. *AI magazine*, 29(3):93–93, 2008.
- [43] Yunsheng Shi, Zhengjie Huang, Shikun Feng, Hui Zhong, Wenjing Wang, and Yu Sun. Masked label prediction: Unified message passing model for semi-supervised classification. In *IJCAI*, 2021.
- [44] Taylor Shin, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace, and Sameer Singh. Autoprompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts. In *EMNLP*, pages 4222–4235, 2020.
- [45] Fan-Yun Sun, Jordan Hoffmann, Vikas Verma, and Jian Tang. InfoGraph: Unsupervised and Semi-supervised Graph-Level Representation Learning via Mutual Information Maximization. In *ICLR*, 2020.
- [46] Mingchen Sun, Kaixiong Zhou, Xin He, Ying Wang, and Xin Wang. GPPT: Graph Pre-Training and Prompt Tuning to Generalize Graph Neural Networks. In *KDD*, pages 1717–1727, 2022.
- [47] 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.
- [48] Xiangguo Sun, Hong Cheng, Bo Liu, Jia Li, Hongyang Chen, Guandong Xu, and Hongzhi Yin. Self-Supervised Hypergraph Representation Learning for Sociological Analysis. *TKDE*, 2023.
- [49] Xiangguo Sun, Hongzhi Yin, Bo Liu, Qing Meng, Jiuxin Cao, Alexander Zhou, and Hongxu Chen. Structure learning via meta-hyperedge for dynamic rumor detection. *IEEE Transactions on Knowledge and Data Engineering*, 2022.
- [50] 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.
- [51] Zhen Tan, Ruocheng Guo, Kaize Ding, and Huan Liu. Virtual Node Tuning for Few-shot Node Classification. In *KDD*, pages 2177–2188, 2023.
- [52] Jianheng Tang, Jiajin Li, Ziqi Gao, and Jia Li. Rethinking graph neural networks for anomaly detection. In *International Conference on Machine Learning*, 2022.
- [53] Jie Tang, Jimeng Sun, Chi Wang, and Zi Yang. Social influence analysis in large-scale networks. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 807–816, 2009.
- [54] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. In *ICLR*, 2017.
- [55] Petar Veličković, William Fedus, William L Hamilton, Pietro Liò, Yoshua Bengio, and R Devon Hjelm. Deep graph infomax. *arXiv preprint arXiv:1809.10341*, 2018.
- [56] Kuansan Wang, Zhihong Shen, Chiyuan Huang, Chieh-Han Wu, Yuxiao Dong, and Anshul Kanakia. Microsoft academic graph: When experts are not enough. *Quantitative Science Studies*, 1(1):396–413, 2020.
- [57] Liyuan Wang, Mingtian Zhang, Zhongfan Jia, Qian Li, Chenglong Bao, Kaisheng Ma, Jun Zhu, and Yi Zhong. Afec: Active forgetting of negative transfer in continual learning. *Advances in Neural Information Processing Systems*, 34:22379–22391, 2021.
- [58] Song Wang, Yushun Dong, Xiao Huang, Chen Chen, and Jundong Li. Faith: Few-shot graph classification with hierarchical task graphs. *arXiv preprint arXiv:2205.02435*, 2022.
- [59] Yingying Wang, Yun Xiong, Xixi Wu, Xiangguo Sun, and Jiawei Zhang. Advanced drug interaction event prediction. *arXiv preprint arXiv:2402.11472*, 2024.
- [60] 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.

- [61] Yaochen Xie, Zhao Xu, and Shuiwang Ji. Self-Supervised Representation Learning via Latent Graph Prediction. In *ICML*, pages 24460–24477, 2022.
- [62] Dongkuan Xu, Wei Cheng, Dongsheng Luo, Haifeng Chen, and Xiang Zhang. InfoGCL: Information-Aware Graph Contrastive Learning. In *NeurIPS*, volume 34, pages 30414–30425, 2021.
- [63] Pinar Yanardag and SVN Vishwanathan. Deep graph kernels. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1365–1374, 2015.
- [64] Yuhao Yang, Lianghao Xia, Da Luo, Kangyi Lin, and Chao Huang. Graphpro: Graph pre-training and prompt learning for recommendation. In *Proceedings of the ACM on Web Conference 2024*, pages 3690–3699, 2024.
- [65] Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. Graph contrastive learning with augmentations. In *NeurIPS*, pages 5812–5823, 2020.
- [66] Haihong Zhao, Aochuan Chen, Xiangguo Sun, Hong Cheng, and Jia Li. All in one and one for all: A simple yet effective method towards cross-domain graph pretraining. In *Proceedings of the 27th ACM SIGKDD international conference on knowledge discovery & data mining (KDD'24)*, 2024.
- [67] Haihong Zhao, Bo Yang, Jiaxu Cui, Qianli Xing, Jiaying Shen, Fujin Zhu, and Jiannong Cao. Effective fault scenario identification for communication networks via knowledge-enhanced graph neural networks. *IEEE Transactions on Mobile Computing*, 2023.
- [68] Haihong Zhao, Chenyi Zi, Yang Liu, Chen Zhang, Yan Zhou, and Jia Li. Weakly supervised anomaly detection via knowledge-data alignment. In *Proceedings of the ACM on Web Conference 2024*, pages 4083–4094, 2024.
- [69] Zinan Zheng, Yang Liu, Jia Li, Jianhua Yao, and Yu Rong. Relaxing continuous constraints of equivariant graph neural networks for broad physical dynamics learning. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 4548–4558, 2024.
- [70] Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. Graph Contrastive Learning with Adaptive Augmentation. In *The Web Conference*, pages 2069–2080, 2021.
- [71] Yun Zhu, Jianhao Guo, and Siliang Tang. SGL-PT: A Strong Graph Learner with Graph Prompt Tuning. *arXiv preprint*, 2023.

A Detailed Description of Methods

End-to-end Graph Models

- **GCN (Graph Convolutional Network [21]):** A method that utilizes convolution operation on the graph to propagate information from a node to its neighboring nodes, enabling the network to learn a representation for each node based on its local neighborhood.
- **GraphSAGE (Graph Sample and Aggregate [13]):** A general inductive learning framework that generates node embeddings by sampling and aggregating features from a node’s local neighborhood.
- **GAT (Graph Attention Networks [54]):** A graph neural network (GNN) framework that incorporates the attention mechanism. It assigns varying levels of importance to different nodes during the neighborhood information aggregation process, allowing the model to focus on the most informative parts.
- **GT (Graph Transformer [43]):** An adaptation of the neural network architecture that applies the principles of the Transformer model to graph-structured data. It uses masks in the self-attention process to leverage the graph structure and enhance model efficiency.

Pre-training

- **DGI[55]:** capitalizes on a self-supervised method for pretraining, which is based on the concept of mutual information (MI). It maximizes the MI between the local augmented instances and the global representation.
- **GraphMAE[14]:** employs the masked autoencoder framework to minimize the difference between the original features of the masked nodes and their reconstructed features. This process enhances the model’s ability to understand and embed graph structures effectively.
- **EdgePreGPPT[46]:** is proposed in GPPT to calculate the dot product as the link probability of the node pair.
- **EdgePreGprompt[33]:** is proposed in Gprompt n, for a set of label-free graphs G , we sample several triplets from each graph to increase the similarity between the contextual subgraphs of linked pair while decreasing unlinked pair.
- **GraphCL[65]:** applies different graph augmentations to exploit the structural information on the graphs and aims to maximize the agreement between different augmentations for graph pre-training.
- **SimGRACE[60]:** tries to perturb the graph model parameter spaces and narrow down the gap between different perturbations for the same graph.

Graph Prompt Methods

- **GPPT (‘Prompt as token’) [46]:** introduces a graph prompting function that transforms a standalone node into a token pair composed of a task token, representing a node label, and a structure token, describing the given node. This reformulates the downstream node classification task to resemble edge prediction. As a result, the pre-trained GNNs can be directly applied without extensive fine-tuning to evaluate the linking probability of the token pair and produce the node classification decision. Since GPPT originally is designed for node classification, in this work, to adapt to graph classification, we follow the core thoughts of GPPT and make the following modifications: (1) Introduce m graph task tokens, one for each graph label. (2) Extract N structure tokens from a graph with N nodes, the same as the GPPT for node tasks. (3) Combine each structure token with all graph tokens to form m token pairs. (4) Calculate an m -dimension link prediction probability vector for the m structure tokens and choose the class with maximum probability as the prediction result. (6) Aggregate the link prediction results using a voting strategy to determine the graph class. (7) Update the learnable tokens based on the prediction loss.
- **Gprompt (‘Prompt as token’) [33]:** employs subgraph similarity as a mechanism to unify different pretext and downstream tasks, including link prediction, node classification, and graph classification. A learnable prompt is subsequently tuned for each downstream task.
- **All-in-one (‘Prompt as graph’) [47]:** involves freezing the parameters of pre-trained graph models and incorporating learnable graph prompts as insert patterns to original induced graphs on the downstream node classification task.
- **GPF (‘Prompt as token’) [8]:** applies a consistent prompt feature to all nodes, regardless of their contextual differences.

- **GPF-plus (‘Prompt as token’) [8]**: enhances the method by assigning prompt features based on node type, ensuring uniformity within each category but without considering the context of individual graphs. Note that GPF and GPF-plus inherently are primarily evaluated in graph classification tasks and are not readily adaptable to node classification. In this work, we address this limitation by inducing a subgraph for each node within a node-level graph. This approach effectively reformulates the node classification task into a graph classification task, allowing for the integration of inserted prompt features.

B Related Work

B.1 Graph pre-training

Pre-training of graphs is a crucial step in graph representation learning for pre-training, prompting, and predicting paradigms, which provides a solid foundation for downstream tasks by utilizing existing information to encode graph structures. It aims to learn some general knowledge for the graph model with easily accessible information to reduce the annotation costs of new tasks. Node-level pre-training obtains local structural representations, which can be used for downstream tasks. Comparative learning learns by maximizing the mutual information (MI) between the original view and the self-supervised view [6, 70, 19]. The prediction model reconstructs perturbed data [13, 14]. However, predictive models are difficult to capture higher-order information. Edge-level pre-training includes distinguishing whether there are edges between node pairs [20, 34], or reconstructing masked edges by restoring including graph reconstruction [39, 61] and contrastive learning targets either local patches of node features and global graph features [55, 45]. Multi-task pre-training addresses multiple optimization objectives, covering various graph-related aspects. This approach enhances generalization and mitigates negative transfer issues. For instance, Hu et al. [16] pre-trained a GNN at both the node and graph levels, allowing the GNN to learn valuable local and global representations simultaneously. Additionally, some studies have utilized contrastive learning at different levels [18, 62], or jointly optimized both contrastive loss and graph reconstruction error [26].

B.2 Graph prompt learning

While Pre-training effectively encodes global information and generates valuable graph-level representations, a significant challenge lies in transferring knowledge from a specific pretext task to downstream tasks with substantial gaps, potentially resulting in negative transfer [40]. Instead of fine-tuning a pre-trained model with an adaptive task head, prompt learning aims to reformulate input data to fit the pretext. Many effective prompt methods were initially developed in the field of NLP, encompassing various approaches such as hand-crafted prompts exemplified by GPT-4 [1], discrete prompts like AutoPrompt [27], and trainable prompts in continuous spaces such as Prefix-Tuning [44]. Due to their remarkable success, prompt-based methods have increasingly been applied to the graph domain [4].

Graph prompt learning usually has two formalities: on the one hand, it can be regarded as tokens [3, 51, 12]. For example, GPF takes a prompt token as some additional features are added to the original graph. Based on that, GPF-plus [8] trains several independent basis vectors and utilizes attentive aggregation of these basis vectors with the assistance of several learnable linear projections. It also solved problems when training graphs have different scales and large-scale input graphs. GPPT [46] defines graph prompts as additional tokens that contain task tokens and structure tokens that align downstream node task and link prediction pretext. Furthermore, some works [71] use graph prompt to assist graph pooling operation (a.k.a Readout), like Gprompt [33] inserts the prompt vector to the graph pooling by element-wise multiplication. On the other hand, graph prompt is regarded as graphs [11, 17]. Such as All-in-one [47] inserts the token graphs as prompt to the original graph and all links between tokens and original graphs, these links can be pre-calculated by the dot product between one token to another token (inner links) or one token to another original node (cross links).

C Detailed Description of Datasets

Homophilic: Cora and Citeseer datasets [42] consist of a diverse collection of computer science publications, where each node is characterized by a bag-of-words representation of papers and a

categorical label indicating the paper topic. Pubmed dataset [35] comprises articles related to diabetes from the PubMed database. Each node in this dataset is represented by an attribute vector containing TF/IDF-weighted word frequencies, accompanied by a label specifying the particular type of diabetes discussed in the publication. ogbn-arxiv [15] is a large-scale directed graph representing the citation network between all Computer Science (CS) arXiv papers indexed by MAG [56]. Each node is an arXiv paper and each directed edge indicates that one paper cites another one. Each paper comes with a 128-dimensional feature vector obtained by averaging the embeddings of words in its title and abstract.

Heterophilic: Cornell, Texas, and Wisconsin are three subdatasets derived from the WebKB dataset [37], compiled from multiple universities’ web pages. Each node within these datasets represents a web page, with edges denoting hyperlinks between pages. The node features are represented as bag-of-words representations of the web pages. Additionally, the web pages are manually categorized into five distinct labels: student, project, course, staff, and faculty. Actor is a co-occurrence network in which each node corresponds to an actor and edges indicate their co-occurrence on the same Wikipedia page [53, 31].

Social networks: IMDB-BINARY [63] is a movie collaboration dataset. The nodes represent actors/actresses, the edges represent collaborations between actors/actresses appearing in the same movie, and the graph labels indicate the genre (Action or Romance). COLLAB [63] is a scientific collaboration dataset. The nodes represent researchers, the edges represent the collaboration relationships between researchers, and the graph labels indicate the field of research (High Energy Physics, Condensed Matter Physics, or Astro Physics).

Biological: PROTEINS [58] is a collection of protein graphs that include the amino acid sequence, conformation, structure, and features such as active sites of the proteins. The nodes represent the secondary structures, and each edge depicts the neighboring relation in the amino-acid sequence or in 3D space. The nodes belong to three categories, and the graphs belong to two classes. ENZYMES [2] is a dataset of 600 enzymes collected from the BRENDA enzyme database. These enzymes are labeled into 6 categories according to their top-level EC enzyme. D&D [7] consists of graph representations of 1,178 proteins. In each graph, nodes represent amino acids, and there is an edge if they are less than six Angstroms apart. Graphs are labeled according to whether they are enzymes or not.

Small Molecule: COX2 [41] is a dataset of molecular structures including 467 cyclooxygenase-2 inhibitors, in which each node is an atom, and each edge represents the chemical bond between atoms, such as single, double, triple or aromatic. All the molecules belong to two categories. BZR [41] is a collection of 405 ligands for benzodiazepine receptors, in which each ligand is represented by a graph. All these ligands belong to 2 categories. MUTAG [22] is a dataset of 188 mutagenic aromatic and heteroaromatic nitro compounds with 7 discrete labels.

D Additional Details

Metrics. In node and graph classification, **AUROC** and **F1-score** are another two important evaluation metrics. AUROC stands for *Area Under the Receiver Operating Characteristic Curve*. It shows how well the model can distinguish between different classes. A score of 1 means perfect classification, while 0.5 indicates the model is guessing randomly. F1-score combines precision and recall into a single number. Precision measures how many of the predicted positives are actually correct, and recall measures how many of the actual positives the model correctly identified. The F1-score ranges from 0 to 1, with 1 being the best. It is particularly useful in cases where class distributions are imbalanced. In this work, we compute the macro average values for each class. For both AUROC and F1-score, in multi-class classification tasks, each class is treated as the positive class once, while the other classes are treated as negative, and the metrics are averaged.

