Can Graph Neural Networks Learn Language with Extremely Weak Text Supervision?

Zihao Li[1], Lecheng Zheng[1], Bowen Jin[1], Dongqi Fu[2], Baoyu Jing[1], Yikun Ban[1], Jingrui He[1], Jiawei Han[1]
[1]University of Illinois Urbana-Champaign, [2]Meta AI

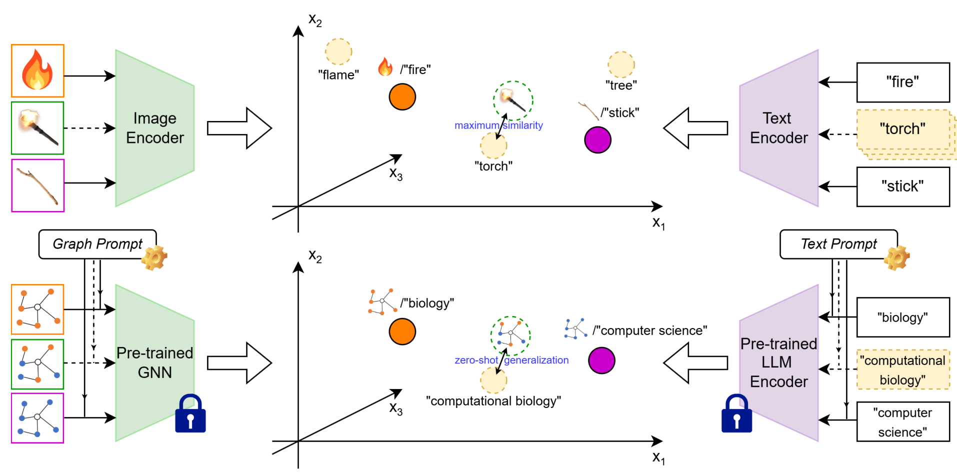

Figure 1: CLIP backbone (top) and this work (bottom). If a research paper cites many papers from biology and computer science, we realize this paper will likely be about computational biology, even if we do not know what exactly computational biology is. CLIP builds image encoders that learn such language dependency by Contrastive Language-Image Pre-training in the same embedding space according to Internet-scale data. However, text supervision is often extremely weak for graphs. This work leverages Multi-modal Prompt Learning for Graph Neural Networks that can effectively teach GNNs language dependency given few training samples with weak text supervision.

Abstract

While great success has been achieved in building vision models with Contrastive Language-Image Pre-training (CLIP) over Internet-scale image-text pairs, building transferable Graph Neural Networks (GNNs) with CLIP pipeline is challenging because of three fundamental issues: the scarcity of labeled data and text supervision, different levels of downstream tasks, and the conceptual gaps between domains. In this work, to address these issues, we leverage multi-modal prompt learning to effectively adapt pre-trained GNN to downstream tasks and data, given only a few semantically labeled samples, each with extremely weak text supervision. Our new paradigm embeds the graphs directly in the same space as the Large Language Models (LLMs) by learning both graph prompts and text prompts simultaneously. To accomplish this, we improve state-of-the-art graph prompt method, and then propose the first graph-language multi-modal prompt learning approach for exploiting the knowledge in pre-trained models. Notably, due to the insufficient supervision for fine-tuning, in our paradigm, the pre-trained GNN and the LLM are kept frozen, so the learnable parameters are much fewer than fine-tuning any pre-trained model. Through extensive experiments on real-world datasets, we demonstrate the superior performance of our paradigm in few-shot, multi-task-level, and cross-domain settings. Moreover, we build the first CLIP-style zero-shot classification prototype that can generalize GNNs to unseen classes with extremely weak text supervision. The code is provided in the supplementary materials.

Preprint.

1 Introduction

Graphs are constructed from real scenarios, but GNNs, optimized according to numerical labels, still do not *understand* what a label represents in the real world. To solve the issue of predetermined numerical categories, Contrastive Language-Image Pre-training (CLIP) [66] leverages natural language supervision by jointly training an image encoder and a text encoder in the same embedding space at scale. CLIP has demonstrated the ability to train high-quality, generalizable vision models [66, 29, 44, 101], which can adapt to diverse downstream tasks. Similar frameworks have been successfully extended to other modalities, including video [84, 4], 3D images [93, 22, 23], speech [68] and audio [20, 79], consistently demonstrating that alignment with text enhances the transferability of encoders. As for graphs, so far, such graph-text alignment has only been explored in the molecular domain [59, 57, 15, 70, 52] and on text-attributed graphs [78, 46, 32, 86, 85], where the paired graph-text data is relatively sufficient for pre-training.

However, extending this paradigm to more general graph data poses significant challenges due to three facts. First, compared with language or vision data, graph data is very scarce and the text supervision is extremely weak [51, 6, 60] for pre-training models. Specifically, besides the number of samples being much smaller than images, many graph datasets are used for classification, where the label names consist of only a few tokens. Second, the task space of graph data could be on node-level [47, 100, 18, 56], edge-level [69, 48, 2], and graph-level [64, 16, 103, 102, 49, 50]. Third, in general, language tokens and visual objects retain the same conceptual meaning across different distributions, but the same graph structure may have distinct interpretations in different domains [17].

Due to the above three reasons, jointly pre-training graph and text encoders is impractical for graph data with extremely weak text supervision. Fortunately, we can deal with the two modalities separately for pre-training: large language models have already been extensively pre-trained, and tremendous efforts have been devoted to pre-train GNNs through self-supervision [83, 26, 54, 38]. However, even with a pre-trained graph model, effectively adapting it to both the semantic embedding space for text alignment and diverse downstream tasks remains non-trivial. This raises a critical question:

How to adapt pre-trained GNNs to the semantic embedding space given limited downstream data, i.e., few available samples and weak text supervision?

This paper aims to answer this question based on the following observations: (1) Semantic text embedding spaces do not necessarily result from joint pre-training. In fact, the embedding spaces of encoder LLMs are inherently semantic and high-quality, as LLMs are trained on massive text data and demonstrate strong reasoning performance. (2) When the downstream data are limited, prompt learning [45, 25, 92, 42, 108] provides a better option than fine-tuning as much fewer parameters not only makes the optimization more efficient but also requires less resource than computing the gradient of a large model. Notably, some works have explored prompt learning for better alignment and obtained improvement in vision prediction [105, 39]. Inspired by these two observations, we propose a prompting-based paradigm with an LLM that, while keeping the parameters of both GNN and LLM frozen, aligns the GNN representations in the LLM's semantic embedding space.

When attempting to adapt the representation from one modality to another, solely prompting a single modality could be sub-optimal, as it limits the adjustment to downstream tasks in the other modality [39]. To this end, we propose Multi-modal Prompt Learning for Graph Neural Networks (Morpher). Given a pre-trained GNN and few-shot semantically labeled graph data with weak text supervision, we assume zeroth-order access to a pre-trained LLM. Then, to leverage its high-quality semantic embedding space, Morpher connects and aligns the graph embeddings to it through prompting on both modalities with a cross-modal projector. Nonetheless, designing such a paradigm is more challenging than vision-language models. First, we lack jointly pre-trained encoders for the two modalities; instead, we only have two encoders whose embedding dimension is possibly different, pre-trained independently in each modality. Second, determining how to prompt the graph modality is non-trivial and remains a trending research topic. Third, the downstream data for GNN usually have much fewer labeled classes and labeled samples than V-L models, and the text supervision is extremely weak. Our contributions towards tackling these challenges are summarized as follows:

- Theoretically, we analyze that, in many cases, state-of-the-art graph prompt [73] is unable to learn good representations of the downstream data. We show that the optimization of the graph prompt is restricted by design. From the theoretical findings, we further improve state-of-the-art graph prompt according to the attention mechanism to prevent failure in optimization.

- To connect and adapt the pre-trained GNN with LLM effectively with extremely weak text supervision, we propose Morpher, a graph-text multi-modal prompt learning paradigm. To the best of our knowledge, this is the first approach to align the representations of GNN and LLM without fine-tuning any of their parameters.
- With extremely weak text supervision, we demonstrate the effectiveness of our improved graph prompt and Morpher on real-world datasets under few-shot, multi-task, and cross-domain settings. To show that GNN learns language dependency through Morpher, we present the first CLIP-style zero-shot generalization prototype where the GNN can correctly predict unseen classes.

2 Background

We use calligraphic letters (e.g., $\mathcal{A}$) for sets, and specifically $\mathcal{G}$ for graphs. We use bold capital letters for matrices (e.g., $\mathbf{A}$). For matrix indices, we use $\mathbf{A}(i,j)$ to denote the entry in the i^{th} row and the j^{th} column. Additionally, $\mathbf{A}(i,:)$ returns the i^{th} row in $\mathbf{A}$.

Graph Neural Networks. We use $\mathcal{G} = (\mathbf{A}, \mathbf{X})$ to denote a graph with node set $\mathcal{V}$ and edge set $\mathcal{E}$, where $\mathbf{A} \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{V}|}$ is the adjacency matrix and $\mathbf{X} \in \mathbb{R}^{|\mathcal{V}|\times d}$ is the node feature matrix. $\mathbf{A}(u,v) = 1$ if there is an edge connecting u and v; otherwise $\mathbf{A}(u,v) = 0$. A Graph Neural Network $f_\phi^g(\cdot)$ with hidden dimension d_g encodes $\mathcal{G}$ into the embedding space: $f_\phi^g(\mathcal{G}) \in \mathbb{R}^{|\mathcal{V}|\times d_g}$, which could preserve both feature and structure information of $\mathcal{G}$. The extracted embeddings $f_\phi^g(\mathcal{G})$ can be used for various downstream tasks such as classification.

Few-shot Prompt Learning. Prompt learning adds learnable tokens to the downstream data and provides a powerful alternative to fine-tuning when the labeled downstream data is scarce. Prompt learning for encoders was first used in NLP. Let $f_\phi^t(\cdot)$ denote the LLM encoder with embedding dimension d_t. For a series of input tokens $\{x_k\}_{k=1}^K$, the LLM encoder embeds it as a matrix $\mathbf{X}_t = f_\phi^t(\{x_k\}_{k=1}^K) \in \mathbb{R}^{K\times d_t}$, then aggregates the representation to a vector $aggre(\mathbf{X}_t) \in \mathbb{R}^{1\times d_t}$ for downstream tasks. Prompt learning initializes a tunable matrix $\mathbf{P}_\theta^t \in \mathbb{R}^{n_t\times d_t}$, where n_t denotes the number of text prompt tokens. Then, this tunable matrix is concatenated with the input tokens' embeddings to form a single matrix $[\mathbf{P}_\theta^t; \mathbf{X}_t]_{dim=0} \in \mathbb{R}^{(K+n_t)\times d_t}$, and the aggregated vector for downstream tasks becomes $aggre([\mathbf{P}_\theta^t; \mathbf{X}_t]_{dim=0})$. In practice, we can train the model to minimize the loss function for downstream tasks, with only the prompt parameters $\mathbf{P}_\theta^{\mathbf{t}}$ being updated.

Our Problem Set-up. Given a pre-trained GNN $f_\phi^g(\cdot)$ with embedding dimension d_g and a pre-train LLM encoder $f_\phi^t(\cdot)$ with embedding dimension d_t. Without loss of generality, we assume the downstream task is graph-level classification, as we will show that the other types of GNN tasks can be reformulated as graph classification. For L-shot graph classification, we are given limited text-labeled pairs $\{(\mathcal{G}_i, t_c)\}_{i=1}^L$ for each class c. Each text label t_c consists of only a few tokens. Assuming $\mathcal{T}$ is the set of all text labels t_c, we are provided a set of test graphs $\{\mathcal{G}_j\}_{j=1}^{L_{test}}$. Using the pre-trained GNN and LLM, we want to correctly predict the text label $t_j \in \mathcal{T}$ for each test graph $\mathcal{G}_j$.

3 Revisiting and Improving Prompt as Graphs

Unlike prompting text data (which appends learnable text tokens to the original text sequence) and prompting image data (which pads a learnable image area above the original image), prompting graph data presents a significant challenge due to the non-euclidean nature of graphs. The recent pioneering work [73] designs the graph prompt still as a graph, then inserts it into the original graph by computing the inner-connections within the prompt graph and the cross-connections between the prompt graph and the original graph. An advantage of prompting at the graph level is that *the downstream tasks of GNN can be reformulated into graph-level tasks*. For the node classification task, we can induce the γ-ego-graph of each node by extracting the subgraph within a pre-defined distance γ. Then, we treat the node label as the induced ego-graph label. Similarly, for the edge classification task, we can extract a subgraph for each edge by extending the node pair to their γ

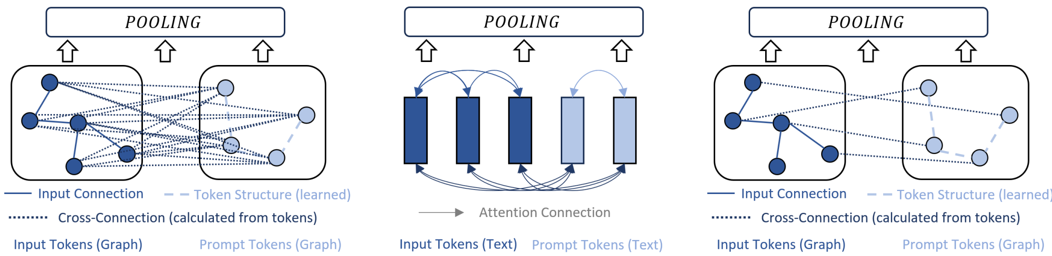

Figure 2: Cross-connections overwhelm inner-connections in current graph prompt design, which may be unstable during training (left); attention in NLP where $3 \times 2 = 6$ cross-connections and $3 + 1 = 4$ inner-connections are balanced (middle); and our balanced graph prompt design (right). **The cross-connections between input and prompt should have a consistent scale with the input connections.**

distance neighborhood, and use the edge label as the induced graph label. By inducing subgraphs, we can reformulate node-level and edge-level downstream tasks to graph-level.

Current Graph Prompt Design. To prompt a graph $\mathcal{G}$, each prompt token is a new node. Let n_g denote the number of prompt tokens and $\mathcal{P} = \{p_i\}_{i=1}^{n_g}$ denote the set of prompt tokens. The graph prompt is formulated by a tunable matrix $\mathbf{P}_\theta^g \in \mathbb{R}^{n_g \times d}$, where d is the node feature dimension of graph $\mathcal{G}$. In other words, each row vector $\mathbf{P}_\theta^g(i,:)$ is the feature of the prompt token p_i. Then, the mechanism to prompt a graph $\mathcal{G} = (\mathbf{A}, \mathbf{X})$ with n nodes and d feature dimension is as follows [73].

- Compute inner-connections to construct the prompt graph $\mathcal{G}_p = (\mathbf{A}_p, \mathbf{X}_p)$. For the feature matrix, we directly set $\mathbf{X}_p = \mathbf{P}_\theta^g$. For two prompt tokens p_i and p_j, the prompt graph will have an edge between them if and only if the dot product of their features is larger than a threshold. In other words, $\mathbf{A}_p(i,j) = 1 \iff \sigma(\mathbf{P}_\theta^g(i,:)\mathbf{P}_\theta^g(j,:)^\top) > \delta_{inner}$, where $\sigma(\cdot)$ is the sigmoid function.
- Compute cross-connections to insert the prompt graph $\mathcal{G}_p$ into the original input graph $\mathcal{G}$. Similarly, for $x_i \in \mathcal{G}$ and $p_j \in \mathcal{G}_p$, there is an edge between them if and only if $\sigma(\mathbf{X}(i,:)\mathbf{P}_\theta^g(j,:)^\top) > \delta_{cross}$.
- Construct the prompted graph (i.e., manipulated graph) $\mathcal{G}_m = (\mathbf{A}_m, \mathbf{X}_m)$.The overall adjacency matrix $\mathbf{A}_m \in \mathbb{R}^{(n+n_g)\times(n+n_g)}$ is constructed from the original adjacency matrix $\mathbf{A}$, the inner edges $\mathbf{A}_p$ and the cross edges. The overall node feature matrix is concatenated from the prompt token features and the original input node features: $\mathbf{X}_m = [\mathbf{P}_\theta^g; \mathbf{X}]_{dim=0} \in \mathbb{R}^{(n+n_g)\times d}$.

Issues associated with the current design. Since not all the GNN backbones can take edge weights [14], the cross-connections in a manipulated graph are discrete[1], thresholded by δ_{cross}. However, the input node features of most real-world datasets are sparse, resulting from the construction process [88, 61, 10]. As shown in Table 6, $||\mathbf{X}(i,:)||_1$ is typically 1. As the initialization of each token feature $\mathbf{P}_\theta^g(i,:)$ is close to $\vec{\mathbf{0}}$, for any node i and token p_j, the dot products $\mathbf{X}(i,:)\mathbf{P}_\theta^g(j,:)^\top$ is close to 0 , and the sigmoid value is very close to 0.5. Consequently, if we want the graph prompt to have cross-connections, we must set $\delta_{cross} < 0.5$. Then, as the sigmoid values are close to 0.5, the cross-connections will be dense, i.e., almost every node in the original graph is connected with every node token in the prompt graph. For two different graphs $\mathcal{G}_1$ and $\mathcal{G}_2$ in the same task, the prompt graph $\mathcal{G}_p$ is identical. Since the GNNs work by aggregating the node features, their embeddings $f_\phi^g(\mathcal{G}_1)$ and $f_\phi^g(\mathcal{G}_2)$ are approximately the same because the features in the prompt graph overwhelm the features in the original graphs due to the dense cross-connections. Then, according to a trivial lemma from optimization, even if $\mathcal{G}_1$ and $\mathcal{G}_2$ have different labels, the task head classifier cannot be trained to distinguish them[2].

Lemma 3.1. *For any classifier $c(\cdot)$, if the identical feature $\mathbf{x}$ has label distribution $p(\cdot)$, then the optimal classification for cross-entropy loss is $Pr(c(\mathbf{x}) = y) = p(y)$. From this, if two graphs have similar embedding but different labels, GNN training may not converge. (Proof in Appendix A)*

Improved Graph Prompt Design. The issue of the current graph prompt is rooted in the imbalance of original connections in the input graph and cross-connections between input and prompt, as shown

[1] In official implementation of [73], adjacency matrices are discrete: either 0 or 1 for each entry.

[2] In fact, when executing the official implementation of [73] on Cora, the training loss does not decrease. Similar problems have been observed by another work [96].