Implementation Details. To ensure a comprehensive evaluation and maintain fairness across a broad spectrum of models, we develop an open-source package named ProG³.

Software and Hardware. For a fair comparison, all methods are implemented by our developed library ProG, which is built upon PyTorch [36], and unless specifically indicated, the encoders for all methods are Graph Convolutional Networks. All experiments are conducted on a Linux server with

³<https://github.com/sheldonresearch/ProG>

GPU (NVIDIA GeForce 3090, NVIDIA GeForce 4090, and NVIDIA A800) and CPU (AMD EPYC 7763), using PyTorch 2.0.1 [36], PyG 2.3.1 [9] and Python 3.9.17.

Hyperparameter Settings. To ensure a comprehensive evaluation process and maintain fairness, we conduct an in-depth analysis of various graph prompt methods. Then, we carefully design a unified random search strategy utilized for selecting a set of optimized hyperparameters for each prompt method on each dataset, going beyond the default settings. Given the significant variations in hyperparameters across different prompt methods, we focus on optimizing three types of general hyperparameters via random search. Concretely, we search for the optimal learning rate within the range $10^{\text{uniform}(-3,-1)}$, and optimal weight decay within $10^{\text{uniform}(-5,-6)}$. Additionally, 32, 64, or 128 batch sizes are selected with equal probability in each trial. Moreover, supervised methods and 'Pre-train & Fine-tune' methods in this work also undergo the same random search process for hyperparameter optimization.

E Additional Experimental Analysis

E.1 Adaptivity of Graph Prompts on Different Graphs.

Various Domains. GPL methods, particularly those of the Prompt as Token type, exhibit *performance inconsistencies across graph-level datasets from various domains*, w.r.t Table 3. Sometimes, these methods *may encounter negative transfer issues* to varying degrees, except for the BZR and DD datasets. GPPT, in particular, faces significant challenges, likely because first-order subgraphs fail to comprehensively capture graph patterns, introducing noise and affecting transferability.

Large-scale Datasets. Referring to Table 2, on the large-scale node-level dataset (i.e., ogbn-arxiv), Gprompt, GPF, and GPF-plus significantly surpass the Supervised and classical 'Pre-train & Fine-tune' methods, demonstrating good scalability. However, the scalability of GPL methods involving subgraph operations, such as GPPT and All-in-one, needs improvement as they fail to fully capture large-scale graph patterns with their subgraph information. Furthermore, regarding the results of the relatively large graph-level dataset (i.e., DD) in Table 3, all GPL methods achieve positive transfer and outperform classical 'Pre-train & Fine-tune' methods. Despite this success, the magnitude of improvement indicates that *the transferability of GPL methods on large-scale datasets requires further enhancement*.

Homophilic & Heterophilic Datasets. Drawing from Table 2, most GPL methods can handle both homophilic and heterophilic graph datasets effectively, with the exception of GPPT. GPPT outperforms the Supervised method on homophilic graph datasets but falls short compared to classical 'Pre-train & Fine-tune' methods. On heterophilic datasets, where node attributes differ significantly, GPPT experiences negative transfer issues, possibly due to instability caused by its clustering algorithms.

Pure Structural Dataset. Observing Table 3, the prompting capability for purely structural graph datasets, such as IMDB-B and COLLAB, still requires improvement. While classical 'Pre-train & Fine-tune' methods achieve positive transfer on these datasets, some variants of the All-in-one method exhibit negative transfer, highlighting *the need for further refinement in handling purely structural graphs*.

E.2 Additional Experimental Results with Different Shots and Metrics

The optimal performance of the graph prompt method, along with other methods, under 3-shot and 5-shot learning settings, is displayed in Tables 7-10. Building on previous experimental outcomes, it is evident that the graph prompt method significantly boosts the effectiveness of various pre-trained graph models in 1/3/5-shot learning settings, thereby enhancing the transferability of pre-trained knowledge in the original 'Pre-train & Fine-tune' method. Specifically, GPF-plus and All-in-one approaches consistently prove to be the most compatible with different pre-trained methods for node and graph tasks, respectively.

Moreover, the experimental results of various methods under 3/5-shot learning settings for three key metrics—Accuracy, F1-score, and AUROC—are comprehensively presented in Tables 11-28.

E.3 Curve of The Shot Number and Accuracy

Figure 5 shows the performance changes of different graph prompt methods with increasing shot numbers on node-level dataset Cora and graph-level dataset ENZYMES. The rest baselines are omitted due to their unsatisfactory performance and unstable training process. Specifically, we observe that the performance of various graph prompt methods generally decreases with the increase in shot numbers and may even be surpassed by Supervised methods. This experimental result aligns with the expectation that graph prompt methods are suited for few-shot learning settings because as the number of samples increases and the sample distribution becomes more comprehensive, the effectiveness of graph prompt methods gradually diminishes.

E.4 Performance with More Kinds of Graph Models.

Table 29 and Table 30 show the performance of various graph prompt methods based on different graph models on the Wisconsin and PROTEINS datasets. Overall, compared to the ‘Pre-train & Fine-tune’ method, graph prompt methods are still generally effective across various pre-trained graph models. Among these, the improvement is relatively notable for GAT. Introducing graph prompt techniques to the pre-trained GAT significantly enhances the transferability of the pre-trained model, especially on node-level datasets.

E.5 Other Heat Maps of Other Graph Prompt Methods

Figure 6 displays the heat maps for various graph prompt methods applied to both node-level and graph-level tasks, offering a more intuitive visualization of the enhancements provided by graph prompt methods over the ‘Pre-train & Fine-tune’ approach. Through an analysis of these heat maps, it becomes evident that, irrespective of the specific graph prompt method or the type of task, node-level pre-training methods consistently outperform in node-level tasks, whereas graph-level pre-training methods exhibit superior performance in graph-level tasks. Moreover, the experimental results provide further evidence of the robust capability of graph prompt methods in effectively addressing the Negative transfer problem.

Table 7: Comparison of different methods across various datasets on 3-shot node classification tasks. The best results for each dataset are highlighted in bold with a dark red background, and the second-best results are underlined with a light red background.

| Methods/Datasets | Cora | Citeseer | Pubmed | Wisconsin | Texas | Actor | Flicker | Ogbn-arxiv |
|-----------------------|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|
| Supervised | 37.79 ±9.16 | 35.18 ±6.86 | 57.33 ±4.64 | 41.03 ±6.40 | 40.78 ±12.55 | 18.62 ±3.46 | 19.03 ±5.08 | 21.01 ±3.44 |
| Pre-train & Fine-tune | 51.97 ±2.84 | 45.08 ±2.09 | 65.40 ±3.00 | 42.40 ±7.77 | 43.13 ±13.79 | 22.11 ±1.97 | 27.34 ±6.61 | 26.08 ±1.89 |
| GPPT | 43.84 ±6.11 | 42.34 ±8.31 | 67.43 ±2.96 | 34.29 ±4.71 | 38.90 ±8.86 | 21.65 ±3.39 | 22.46 ±4.05 | 23.35 ±0.70 |
| All-in-one | 48.09 ±4.83 | 48.09 ±8.18 | 65.79 ±5.79 | 89.62 ±4.38 | 88.69 ±1.08 | 24.23 ±1.39 | <u>29.92 ±1.52</u> | 16.79 ±6.81 |
| Gprompt | 63.78 ±5.77 | <u>60.00 ±6.18</u> | 66.68 ±3.53 | 92.52 ±5.38 | 0.50 ±0.41 | 29.67 ±2.53 | 19.59 ±3.20 | 82.07 ±3.19 |
| GPF | 34.84 ±19.83 | 25.92 ±12.30 | 71.20 ±2.82 | 98.92 ±0.79 | <u>98.49 ±1.45</u> | <u>37.44 ±3.43</u> | <u>29.90 ±3.01</u> | <u>77.50 ±7.23</u> |
| GPF-plus | <u>56.38 ±5.37</u> | 72.48 ±5.63 | <u>70.85 ±4.03</u> | <u>97.44 ±4.52</u> | <u>98.83 ±0.85</u> | <u>43.59 ±4.52</u> | 24.71 ±6.77 | 65.97 ±1.11 |

Table 8: Comparison of different methods across various datasets on 3-shot graph classification tasks.

| Methods/Datasets | IMDB-B | COLLAB | PROTEINS | MUTAG | ENZYMES | COX2 | BZR | DD |
|-----------------------|--------------------|--------------------|--------------------|--------------------|--------------------|---------------------|---------------------|--------------------|
| Supervised | 53.33 ±6.61 | 50.77 ±2.44 | 61.33 ±2.89 | 59.47 ±8.34 | 15.96 ±1.64 | 65.15 ±18.61 | 52.35 ±8.12 | 59.77 ±1.10 |
| Pre-train & Fine-tune | 66.10 ±0.70 | 56.10 ±3.46 | 62.72 ±2.39 | 59.87 ±8.78 | 22.71 ±0.86 | <u>69.97 ±13.89</u> | 52.22 ±10.64 | <u>59.70 ±0.98</u> |
| GPPT | 59.48 ±5.42 | 50.88 ±6.31 | 64.74 ±1.99 | 64.13 ±18.31 | 19.12 ±2.43 | 71.90 ±14.28 | 78.46 ±1.57 | 59.00 ±6.34 |
| All-in-one | 65.67 ±0.58 | <u>57.12 ±1.99</u> | 69.84 ±6.02 | 85.60 ±2.22 | <u>23.96 ±0.62</u> | 66.06 ±18.23 | 61.98 ±11.32 | 58.96 ±5.93 |
| Gprompt | 64.35 ±1.21 | 54.95 ±9.47 | <u>64.94 ±2.92</u> | 66.53 ±14.84 | 22.08 ±3.57 | 51.53 ±13.08 | 54.63 ±2.95 | 55.99 ±7.53 |
| GPF | 65.97 ±0.69 | 53.87 ±3.44 | 63.35 ±2.45 | 74.27 ±1.55 | 23.87 ±3.45 | 65.31 ±19.45 | 74.38 ±11.62 | 59.07 ±0.65 |
| GPF-plus | 64.38 ±2.30 | <u>56.50 ±3.71</u> | 63.55 ±1.85 | <u>75.20 ±3.64</u> | <u>24.46 ±2.27</u> | 65.25 ±18.07 | 71.67 ±14.87 | 59.51 ±0.62 |

Table 9: Comparison of different methods across various datasets on 5-shot node classification tasks.

| Methods/Datasets | Cora | Citeseer | Pubmed | Wisconsin | Texas | Actor | flicker | Ogbn-arxiv |
|-----------------------|-------------------------|-------------------------|-------------------------|-------------------------|--------------------------|-------------------------|-------------------------|-------------------------|
| Supervised | 50.25 \pm 8.37 | 41.22 \pm 6.30 | 67.88 \pm 2.18 | 39.43 \pm 5.86 | 43.91 \pm 6.47 | 21.92 \pm 1.86 | 24.28 \pm 4.49 | 22.38 \pm 3.05 |
| Pre-train & Fine-tune | 62.66 \pm 3.55 | 39.54 \pm 3.54 | 70.91 \pm 4.87 | 42.97 \pm 8.99 | 47.19 \pm 7.37 | 22.92 \pm 1.22 | 35.12 \pm 4.72 | 28.84 \pm 3.11 |
| GPPT | 51.98 \pm 3.43 | 45.77 \pm 7.41 | 66.97 \pm 3.70 | 37.00 \pm 3.19 | 48.82 \pm 5.15 | 21.58 \pm 0.84 | 24.32 \pm 16.36 | 28.90 \pm 1.64 |
| All-in-one | 30.36 \pm 13.48 | 27.93 \pm 10.59 | 46.16 \pm 15.83 | 87.16 \pm 3.02 | 73.28 \pm 9.91 | 21.49 \pm 3.02 | 27.28 \pm 6.43 | 13.01 \pm 6.29 |
| Gprompt | 69.03 \pm 3.61 | 66.13 \pm 1.64 | 67.87 \pm 2.08 | 78.22 \pm 37.33 | 39.32 \pm 47.08 | 34.67 \pm 1.28 | 22.83 \pm 3.43 | 85.40 \pm 0.79 |
| GPF | 35.43 \pm 1.02 | 25.12 \pm 3.01 | 68.96 \pm 3.99 | 98.26 \pm 1.19 | 98.42 \pm 0.36 | 44.07 \pm 3.94 | 25.19 \pm 6.78 | 71.83 \pm 9.37 |
| GPF-plus | 63.28 \pm 4.69 | 75.73 \pm 2.19 | 69.59 \pm 4.33 | 99.01 \pm 1.43 | 99.12 \pm 0.95 | 44.58 \pm 5.95 | 39.62 \pm 8.22 | 66.88 \pm 6.14 |

Table 10: Comparison of different methods across various datasets on 5-shot graph classification tasks.

| Methods/Datasets | IMDB-B | COLLAB | PROTEINS | MUTAG | ENZYMES | COX2 | BZR | DD |
|-----------------------|-------------------------|-------------------------|-------------------------|-------------------------|-------------------------|--------------------------|--------------------------|-------------------------|
| Supervised | 62.60 \pm 4.01 | 55.23 \pm 4.26 | 62.90 \pm 5.03 | 73.47 \pm 3.92 | 25.67 \pm 0.48 | 64.99 \pm 10.42 | 51.48 \pm 2.29 | 63.59 \pm 2.86 |
| Pre-train & Fine-tune | 65.40 \pm 3.33 | 60.72 \pm 2.09 | 63.33 \pm 4.13 | 75.33 \pm 1.89 | 27.46 \pm 1.29 | 73.19 \pm 9.53 | 72.96 \pm 11.98 | 64.71 \pm 3.22 |
| GPPT | 66.37 \pm 3.59 | 54.05 \pm 4.58 | 58.27 \pm 4.63 | 70.53 \pm 3.90 | 22.17 \pm 2.34 | 67.88 \pm 17.34 | 69.63 \pm 14.96 | 60.02 \pm 3.24 |
| All-in-one | 63.62 \pm 2.30 | 57.86 \pm 5.88 | 71.37 \pm 4.89 | 84.67 \pm 0.73 | 26.71 \pm 2.17 | 62.95 \pm 8.57 | 62.78 \pm 10.18 | 63.44 \pm 1.35 |
| Gprompt | 66.70 \pm 3.87 | 60.76 \pm 5.08 | 62.94 \pm 1.38 | 73.07 \pm 2.13 | 21.46 \pm 2.27 | 53.35 \pm 7.75 | 59.38 \pm 14.43 | 58.28 \pm 2.18 |
| GPF | 67.80 \pm 5.58 | 59.65 \pm 6.25 | 63.37 \pm 4.37 | 74.00 \pm 3.65 | 27.00 \pm 0.78 | 66.27 \pm 14.57 | 61.05 \pm 11.51 | 61.06 \pm 2.63 |
| GPF-plus | 68.13 \pm 3.31 | 60.68 \pm 4.67 | 63.51 \pm 2.89 | 73.87 \pm 3.51 | 26.87 \pm 1.89 | 72.87 \pm 10.17 | 71.54 \pm 14.81 | 64.80 \pm 3.45 |