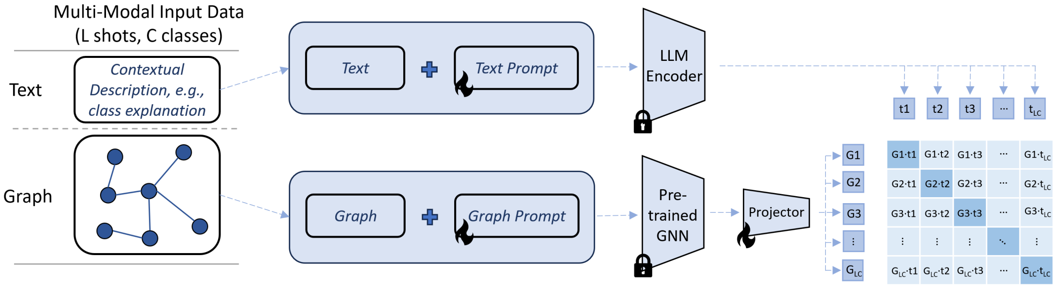

Figure 3: Similar to CLIP backbone, Morpher adapts the graph representations to semantic space through multi-modal prompt learning, even if the GNN and LLM are not jointly trained and are kept frozen.

in Figure 2 (left). We also visualize the standard NLP attention mechanism [77] in Figure 2 (middle). After we prepend a sequence of text prompt tokens $\{p_i^t\}$, the features of the text prompt tokens will be densely aggregated to the features of the original text tokens. In other words, the "cross-connection" between the text prompt sequence and the input sequence is dense. However, such a dense connection does not cause the prompt feature to overwhelm the input, because the features in the input sequence are also aggregated in a dense manner. Inspired by this, the number of cross-attention between input and prompt should approximate the number of input connections. Since the connection of a graph dataset is often sparse, we should also constrain the cross-connections between the prompt graph and the input graph to be sparse as well.

Nonetheless, "sparse" is a wide concept to implement: if the cross-connections are too dense, the prompt graph will dominate the input graph; but if the cross-attention is too sparse, the prompt graph will be limited to manipulating the input graph. We deem that a balance could be achieved by approximately equalizing the number of cross-connections with that of connections in the input graph, i.e., n_e. Therefore, we set the number of cross-connections to at most n_e by connecting each node in the input graph with at most $\lfloor \frac{n_e}{a} \rfloor$ prompt tokens. Then, we can safely use a small δ_{cross} and cosine similarity $\frac{\mathbf{X}(i,:)\cdot\mathbf{P}_\theta^g(j,:)^\top}{\|\mathbf{X}(i,:)\|_2\|\mathbf{P}_\theta^g(j,:)\|_2}$ instead of $\sigma(\mathbf{X}(i,:)\mathbf{P}_\theta^g(j,:)^\top)$ to calculate the cross-connections. We demonstrate that our improved graph prompt works better in the later experiments.

4 Multi-modal Prompt Learning for GNNs

To adapt the GNN embeddings to the LLM's semantic embedding space and leverage the additional weak supervision provided by the text associated with graph labels, we explore the potential of multi-modal prompt learning for both graphs and language. This approach is motivated by the intuition that only prompting on the graph data may limit the flexibility to adjust the LLM representation space. The overall paradigm of Morpher is illustrated in Figure 3. Given the data $\{(\mathcal{G}_i, t_i)\}_{i=1}^{L\times C}$, we aim to align graph embedding $\mathrm{readout}(f_\phi^g(\mathcal{G}_i))$ with $\mathrm{readout}(f_\phi^t(\mathrm{Tokenize}(t_i)))$. Yet one direct issue is that, $\mathrm{readout}(f_\phi^g(\mathcal{G}_i)) \in \mathbb{R}^{1\times d_g}$ and $\mathrm{readout}(f_\phi^t(\mathrm{Tokenize}(t_i))) \in \mathbb{R}^{1\times d_t}$ may have distinct dimensions. To address this issue, we adopt a cross-modal projector that learns to map the graph embedding space to the text embedding space. For an input d_g-dimensional graph embedding $\mathbf{v}$, the projector maps it to a vector $\widetilde{\mathbf{v}}$ in the d_t-dimensional text embedding space:

$$\widetilde{\mathbf{v}} = \mathrm{Proj}_\theta(\mathbf{v}) := \tanh(\mathbf{W}\mathbf{v} + \mathbf{b}) \in \mathbb{R}^{1\times d_t} \tag{1}$$

As discussed in Sections 2 and 3, we introduce the text prompt $\mathbf{P}_\theta^t \in \mathbb{R}^{n_t\times d_t}$ with n_t text prompt tokens and the graph prompt $\mathbf{P}_\theta^g \in \mathbb{R}^{n_g\times d}$ with n_g graph prompt tokens. Let $\psi_g(\cdot, \mathbf{P}_\theta^g)$ be the graph prompting function, e.g., given any graph $\mathcal{G}$, the manipulated graph $\mathcal{G}_m = \psi_g(\mathcal{G}, \mathbf{P}_\theta^g)$.

Let $\omega_t(\cdot, \mathbf{P}_\theta^t)$ be the prompted text embedding given input text t. For the text prompt methods we choose, the prompted embedding is

$$\omega_t(t, \mathbf{P}_\theta^t) = [\mathbf{P}_\theta^t; f_\phi^t(\mathrm{Tokenize}(t))]_{dim=0} \in \mathbb{R}^{(len(\mathrm{Tokenize}(t))+n_t)\times d_t} \tag{2}$$

Let $\omega_g(\cdot, \mathbf{P}_\theta^g)$ be the prompted graph embedding given input graph $\mathcal{G}$, then we have:

$$\omega_g(\mathcal{G}, \mathbf{P}_\theta^g) = f_\phi^g(\mathcal{G}_m) = f_\phi^g(\psi_g(\mathcal{G}, \mathbf{P}_\theta^g)) \in \mathbb{R}^{(n+n_g)\times d_g} \tag{3}$$

For the whole prompted text and the whole prompted graph of the sample $(\mathcal{G}_i, t_i)$, we apply readout (e.g., mean-pooling, max-pooling, etc.) to get their embedding:

$$\mathbf{h}_i^t = \text{readout}(\omega_t(t_i, \mathbf{P}_\theta^t)) \in \mathbb{R}^{1\times d_t},\ \mathbf{h}_i^{\mathcal{G}} = \text{readout}(\omega_g(\mathcal{G}_i, \mathbf{P}_\theta^g)) \in \mathbb{R}^{1\times d_g} \tag{4}$$

For the given data $(\mathcal{G}_i, t_i)$, we compute the normalized embedding of prompted $\mathcal{G}_i$ and project it to the text embedding space through the projector:

$$\mathbf{z}_{norm,i}^{\mathcal{G}} = \frac{\mathbf{h}_i^{\mathcal{G}}}{||\mathbf{h}_i^{\mathcal{G}}||_2} = \frac{\text{readout}(\omega_g(\mathcal{G}_i, \mathbf{P}_\theta^g))}{||\text{readout}(\omega_g(\mathcal{G}_i, \mathbf{P}_\theta^g))||_2}, \quad \mathbf{z}_i^{\mathcal{G}} = \text{Proj}_\theta(\mathbf{z}_{norm,i}^{\mathcal{G}}) \tag{5}$$

For the text embeddings, since for limited data the set $\mathcal{T} = \{t_i\}_{i=1}^C$ may contain texts that are semantically close as discussed in Appendix B.2, we extract a subspace in the text embedding space by normalizing the embedding as follows. We further normalize the text embeddings to the unit sphere, as standard practice in NLP.

$$\mu = \frac{1}{L}\sum_{i=1}^{L}\mathbf{h}_i^t, \quad \mathbf{h}_{norm,i}^t = \mathbf{h}_i^t - \mu \tag{6}$$

$$\mathbf{z}_i^t = \frac{\mathbf{h}_{norm,i}^t}{||\mathbf{h}_{norm,i}^t||_2} = \frac{\text{readout}(\omega_t(t_i, \mathbf{P}_\theta^t)) - \mu}{||\text{readout}(\omega_t(t_i, \mathbf{P}_\theta^t)) - \mu||_2} \tag{7}$$

Finally, we use the in-batch similarity-based contrastive loss [99] to train text prompts, graph prompts, and the projector as shown below, to adapt the pre-trained GNN representations to LLM.

$$\mathcal{L}_{G\to T} = -\frac{1}{B}\sum_{i=1}^{B}\log\frac{\exp(\mathbf{z}_i^{\mathcal{G}}\cdot\mathbf{z}_i^t/\tau)}{\sum_{j=1}^{B}\exp(\mathbf{z}_i^{\mathcal{G}}\cdot\mathbf{z}_j^t/\tau)} \tag{8}$$

During inference stage, for an input graph $\mathcal{G}_i$ and text label candidates $\mathcal{T} = \{t_i\}_{i=1}^C$, we compute the embedding $\mathbf{z}_i^{\mathcal{G}} = \text{Proj}_\theta(\frac{\text{readout}(\omega_g(\mathcal{G}_i, \mathbf{P}_\theta^g))}{||\text{readout}(\omega_g(\mathcal{G}_i, \mathbf{P}_\theta^g))||_2})$ using trained $\mathbf{P}_\theta^g$ and $\text{Proj}_\theta(\cdot)$. Then, we compute $\mathbf{z}_i^t$ as Equation 7 and 8. Finally, $\mathcal{G}_i$ will be classified to associate with text label $\arg\max_{1\le i\le C}(\mathbf{z}_i^{\mathcal{G}}\cdot\mathbf{z}_i^t)$.

5 Experiments

We evaluate our Morpher and the improved graph prompt through extensive experiments. In particular, we show that, compared to state-of-the-art baseline methods, they both more effectively adapt pre-trained GNNs to the specific downstream classification task, and introducing the text modality brings Morpher additional advantages over others. We use RoBERTa [53] as the LLM encoder for Morpher in the main experiments. We also validate the performance of Morpher with ELECTRA [8] and DistilBERT [67] in section 5.6 and Appendix C.3.