Table 11: Node classification performance Accuracy (%) (1-shot)

| Training schemes | Methods | Cora | CiteSeer | Pubmed | Wisconsin | Texas | Actor | Ogbn-arxiv |
|------------------------|----------------|-------------------|-------------------|-------------------|-------------------|-------------------|------------------|-------------------|
| self-supervise | GCN | 26.56 \pm 5.55 | 21.78 \pm 7.32 | 39.37 \pm 16.34 | 41.60 \pm 3.10 | 37.97 \pm 5.80 | 20.57 \pm 4.47 | 10.99 \pm 3.19 |
| Pretrain + Fine-tuning | DGI | 33.15 \pm 7.84 | 21.64 \pm 3.92 | 42.01 \pm 12.54 | 37.49 \pm 5.13 | 45.31 \pm 5.01 | 19.76 \pm 3.53 | 7.21 \pm 2.91 |
| | GraphMAE | 32.93 \pm 3.17 | 21.26 \pm 3.57 | 42.99 \pm 14.25 | 36.80 \pm 7.17 | 37.81 \pm 8.62 | 19.86 \pm 2.70 | 12.35 \pm 3.60 |
| | EdgePreGPPT | 38.12 \pm 5.29 | 18.09 \pm 5.39 | 46.74 \pm 14.09 | 35.31 \pm 9.31 | 47.66 \pm 2.37 | 19.17 \pm 2.53 | 16.21 \pm 3.82 |
| | EdgePreGprompt | 35.57 \pm 5.83 | 22.28 \pm 3.80 | 41.50 \pm 7.54 | 40.69 \pm 4.13 | 40.62 \pm 7.95 | 20.74 \pm 4.12 | 14.83 \pm 2.38 |
| | GCL | 52.61 \pm 1.73 | 27.02 \pm 4.31 | 42.49 \pm 11.29 | 33.94 \pm 7.74 | 40.31 \pm 13.68 | 20.19 \pm 1.98 | 4.65 \pm 1.19 |
| | SimGRACE | 40.40 \pm 4.66 | 35.05 \pm 4.37 | 37.59 \pm 8.17 | 37.37 \pm 3.68 | 46.88 \pm 4.64 | 19.78 \pm 1.89 | 8.13 \pm 3.26 |
| GPPTPrompt | DGI | 30.47 \pm 3.53 | 37.26 \pm 6.17 | 35.62 \pm 7.74 | 29.94 \pm 10.40 | 29.29 \pm 14.57 | 21.76 \pm 2.00 | 3.80 \pm 6.19 |
| | GraphMAE | 27.39 \pm 10.26 | 21.54 \pm 4.00 | 48.31 \pm 17.72 | 29.83 \pm 9.34 | 25.04 \pm 10.38 | 22.58 \pm 1.97 | 4.35 \pm 6.01 |
| | EdgePreGPPT | 30.37 \pm 4.30 | 21.06 \pm 4.37 | 39.64 \pm 7.64 | 23.89 \pm 5.40 | 30.39 \pm 8.96 | 19.85 \pm 0.76 | 14.65 \pm 3.07 |
| | EdgePreGprompt | 25.52 \pm 4.42 | 21.85 \pm 4.30 | 46.20 \pm 10.76 | 30.40 \pm 6.81 | 22.68 \pm 12.82 | 21.52 \pm 1.13 | 2.05 \pm 1.43 |
| | GCL | 43.15 \pm 9.44 | 26.73 \pm 4.12 | 38.34 \pm 11.59 | 25.03 \pm 5.37 | 31.81 \pm 15.33 | 22.51 \pm 1.73 | 7.15 \pm 4.12 |
| | SimGRACE | 27.86 \pm 2.79 | 25.06 \pm 4.90 | 36.70 \pm 9.26 | 29.83 \pm 6.44 | 25.67 \pm 8.01 | 20.97 \pm 2.30 | 5.50 \pm 5.10 |
| ALL-in-one | DGI | 47.52 \pm 2.50 | 39.37 \pm 3.12 | 38.74 \pm 2.15 | 69.54 \pm 8.98 | 57.38 \pm 12.82 | 21.03 \pm 1.96 | 1.93 \pm 1.48 |
| | GraphMAE | 23.09 \pm 4.92 | 18.08 \pm 5.23 | 33.19 \pm 11.98 | 67.44 \pm 10.22 | 52.82 \pm 11.47 | 23.31 \pm 2.01 | 4.19 \pm 6.04 |
| | EdgePreGPPT | 49.63 \pm 5.26 | 35.06 \pm 2.37 | 40.73 \pm 11.32 | 78.24 \pm 16.68 | 58.62 \pm 5.54 | 21.49 \pm 1.27 | 5.62 \pm 3.95 |
| | EdgePreGprompt | 37.39 \pm 3.31 | 28.85 \pm 4.32 | 35.53 \pm 9.07 | 71.40 \pm 12.32 | 39.71 \pm 25.31 | 20.49 \pm 3.90 | 13.01 \pm 6.29 |
| | GCL | 52.39 \pm 10.17 | 37.37 \pm 4.15 | 45.17 \pm 6.93 | 42.19 \pm 1.05 | 65.49 \pm 7.06 | 24.61 \pm 2.80 | 6.70 \pm 6.01 |
| | SimGRACE | 35.99 \pm 2.76 | 40.41 \pm 2.80 | 30.23 \pm 7.03 | 67.44 \pm 8.42 | 59.22 \pm 20.17 | 21.03 \pm 2.23 | 5.72 \pm 1.61 |
| Gprompt | DGI | 36.46 \pm 5.39 | 36.25 \pm 10.26 | 33.65 \pm 5.29 | 74.00 \pm 4.40 | 31.00 \pm 37.32 | 23.85 \pm 3.52 | 48.47 \pm 5.87 |
| | GraphMAE | 50.58 \pm 7.34 | 42.84 \pm 10.59 | 39.74 \pm 15.35 | 78.51 \pm 9.67 | 23.31 \pm 29.49 | 22.34 \pm 3.57 | 66.66 \pm 5.04 |
| | EdgePreGPPT | 46.96 \pm 6.22 | 40.15 \pm 7.04 | 35.46 \pm 14.12 | 83.80 \pm 2.44 | 30.52 \pm 36.73 | 23.50 \pm 4.16 | 75.72 \pm 4.95 |
| | EdgePreGprompt | 48.11 \pm 9.89 | 48.07 \pm 5.62 | 33.54 \pm 16.66 | 81.93 \pm 2.61 | 33.25 \pm 40.11 | 19.89 \pm 1.38 | 70.55 \pm 7.66 |
| | GCL | 56.66 \pm 11.22 | 45.81 \pm 7.04 | 39.37 \pm 14.95 | 82.16 \pm 6.25 | 29.55 \pm 35.56 | 25.26 \pm 1.10 | 51.20 \pm 6.40 |
| | SimGRACE | 46.34 \pm 6.75 | 53.21 \pm 10.94 | 35.58 \pm 9.03 | 68.95 \pm 10.71 | 30.20 \pm 36.49 | 24.49 \pm 4.38 | 52.70 \pm 5.30 |
| GPF | DGI | 27.83 \pm 18.89 | 16.50 \pm 4.57 | 38.33 \pm 8.13 | 74.13 \pm 1.88 | 78.45 \pm 2.57 | 28.17 \pm 4.81 | 17.85 \pm 8.23 |
| | GraphMAE | 38.57 \pm 5.41 | 25.61 \pm 3.27 | 48.52 \pm 13.23 | 88.33 \pm 5.48 | 78.91 \pm 18.31 | 38.37 \pm 5.82 | 48.00 \pm 3.95 |
| | EdgePreGPPT | 15.29 \pm 8.41 | 12.33 \pm 5.33 | 43.78 \pm 6.02 | 88.67 \pm 5.78 | 86.15 \pm 5.31 | 25.66 \pm 3.33 | 78.37 \pm 5.47 |
| | EdgePreGprompt | 26.60 \pm 13.92 | 31.16 \pm 8.05 | 48.98 \pm 11.57 | 82.04 \pm 4.20 | 86.26 \pm 2.52 | 25.27 \pm 5.65 | 54.27 \pm 11.49 |
| | GCL | 23.16 \pm 5.11 | 16.77 \pm 1.39 | 49.99 \pm 8.86 | 59.50 \pm 13.30 | 87.40 \pm 3.40 | 20.68 \pm 6.70 | 23.55 \pm 0.45 |
| | SimGRACE | 32.01 \pm 11.21 | 19.43 \pm 2.10 | 37.27 \pm 6.09 | 79.48 \pm 2.62 | 87.39 \pm 3.05 | 28.70 \pm 3.35 | 37.56 \pm 6.10 |
| GPF-plus | DGI | 17.29 \pm 6.18 | 26.60 \pm 13.24 | 34.02 \pm 11.94 | 76.34 \pm 6.66 | 79.54 \pm 1.89 | 22.42 \pm 9.66 | 16.83 \pm 10.02 |
| | GraphMAE | 54.26 \pm 7.48 | 59.67 \pm 11.87 | 46.64 \pm 18.57 | 89.17 \pm 9.79 | 94.09 \pm 4.40 | 26.58 \pm 7.84 | 49.81 \pm 2.62 |
| | EdgePreGPPT | 28.49 \pm 18.73 | 28.04 \pm 14.31 | 46.51 \pm 15.84 | 91.03 \pm 4.11 | 92.14 \pm 5.46 | 29.32 \pm 8.56 | 71.98 \pm 12.23 |
| | EdgePreGprompt | 55.77 \pm 10.30 | 49.43 \pm 8.21 | 42.79 \pm 18.18 | 85.44 \pm 2.23 | 87.70 \pm 3.77 | 22.68 \pm 3.64 | 57.44 \pm 6.95 |
| | GCL | 34.18 \pm 17.71 | 28.86 \pm 22.88 | 37.02 \pm 11.29 | 68.57 \pm 8.77 | 95.83 \pm 4.19 | 22.82 \pm 4.99 | 32.11 \pm 4.86 |
| | SimGRACE | 21.33 \pm 14.86 | 24.61 \pm 21.21 | 35.90 \pm 9.06 | 73.64 \pm 5.25 | 90.74 \pm 3.12 | 20.51 \pm 4.24 | 46.71 \pm 3.17 |

Table 14: Node classification performance Accuracy (%) (3-shot)

| Training schemes | Methods | Cora | CiteSeer | Pubmed | Wisconsin | Texas | Actor | Ogbn-arxiv |
|------------------------|-----------------|--------------|--------------|--------------|--------------|--------------|-------------|--------------|
| self-supervise | GCN | 37.79 ±9.16 | 35.18 ±6.86 | 57.33 ±4.64 | 41.03 ±6.40 | 40.78 ±12.55 | 18.62 ±3.46 | 19.03 ±5.08 |
| Pretrain + Fine-tuning | DGI | 45.84 ±3.29 | 34.91 ±10.07 | 63.00 ±6.83 | 39.43 ±6.93 | 43.13 ±13.79 | 20.27 ±1.70 | 21.85 ±5.37 |
| | GraphMAE | 45.15 ±4.77 | 28.59 ±6.41 | 65.40 ±3.00 | 41.49 ±5.19 | 40.94 ±13.91 | 19.03 ±3.17 | 19.63 ±7.64 |
| | EdgePreGPPT | 51.97 ±2.84 | 33.19 ±6.79 | 64.38 ±3.59 | 42.40 ±7.77 | 34.69 ±12.07 | 20.83 ±2.24 | 20.51 ±4.02 |
| | EdgePredGprompt | 40.33 ±6.62 | 33.39 ±5.51 | 63.49 ±3.17 | 41.37 ±7.28 | 34.53 ±10.09 | 19.29 ±1.95 | 27.34 ±6.61 |
| | GCL | 49.39 ±9.15 | 38.40 ±3.06 | 62.79 ±3.21 | 41.26 ±6.26 | 40.31 ±13.36 | 21.06 ±1.27 | 17.37 ±7.34 |
| | SimGRACE | 43.61 ±5.41 | 45.08 ±2.09 | 62.66 ±3.21 | 40.11 ±8.04 | 40.94 ±13.73 | 22.11 ±1.97 | 16.06 ±2.05 |
| GPPTPrompt | DGI | 37.46 ±6.27 | 42.34 ±8.31 | 44.36 ±3.67 | 34.29 ±4.71 | 37.80 ±8.98 | 21.42 ±3.13 | 14.34 ±14.10 |
| | GraphMAE | 30.93 ±3.64 | 20.76 ±2.64 | 67.43 ±2.96 | 33.60 ±2.30 | 37.64 ±5.88 | 21.65 ±3.39 | 17.24 ±13.25 |
| | EdgePreGPPT | 35.05 ±2.95 | 24.26 ±2.55 | 58.66 ±5.93 | 29.37 ±6.22 | 38.90 ±7.56 | 20.35 ±0.43 | 22.46 ±4.05 |
| | EdgePreGprompt | 27.94 ±5.07 | 23.21 ±2.95 | 64.98 ±5.35 | 33.49 ±5.49 | 36.06 ±12.01 | 19.85 ±0.31 | 18.73 ±4.77 |
| | GCL | 43.84 ±6.11 | 27.09 ±4.57 | 51.30 ±6.35 | 23.89 ±5.35 | 38.90 ±8.86 | 20.60 ±1.10 | 13.04 ±6.44 |
| | SimGRACE | 29.66 ±5.49 | 29.63 ±1.87 | 43.95 ±3.55 | 31.66 ±6.21 | 34.65 ±6.97 | 21.08 ±1.16 | 14.13 ±13.89 |
| ALL-in-one | DGI | 33.80 ±6.36 | 31.08 ±2.95 | 48.09 ±2.81 | 74.71 ±2.35 | 31.94 ±7.00 | 24.23 ±1.39 | 29.92 ±1.52 |
| | GraphMAE | 19.25 ±3.11 | 19.02 ±5.19 | 64.03 ±4.70 | 76.90 ±2.98 | 65.93 ±17.42 | 19.56 ±2.48 | 13.15 ±4.63 |
| | EdgePreGPPT | 48.09 ±4.83 | 34.29 ±1.85 | 58.36 ±4.63 | 89.62 ±4.38 | 88.69 ±1.08 | 20.91 ±3.37 | 14.41 ±5.57 |
| | EdgePreGprompt | 29.54 ±6.25 | 24.63 ±3.57 | 65.79 ±5.70 | 84.29 ±2.32 | 73.89 ±16.92 | 20.70 ±2.58 | 12.91 ±2.08 |
| | GCL | 27.89 ±18.34 | 43.96 ±6.69 | 51.44 ±4.46 | 67.06 ±9.70 | 78.86 ±4.06 | 22.14 ±2.57 | 31.15 ±2.25 |
| | SimGRACE | 25.16 ±11.78 | 48.09 ±8.18 | 47.87 ±3.53 | 76.78 ±1.28 | 82.93 ±5.04 | 21.99 ±1.17 | 15.47 ±4.05 |
| Gprompt | DGI | 37.42 ±7.07 | 41.19 ±5.30 | 38.69 ±1.54 | 71.54 ±10.15 | 0.50 ±0.41 | 26.64 ±4.45 | 19.40 ±5.30 |
| | GraphMAE | 63.78 ±5.77 | 57.15 ±3.90 | 62.47 ±4.29 | 86.48 ±7.48 | 0.50 ±0.41 | 29.21 ±3.27 | 18.90 ±3.10 |
| | EdgePreGPPT | 51.29 ±7.07 | 34.79 ±4.07 | 53.63 ±7.62 | 89.10 ±4.09 | 0.50 ±0.41 | 26.44 ±3.36 | 18.14 ±3.54 |
| | EdgePreGprompt | 53.37 ±6.30 | 56.19 ±2.95 | 66.68 ±3.53 | 92.52 ±5.38 | 0.50 ±0.41 | 21.78 ±1.85 | 19.59 ±3.20 |
| | GCL | 56.61 ±8.02 | 56.28 ±5.89 | 50.33 ±5.18 | 85.83 ±4.53 | 0.50 ±0.41 | 29.67 ±2.53 | 18.94 ±2.93 |
| | SimGRACE | 45.22 ±5.69 | 60.00 ±6.18 | 43.58 ±5.78 | 82.84 ±1.52 | 0.50 ±0.41 | 25.07 ±1.10 | 15.74 ±4.18 |
| GPF | DGI | 25.14 ±19.07 | 25.92 ±12.30 | 38.49 ±14.50 | 76.22 ±5.46 | 44.94 ±4.84 | 25.03 ±1.66 | 29.90 ±3.01 |
| | GraphMAE | 33.93 ±8.01 | 23.91 ±5.11 | 71.20 ±2.82 | 98.92 ±0.79 | 94.67 ±3.28 | 37.44 ±3.43 | 22.22 ±5.44 |
| | EdgePreGPPT | 23.05 ±9.03 | 18.81 ±4.35 | 65.06 ±2.22 | 96.63 ±2.53 | 94.85 ±1.67 | 30.47 ±3.57 | 19.76 ±4.75 |
| | EdgePreGprompt | 32.57 ±9.57 | 21.75 ±2.88 | 67.57 ±14.07 | 94.58 ±5.67 | 98.49 ±1.45 | 25.64 ±8.58 | 15.72 ±5.16 |
| | GCL | 34.84 ±19.83 | 23.02 ±5.56 | 49.15 ±6.46 | 56.93 ±12.66 | 94.66 ±3.87 | 21.95 ±6.54 | 8.90 ±2.80 |
| | SimGRACE | 27.80 ±18.09 | 21.20 ±11.25 | 31.75 ±13.86 | 86.10 ±0.76 | 97.83 ±1.13 | 22.82 ±3.16 | 10.71 ±1.11 |
| GPF-plus | DGI | 20.80 ±13.69 | 24.46 ±17.23 | 51.70 ±11.29 | 85.61 ±8.19 | 72.61 ±22.35 | 23.99 ±6.98 | 16.16 ±6.23 |
| | GraphMAE | 56.38 ±5.37 | 72.48 ±5.63 | 70.85 ±4.03 | 97.44 ±4.52 | 97.00 ±0.68 | 43.59 ±4.52 | 22.50 ±14.10 |
| | EdgePreGPPT | 32.62 ±23.21 | 22.06 ±21.01 | 67.20 ±1.67 | 97.34 ±2.33 | 98.83 ±0.85 | 34.38 ±5.23 | 23.56 ±5.67 |
| | EdgePreGprompt | 52.74 ±7.08 | 71.56 ±8.03 | 68.84 ±4.33 | 96.96 ±4.60 | 97.01 ±1.62 | 30.87 ±6.61 | 24.71 ±6.77 |
| | GCL | 30.34 ±21.41 | 27.17 ±24.65 | 61.23 ±11.39 | 62.76 ±14.68 | 97.18 ±1.93 | 36.87 ±2.45 | 5.42 ±0.82 |
| | SimGRACE | 30.95 ±19.32 | 28.97 ±20.66 | 41.96 ±15.64 | 80.64 ±3.99 | 97.34 ±1.84 | 34.89 ±6.03 | 13.16 ±5.89 |

Table 15: Node classification performance F1-score (3-shot)

| Training schemes | Methods | Cora | CiteSeer | Pubmed | Wisconsin | Texas | Actor | Ogbn-arxiv |
|------------------------|-----------------|--------------|--------------|--------------|--------------|--------------|--------------|-------------|
| self-supervise | GCN | 37.64 ±6.32 | 28.68 ±8.09 | 53.04 ±6.01 | 34.90 ±5.93 | 29.16 ±7.77 | 13.69 ±2.87 | 15.21 ±2.66 |
| Pretrain + Fine-tuning | DGI | 41.76 ±1.48 | 27.23 ±9.82 | 61.18 ±10.18 | 35.01 ±5.55 | 31.12 ±8.67 | 16.05 ±0.75 | 16.45 ±1.97 |
| | GraphMAE | 45.70 ±6.35 | 20.60 ±9.72 | 64.33 ±3.42 | 34.08 ±4.01 | 28.29 ±8.05 | 16.30 ±2.37 | 14.26 ±4.46 |
| | EdgePreGPPT | 49.76 ±2.52 | 28.14 ±6.04 | 63.62 ±4.81 | 35.03 ±5.65 | 25.55 ±6.58 | 17.79 ±1.62 | 16.68 ±2.65 |
| | EdgePredGprompt | 41.69 ±5.58 | 23.81 ±4.86 | 62.48 ±4.02 | 35.42 ±6.02 | 26.05 ±6.95 | 12.98 ±2.02 | 19.33 ±2.46 |
| | GCL | 47.79 ±9.36 | 34.47 ±3.44 | 63.10 ±2.93 | 22.42 ±6.61 | 29.45 ±8.86 | 19.91 ±1.08 | 7.46 ±3.10 |
| | SimGRACE | 41.71 ±4.09 | 42.01 ±1.56 | 63.14 ±3.43 | 34.28 ±5.18 | 29.55 ±7.96 | 17.92 ±1.61 | 11.68 ±1.37 |
| GPPTPrompt | DGI | 33.94 ±7.56 | 36.13 ±8.65 | 44.13 ±3.80 | 30.91 ±2.41 | 34.81 ±7.09 | 19.30 ±2.98 | 3.50 ±2.51 |
| | GraphMAE | 25.31 ±6.18 | 6.91 ±2.93 | 67.61 ±2.62 | 29.51 ±3.03 | 33.87 ±3.28 | 18.03 ±3.21 | 6.05 ±2.21 |
| | EdgePreGPPT | 32.77 ±2.69 | 19.54 ±1.23 | 58.07 ±6.41 | 27.20 ±3.72 | 33.65 ±5.16 | 19.32 ±0.53 | 17.24 ±1.62 |
| | EdgePreGprompt | 20.10 ±4.36 | 20.60 ±1.98 | 62.73 ±4.96 | 29.19 ±4.29 | 30.98 ±7.36 | 18.99 ±0.70 | 15.47 ±2.95 |
| | GCL | 41.25 ±4.50 | 25.71 ±4.25 | 50.27 ±6.65 | 15.65 ±5.40 | 34.61 ±6.65 | 19.81 ±1.02 | 5.96 ±3.82 |
| | SimGRACE | 27.19 ±4.01 | 27.09 ±2.43 | 42.79 ±3.17 | 26.79 ±4.55 | 32.38 ±5.73 | 19.02 ±0.50 | 4.01 ±2.59 |
| ALL-in-one | DGI | 22.05 ±4.15 | 25.31 ±3.49 | 42.82 ±4.66 | 63.22 ±7.20 | 37.71 ±11.00 | 11.04 ±2.36 | 15.79 ±0.87 |
| | GraphMAE | 8.10 ±2.43 | 7.36 ±2.81 | 62.24 ±5.80 | 51.62 ±2.63 | 58.91 ±14.27 | 11.52 ±0.97 | 9.72 ±3.20 |
| | EdgePreGPPT | 41.54 ±1.81 | 27.24 ±3.58 | 56.20 ±5.02 | 80.48 ±6.96 | 82.89 ±2.60 | 9.51 ±3.89 | 9.61 ±4.38 |
| | EdgePreGprompt | 20.33 ±9.09 | 12.09 ±3.36 | 63.18 ±7.52 | 79.59 ±6.58 | 68.71 ±14.50 | 7.89 ±2.38 | 9.96 ±1.24 |
| | GCL | 14.85 ±18.09 | 33.38 ±6.65 | 50.52 ±3.48 | 61.79 ±11.38 | 69.09 ±6.96 | 16.40 ±1.47 | 14.33 ±0.41 |
| | SimGRACE | 12.71 ±11.42 | 40.22 ±8.61 | 44.79 ±2.93 | 64.88 ±4.35 | 79.46 ±5.10 | 13.70 ±2.29 | 10.07 ±1.80 |
| Gprompt | DGI | 32.92 ±6.82 | 37.71 ±5.52 | 37.90 ±1.53 | 60.12 ±10.43 | 0.20 ±0.16 | 26.39 ±4.24 | 14.59 ±2.78 |
| | GraphMAE | 59.30 ±7.36 | 53.60 ±3.06 | 61.33 ±4.81 | 79.06 ±11.48 | 0.20 ±0.16 | 29.02 ±3.53 | 14.78 ±1.50 |
| | EdgePreGPPT | 46.21 ±6.52 | 29.81 ±3.69 | 48.94 ±10.06 | 80.60 ±7.17 | 0.20 ±0.16 | 24.73 ±3.32 | 15.68 ±2.29 |
| | EdgePreGprompt | 52.43 ±6.41 | 53.88 ±3.50 | 65.02 ±2.80 | 91.03 ±5.30 | 0.20 ±0.16 | 20.40 ±1.65 | 15.56 ±1.69 |
| | GCL | 51.54 ±8.49 | 52.46 ±5.08 | 49.16 ±5.73 | 65.08 ±11.12 | 0.20 ±0.16 | 28.21 ±2.34 | 14.80 ±1.34 |
| | SimGRACE | 40.83 ±3.88 | 56.25 ±7.01 | 42.59 ±5.51 | 76.68 ±3.65 | 0.20 ±0.16 | 25.41 ±1.35 | 12.58 ±1.91 |
| GPF | DGI | 12.93 ±16.42 | 12.92 ±13.22 | 32.76 ±17.33 | 67.09 ±7.72 | 46.87 ±5.91 | 15.68 ±9.94 | 14.93 ±1.60 |
| | GraphMAE | 19.59 ±5.22 | 14.48 ±5.04 | 70.15 ±2.68 | 94.99 ±4.12 | 83.16 ±10.27 | 31.56 ±8.62 | 5.27 ±5.58 |
| | EdgePreGPPT | 9.90 ±9.04 | 7.40 ±3.59 | 63.04 ±3.09 | 92.67 ±6.23 | 80.32 ±6.04 | 22.38 ±8.76 | 14.77 ±2.22 |
| | EdgePreGprompt | 21.59 ±5.95 | 11.79 ±3.50 | 62.02 ±21.60 | 93.30 ±5.43 | 86.36 ±10.64 | 22.04 ±11.41 | 11.51 ±3.42 |
| | GCL | 18.90 ±20.40 | 8.10 ±4.95 | 47.10 ±8.36 | 30.08 ±6.91 | 81.66 ±11.83 | 14.24 ±8.36 | 4.28 ±1.13 |
| | SimGRACE | 13.21 ±14.54 | 10.26 ±8.85 | 25.30 ±17.26 | 77.51 ±2.92 | 85.34 ±10.18 | 15.37 ±7.26 | 5.41 ±1.24 |
| GPF-plus | DGI | 10.87 ±13.23 | 14.35 ±19.57 | 43.25 ±20.47 | 78.25 ±14.06 | 63.25 ±15.75 | 14.70 ±3.64 | 10.41 ±3.06 |
| | GraphMAE | 55.76 ±3.81 | 70.45 ±6.54 | 69.78 ±3.58 | 97.33 ±2.96 | 83.31 ±9.41 | 45.02 ±4.47 | 12.22 ±4.77 |
| | EdgePreGPPT | 20.90 ±24.90 | 15.30 ±23.70 | 65.43 ±1.28 | 94.18 ±5.80 | 86.54 ±10.00 | 32.74 ±6.14 | 14.90 ±2.10 |
| | EdgePreGprompt | 52.13 ±7.14 | 68.79 ±9.94 | 67.98 ±3.69 | 94.78 ±4.60 | 85.14 ±8.73 | 30.59 ±7.43 | 15.43 ±2.65 |
| | GCL | 21.84 ±23.15 | 18.50 ±28.28 | 59.42 ±13.43 | 47.29 ±15.14 | 84.24 ±8.20 | 30.93 ±5.62 | 4.37 ±1.56 |
| | SimGRACE | 20.11 ±22.26 | 18.43 ±24.95 | 27.58 ±21.15 | 61.92 ±12.87 | 85.24 ±9.54 | 30.11 ±6.73 | 6.91 ±2.30 |

Table 16: Node classification performance AUROC (3-shot)