Datasets. We use real-world graph datasets from PyTorch Geometric [14], including one molecular dataset MUTAG [61]; two bioinformatic datasets ENZYMES and PROTEINS [3]; one computer vision dataset MSRC_21C [63]; three citation network datasets Cora, CiteSeer and PubMed [88]. We use real-world class names as text labels. The text supervision is extremely weak, as each text label contains no more than five words. More details are summarized in Appendix B.

Pre-trained algorithms and GNN backbones. To pretrain GNNs for evaluation, we adopt GraphCL [89] and SimGRACE [82] to pre-train three widely used GNN backbones: GCN [41], GAT [90] and GraphTransformer (GT) [40]. Additionally, in Appendix C.4, we verify the effectiveness of our methods on GNNs pre-trained using GraphMAE [24] and MVGRL [21], two other representative GNN self-supervised learning algorithms. For each dataset, to pre-train GNNs, we leverage self-supervised learning methods on all the graphs without any label information.

Baselines and metrics. We compare our methods with the following baselines: (1) training a GNN from scratch supervised by few-shot data (*"supervised"*); (2) fine-tuning a task head together with pre-trained GNN (*"fine-tune"*). We allow GNNs to be tunable for "*supervised*" and "*fine-tune*"; (3) state-of-the-art graph prompting algorithms: All-in-one (*"AIO"*) [73], which is the only graph prompting algorithm that supports multiple tasks in node-level, edge-level and graph-level to the best of our knowledge; GPF-plus [11] which prompt on graph features and Gprompt [55] which is based on subgraph similarity. We use accuracy and weighted F1 as classification performance metrics.

Table 1: Few-shot graph classification performance (%). IMP (%): the average improvement (absolute value) compared to the **best result** among all the baseline methods.

Training schemes	GNN pretraining	MUTAG		ENZYMES		PROTEINS		MSRC_21C	
		Acc	F1	Acc	F1	Acc	F1	Acc	F1
Supervised	N/A + GCN	66.00	66.67	16.67	8.68	65.89	60.77	38.85	35.32
	N/A + GAT	66.00	65.69	16.45	4.65	64.75	64.08	41.14	39.86
	N/A + GT	66.66	66.26	15.62	4.22	62.81	57.12	38.28	41.62
Pre-train + Fine-tune	GraphCL+GCN	70.00	70.23	17.91	11.82	65.89	61.23	40.00	43.89
	GraphCL+GAT	70.00	69.73	17.91	10.46	65.16	63.92	44.57	45.74
	GraphCL+GT	68.00	67.81	17.70	8.99	63.28	56.41	41.71	43.73
	SimGRACE+GCN	66.67	67.27	17.29	8.78	66.82	64.70	40.57	43.84
	SimGRACE+GAT	70.67	69.10	16.87	7.18	65.42	63.65	42.85	42.37
	SimGRACE+GT	69.33	69.77	16.24	6.08	65.98	62.31	39.42	40.78
AIO [73]	GraphCL+GCN	64.67	39.27	17.50	4.97	61.35	44.93	3.59	10.09
	GraphCL+GAT	64.67	39.27	17.50	4.97	59.21	37.19	14.37	3.11
	GraphCL+GT	73.33	72.06	18.33	9.09	40.79	28.97	17.96	8.30
	SimGRACE+GCN	64.67	39.27	16.04	4.61	67.42	60.87	34.73	18.16
	SimGRACE+GAT	64.67	39.27	16.04	4.61	59.21	37.19	7.78	1.79
	SimGRACE+GT	36.00	27.26	17.50	8.15	50.56	49.34	32.34	15.13
GPF-plus [11]	GraphCL+GCN	68.67	67.27	16.88	15.48	64.75	61.45	47.42	29.02
	GraphCL+GAT	68.67	62.84	16.45	13.23	65.89	60.07	47.42	26.28
	GraphCL+GT	69.33	67.87	18.12	15.56	59.66	37.37	41.71	21.35
	SimGRACE+GCN	65.33	39.52	18.96	15.83	65.16	58.80	45.71	23.32
	SimGRACE+GAT	69.33	66.72	18.54	12.58	63.28	53.50	42.85	21.40
	SimGRACE+GT	70.00	67.31	17.91	14.69	64.83	52.97	34.13	20.13
Gprompt [55]	GraphCL+GCN	73.33	66.93	17.91	8.44	61.01	60.01	1.80	0.21
	GraphCL+GAT	64.67	62.63	17.08	14.18	50.56	50.55	1.80	0.22
	GraphCL+GT	70.67	70.02	17.91	9.64	63.28	58.65	1.80	0.21
	SimGRACE+GCN	65.33	39.52	17.29	14.48	52.70	52.68	1.80	0.21
	SimGRACE+GAT	67.33	65.88	16.25	11.31	59.10	58.72	1.80	0.21
	SimGRACE+GT	73.33	67.84	16.87	13.54	64.75	62.37	1.80	0.223
Improved AIO (Ours)	GraphCL+GCN	77.33	77.74	18.13	11.98	65.89	65.97	42.85	45.91
	GraphCL+GAT	74.67	75.51	18.33	11.26	65.76	66.05	46.85	51.39
	GraphCL+GT	74.67	74.67	19.16	9.04	68.12	68.18	42.85	43.54
	SimGRACE+GCN	68.00	69.01	17.91	9.02	66.82	66.40	44.57	49.24
	SimGRACE+GAT	77.33	77.20	18.75	9.39	66.91	65.49	45.14	42.31
	SimGRACE+GT	71.33	72.06	18.95	11.25	68.59	68.84	40.57	42.82
Morpher (Ours)	GraphCL+GCN	78.67	78.09	20.41	15.20	67.47	66.40	45.14	49.62
	GraphCL+GAT	79.33	79.15	23.12	18.01	70.89	70.30	50.85	54.48
	GraphCL+GT	76.00	76.51	19.58	13.28	73.53	72.48	45.71	48.41
	SimGRACE+GCN	69.33	70.27	19.79	14.94	67.10	66.15	45.71	51.24
	SimGRACE+GAT	78.00	77.65	20.21	16.27	68.12	67.26	45.71	51.13
	SimGRACE+GT	74.00	74.84	19.16	14.29	71.76	71.75	44.00	48.16
IMP of ImprovedAIO		2.00 $\uparrow$	5.01 $\uparrow$	0.52 $\uparrow$	4.41 $\downarrow$	2.01 $\uparrow$	4.37 $\uparrow$	0.28 $\downarrow$	2.50 $\uparrow$
IMP of Morpher		4.00 $\uparrow$	6.73 $\uparrow$	2.36 $\uparrow$	0.60 $\uparrow$	4.81 $\uparrow$	6.61 $\uparrow$	2.66 $\uparrow$	7.14 $\uparrow$

5.1 Few-shot Learning

We investigate the ability of our improved graph prompt ("*ImprovedAIO*") and Multimodal prompt ("*Morpher*") to adapt frozen pre-trained GNNs using few-shot data. We focus on graph-level classification here and will further investigate the few-shot learning ability at other task levels in Section 5.2. Our few-shot learning setting is more challenging than existing works [73, 72] as we only allow no more than 10 labeled training and validation samples for each class. The results are shown in Table 1, where we report the average performance of 5 runs and calculate the absolute average improvement of our methods. From the results, given the same pre-trained GNN, our ImprovedAIO outperforms all the existing baseline methods except for ENZYMES F1 and MSRC_21C accuracy. Yet the performance of our ImprovedAIO on ENZYMES F1 and MSRC_21C accuracy is clearly better than those of the original AIO. Our Morpher can achieve an absolute accuracy improvement of 0.60% to 7.14% over the baselines across all datasets. Supervised by very limited labeled data, training a GNN from scratch is sub-optimal. Passing a GNN pre-trained on the dataset and fine-tuning it with a task head achieves sub-optimal but better results as the pre-trained GNN learns generalizable representations over the dataset through self-supervised learning. To mitigate the gap between the pre-training task and downstream tasks, AIO [73] proposes to learn graph prompts for downstream data. However, as we discussed in Section 3, when the node features are sparse vectors, the optimization would fail. Using the official implementation of AIO, we observe that the loss value tends to fluctuate, and the performance of AIO is usually even worse than supervised training. By restricting the cross-connections, our ImprovedAIO becomes more stable and constantly outperforms the fine-tuning baseline. Compared to the aforementioned methods, Morpher demonstrated superior performance due to its capability to adapt both graph and language representation spaces dynamically.

5.2 Morpher Supports Multiple-level Tasks

Inherited from AIO, our ImprovedAIO and Morpher also support adaptation to downstream tasks at node-level and edge-level, because they can be reformulated into graph-level tasks as discussed in Section 3. We demonstrate the performance of node classification and link prediction on Cora and CiteSeer. For node classification, we reformulate it to graph classification by inducing an ego-graph with 10 to 30 nodes centered at the node to classify. Each ego-graph has the same label as the center node. For edge classification, we randomly sample 200 edges from the graph, then create 200 negative samples by replacing one node in each edge. We label each graph according to whether it is a positive or negative sample.

Table 2: Node-level, edge-level performance.

Dataset		Cora		CiteSeer	
Tasks	Methods	Acc	F1	Acc	F1
Node Level	Supervised	52.83	47.73	63.91	64.82
	Fine-tune	56.37	55.04	64.87	66.42
	AIO [73]	14.69	7.10	18.93	6.92
	ImprovedAIO	58.46	55.10	66.44	66.53
	Morpher	61.26	62.36	68.20	68.56
Edge Level	Supervised	51.78	50.62	52.14	50.81
	Fine-tune	52.50	51.00	52.50	51.12
	AIO [73]	50.00	33.33	50.00	33.33
	ImprovedAIO	54.64	54.57	53.92	53.55
	Morpher	55.71	55.05	55.35	55.05

We use GraphCL+GCN to pre-train the GNN and report the mean performance in Table 2. The results are consistent with graph-level performance, where ImprovedAIO and Morpher outperform existing methods, with Morpher achieving slightly better performance than ImprovedAIO. Additionally, the training of the original AIO fails on both datasets due to the sparse node feature vectors.