| Training schemes | Methods | Cora | CiteSeer | Pubmed | Wisconsin | Texas | Actor | Ogbn-arxiv |
|------------------------|-----------------|--------------|--------------|--------------|--------------|--------------|-------------|-------------|
| self-supervise | GCN | 86.20 ±2.47 | 76.31 ±3.45 | 80.24 ±2.82 | 66.44 ±6.12 | 59.85 ±12.05 | 51.48 ±0.74 | 59.17 ±1.26 |
| Pretrain + Fine-tuning | DGI | 84.23 ±2.75 | 74.32 ±4.21 | 82.28 ±5.43 | 67.49 ±3.80 | 55.28 ±10.97 | 51.93 ±1.20 | 61.28 ±1.40 |
| | GraphMAE | 86.80 ±2.35 | 75.07 ±2.48 | 84.70 ±1.28 | 66.58 ±6.51 | 58.48 ±11.37 | 51.78 ±1.01 | 59.12 ±1.66 |
| | EdgePreGPPT | 85.62 ±1.41 | 72.39 ±3.60 | 83.03 ±2.89 | 66.70 ±4.04 | 54.49 ±12.28 | 52.25 ±0.89 | 60.55 ±1.73 |
| | EdgePredGprompt | 85.85 ±0.79 | 76.73 ±2.26 | 83.40 ±2.74 | 67.27 ±4.41 | 58.43 ±10.76 | 51.24 ±0.96 | 61.30 ±1.07 |
| | GCL | 87.71 ±1.59 | 75.10 ±2.66 | 81.83 ±2.15 | 61.56 ±7.86 | 57.10 ±12.53 | 52.03 ±1.09 | 54.66 ±2.69 |
| | SimGRACE | 80.39 ±2.84 | 76.94 ±1.30 | 79.56 ±3.24 | 67.34 ±5.82 | 59.02 ±11.26 | 51.60 ±0.77 | 56.28 ±1.14 |
| GPPTPrompt | DGI | 70.88 ±2.54 | 73.97 ±5.46 | 61.07 ±2.93 | 66.79 ±2.66 | 63.59 ±7.91 | 51.34 ±1.54 | 50.34 ±0.68 |
| | GraphMAE | 80.91 ±2.38 | 69.11 ±3.46 | 83.10 ±1.03 | 63.77 ±3.61 | 60.95 ±3.13 | 51.22 ±1.38 | 51.84 ±3.86 |
| | EdgePreGPPT | 68.71 ±2.02 | 59.01 ±2.73 | 74.22 ±4.90 | 57.60 ±1.78 | 58.20 ±4.72 | 50.23 ±0.31 | 58.95 ±1.29 |
| | EdgePreGprompt | 68.29 ±3.52 | 58.07 ±2.40 | 81.15 ±4.81 | 61.93 ±5.66 | 61.91 ±5.28 | 50.27 ±0.63 | 56.16 ±2.47 |
| | GCL | 75.26 ±3.86 | 60.58 ±4.04 | 71.43 ±2.47 | 53.79 ±2.21 | 61.73 ±5.50 | 50.96 ±0.66 | 50.69 ±1.68 |
| | SimGRACE | 66.90 ±4.18 | 65.72 ±2.09 | 62.58 ±2.08 | 66.87 ±2.38 | 60.35 ±7.62 | 50.53 ±0.57 | 50.07 ±0.24 |
| ALL-in-one | DGI | 71.15 ±3.33 | 65.99 ±0.90 | 66.83 ±1.02 | 81.86 ±0.82 | 46.05 ±2.66 | 52.80 ±2.23 | 55.31 ±0.33 |
| | GraphMAE | 77.89 ±3.94 | 63.31 ±2.05 | 84.13 ±1.99 | 81.57 ±1.86 | 66.49 ±5.26 | 50.76 ±1.17 | 55.43 ±0.07 |
| | EdgePreGPPT | 80.90 ±2.72 | 68.74 ±3.28 | 76.23 ±0.81 | 96.17 ±3.82 | 75.39 ±1.30 | 52.98 ±1.21 | 56.37 ±0.72 |
| | EdgePreGprompt | 77.50 ±2.96 | 69.84 ±1.80 | 85.03 ±1.73 | 91.35 ±2.44 | 72.04 ±6.05 | 49.89 ±1.95 | 54.08 ±0.15 |
| | GCL | 65.18 ±11.80 | 77.64 ±2.06 | 73.05 ±1.97 | 82.74 ±3.64 | 72.60 ±1.20 | 51.67 ±1.31 | 51.63 ±0.27 |
| | SimGRACE | 59.44 ±9.08 | 78.11 ±1.17 | 64.76 ±2.62 | 84.99 ±2.40 | 92.99 ±1.95 | 51.64 ±2.25 | 51.35 ±0.09 |
| Gprompt | DGI | 62.24 ±2.99 | 75.87 ±3.70 | 55.46 ±1.89 | 87.72 ±5.97 | 49.44 ±8.34 | 55.78 ±3.19 | 52.89 ±2.11 |
| | GraphMAE | 80.27 ±5.61 | 74.38 ±3.36 | 84.37 ±1.75 | 96.34 ±2.81 | 49.44 ±8.34 | 58.52 ±2.66 | 53.56 ±1.76 |
| | EdgePreGPPT | 81.41 ±5.40 | 65.99 ±4.06 | 72.84 ±0.48 | 98.59 ±1.16 | 49.44 ±8.34 | 56.86 ±2.69 | 56.05 ±3.09 |
| | EdgePreGprompt | 83.15 ±3.91 | 82.01 ±1.78 | 84.45 ±1.80 | 99.32 ±0.66 | 49.44 ±8.34 | 51.33 ±2.32 | 54.94 ±1.15 |
| | GCL | 83.25 ±3.22 | 83.20 ±0.97 | 71.35 ±3.39 | 91.20 ±4.59 | 49.44 ±8.34 | 59.78 ±3.41 | 53.58 ±1.71 |
| | SimGRACE | 79.02 ±2.55 | 86.57 ±1.86 | 61.89 ±6.64 | 94.09 ±3.00 | 49.44 ±8.34 | 56.13 ±1.59 | 52.64 ±1.18 |
| GPF | DGI | 60.20 ±13.06 | 59.79 ±13.56 | 59.52 ±12.99 | 84.92 ±3.03 | 71.79 ±11.35 | 54.39 ±2.39 | 55.76 ±1.34 |
| | GraphMAE | 82.70 ±2.62 | 72.25 ±4.60 | 87.91 ±1.01 | 99.59 ±0.81 | 91.54 ±9.54 | 67.24 ±4.73 | 55.48 ±0.92 |
| | EdgePreGPPT | 65.44 ±9.07 | 67.13 ±1.95 | 81.07 ±0.78 | 99.62 ±0.64 | 84.29 ±6.85 | 58.43 ±3.84 | 58.43 ±1.11 |
| | EdgePreGprompt | 83.18 ±3.22 | 69.63 ±7.04 | 85.38 ±6.75 | 99.56 ±0.58 | 87.93 ±6.62 | 56.19 ±3.18 | 54.51 ±2.74 |
| | GCL | 73.76 ±9.73 | 68.34 ±10.91 | 75.10 ±2.04 | 52.13 ±1.68 | 87.77 ±8.39 | 56.96 ±4.69 | 51.48 ±0.80 |
| | SimGRACE | 63.96 ±7.80 | 63.26 ±11.65 | 58.71 ±6.49 | 95.66 ±0.57 | 90.93 ±9.06 | 55.06 ±3.04 | 49.92 ±0.31 |
| GPF-plus | DGI | 55.30 ±13.20 | 55.81 ±15.74 | 66.54 ±15.31 | 93.25 ±5.92 | 77.44 ±7.96 | 57.09 ±3.17 | 55.62 ±1.15 |
| | GraphMAE | 87.26 ±3.14 | 91.53 ±1.55 | 87.12 ±1.71 | 99.99 ±0.02 | 86.23 ±6.11 | 72.39 ±1.82 | 58.01 ±1.44 |
| | EdgePreGPPT | 73.87 ±10.93 | 56.16 ±16.39 | 82.80 ±1.90 | 99.75 ±0.50 | 87.25 ±8.99 | 67.15 ±3.76 | 57.48 ±1.34 |
| | EdgePreGprompt | 85.81 ±3.03 | 91.47 ±1.96 | 85.13 ±2.57 | 99.73 ±0.51 | 88.28 ±9.02 | 65.17 ±4.06 | 56.91 ±3.33 |
| | GCL | 80.93 ±6.55 | 61.66 ±15.01 | 80.00 ±7.05 | 78.70 ±14.44 | 81.41 ±2.06 | 69.40 ±1.84 | 51.15 ±0.58 |
| | SimGRACE | 66.36 ±16.56 | 60.91 ±15.05 | 61.66 ±12.28 | 90.60 ±4.75 | 91.16 ±9.14 | 66.44 ±4.39 | 49.94 ±1.04 |

Table 17: Node classification performance Accuracy (%) (5-shot)

| Training schemes | Methods | Cora | CiteSeer | Pubmed | Wisconsin | Texas | Actor | Ogbn-arxiv |
|------------------------|-----------------|--------------|--------------|--------------|--------------|--------------|-------------|-------------|
| self-supervise | GCN | 50.25 ±8.37 | 41.22 ±6.30 | 67.88 ±2.18 | 39.43 ±5.86 | 43.91 ±6.47 | 21.92 ±1.86 | 22.38 ±3.05 |
| Pretrain + Fine-tuning | DGI | 48.79 ±8.51 | 35.91 ±4.94 | 61.44 ±7.93 | 40.00 ±6.31 | 47.19 ±7.37 | 21.48 ±1.71 | 12.48 ±6.15 |
| | GraphMAE | 54.09 ±4.21 | 32.75 ±6.35 | 70.04 ±4.57 | 40.69 ±6.73 | 43.44 ±8.48 | 22.92 ±1.22 | 24.08 ±1.45 |
| | EdgePreGPPT | 54.34 ±3.78 | 38.97 ±7.35 | 70.91 ±4.87 | 40.69 ±6.77 | 44.53 ±11.05 | 21.46 ±1.00 | 28.84 ±3.11 |
| | EdgePredGprompt | 49.04 ±8.99 | 32.08 ±8.24 | 70.44 ±5.04 | 41.03 ±6.58 | 45.78 ±8.07 | 22.46 ±1.99 | 25.94 ±2.63 |
| | GCL | 62.66 ±3.55 | 39.54 ±3.54 | 66.84 ±5.78 | 42.97 ±8.99 | 46.41 ±5.47 | 21.99 ±1.61 | 13.42 ±3.73 |
| | SimGRACE | 45.13 ±7.81 | 38.90 ±6.03 | 62.65 ±6.29 | 41.49 ±5.77 | 46.09 ±8.71 | 22.77 ±1.56 | 9.45 ±1.62 |
| GPPTPrompt | DGI | 43.68 ±7.12 | 45.77 ±7.41 | 47.39 ±10.22 | 36.29 ±3.97 | 48.82 ±5.15 | 20.91 ±1.36 | 8.85 ±6.21 |
| | GraphMAE | 31.50 ±11.89 | 19.93 ±6.60 | 66.97 ±3.70 | 37.00 ±3.19 | 42.68 ±6.91 | 22.10 ±0.86 | 8.90 ±6.25 |
| | EdgePreGPPT | 32.94 ±2.82 | 26.93 ±1.56 | 63.45 ±3.63 | 34.00 ±5.69 | 36.85 ±3.24 | 20.42 ±0.81 | 28.90 ±1.64 |
| | EdgePreGprompt | 30.55 ±4.59 | 28.46 ±4.25 | 63.33 ±4.18 | 32.57 ±3.31 | 39.21 ±6.09 | 20.07 ±0.45 | 4.01 ±0.69 |
| | GCL | 51.98 ±3.43 | 26.68 ±2.95 | 57.59 ±2.89 | 30.43 ±8.83 | 45.20 ±2.20 | 21.25 ±0.78 | 14.15 ±5.26 |
| | SimGRACE | 34.93 ±2.55 | 29.51 ±2.60 | 42.87 ±1.57 | 33.00 ±5.27 | 41.42 ±6.80 | 21.58 ±0.84 | 9.11 ±3.50 |
| ALL-in-one | DGI | 30.45 ±0.19 | 21.82 ±4.73 | 43.60 ±5.24 | 79.95 ±4.01 | 62.50 ±12.39 | 19.83 ±3.14 | 2.65 ±0.09 |
| | GraphMAE | 25.02 ±7.58 | 19.38 ±4.38 | 46.16 ±15.83 | 82.43 ±2.84 | 66.47 ±17.32 | 15.03 ±3.13 | 0.24 ±0.00 |
| | EdgePreGPPT | 30.36 ±13.48 | 25.83 ±9.32 | 37.61 ±20.58 | 88.55 ±3.52 | 47.98 ±27.78 | 21.49 ±3.02 | 1.59 ±1.30 |
| | EdgePreGprompt | 21.59 ±7.30 | 18.75 ±1.86 | 39.36 ±11.45 | 87.16 ±3.02 | 46.32 ±29.89 | 21.23 ±5.64 | 13.01 ±6.29 |
| | GCL | 25.63 ±17.68 | 27.93 ±10.59 | 42.86 ±7.21 | 84.75 ±8.86 | 73.28 ±9.91 | 20.83 ±8.04 | 6.70 ±6.01 |
| | SimGRACE | 15.20 ±9.51 | 25.19 ±12.95 | 21.59 ±2.58 | 78.90 ±2.26 | 64.63 ±25.59 | 21.26 ±6.24 | 4.80 ±5.17 |
| Gprompt | DGI | 27.81 ±10.28 | 52.84 ±2.36 | 37.83 ±5.71 | 57.24 ±27.27 | 34.47 ±41.15 | 26.07 ±4.20 | 56.59 ±4.05 |
| | GraphMAE | 66.82 ±3.98 | 60.07 ±3.98 | 66.66 ±2.25 | 74.89 ±35.83 | 36.73 ±43.95 | 30.71 ±2.12 | 76.98 ±3.88 |
| | EdgePreGPPT | 53.15 ±2.85 | 36.15 ±5.00 | 62.18 ±4.88 | 78.22 ±37.33 | 37.59 ±45.03 | 28.76 ±3.95 | 85.40 ±0.79 |
| | EdgePreGprompt | 62.63 ±3.26 | 65.34 ±4.60 | 67.87 ±2.08 | 78.01 ±37.38 | 38.45 ±46.03 | 21.86 ±2.13 | 81.85 ±3.16 |
| | GCL | 69.03 ±3.61 | 61.27 ±5.37 | 57.21 ±2.78 | 65.72 ±31.30 | 36.38 ±43.47 | 34.67 ±1.28 | 57.52 ±4.19 |
| | SimGRACE | 51.27 ±6.05 | 66.13 ±1.64 | 44.78 ±6.52 | 68.34 ±32.42 | 39.32 ±47.08 | 30.06 ±3.36 | 57.36 ±3.36 |
| GPF | DGI | 29.57 ±20.89 | 24.55 ±12.61 | 52.43 ±6.94 | 83.21 ±5.19 | 89.30 ±1.58 | 35.40 ±8.78 | 20.72 ±0.26 |
| | GraphMAE | 35.43 ±1.02 | 25.12 ±3.01 | 68.96 ±3.99 | 96.30 ±5.12 | 92.44 ±3.04 | 44.07 ±3.94 | 45.20 ±2.03 |
| | EdgePreGPPT | 15.39 ±8.19 | 18.11 ±2.78 | 58.67 ±3.21 | 98.26 ±1.19 | 89.47 ±1.91 | 30.99 ±4.68 | 71.83 ±9.37 |
| | EdgePreGprompt | 31.58 ±18.16 | 24.55 ±8.28 | 66.25 ±8.53 | 96.30 ±5.12 | 90.18 ±1.57 | 31.92 ±5.75 | 28.60 ±3.11 |
| | GCL | 28.60 ±11.19 | 17.69 ±1.35 | 52.47 ±6.73 | 69.15 ±21.44 | 98.42 ±0.36 | 29.13 ±2.63 | 23.90 ±0.12 |
| | SimGRACE | 18.94 ±12.60 | 22.45 ±3.45 | 40.35 ±1.71 | 92.29 ±3.36 | 91.73 ±3.84 | 30.28 ±2.64 | 33.77 ±8.48 |
| GPF-plus | DGI | 27.23 ±14.61 | 26.31 ±11.48 | 47.02 ±14.51 | 83.86 ±17.24 | 96.18 ±4.12 | 36.03 ±7.49 | 16.86 ±3.30 |
| | GraphMAE | 63.28 ±4.69 | 75.73 ±2.19 | 69.59 ±4.33 | 99.01 ±1.43 | 99.12 ±0.95 | 44.58 ±5.95 | 47.79 ±1.09 |
| | EdgePreGPPT | 22.44 ±23.88 | 13.63 ±5.27 | 66.43 ±3.28 | 98.52 ±2.07 | 96.18 ±4.12 | 37.15 ±8.48 | 66.88 ±6.14 |
| | EdgePreGprompt | 66.22 ±6.20 | 64.49 ±14.12 | 68.10 ±4.56 | 98.64 ±2.14 | 97.74 ±2.47 | 41.98 ±4.70 | 51.25 ±1.31 |
| | GCL | 47.71 ±22.44 | 29.16 ±16.66 | 64.53 ±4.15 | 70.98 ±19.27 | 99.12 ±0.95 | 36.99 ±6.59 | 25.74 ±1.70 |
| | SimGRACE | 27.79 ±21.51 | 28.37 ±22.54 | 50.25 ±10.04 | 89.57 ±4.22 | 95.83 ±3.85 | 38.23 ±2.34 | 44.70 ±2.69 |