5.3 Domain Transfer

A key problem of the graph foundation model is whether we can adapt the pre-trained models to other data domains. Here, we explore the potential of using Morpher for such adaptation. We pre-train GNNs on ENZYMES or CiteSeer datasets, then test the classification performance on MUTAG and PubMed and report the results in Table 3. We unify the pre-train feature dimension with the downstream feature dimension by padding zeros or SVD reduction. From the results, Morpher demonstrates the best transferability, followed by ImprovedAIO. Also, compared to the results on MUTAG in Table 1, all three methods have worse performances, because the GNNs are pre-trained on other datasets instead of MUTAG.

Table 3: Domain Transfer Performance.

Target Domain		MUTAG		PubMed	
Target Task		graph-level		node-level	
Source	Methods	Acc	F1	Acc	F1
ENZYMES (graph-level)	Fine-tune	68.00	55.04	47.57	36.07
	ImprovedAIO	70.67	64.07	50.28	50.51
	Morpher	72.67	73.29	54.42	53.96
CiteSeer (node-level)	Fine-tune	71.33	62.19	48.71	40.66
	ImprovedAIO	74.00	73.76	52.57	51.29
	Morpher	76.67	77.04	58.29	57.54

5.4 Zero-shot Classification Prototype

An advantage of adapting pre-trained GNNs to the semantic embedding space is that GNNs might be empowered to “reasoning”. Here, we conduct a novel experiment that generalizes GNN to an unseen class. Since no real-world data is available for this setting, we synthetically create three datasets, ZERO-Cora, ZERO-CiteSeer, and ZERO-PubMed, all from real-world connections. We aim to simulate a citation network with two research areas and an interdisciplinary research area in between. For each citation network, we randomly sample 120 nodes and induce their 2-hop ego-graphs, then replace the node features in 10 ego-graphs with $[1, 0]$ and another 10 ego-graphs

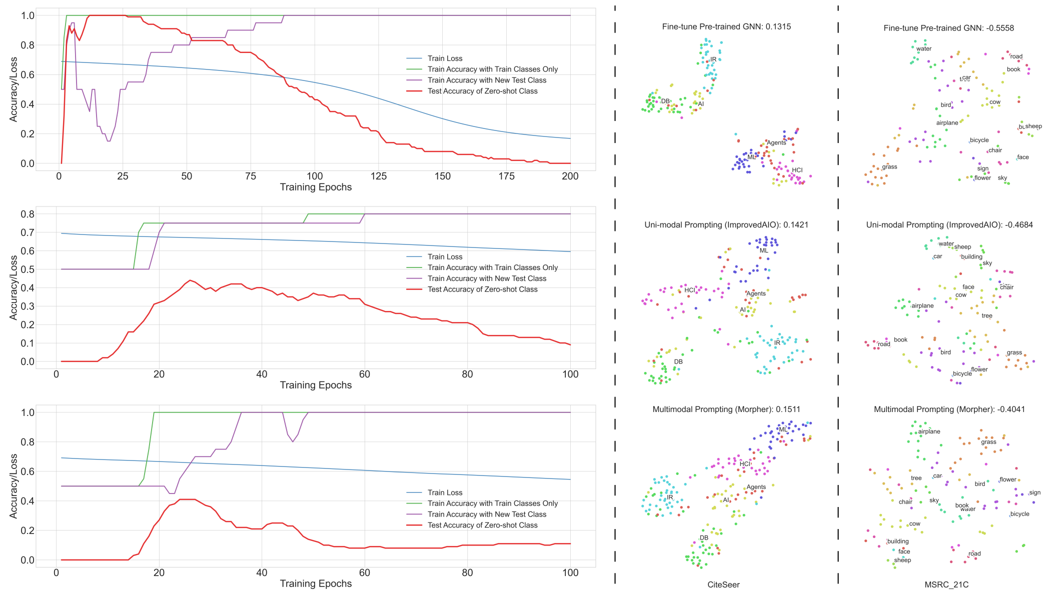

Figure 4: Results of novel class generalization (left); t-SNE embedding plots on CiteSeer, MSRC_21C (right). Train accuracy with train classes only is the accuracy of predicting the training graphs from the two training classes. Train accuracy with new test classes is the accuracy of predicting the training graphs from all three classes. Test Accuracy of zero-shot class is the accuracy of predicting the testing graphs from all three classes.

with $[0, 1]$ to construct 20 training graph samples. For the remaining ego-graphs, we uniformly randomly replace the node features with $[1, 0]$ and $[0, 1]$ to construct 100 testing graph samples. We assign text labels of the first research area (e.g., "biology") to the $[1, 0]$ training graphs, the second research area (e.g., "informatics") to the $[0, 1]$ training graphs, and the interdisciplinary area (e.g., "bioinformatics") to the testing graphs. Intuitively, the nodes with feature $[1, 0]$ are papers in the first area, and other nodes with feature $[0, 1]$ are in the second area, which makes the datasets rational.

For each dataset, using GraphCL+GCN, we pre-train GNNs on all graphs. Then, we train Morpher on the training graphs, only knowing the text labels of the two training classes. Since we do not have validation data in zero-shot learning, we report the results of each epoch in Figure 4 (left). We observe that, while Morpher quickly adapts the GNN to downstream training data, the CLIP-like framework can predict the graphs in the novel class with good accuracy (red curve). Moreover, the training samples can be classified correctly from training and novel classes. Before the training overfits, there is a period when Morpher can distinguish all the graphs from the training and novel classes with high accuracy.

Such zero-shot novel-class generalization ability validates Morpher's alignment between graph embeddings and text embeddings. When Morpher is trained on two classes of graphs with text labels of biology and informatics, a graph-in-the-middle will be classified as text-in-the-middle: bioinformatics, even if "bioinformatics" is an unseen label. The correspondence of in-the-middle graphs and texts shows the benefit and novelty of Morpher. To the best of our knowledge, this is the first zero-shot classification prototype that generalizes GNN to unseen classes.

5.5 Efficiency and Embedding Analysis

Without fine-tuning the GNN or LLM, the prompt-based methods have better parameter efficiency. As shown in Figure 5 (left), our ImprovedAIO and Morpher require similar numbers of parameters with AIO [73], which is 0.032% to 0.46% compared to either tune the LLM (RoBERTa) or GNN (GCN). Due to such parameter efficiency, our methods learn better graph representations given few-shot data. We visualize the graph embeddings of CiteSeer and MSRC_21C in Figure 4 and calculate the silhouette score, a metric for cluster quality ($\uparrow$) ranged in $[-1, 1]$. It turns out that our multimodal prompting leads to better adaptation.

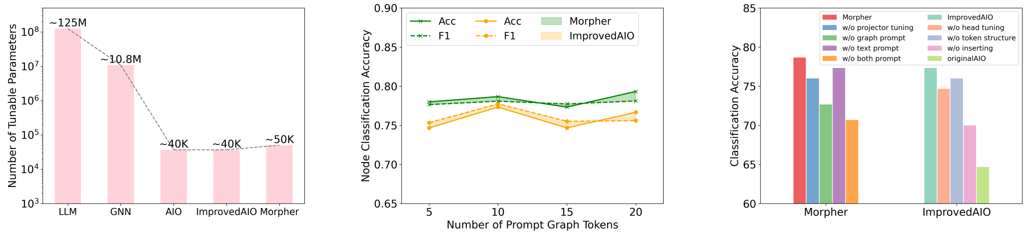

Figure 5: Efficiency comparison (left), parameter study (middle) and ablation study (right).

5.6 Hyperparameter and Ablation Study

We conduct the hyperparameter study by choosing and testing various numbers of graph prompt tokens for both ImprovedAIO and Morpher. The results are shown in Figure 5 (middle), from which we can observe that both methods are generally stable, and Morpher constantly outperforms ImprovedAIO under different choices. To verify the necessity of each component in our design, we compare Morpher and ImprovedAIO with multiple variants, respectively, and report the result in Figure 5 (right). We observe that removing any component would result in a performance drop. Additionally, our comparison of Morpher with ImprovedAIO throughout the experiments demonstrates that our multimodal design would lead to improvement over the uni-modal prompting of GNNs.

In the main experiments, we use RoBERTa as Morpher's text encoder. We also conduct experiments to verify the effectiveness of our proposed Morpher with ELECTRA [8] and DistilBERT [67] as the text encoder. Due to space limitation, we only show the F1 score of using ELECTRA in Table 4, and more detailed experiment data can be found in Appendix C.3. In general, using ELECTRA and DistilBERT results in similar performance compared to using RoBERTa, showing the robustness of Morpher with respect to the language encoder.

Table 4: Effectiveness (F1 score) of Morpher with ELECTRA [8] as the text encoder.

GNN pretraining	MUTAG	ENZYMES	PROTEINS	MSRC_21C
GraphCL + GCN	78.17	15.79	65.66	47.19
GraphCL + GAT	75.75	11.37	65.66	49.01
GraphCL + GT	77.04	14.68	72.70	44.09
SimGRACE + GCN	70.99	12.41	67.77	48.44
SimGRACE + GAT	77.51	13.31	67.78	49.43
SimGRACE + GT	73.55	15.76	70.28	44.50

As for the robustness with respect to the pre-trained GNNs, in the main experiments, we adopt two pre-train methods, GraphCL and SimGRACE to pre-train three different GNN architectures: GCN, GAT and GT. We further conduct experiments using GNNs pre-trained from GraphMAE [24] and MVGRL [21]. Due to the space limitation, we report the results and discuss in Appendix C.4.