Table 18: Node classification performance F1-score (5-shot)

| Training schemes | Methods | Cora | CiteSeer | Pubmed | Wisconsin | Texas | Actor | Ogbn-arxiv |
|------------------------|-----------------|--------------|--------------|--------------|--------------|--------------|--------------|-------------|
| self-supervise | GCN | 51.42 ±8.70 | 34.20 ±6.12 | 67.45 ±2.25 | 34.31 ±4.41 | 35.02 ±8.10 | 16.89 ±1.20 | 14.76 ±1.13 |
| Pretrain + Fine-tuning | DGI | 47.53 ±9.44 | 28.71 ±2.76 | 59.33 ±10.23 | 34.71 ±5.28 | 38.80 ±7.04 | 16.49 ±1.42 | 4.78 ±2.53 |
| | GraphMAE | 55.17 ±5.27 | 29.67 ±6.94 | 69.58 ±4.63 | 34.69 ±5.63 | 34.57 ±9.23 | 16.92 ±1.76 | 12.97 ±0.32 |
| | EdgePreGPPT | 55.50 ±3.62 | 31.31 ±8.86 | 70.31 ±4.65 | 34.86 ±4.83 | 36.63 ±7.69 | 19.42 ±1.02 | 20.19 ±1.27 |
| | EdgePredGprompt | 49.18 ±8.50 | 27.04 ±6.83 | 69.46 ±4.79 | 34.90 ±5.77 | 36.38 ±7.04 | 18.28 ±1.91 | 18.51 ±1.18 |
| | GCL | 61.32 ±3.43 | 35.95 ±3.18 | 66.16 ±5.43 | 26.66 ±5.78 | 33.81 ±3.28 | 20.44 ±1.45 | 6.96 ±1.07 |
| | SimGRACE | 45.33 ±4.88 | 34.74 ±7.40 | 62.66 ±6.39 | 35.86 ±4.86 | 36.86 ±7.72 | 20.63 ±1.13 | 4.74 ±0.48 |
| GPPTPrompt | DGI | 40.79 ±7.28 | 39.86 ±7.94 | 44.67 ±12.25 | 32.76 ±3.66 | 39.86 ±4.07 | 19.28 ±0.85 | 0.39 ±0.27 |
| | GraphMAE | 24.55 ±7.97 | 7.49 ±4.08 | 66.55 ±3.62 | 33.86 ±3.88 | 36.66 ±2.42 | 19.92 ±0.84 | 0.42 ±0.29 |
| | EdgePreGPPT | 32.50 ±2.42 | 19.95 ±1.98 | 62.42 ±4.78 | 30.57 ±4.59 | 33.93 ±2.76 | 19.89 ±0.56 | 18.11 ±1.03 |
| | EdgePreGprompt | 22.20 ±3.24 | 25.96 ±4.33 | 61.22 ±5.46 | 30.39 ±2.77 | 35.84 ±3.35 | 19.22 ±0.33 | 2.56 ±1.55 |
| | GCL | 50.60 ±2.51 | 25.00 ±2.05 | 57.27 ±2.84 | 18.78 ±5.00 | 35.83 ±1.96 | 20.34 ±0.74 | 3.03 ±1.03 |
| | SimGRACE | 31.56 ±2.75 | 28.27 ±1.68 | 40.87 ±2.90 | 27.06 ±3.81 | 36.11 ±2.81 | 20.45 ±0.51 | 1.63 ±0.40 |
| ALL-in-one | DGI | 6.67 ±0.03 | 10.68 ±7.84 | 26.81 ±9.85 | 60.73 ±6.34 | 31.59 ±18.97 | 10.66 ±3.88 | 0.21 ±0.04 |
| | GraphMAE | 6.77 ±3.23 | 8.66 ±4.25 | 33.97 ±19.42 | 58.02 ±3.64 | 49.27 ±9.03 | 8.31 ±2.80 | 0.01 ±0.00 |
| | EdgePreGPPT | 15.19 ±16.25 | 13.46 ±10.23 | 31.47 ±24.42 | 78.97 ±11.06 | 29.95 ±20.92 | 10.10 ±3.63 | 0.53 ±0.63 |
| | EdgePreGprompt | 4.99 ±1.38 | 5.28 ±0.47 | 27.49 ±15.02 | 68.69 ±3.29 | 25.44 ±17.47 | 8.08 ±2.12 | 0.56 ±0.27 |
| | GCL | 14.88 ±18.86 | 14.98 ±12.03 | 31.71 ±13.56 | 74.33 ±11.49 | 49.70 ±21.87 | 10.10 ±6.01 | 0.75 ±1.07 |
| | SimGRACE | 7.21 ±7.14 | 13.71 ±14.05 | 13.42 ±3.26 | 66.26 ±6.68 | 45.47 ±21.83 | 11.42 ±5.14 | 0.52 ±0.70 |
| Gprompt | DGI | 19.96 ±9.68 | 47.84 ±2.39 | 37.26 ±5.45 | 48.86 ±23.99 | 29.50 ±36.22 | 24.02 ±4.65 | 49.69 ±1.68 |
| | GraphMAE | 61.57 ±5.38 | 56.39 ±3.60 | 65.31 ±2.36 | 70.18 ±34.72 | 36.05 ±43.75 | 30.08 ±1.88 | 66.82 ±2.68 |
| | EdgePreGPPT | 51.03 ±3.08 | 31.15 ±3.86 | 60.74 ±4.30 | 73.65 ±36.66 | 34.07 ±41.83 | 27.29 ±3.43 | 79.73 ±1.29 |
| | EdgePreGprompt | 60.77 ±3.39 | 63.42 ±5.11 | 66.46 ±2.30 | 77.08 ±37.97 | 34.45 ±41.99 | 21.03 ±2.04 | 79.20 ±1.95 |
| | GCL | 64.22 ±4.08 | 57.65 ±5.74 | 56.66 ±2.35 | 51.82 ±25.62 | 31.38 ±38.22 | 32.99 ±1.00 | 49.82 ±2.81 |
| | SimGRACE | 45.98 ±5.22 | 63.84 ±1.37 | 42.20 ±5.64 | 62.12 ±30.55 | 38.47 ±46.70 | 29.63 ±3.89 | 53.41 ±2.08 |
| GPF | DGI | 18.44 ±18.83 | 13.25 ±11.04 | 47.23 ±9.28 | 75.78 ±14.28 | 77.01 ±10.57 | 28.58 ±14.86 | 11.02 ±0.99 |
| | GraphMAE | 23.17 ±2.53 | 12.78 ±1.23 | 67.88 ±4.07 | 87.75 ±13.24 | 82.81 ±10.55 | 43.09 ±4.07 | 16.76 ±2.76 |
| | EdgePreGPPT | 3.69 ±1.67 | 5.09 ±0.66 | 53.91 ±7.41 | 94.37 ±8.31 | 80.86 ±9.83 | 23.04 ±9.22 | 55.01 ±3.85 |
| | EdgePreGprompt | 20.17 ±13.79 | 12.79 ±6.88 | 62.47 ±11.25 | 89.10 ±11.52 | 79.88 ±10.50 | 30.44 ±9.87 | 18.77 ±0.29 |
| | GCL | 11.74 ±6.74 | 5.99 ±2.19 | 46.46 ±14.28 | 51.01 ±22.81 | 86.64 ±9.95 | 22.59 ±5.87 | 18.74 ±0.78 |
| | SimGRACE | 10.17 ±12.35 | 9.91 ±2.56 | 29.36 ±8.75 | 71.38 ±2.97 | 81.14 ±11.49 | 22.30 ±8.60 | 12.31 ±4.13 |
| GPF-plus | DGI | 13.92 ±14.65 | 12.41 ±13.45 | 40.73 ±16.83 | 71.22 ±13.69 | 84.85 ±10.28 | 32.05 ±9.40 | 6.42 ±2.47 |
| | GraphMAE | 61.55 ±4.90 | 74.86 ±2.27 | 68.34 ±4.25 | 94.48 ±8.79 | 86.93 ±10.70 | 44.28 ±6.46 | 24.14 ±2.16 |
| | EdgePreGPPT | 15.91 ±26.40 | 4.26 ±1.11 | 61.79 ±8.54 | 93.18 ±10.98 | 82.10 ±9.29 | 35.11 ±10.76 | 56.15 ±4.28 |
| | EdgePreGprompt | 65.96 ±5.07 | 57.16 ±18.91 | 66.11 ±4.85 | 93.46 ±10.79 | 85.56 ±9.53 | 39.84 ±6.02 | 32.07 ±2.05 |
| | GCL | 38.84 ±26.82 | 18.53 ±21.72 | 63.06 ±3.88 | 65.13 ±17.26 | 86.59 ±11.04 | 33.23 ±7.80 | 19.46 ±1.47 |
| | SimGRACE | 19.60 ±23.99 | 18.27 ±26.81 | 36.93 ±16.65 | 69.27 ±9.83 | 84.63 ±10.23 | 35.78 ±3.00 | 24.02 ±2.29 |

Table 19: Node classification performance AUROC (5-shot)

| Training schemes | Methods | Cora | CiteSeer | Pubmed | Wisconsin | Texas | Actor | Ogbn-arxiv |
|------------------------|-----------------|--------------|--------------|--------------|--------------|--------------|-------------|-------------|
| self-supervise | GCN | 88.39 ±2.56 | 78.16 ±2.95 | 84.93 ±2.77 | 65.96 ±5.03 | 66.32 ±8.89 | 51.44 ±0.62 | 78.83 ±1.37 |
| Pretrain + Fine-tuning | DGI | 84.60 ±5.40 | 74.29 ±3.78 | 81.26 ±3.74 | 67.40 ±4.97 | 66.62 ±8.12 | 51.78 ±0.48 | 62.86 ±4.35 |
| | GraphMAE | 90.30 ±1.62 | 77.04 ±3.75 | 86.09 ±2.54 | 65.32 ±5.26 | 63.95 ±9.63 | 51.73 ±0.46 | 75.65 ±0.78 |
| | EdgePreGPPT | 86.88 ±1.90 | 75.17 ±2.72 | 86.23 ±2.32 | 67.71 ±3.54 | 69.36 ±8.52 | 51.34 ±0.65 | 83.86 ±1.25 |
| | EdgePredGprompt | 87.41 ±2.15 | 73.79 ±5.07 | 85.66 ±3.26 | 66.04 ±5.12 | 66.70 ±8.92 | 52.40 ±0.91 | 82.34 ±1.45 |
| | GCL | 90.49 ±1.49 | 76.54 ±1.56 | 83.07 ±3.73 | 57.71 ±8.14 | 65.57 ±7.78 | 52.68 ±0.69 | 66.70 ±0.95 |
| | SimGRACE | 85.45 ±2.29 | 76.95 ±2.79 | 80.86 ±3.17 | 67.73 ±4.47 | 66.11 ±7.69 | 52.52 ±0.92 | 60.36 ±0.72 |
| GPPTPrompt | DGI | 76.38 ±6.02 | 76.59 ±3.18 | 63.76 ±10.00 | 69.27 ±1.49 | 66.52 ±3.33 | 51.27 ±1.04 | 50.09 ±0.25 |
| | GraphMAE | 84.77 ±3.18 | 69.69 ±1.62 | 82.56 ±1.84 | 66.53 ±4.44 | 63.43 ±2.99 | 50.92 ±0.44 | 63.55 ±0.53 |
| | EdgePreGPPT | 69.44 ±2.15 | 60.14 ±1.47 | 77.28 ±3.58 | 62.04 ±3.27 | 61.22 ±2.13 | 50.85 ±0.37 | 85.67 ±0.67 |
| | EdgePreGprompt | 69.60 ±0.83 | 62.44 ±2.17 | 81.98 ±4.11 | 63.83 ±2.92 | 65.86 ±2.34 | 50.32 ±0.54 | 71.01 ±5.38 |
| | GCL | 82.06 ±2.24 | 61.54 ±2.42 | 76.26 ±1.36 | 55.23 ±2.54 | 62.83 ±4.62 | 50.87 ±0.69 | 58.53 ±1.53 |
| | SimGRACE | 70.54 ±2.07 | 64.49 ±1.54 | 60.67 ±1.80 | 66.38 ±0.42 | 62.69 ±1.92 | 51.00 ±0.54 | 51.46 ±0.26 |
| ALL-in-one | DGI | 42.65 ±0.41 | 57.45 ±7.01 | 57.38 ±8.87 | 85.53 ±2.93 | 68.87 ±6.32 | 51.24 ±2.32 | 54.76 ±0.75 |
| | GraphMAE | 80.75 ±0.64 | 57.95 ±1.98 | 81.32 ±2.25 | 83.91 ±5.14 | 72.30 ±3.24 | 53.43 ±0.71 | 53.46 ±2.41 |
| | EdgePreGPPT | 62.42 ±11.37 | 62.25 ±7.23 | 63.45 ±13.43 | 95.30 ±4.46 | 78.68 ±10.06 | 51.90 ±0.91 | 54.22 ±1.57 |
| | EdgePreGprompt | 77.24 ±3.51 | 64.14 ±1.01 | 75.97 ±3.33 | 91.72 ±1.62 | 80.10 ±10.86 | 50.40 ±3.16 | 75.81 ±2.83 |
| | GCL | 65.99 ±10.83 | 65.27 ±11.57 | 69.76 ±4.24 | 88.87 ±4.34 | 76.09 ±6.05 | 54.21 ±2.36 | 55.77 ±4.78 |
| | SimGRACE | 59.53 ±3.53 | 64.55 ±7.97 | 47.71 ±3.86 | 86.55 ±2.81 | 83.01 ±10.84 | 52.44 ±1.68 | 53.56 ±7.86 |
| Gprompt | DGI | 70.76 ±4.67 | 80.01 ±2.22 | 54.50 ±2.51 | 84.53 ±17.37 | 65.41 ±9.24 | 54.02 ±2.61 | 92.66 ±0.73 |
| | GraphMAE | 72.59 ±7.56 | 76.04 ±5.23 | 82.88 ±2.18 | 89.40 ±19.62 | 66.34 ±10.38 | 56.23 ±2.46 | 96.45 ±0.14 |
| | EdgePreGPPT | 85.57 ±0.39 | 68.97 ±4.56 | 76.57 ±1.08 | 89.57 ±19.71 | 66.48 ±10.55 | 58.85 ±2.98 | 98.40 ±0.08 |
| | EdgePreGprompt | 88.49 ±0.88 | 84.82 ±2.89 | 83.58 ±1.29 | 89.71 ±19.78 | 66.52 ±10.61 | 52.51 ±2.39 | 97.79 ±0.15 |
| | GCL | 88.70 ±2.05 | 81.00 ±6.61 | 74.92 ±2.53 | 83.64 ±16.83 | 66.33 ±10.37 | 63.03 ±3.07 | 94.54 ±0.37 |
| | SimGRACE | 78.28 ±3.96 | 89.06 ±0.86 | 59.22 ±5.76 | 86.22 ±18.17 | 66.68 ±10.79 | 58.88 ±3.13 | 95.27 ±0.28 |
| GPF | DGI | 62.20 ±15.24 | 62.32 ±17.75 | 67.14 ±11.86 | 89.86 ±8.25 | 78.39 ±1.66 | 65.97 ±7.26 | 74.59 ±0.97 |
| | GraphMAE | 80.34 ±2.03 | 67.33 ±3.16 | 86.64 ±2.02 | 97.72 ±3.33 | 89.25 ±8.10 | 73.06 ±2.24 | 74.39 ±1.76 |
| | EdgePreGPPT | 49.95 ±0.10 | 52.76 ±6.92 | 74.91 ±5.59 | 99.98 ±0.03 | 83.65 ±7.64 | 59.31 ±5.01 | 90.96 ±1.32 |
| | EdgePreGprompt | 73.56 ±5.48 | 76.82 ±4.28 | 85.50 ±1.95 | 97.15 ±4.98 | 86.81 ±8.42 | 60.43 ±6.30 | 75.36 ±2.61 |
| | GCL | 79.04 ±5.41 | 64.25 ±7.12 | 70.90 ±6.31 | 73.84 ±18.96 | 83.62 ±4.75 | 57.39 ±3.99 | 75.19 ±0.54 |
| | SimGRACE | 60.98 ±9.22 | 64.93 ±3.80 | 58.21 ±3.99 | 90.38 ±1.52 | 90.43 ±8.89 | 61.68 ±2.02 | 60.89 ±3.56 |
| GPF-plus | DGI | 60.30 ±14.39 | 54.40 ±11.09 | 73.47 ±9.02 | 95.86 ±3.95 | 89.08 ±8.43 | 68.06 ±2.70 | 67.37 ±3.03 |
| | GraphMAE | 89.99 ±1.39 | 92.09 ±1.11 | 85.69 ±3.12 | 99.93 ±0.14 | 87.29 ±7.92 | 73.10 ±2.68 | 81.99 ±0.56 |
| | EdgePreGPPT | 65.26 ±14.38 | 55.10 ±2.95 | 82.94 ±2.55 | 99.96 ±0.05 | 86.89 ±8.72 | 70.17 ±4.50 | 93.73 ±1.05 |
| | EdgePreGprompt | 91.09 ±1.03 | 82.22 ±12.12 | 83.84 ±3.91 | 99.96 ±0.08 | 83.88 ±7.87 | 72.25 ±1.48 | 81.54 ±1.05 |
| | GCL | 84.11 ±7.92 | 61.55 ±15.26 | 81.38 ±3.75 | 91.95 ±9.63 | 86.24 ±7.72 | 69.74 ±1.87 | 77.42 ±2.14 |
| | SimGRACE | 66.55 ±14.49 | 63.11 ±14.51 | 65.07 ±10.97 | 88.17 ±7.54 | 89.58 ±9.03 | 68.81 ±1.60 | 69.60 ±2.37 |

Table 28: Graph classification performance AUROC (5-shot)

| Training schemes | Methods | IMDB-B | COLLAB | PROTEINS | MUTAG | ENZYMES | COX2 | BZR | DD |
|------------------------|----------------|-------------------|-------------------|-------------------|-------------------|------------------|-------------------|-------------------|------------------|
| self-supervise | GCN | 66.66 \pm 4.29 | 65.98 \pm 11.45 | 60.55 \pm 1.27 | 68.80 \pm 5.31 | 59.06 \pm 0.90 | 53.16 \pm 1.88 | 63.16 \pm 1.91 | 64.75 \pm 3.62 |
| Pretrain + Fine-tuning | DGI | 50.02 \pm 0.04 | 81.66 \pm 0.61 | 60.91 \pm 1.77 | 68.55 \pm 5.83 | 58.93 \pm 1.27 | 51.76 \pm 2.64 | 52.81 \pm 2.18 | 63.61 \pm 4.41 |
| | GraphMAE | 57.68 \pm 8.77 | 69.01 \pm 10.12 | 59.92 \pm 2.11 | 74.33 \pm 4.22 | 58.39 \pm 0.82 | 51.25 \pm 2.21 | 57.13 \pm 7.05 | 64.45 \pm 3.57 |
| | EdgePreGPPT | 61.63 \pm 14.66 | 76.71 \pm 9.50 | 60.88 \pm 1.39 | 72.84 \pm 3.58 | 59.64 \pm 0.79 | 53.48 \pm 3.98 | 60.58 \pm 6.74 | 63.82 \pm 3.08 |
| | EdgePreGprompt | 66.78 \pm 4.49 | 81.42 \pm 0.65 | 56.93 \pm 4.04 | 72.62 \pm 1.98 | 59.65 \pm 1.08 | 51.34 \pm 2.69 | 50.75 \pm 1.31 | 62.82 \pm 3.74 |
| | GCL | 69.68 \pm 5.15 | 76.71 \pm 2.70 | 61.80 \pm 2.09 | 70.63 \pm 4.24 | 57.49 \pm 0.69 | 55.91 \pm 4.67 | 50.86 \pm 1.72 | 64.37 \pm 3.44 |
| | SimGRACE | 64.69 \pm 7.42 | 72.81 \pm 3.58 | 61.49 \pm 2.04 | 71.48 \pm 3.93 | 55.98 \pm 2.05 | 61.95 \pm 2.47 | 48.92 \pm 2.16 | 57.39 \pm 2.98 |
| GPPTPrompt | DGI | 50.48 \pm 0.70 | 70.62 \pm 7.22 | 55.92 \pm 14.32 | 79.97 \pm 4.66 | 52.20 \pm 2.26 | 51.22 \pm 2.96 | 47.75 \pm 9.82 | 53.81 \pm 9.82 |
| | GraphMAE | 50.63 \pm 0.56 | 50.56 \pm 3.78 | 60.56 \pm 6.96 | 77.76 \pm 5.00 | 53.63 \pm 1.06 | 50.60 \pm 6.83 | 53.55 \pm 6.08 | 58.60 \pm 5.56 |
| | EdgePreGPPT | 69.02 \pm 4.94 | 68.78 \pm 9.45 | 68.50 \pm 9.36 | 72.44 \pm 6.57 | 53.43 \pm 1.03 | 53.02 \pm 8.32 | 47.93 \pm 5.96 | 51.76 \pm 5.53 |
| | EdgePreGprompt | 50.63 \pm 0.56 | 56.85 \pm 1.07 | 45.95 \pm 18.22 | 24.75 \pm 25.84 | 53.66 \pm 1.73 | 52.56 \pm 3.29 | 49.31 \pm 5.04 | 61.01 \pm 3.67 |
| | GCL | 59.75 \pm 10.88 | 76.73 \pm 1.15 | 69.17 \pm 8.03 | 77.80 \pm 5.27 | 53.37 \pm 1.20 | 56.22 \pm 8.12 | 50.45 \pm 6.37 | 57.89 \pm 3.91 |
| | SimGRACE | 61.35 \pm 10.03 | 70.93 \pm 4.01 | 59.76 \pm 13.58 | 77.27 \pm 7.67 | 53.63 \pm 1.06 | 57.78 \pm 6.95 | 59.49 \pm 6.13 | 57.49 \pm 4.92 |
| ALL-in-one | DGI | 61.66 \pm 12.01 | 80.99 \pm 1.96 | 79.39 \pm 1.76 | 90.73 \pm 0.46 | 58.71 \pm 0.88 | 49.33 \pm 1.56 | 52.97 \pm 11.15 | 52.92 \pm 1.58 |
| | GraphMAE | 60.44 \pm 13.36 | 50.00 \pm 0.00 | 73.82 \pm 1.89 | 89.59 \pm 1.17 | 53.75 \pm 1.13 | 54.24 \pm 10.01 | 55.73 \pm 11.76 | 64.55 \pm 1.63 |
| | EdgePreGPPT | 69.53 \pm 3.98 | 81.50 \pm 1.45 | 77.60 \pm 0.81 | 91.73 \pm 0.49 | 54.18 \pm 3.09 | 52.22 \pm 5.14 | 39.69 \pm 2.87 | 58.41 \pm 0.60 |
| | EdgePreGprompt | 61.78 \pm 11.31 | 81.93 \pm 1.28 | 66.26 \pm 1.84 | 87.58 \pm 0.76 | 55.09 \pm 2.15 | 61.86 \pm 2.18 | 50.64 \pm 4.57 | 64.95 \pm 0.92 |
| | GCL | 65.55 \pm 5.57 | 77.86 \pm 4.19 | 73.65 \pm 4.67 | 85.46 \pm 1.40 | 59.50 \pm 1.10 | 49.77 \pm 1.51 | 52.63 \pm 2.91 | 59.95 \pm 4.37 |
| | SimGRACE | 66.52 \pm 3.18 | 75.84 \pm 0.40 | 77.13 \pm 0.44 | 84.04 \pm 2.22 | 55.76 \pm 0.82 | 62.42 \pm 1.89 | 54.60 \pm 2.87 | 55.43 \pm 0.66 |
| Gprompt | DGI | 53.91 \pm 13.03 | 70.52 \pm 5.99 | 57.55 \pm 7.36 | 64.03 \pm 19.95 | 53.87 \pm 2.02 | 50.06 \pm 2.93 | 56.36 \pm 6.92 | 54.21 \pm 4.23 |
| | GraphMAE | 55.46 \pm 7.98 | 44.83 \pm 0.15 | 54.75 \pm 7.17 | 63.37 \pm 9.48 | 52.82 \pm 1.11 | 49.40 \pm 6.97 | 54.99 \pm 5.88 | 51.72 \pm 5.54 |
| | EdgePreGPPT | 70.26 \pm 4.44 | 73.55 \pm 4.01 | 67.07 \pm 0.80 | 66.04 \pm 10.33 | 53.32 \pm 2.77 | 52.47 \pm 6.25 | 52.16 \pm 2.83 | 53.26 \pm 4.39 |
| | EdgePreGprompt | 56.66 \pm 8.48 | 75.59 \pm 3.42 | 56.24 \pm 7.10 | 74.43 \pm 4.69 | 54.57 \pm 2.69 | 50.60 \pm 3.01 | 53.90 \pm 2.28 | 56.36 \pm 5.42 |
| | GCL | 52.62 \pm 12.32 | 75.45 \pm 3.52 | 61.03 \pm 5.77 | 64.43 \pm 10.15 | 54.04 \pm 2.81 | 50.64 \pm 7.63 | 53.71 \pm 2.16 | 56.22 \pm 3.97 |
| | SimGRACE | 57.24 \pm 12.21 | 73.41 \pm 3.61 | 60.58 \pm 5.34 | 64.67 \pm 8.28 | 55.02 \pm 1.57 | 56.79 \pm 5.66 | 56.83 \pm 6.49 | 53.99 \pm 6.27 |
| GPF | DGI | 61.28 \pm 14.33 | 80.91 \pm 1.35 | 59.36 \pm 1.27 | 68.29 \pm 4.20 | 57.38 \pm 0.69 | 55.65 \pm 2.89 | 69.56 \pm 8.43 | 52.10 \pm 1.18 |
| | GraphMAE | 71.36 \pm 4.86 | 50.00 \pm 0.00 | 60.79 \pm 1.35 | 73.75 \pm 6.18 | 57.51 \pm 1.83 | 55.80 \pm 3.40 | 57.87 \pm 6.08 | 59.78 \pm 6.01 |
| | EdgePreGPPT | 72.09 \pm 6.35 | 81.60 \pm 1.83 | 63.33 \pm 6.33 | 71.14 \pm 4.08 | 57.09 \pm 1.47 | 45.44 \pm 3.51 | 40.30 \pm 2.41 | 57.56 \pm 1.55 |
| | EdgePreGprompt | 66.56 \pm 4.93 | 82.11 \pm 0.58 | 59.32 \pm 1.46 | 78.72 \pm 2.44 | 56.88 \pm 0.88 | 57.11 \pm 3.21 | 38.86 \pm 3.08 | 59.75 \pm 5.73 |
| | GCL | 61.77 \pm 14.84 | 76.32 \pm 2.61 | 58.70 \pm 2.93 | 75.54 \pm 4.82 | 57.23 \pm 1.32 | 50.87 \pm 1.24 | 54.74 \pm 5.41 | 51.02 \pm 3.03 |
| | SimGRACE | 66.69 \pm 4.51 | 78.63 \pm 3.50 | 60.84 \pm 0.83 | 73.60 \pm 4.98 | 57.64 \pm 0.64 | 55.74 \pm 2.01 | 44.52 \pm 2.78 | 48.14 \pm 1.17 |
| GPF-plus | DGI | 67.76 \pm 7.65 | 80.06 \pm 1.94 | 60.56 \pm 1.11 | 75.93 \pm 5.59 | 57.99 \pm 1.37 | 56.06 \pm 2.35 | 67.92 \pm 4.56 | 62.75 \pm 3.59 |
| | GraphMAE | 68.56 \pm 7.52 | 50.00 \pm 0.00 | 60.88 \pm 1.41 | 74.54 \pm 6.62 | 56.78 \pm 1.51 | 48.93 \pm 2.41 | 54.11 \pm 8.51 | 62.56 \pm 3.54 |
| | EdgePreGPPT | 71.71 \pm 3.89 | 82.51 \pm 0.50 | 59.30 \pm 1.96 | 77.03 \pm 2.00 | 56.17 \pm 0.68 | 49.53 \pm 4.57 | 49.78 \pm 0.43 | 63.49 \pm 4.00 |
| | EdgePreGprompt | 67.36 \pm 7.32 | 81.68 \pm 0.93 | 54.63 \pm 1.53 | 75.31 \pm 2.84 | 60.04 \pm 0.95 | 50.95 \pm 1.90 | 48.60 \pm 2.80 | 64.20 \pm 3.47 |
| | GCL | 61.80 \pm 14.82 | 80.57 \pm 0.98 | 61.23 \pm 1.64 | 69.47 \pm 3.56 | 57.95 \pm 1.27 | 51.87 \pm 1.80 | 64.97 \pm 3.46 | 61.94 \pm 4.01 |
| | SimGRACE | 61.85 \pm 14.76 | 77.71 \pm 3.28 | 61.07 \pm 1.57 | 70.13 \pm 2.87 | 57.30 \pm 0.70 | 63.15 \pm 2.98 | 49.11 \pm 1.78 | 57.21 \pm 2.94 |

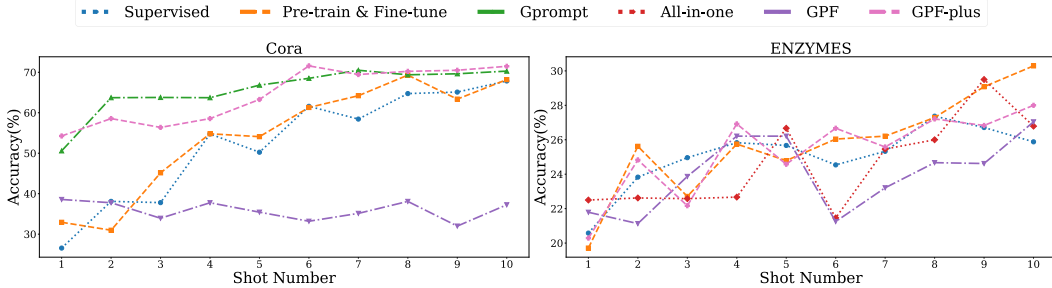


Figure 5: The curve of the shot number from 1 to 10 and the accuracy (%) on Cora and ENZYMES.