5.7 Morpher on MolecureNet with More Text Supervision

We demonstrate that, though not specifically designed for any downstream applications, the Morpher framework has the potential to be used in various tasks where there is more text supervision compared to previous experiments. As for a case study, We use bace (inhibitors of human beta-secretase), tox21 (toxicology in the 21st century) and hiv (inhibit HIV replication) from MolecureNet [81]. These three datasets have 1513, 7831, and 41127 graphs to classify, respectively. In these datasets, each graph label is associated with a text description. The tasks on bace and hiv are bio-activity prediction and the task on tox21 is toxicity prediction. To adopt Morpher, we use GraphCL to pre-train the GAT model and initialize the text prompts and text labels using those from GIMLET [95].

Table 5: AUC-ROC (↑) on MolecureNet (bace, tox21, hiv). Morpher-K denotes K shots.

Dataset	KVPLM	MoMu	Galactica-1.3B	GIMLET-64M-50-shots	GAT-1M-supervised	Morpher-10	Morpher-20	Morpher-50
bace	0.5126	0.6656	0.5648	0.729	0.697	0.6231	0.6513	0.6858
tox21	0.4917	0.5757	0.4946	0.652	0.754	0.6769	0.7275	0.7459
hiv	0.6120	0.5026	0.3385	0.721	0.729	0.5742	0.7034	0.7283

KVPLM [91], MoMu [70], Galactica-1.3B [76] are zero-shot predictors for the three tasks; GIMLET-64M-50-shots is the GIMLET [95] model fine-tuned on 50 additional training samples[3]; GAT-1M-

[3]the performance of GIMLET and other baselines are directly from the GIMLET paper [95].

fully-supervised uses all the training data to train a GAT. Our Morpher-k-shots uses only k training samples. From the results, first, using only 10 training samples, Morpher can outperform the zero-shot baselines KVPLM, MoMu, and Galactica-1.3B. Second, using only 50 shots, Morpher can achieve similar performance with the fully supervised GAT. Third, using the same amount of few-shot data (50 shots), Morpher-50 outperforms GIMLET-64M-50-shots on tox21 and hiv, the two largest datasets among the three. This means our graph-text multi-modal prompt learning, with much fewer learnable parameters ($\sim 50K$), is more sample-efficient than fine-tuning language model encoder.

6 Related Work

GNN Pre-training. Recently, a surge of graph pre-training strategies have emerged to address the issue of label scarcity in graph representation learning [26, 58, 72, 43, 104, 1, 36]. The main idea of pre-trained graph models is to capture general graph information across different tasks and transfer this knowledge to the target task using techniques such as contrastive predictive coding [40, 12, 13, 65, 82], context prediction [62, 27], prompt tuning [72, 11], and mutual information maximization [62, 71, 30, 35, 37]. For instance, [26] proposes to learn transferable structural information from three levels of graph topology, including node-level, subgraph-level, and graph-level. Different from these approaches, this paper aims to build up foundational GNNs by leveraging multi-modal prompt learning techniques.

Graph Prompt Learning. Prompting is now mainstream for adapting NLP tasks, and recent studies exploring prompt learning for GNNs mark a thriving research area [74, 80]. It is a promising way to adapt GNNs to downstream tasks through token-level [11, 75, 5, 72, 107] or graph-level [73, 28, 19] prompting. Among all the existing methods, All-in-one (AIO) [73] is the only algorithm to learn tunable graph prompts for node-level, edge-level or graph-level downstream tasks given few-shot labeled data (Table 8). Based on our improved AIO, we present a pioneer study to explore learning prompts in multiple modalities simultaneously while keeping the pre-trained models frozen.

LLM on Graphs. Inspired by the advances of large language models in NLP [98], researchers have begun to explore their potential for graph-related tasks [31]. Current approaches can be divided into two main categories. The first category employs LLMs as pre-trained feature extractors to enhance GNNs [9, 7, 106]. For example, GLEM [97] proposes to input the language representation as initial features for the GNN and train them iteratively. The second category focuses on integrating graph structures directly into LLM architectures [87, 94, 34]. A notable example is Patton [33], which pre-trains a joint architecture on text-attributed graphs. Despite these advancements, none of them have explored the collaboration between LLMs and GNNs under graph prompt learning.

7 Conclusion

In this work, we introduce Morpher, the first multimodal prompt learning paradigm that can semantically adapt pre-trained GNNs to downstream tasks with the help of LLM, while keeping both the pre-trained models frozen. To build Morpher, we first analyze the limitations of the state-of-the-art graph prompting technique and propose an improved version. Through extensive experiments, we demonstrate that our improved AIO can achieve outperformance, and our Morpher has further improvements in few-shot, multi-level task, or domain transfer settings. Additionally, using Morpher, we build the first GNN zero-shot classifier prototype that can be generalized to novel testing classes.

References

[1] Yikun Ban, Yuchen Yan, Arindam Banerjee, and Jingrui He. Ee-net: Exploitation-exploration neural networks in contextual bandits. *arXiv preprint arXiv:2110.03177*, 2021. 11

[2] Yikun Ban, Jiaru Zou, Zihao Li, Yunzhe Qi, Dongqi Fu, Jian Kang, Hanghang Tong, and Jingrui He. Pagerank bandits for link prediction. *arXiv preprint arXiv:2411.01410*, 2024. 2

[3] Karsten M. Borgwardt, Cheng Soon Ong, Stefan Schönauer, S. V. N. Vishwanathan, Alexander J. Smola, and Hans-Peter Kriegel. Protein function prediction via graph kernels. In *Proceedings Thirteenth International Conference on Intelligent Systems for Molecular Biology 2005, Detroit, MI, USA, 25-29 June 2005*, pp. 47–56, 2005. doi: 10.1093/BIOINFORMATICS/BTI1007. URL https://doi.org/10.1093/bioinformatics/bti1007. 6

[4] Digbalay Bose, Rajat Hebbar, Krishna Somandepalli, Haoyang Zhang, Yin Cui, Kree Cole-McLaughlin, Huisheng Wang, and Shrikanth Narayanan. Movieclip: Visual scene recognition in movies. In *IEEE/CVF Winter Conference on Applications of Computer Vision, WACV 2023, Waikoloa, HI, USA, January 2-7, 2023*, pp. 2082–2091. IEEE, 2023. doi: 10.1109/WACV56688.2023.00212. URL https://doi.org/10.1109/WACV56688.2023.00212. 2

[5] 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. *CoRR*, abs/2310.14845, 2023. doi: 10.48550/ARXIV.2310.14845. URL https://doi.org/10.48550/arXiv.2310.14845. 11, 24

[6] Zefeng Chen, Wensheng Gan, Jiayang Wu, Kaixia Hu, and Hong Lin. Data scarcity in recommendation systems: A survey. *CoRR*, abs/2312.10073, 2023. doi: 10.48550/ARXIV.2312.10073. URL https://doi.org/10.48550/arXiv.2312.10073. 2

[7] Eli Chien, Wei-Cheng Chang, Cho-Jui Hsieh, Hsiang-Fu Yu, Jiong Zhang, Olgica Milenkovic, and Inderjit S Dhillon. Node feature extraction by self-supervised multi-scale neighborhood prediction. *arXiv preprint arXiv:2111.00064*, 2021. 11

[8] Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. ELECTRA: pre-training text encoders as discriminators rather than generators. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL https://openreview.net/forum?id=r1xMH1BtvB. 6, 10, 24, 25

[9] Keyu Duan, Qian Liu, Tat-Seng Chua, Shuicheng Yan, Wei Tsang Ooi, Qizhe Xie, and Junxian He. Simteg: A frustratingly simple approach improves textual graph learning. *arXiv preprint arXiv:2308.02565*, 2023. 11

[10] Vijay Prakash Dwivedi, Chaitanya K. Joshi, Anh Tuan Luu, Thomas Laurent, Yoshua Bengio, and Xavier Bresson. Benchmarking graph neural networks. *J. Mach. Learn. Res.*, 24:43:1–43:48, 2023. URL http://jmlr.org/papers/v24/22-0567.html. 4

[11] Taoran Fang, Yunchao Zhang, Yang Yang, Chunping Wang, and Lei Chen. Universal prompt tuning for graph neural networks. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/a4a1ee071ce0fe63b83bce507c9dc4d7-Abstract-Conference.html. 6, 7, 11, 24

[12] Shengyu Feng, Baoyu Jing, Yada Zhu, and Hanghang Tong. Adversarial graph contrastive learning with information regularization. In *WWW '22: The ACM Web Conference 2022, Virtual Event, Lyon, France, April 25 - 29, 2022*, pp. 1362–1371. ACM, 2022. 11

[13] Shengyu Feng, Baoyu Jing, Yada Zhu, and Hanghang Tong. Ariel: Adversarial graph contrastive learning. *ACM Transactions on Knowledge Discovery from Data*, 18(4):1–22, 2024. 11

[14] Matthias Fey and Jan Eric Lenssen. Fast graph representation learning with pytorch geometric. *CoRR*, abs/1903.02428, 2019. URL http://arxiv.org/abs/1903.02428. 4, 6

[15] Dongqi Fu and Jingrui He. DPPIN: A biological repository of dynamic protein-protein interaction network data. In *IEEE Big Data*, 2022. 2

[16] Dongqi Fu, Liri Fang, Ross Maciejewski, Vetle I. Torvik, and Jingrui He. Meta-learned metrics over multi-evolution temporal graphs. In *KDD*, 2022. 2

[17] Dongqi Fu, Liri Fang, Zihao Li, Hanghang Tong, Vetle I. Torvik, and Jingrui He. Parametric graph representations in the era of foundation models: A survey and position. *CoRR*, abs/2410.12126, 2024. doi: 10.48550/ARXIV.2410.12126. URL https://doi.org/10.48550/arXiv.2410.12126. 2

[18] Dongqi Fu, Zhigang Hua, Yan Xie, Jin Fang, Si Zhang, Kaan Sancak, Hao Wu, Andrey Malevich, Jingrui He, and Bo Long. Vcr-graphormer: A mini-batch graph transformer via virtual connections. *CoRR*, abs/2403.16030, 2024. doi: 10.48550/ARXIV.2403.16030. URL https://doi.org/10.48550/arXiv.2403.16030. 2

[19] Qingqing Ge, Zeyuan Zhao, Yiding Liu, Anfeng Cheng, Xiang Li, Shuaiqiang Wang, and Dawei Yin. Enhancing graph neural networks with structure-based prompt. *CoRR*, abs/2310.17394, 2023. doi: 10.48550/ARXIV.2310.17394. URL https://doi.org/10.48550/arXiv.2310.17394. 11, 24

[20] Andrey Guzhov, Federico Raue, Jörn Hees, and Andreas Dengel. Audioclip: Extending clip to image, text and audio. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2022, Virtual and Singapore, 23-27 May 2022*, pp. 976–980. IEEE, 2022. doi: 10.1109/ICASSP43922.2022.9747631. URL https://doi.org/10.1109/ICASSP43922.2022.9747631. 2

[21] Kaveh Hassani and Amir Hosein Khas Ahmadi. Contrastive multi-view representation learning on graphs. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pp. 4116–4126. PMLR, 2020. URL http://proceedings.mlr.press/v119/hassani20a.html. 6, 10, 25, 26

[22] Georg Hess, Adam Tonderski, Christoffer Petersson, Kalle Åström, and Lennart Svensson. Lidarclip or: How I learned to talk to point clouds. In *IEEE/CVF Winter Conference on Applications of Computer Vision, WACV 2024, Waikoloa, HI, USA, January 3-8, 2024*, pp. 7423–7432. IEEE, 2024. doi: 10.1109/WACV57701.2024.00727. URL https://doi.org/10.1109/WACV57701.2024.00727. 2

[23] Fangzhou Hong, Mingyuan Zhang, Liang Pan, Zhongang Cai, Lei Yang, and Ziwei Liu. Avatarclip: zero-shot text-driven generation and animation of 3d avatars. *ACM Trans. Graph.*, 41(4):161:1–161:19, 2022. doi: 10.1145/3528223.3530094. URL https://doi.org/10.1145/3528223.3530094. 2

[24] Zhenyu Hou, Xiao Liu, Yukuo Cen, Yuxiao Dong, Hongxia Yang, Chunjie Wang, and Jie Tang. Graphmae: Self-supervised masked graph autoencoders. In Aidong Zhang and Huzefa Rangwala (eds.), *KDD '22: The 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, August 14 - 18, 2022*, pp. 594–604. ACM, 2022. doi: 10.1145/3534678.3539321. URL https://doi.org/10.1145/3534678.3539321. 6, 10, 25

[25] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for NLP. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2790–2799. PMLR, 2019. URL http://proceedings.mlr.press/v97/houlsby19a.html. 2

[26] Weihua Hu, Bowen Liu, Joseph Gomes, Marinka Zitnik, Percy Liang, Vijay S. Pande, and Jure Leskovec. Strategies for pre-training graph neural networks. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL https://openreview.net/forum?id=HJlWWJSFDH. 2, 11

[27] Ziniu Hu, Yuxiao Dong, Kuansan Wang, Kai-Wei Chang, and Yizhou Sun. GPT-GNN: generative pre-training of graph neural networks. In Rajesh Gupta, Yan Liu, Jiliang Tang, and B. Aditya Prakash (eds.), *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*, pp. 1857–1867. ACM, 2020. 11

[28] Qian Huang, Hongyu Ren, Peng Chen, Gregor Krzmanc, Daniel Zeng, Percy Liang, and Jure Leskovec. PRODIGY: enabling in-context learning over graphs. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/34dce0dc3121951dd0399ba02c0f0d06-Abstract-Conference.html. 11, 24

[29] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V. Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pp. 4904–4916. PMLR, 2021. URL http://proceedings.mlr.press/v139/jia21b.html. 2

[30] Xunqiang Jiang, Yuanfu Lu, Yuan Fang, and Chuan Shi. Contrastive pre-training of gnns on heterogeneous graphs. In *CIKM '21: The 30th ACM International Conference on Information and Knowledge Management, Virtual Event, Queensland, Australia, November 1 - 5, 2021*, pp. 803–812. ACM, 2021. 11

[31] Bowen Jin, Gang Liu, Chi Han, Meng Jiang, Heng Ji, and Jiawei Han. Large language models on graphs: A comprehensive survey. *arXiv preprint arXiv:2312.02783*, 2023. 11

[32] Bowen Jin, Wentao Zhang, Yu Zhang, Yu Meng, Xinyang Zhang, Qi Zhu, and Jiawei Han. Patton: Language model pretraining on text-rich networks. In Anna Rogers, Jordan L. Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pp. 7005–7020. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.ACL-LONG.387. URL https://doi.org/10.18653/v1/2023.acl-long.387. 2

[33] Bowen Jin, Wentao Zhang, Yu Zhang, Yu Meng, Xinyang Zhang, Qi Zhu, and Jiawei Han. Patton: Language model pretraining on text-rich networks. *arXiv preprint arXiv:2305.12268*, 2023. 11

[34] Bowen Jin, Yu Zhang, Qi Zhu, and Jiawei Han. Heterformer: Transformer-based deep node representation learning on heterogeneous text-rich networks. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 1020–1031, 2023. 11

[35] Baoyu Jing, Chanyoung Park, and Hanghang Tong. HDMI: high-order deep multiplex infomax. In *WWW '21: The Web Conference 2021, Virtual Event / Ljubljana, Slovenia, April 19-23, 2021*, pp. 2414–2424. ACM / IW3C2, 2021. 11

[36] Baoyu Jing, Shengyu Feng, Yuejia Xiang, Xi Chen, Yu Chen, and Hanghang Tong. X-goal: Multiplex heterogeneous graph prototypical contrastive learning. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pp. 894–904, 2022. 11

[37] Baoyu Jing, Yuchen Yan, Yada Zhu, and Hanghang Tong. COIN: co-cluster infomax for bipartite graphs. *CoRR*, abs/2206.00006, 2022. doi: 10.48550/ARXIV.2206.00006. URL https://doi.org/10.48550/arXiv.2206.00006. 11

[38] Baoyu Jing, Yuchen Yan, Kaize Ding, Chanyoung Park, Yada Zhu, Huan Liu, and Hanghang Tong. Sterling: Synergistic representation learning on bipartite graphs. In Michael J. Wooldridge, Jennifer G. Dy, and Sriraam Natarajan (eds.), *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada*, pp. 12976–12984. AAAI Press, 2024. doi: 10.1609/AAAI.V38I12.29195. URL https://doi.org/10.1609/aaai.v38i12.29195. 2

[39] Muhammad Uzair Khattak, Hanoona Abdul Rasheed, Muhammad Maaz, Salman H. Khan, and Fahad Shahbaz Khan. Maple: Multi-modal prompt learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, pp. 19113–19122. IEEE, 2023. doi: 10.1109/CVPR52729.2023.01832. URL https://doi.org/10.1109/CVPR52729.2023.01832. 2

[40] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. 6, 11

[41] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *CoRR*, abs/1609.02907, 2016. URL http://arxiv.org/abs/1609.02907. 6

[42] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pp. 3045–3059. Association for Computational Linguistics, 2021. doi: 10.18653/V1/2021.EMNLP-MAIN.243. URL https://doi.org/10.18653/v1/2021.emnlp-main.243. 2

[43] Bolian Li, Baoyu Jing, and Hanghang Tong. Graph communal contrastive learning. In *WWW '22: The ACM Web Conference 2022, Virtual Event, Lyon, France, April 25 - 29, 2022*, pp. 1203–1213. ACM, 2022. 11

[44] Junnan Li, Dongxu Li, Caiming Xiong, and Steven C. H. Hoi. BLIP: bootstrapping language-image pre-training for unified vision-language understanding and generation. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato (eds.), *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pp. 12888–12900. PMLR, 2022. URL https://proceedings.mlr.press/v162/li22n.html. 2

[45] Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (eds.), *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pp. 4582–4597. Association for Computational Linguistics, 2021. doi: 10.18653/V1/2021.ACL-LONG.353. URL https://doi.org/10.18653/v1/2021.acl-long.353. 2

[46] Yichuan Li, Kaize Ding, and Kyumin Lee. GRENADE: graph-centric language model for self-supervised representation learning on text-attributed graphs. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pp. 2745–2757. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.FINDINGS-EMNLP.181. URL https://doi.org/10.18653/v1/2023.findings-emnlp.181. 2

[47] Zihao Li, Dongqi Fu, and Jingrui He. Everything evolves in personalized pagerank. In Ying Ding, Jie Tang, Juan F. Sequeda, Lora Aroyo, Carlos Castillo, and Geert-Jan Houben (eds.), *Proceedings of the ACM Web Conference 2023, WWW 2023, Austin, TX, USA, 30 April 2023 - 4 May 2023*, pp. 3342–3352. ACM, 2023. doi: 10.1145/3543507.3583474. URL https://doi.org/10.1145/3543507.3583474. 2