Table 29: Node classification performance Accuracy (%) across different graph models on Wisconsin (1-shot setting). The **Accuracy** of supervised methods: **GAT** is $34.51_{\pm 18.02}$, **GraphSAGE** is $25.37_{\pm 5.61}$, **GT** is $20.91_{\pm 7.07}$

| Pre-train & Fine-tune | | | | | | |
|-----------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|
| Model | DGI | GraphMAE | EdgePreGPPT | EdgePreGprompt | GCL | SimGRACE |
| GAT | $16.00_{\pm 6.24}$ | $37.60_{\pm 10.69}$ | $20.00_{\pm 3.82}$ | $33.37_{\pm 4.76}$ | $18.86_{\pm 1.88}$ | $28.00_{\pm 9.40}$ |
| GraphSAGE | $40.69_{\pm 9.46}$ | $43.77_{\pm 12.43}$ | $26.06_{\pm 5.38}$ | $29.94_{\pm 3.75}$ | $36.57_{\pm 4.88}$ | $9.37_{\pm 2.72}$ |
| GT | $25.71_{\pm 3.07}$ | $39.77_{\pm 8.42}$ | $23.20_{\pm 2.65}$ | $28.23_{\pm 6.64}$ | $11.77_{\pm 1.06}$ | $14.51_{\pm 5.08}$ |
| GPPT | | | | | | |
| Model | DGI | GraphMAE | EdgePreGPPT | EdgePreGprompt | GCL | SimGRACE |
| GAT | $22.17_{\pm 6.13}$ | $33.94_{\pm 7.76}$ | $23.43_{\pm 4.46}$ | $37.94_{\pm 7.11}$ | $26.86_{\pm 6.12}$ | $29.83_{\pm 8.04}$ |
| GraphSAGE | $26.51_{\pm 8.00}$ | $30.51_{\pm 5.40}$ | $21.49_{\pm 5.17}$ | $24.23_{\pm 6.55}$ | $20.91_{\pm 7.11}$ | $25.37_{\pm 7.22}$ |
| GT | $27.20_{\pm 10.48}$ | $29.83_{\pm 5.80}$ | $28.00_{\pm 6.01}$ | $23.31_{\pm 3.01}$ | $27.66_{\pm 0.69}$ | $25.03_{\pm 5.43}$ |
| All-in-one | | | | | | |
| Model | DGI | GraphMAE | EdgePreGPPT | EdgePreGprompt | GCL | SimGRACE |
| GAT | $69.44_{\pm 5.19}$ | $36.25_{\pm 10.63}$ | $91.25_{\pm 4.33}$ | $92.65_{\pm 3.75}$ | $42.85_{\pm 9.16}$ | $36.61_{\pm 14.86}$ |
| GraphSAGE | $94.73_{\pm 5.86}$ | $97.55_{\pm 0.78}$ | $98.60_{\pm 0.87}$ | $100.00_{\pm 0.00}$ | $83.01_{\pm 18.37}$ | $85.12_{\pm 16.29}$ |
| GT | $68.59_{\pm 11.78}$ | $98.48_{\pm 0.70}$ | $96.74_{\pm 1.01}$ | $100.00_{\pm 0.00}$ | $70.23_{\pm 8.80}$ | $45.70_{\pm 11.83}$ |
| Gprompt | | | | | | |
| Model | DGI | GraphMAE | EdgePreGPPT | EdgePreGprompt | GCL | SimGRACE |
| GAT | $58.25_{\pm 13.83}$ | $67.77_{\pm 15.91}$ | $94.17_{\pm 2.26}$ | $84.28_{\pm 3.63}$ | $80.11_{\pm 16.65}$ | $57.18_{\pm 12.60}$ |
| GraphSAGE | $66.48_{\pm 12.88}$ | $83.49_{\pm 15.93}$ | $87.52_{\pm 3.79}$ | $82.16_{\pm 2.64}$ | $65.50_{\pm 6.48}$ | $72.61_{\pm 5.97}$ |
| GT | $56.03_{\pm 7.33}$ | $73.50_{\pm 9.72}$ | $76.97_{\pm 13.39}$ | $80.07_{\pm 2.84}$ | $59.31_{\pm 10.17}$ | $69.30_{\pm 10.57}$ |
| GPF | | | | | | |
| Model | DGI | GraphMAE | EdgePreGPPT | EdgePreGprompt | GCL | SimGRACE |
| GAT | $63.64_{\pm 3.91}$ | $78.59_{\pm 18.57}$ | $97.90_{\pm 0.79}$ | $91.84_{\pm 5.19}$ | $74.93_{\pm 4.49}$ | $65.67_{\pm 15.07}$ |
| GraphSAGE | $68.12_{\pm 13.96}$ | $67.66_{\pm 13.37}$ | $74.06_{\pm 14.59}$ | $72.45_{\pm 10.14}$ | $59.69_{\pm 21.37}$ | $78.37_{\pm 14.84}$ |
| GT | $39.85_{\pm 4.83}$ | $71.26_{\pm 14.43}$ | $72.67_{\pm 13.36}$ | $81.33_{\pm 3.41}$ | $78.19_{\pm 2.19}$ | $67.90_{\pm 10.53}$ |
| GPF-plus | | | | | | |
| Model | DGI | GraphMAE | EdgePreGPPT | EdgePreGprompt | GCL | SimGRACE |
| GAT | $92.62_{\pm 8.60}$ | $96.37_{\pm 4.44}$ | $99.29_{\pm 1.41}$ | $97.88_{\pm 4.24}$ | $86.12_{\pm 7.73}$ | $94.17_{\pm 2.63}$ |
| GraphSAGE | $77.85_{\pm 3.30}$ | $87.88_{\pm 0.90}$ | $99.30_{\pm 0.86}$ | $96.16_{\pm 5.09}$ | $85.93_{\pm 19.52}$ | $72.47_{\pm 8.33}$ |
| GT | $91.69_{\pm 6.84}$ | $98.60_{\pm 1.02}$ | $99.30_{\pm 0.86}$ | $96.13_{\pm 4.27}$ | $58.83_{\pm 18.53}$ | $89.04_{\pm 4.25}$ |

Table 30: Graph classification performance Accuracy (%) across different graph models on PROTEINS (1-shot setting). The **Accuracy** of supervised methods: **GAT** is $48.34_{\pm 9.96}$, **GraphSAGE** is $60.54_{\pm 2.95}$, **GT** is $61.44_{\pm 2.48}$

| Pre-train & Fine-tune | | | | | | |
|-----------------------|---------------------|---------------------|---------------------|--------------------|---------------------|--------------------|
| Model | DGI | GraphMAE | EdgePreGPPT | EdgePreGprompt | GCL | SimGRACE |
| GAT | $58.34_{\pm 6.52}$ | $61.06_{\pm 4.13}$ | $63.75_{\pm 3.71}$ | $54.09_{\pm 4.03}$ | $60.04_{\pm 3.06}$ | $58.65_{\pm 6.71}$ |
| GraphSAGE | $60.70_{\pm 4.08}$ | $60.56_{\pm 5.12}$ | $61.60_{\pm 1.78}$ | $63.21_{\pm 1.80}$ | $61.80_{\pm 3.77}$ | $58.56_{\pm 1.84}$ |
| GT | $53.87_{\pm 4.81}$ | $60.00_{\pm 3.99}$ | $64.92_{\pm 3.19}$ | $56.58_{\pm 3.28}$ | $62.88_{\pm 1.82}$ | $60.00_{\pm 1.60}$ |
| GPPT | | | | | | |
| Model | DGI | GraphMAE | EdgePreGPPT | EdgePreGprompt | GCL | SimGRACE |
| GAT | $57.71_{\pm 8.98}$ | $57.80_{\pm 10.55}$ | $58.04_{\pm 9.92}$ | $54.97_{\pm 7.45}$ | $52.29_{\pm 7.83}$ | $55.15_{\pm 9.84}$ |
| GraphSAGE | $56.56_{\pm 6.73}$ | $57.73_{\pm 7.95}$ | $58.63_{\pm 11.78}$ | $56.94_{\pm 5.67}$ | $58.00_{\pm 7.80}$ | $54.74_{\pm 6.59}$ |
| GT | $53.08_{\pm 7.56}$ | $57.35_{\pm 8.58}$ | $60.27_{\pm 3.92}$ | $55.51_{\pm 7.68}$ | $56.18_{\pm 5.79}$ | $55.87_{\pm 7.69}$ |
| All-in-one | | | | | | |
| Model | DGI | GraphMAE | EdgePreGPPT | EdgePreGprompt | GCL | SimGRACE |
| GAT | $60.04_{\pm 3.84}$ | $60.00_{\pm 6.04}$ | $62.11_{\pm 2.85}$ | $63.21_{\pm 2.22}$ | $58.36_{\pm 4.93}$ | $59.37_{\pm 5.59}$ |
| GraphSAGE | $59.53_{\pm 4.94}$ | $60.70_{\pm 4.89}$ | $63.12_{\pm 1.59}$ | $59.98_{\pm 8.46}$ | $62.22_{\pm 3.81}$ | $62.04_{\pm 2.07}$ |
| GT | $57.39_{\pm 3.66}$ | $58.92_{\pm 6.61}$ | $62.61_{\pm 4.08}$ | $60.20_{\pm 7.55}$ | $62.81_{\pm 1.63}$ | $50.52_{\pm 6.17}$ |
| Gprompt | | | | | | |
| Model | DGI | GraphMAE | EdgePreGPPT | EdgePreGprompt | GCL | SimGRACE |
| GAT | $61.08_{\pm 6.19}$ | $63.03_{\pm 2.61}$ | $64.47_{\pm 4.30}$ | $61.48_{\pm 3.34}$ | $59.12_{\pm 6.84}$ | $58.13_{\pm 7.27}$ |
| GraphSAGE | $61.35_{\pm 2.21}$ | $59.48_{\pm 9.19}$ | $60.92_{\pm 3.16}$ | $63.30_{\pm 1.43}$ | $55.26_{\pm 2.61}$ | $63.21_{\pm 2.66}$ |
| GT | $56.65_{\pm 5.81}$ | $60.99_{\pm 1.62}$ | $61.87_{\pm 5.60}$ | $55.33_{\pm 3.69}$ | $54.81_{\pm 7.62}$ | $58.97_{\pm 1.16}$ |
| GPF | | | | | | |
| Model | DGI | GraphMAE | EdgePreGPPT | EdgePreGprompt | GCL | SimGRACE |
| GAT | $66.11_{\pm 5.18}$ | $65.62_{\pm 5.38}$ | $56.38_{\pm 8.64}$ | $65.55_{\pm 6.65}$ | $59.71_{\pm 7.66}$ | $67.42_{\pm 6.26}$ |
| GraphSAGE | $64.20_{\pm 7.63}$ | $67.87_{\pm 4.32}$ | $58.18_{\pm 9.06}$ | $64.49_{\pm 6.80}$ | $60.25_{\pm 2.91}$ | $62.94_{\pm 2.29}$ |
| GT | $65.80_{\pm 7.42}$ | $60.16_{\pm 5.81}$ | $64.54_{\pm 7.18}$ | $61.21_{\pm 2.91}$ | $58.74_{\pm 5.51}$ | $59.57_{\pm 2.93}$ |
| GPF-plus | | | | | | |
| Model | DGI | GraphMAE | EdgePreGPPT | EdgePreGprompt | GCL | SimGRACE |
| GAT | $56.20_{\pm 12.87}$ | $57.35_{\pm 11.28}$ | $56.25_{\pm 8.61}$ | $53.24_{\pm 4.79}$ | $57.48_{\pm 11.74}$ | $57.48_{\pm 9.63}$ |
| GraphSAGE | $56.22_{\pm 9.08}$ | $57.55_{\pm 10.56}$ | $56.31_{\pm 9.26}$ | $57.71_{\pm 9.60}$ | $53.89_{\pm 9.47}$ | $55.89_{\pm 4.30}$ |
| GT | $53.39_{\pm 5.23}$ | $57.37_{\pm 10.95}$ | $57.39_{\pm 11.88}$ | $52.61_{\pm 5.30}$ | $57.62_{\pm 12.27}$ | $56.16_{\pm 5.07}$ |

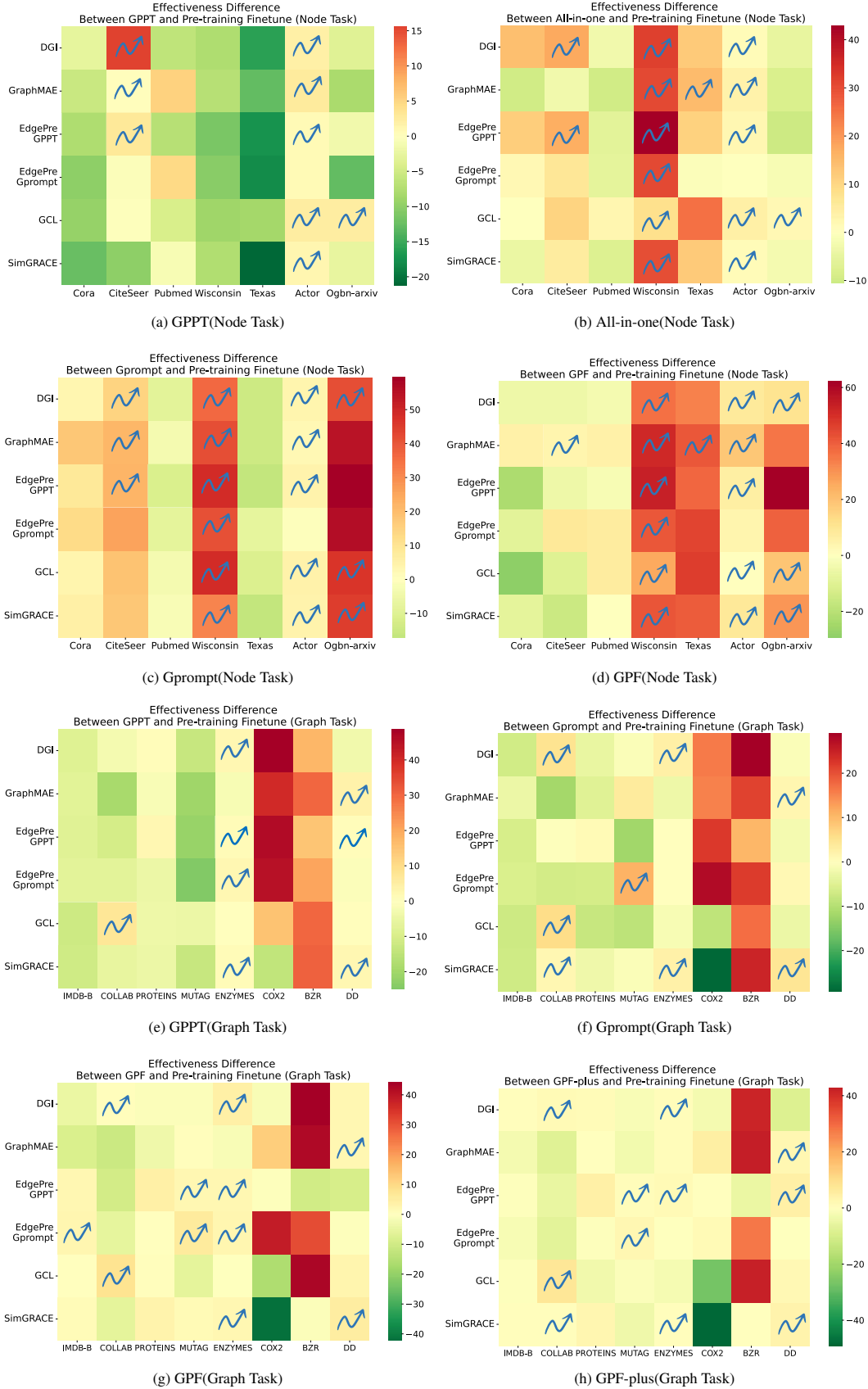


Figure 6: Heat maps of graph prompt methods on different node/graph-level datasets in 1-shot settings, excluding GPF-plus on the node task and All-in-one on the graph tasks.

Checklist

1. For all authors...
 - (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
 - (b) Did you describe the limitations of your work? [Yes] Please see Section 6
 - (c) Did you discuss any potential negative societal impacts of your work? [N/A]
 - (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
 - (a) Did you state the full set of assumptions of all theoretical results? [N/A]
 - (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments (e.g. for benchmarks)...
 - (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
 - (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
 - (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
 - (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
 - (a) If your work uses existing assets, did you cite the creators? [Yes]
 - (b) Did you mention the license of the assets? [N/A]
 - (c) Did you include any new assets either in the supplemental material or as a URL? [No]
 - (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A] All datasets are public.
 - (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
 - (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
 - (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
 - (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]