[48] Zihao Li, Yuyi Ao, and Jingrui He. Sphere: Expressive and interpretable knowledge graph embedding for set retrieval. In Grace Hui Yang, Hongning Wang, Sam Han, Claudia Hauff, Guido Zuccon, and Yi Zhang (eds.), *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2024, Washington DC, USA, July 14-18, 2024*, pp. 2629–2634. ACM, 2024. doi: 10.1145/3626772.3657910. URL https://doi.org/10.1145/3626772.3657910. 2

[49] Zihao Li, Dongqi Fu, Hengyu Liu, and Jingrui He. Hypergraphs as weighted directed self-looped graphs: Spectral properties, clustering, cheeger inequality. *arXiv preprint arXiv:2411.03331*, 2024. 2

[50] Zihao Li, Dongqi Fu, Hengyu Liu, and Jingrui He. Provably extending pagerank-based local clustering algorithm to weighted directed graphs with self-loops and to hypergraphs. *arXiv preprint arXiv:2412.03008*, 2024. 2

[51] Jiawei Liu, Cheng Yang, Zhiyuan Lu, Junze Chen, Yibo Li, Mengmei Zhang, Ting Bai, Yuan Fang, Lichao Sun, Philip S. Yu, and Chuan Shi. Towards graph foundation models: A survey and beyond. *CoRR*, abs/2310.11829, 2023. doi: 10.48550/ARXIV.2310.11829. URL https://doi.org/10.48550/arXiv.2310.11829. 2

[52] Pengfei Liu, Yiming Ren, and Zhixiang Ren. Git-mol: A multi-modal large language model for molecular science with graph, image, and text. *CoRR*, abs/2308.06911, 2023. doi: 10.48550/ARXIV.2308.06911. URL https://doi.org/10.48550/arXiv.2308.06911. 2

[53] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692, 2019. URL http://arxiv.org/abs/1907.11692. 6

[54] Yixin Liu, Ming Jin, Shirui Pan, Chuan Zhou, Yu Zheng, Feng Xia, and Philip S. Yu. Graph self-supervised learning: A survey. *IEEE Trans. Knowl. Data Eng.*, 35(6):5879–5900, 2023. doi: 10.1109/TKDE.2022.3172903. URL https://doi.org/10.1109/TKDE.2022.3172903. 2

[55] Zemin Liu, Xingtong Yu, Yuan Fang, and Xinming Zhang. Graphprompt: Unifying pre-training and downstream tasks for graph neural networks. In Ying Ding, Jie Tang, Juan F. Sequeda, Lora Aroyo, Carlos Castillo, and Geert-Jan Houben (eds.), *Proceedings of the ACM Web Conference 2023, WWW 2023, Austin, TX, USA, 30 April 2023 - 4 May 2023*, pp. 417–428. ACM, 2023. doi: 10.1145/3543507.3583386. URL https://doi.org/10.1145/3543507.3583386. 6, 7, 24

[56] Zhining Liu, Ruizhong Qiu, Zhichen Zeng, Hyunsik Yoo, David Zhou, Zhe Xu, Yada Zhu, Kommy Weldemariam, Jingrui He, and Hanghang Tong. Class-imbalanced graph learning without class rebalancing. In *Forty-first International Conference on Machine Learning*, 2024. 2

[57] Zhiyuan Liu, Sihang Li, Yanchen Luo, Hao Fei, Yixin Cao, Kenji Kawaguchi, Xiang Wang, and Tat-Seng Chua. Molca: Molecular graph-language modeling with cross-modal projector and uni-modal adapter. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pp. 15623–15638. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.EMNLP-MAIN.966. URL https://doi.org/10.18653/v1/2023.emnlp-main.966. 2

[58] Yuanfu Lu, Xunqiang Jiang, Yuan Fang, and Chuan Shi. Learning to pre-train graph neural networks. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pp. 4276–4284. AAAI Press, 2021. 11

[59] Yizhen Luo, Kai Yang, Massimo Hong, Xing Yi Liu, and Zaiqing Nie. Molfm: A multimodal molecular foundation model. *CoRR*, abs/2307.09484, 2023. doi: 10.48550/ARXIV.2307.09484. URL https://doi.org/10.48550/arXiv.2307.09484. 2

[60] Sahil Manchanda, Shubham Gupta, Sayan Ranu, and Srikanta J. Bedathur. Generative modeling of labeled graphs under data scarcity. In Soledad Villar and Benjamin Chamberlain (eds.), *Learning on Graphs Conference, 27-30 November 2023, Virtual Event*, volume 231 of *Proceedings of Machine Learning Research*, pp. 32. PMLR, 2023. URL https://proceedings.mlr.press/v231/manchanda24a.html. 2

[61] Christopher Morris, Nils M. Kriege, Franka Bause, Kristian Kersting, Petra Mutzel, and Marion Neumann. Tudataset: A collection of benchmark datasets for learning with graphs. *CoRR*, abs/2007.08663, 2020. URL https://arxiv.org/abs/2007.08663. 4, 6

[62] Nicolò Navarin, Dinh Van Tran, and Alessandro Sperduti. Pre-training graph neural networks with kernels. *CoRR*, abs/1811.06930, 2018. 11

[63] Marion Neumann, Roman Garnett, Christian Bauckhage, and Kristian Kersting. Propagation kernels: efficient graph kernels from propagated information. *Mach. Learn.*, 102(2): 209–245, 2016. doi: 10.1007/S10994-015-5517-9. URL https://doi.org/10.1007/s10994-015-5517-9. 6

[64] Yunzhe Qi, Yikun Ban, and Jingrui He. Graph neural bandits. In Ambuj K. Singh, Yizhou Sun, Leman Akoglu, Dimitrios Gunopulos, Xifeng Yan, Ravi Kumar, Fatma Ozcan, and Jieping Ye (eds.), *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2023, Long Beach, CA, USA, August 6-10, 2023*, pp. 1920–1931. ACM, 2023. doi: 10.1145/3580305.3599371. URL https://doi.org/10.1145/3580305.3599371. 2

[65] Jiezhong Qiu, Qibin Chen, Yuxiao Dong, Jing Zhang, Hongxia Yang, Ming Ding, Kuansan Wang, and Jie Tang. GCC: graph contrastive coding for graph neural network pre-training. In Rajesh Gupta, Yan Liu, Jiliang Tang, and B. Aditya Prakash (eds.), *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*, pp. 1150–1160. ACM, 2020. 11

[66] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8748–8763. PMLR, 2021. URL http://proceedings.mlr.press/v139/radford21a.html. 2

[67] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. *CoRR*, abs/1910.01108, 2019. URL http://arxiv.org/abs/1910.01108. 6, 10, 24, 25

[68] Yi-Jen Shih, Hsuan-Fu Wang, Heng-Jui Chang, Layne Berry, Hung-yi Lee, and David Harwath. Speechclip: Integrating speech with pre-trained vision and language model. In *IEEE Spoken Language Technology Workshop, SLT 2022, Doha, Qatar, January 9-12, 2023*, pp. 715–722. IEEE, 2022. doi: 10.1109/SLT54892.2023.10022954. URL https://doi.org/10.1109/SLT54892.2023.10022954. 2

[69] Harry Shomer, Yao Ma, Haitao Mao, Juanhui Li, Bo Wu, and Jiliang Tang. Lpformer: An adaptive graph transformer for link prediction, 2024. 2

[70] Bing Su, Dazhao Du, Zhao Yang, Yujie Zhou, Jiangmeng Li, Anyi Rao, Hao Sun, Zhiwu Lu, and Ji-Rong Wen. A molecular multimodal foundation model associating molecule graphs with natural language. *CoRR*, abs/2209.05481, 2022. doi: 10.48550/ARXIV.2209.05481. URL https://doi.org/10.48550/arXiv.2209.05481. 2, 10

[71] Fan-Yun Sun, Jordan Hoffmann, Vikas Verma, and Jian Tang. Infograph: Unsupervised and semi-supervised graph-level representation learning via mutual information maximization. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. 11

[72] Mingchen Sun, Kaixiong Zhou, Xin He, Ying Wang, and Xin Wang. GPPT: graph pre-training and prompt tuning to generalize graph neural networks. In Aidong Zhang and Huzefa Rangwala (eds.), *KDD '22: The 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, August 14 - 18, 2022*, pp. 1717–1727. ACM, 2022. doi: 10.1145/3534678.3539249. URL https://doi.org/10.1145/3534678.3539249. 7, 11, 24

[73] Xiangguo Sun, Hong Cheng, Jia Li, Bo Liu, and Jihong Guan. All in one: Multi-task prompting for graph neural networks. In Ambuj K. Singh, Yizhou Sun, Leman Akoglu, Dimitrios Gunopulos, Xifeng Yan, Ravi Kumar, Fatma Ozcan, and Jieping Ye (eds.), *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2023, Long Beach, CA, USA, August 6-10, 2023*, pp. 2120–2131. ACM, 2023. doi: 10.1145/3580305.3599256. URL https://doi.org/10.1145/3580305.3599256. 2, 3, 4, 6, 7, 8, 9, 11, 24

[74] Xiangguo Sun, Jiawen Zhang, Xixi Wu, Hong Cheng, Yun Xiong, and Jia Li. Graph prompt learning: A comprehensive survey and beyond. *CoRR*, abs/2311.16534, 2023. doi: 10.48550/ARXIV.2311.16534. URL https://doi.org/10.48550/arXiv.2311.16534. 11

[75] Zhen Tan, Ruocheng Guo, Kaize Ding, and Huan Liu. Virtual node tuning for few-shot node classification. In Ambuj K. Singh, Yizhou Sun, Leman Akoglu, Dimitrios Gunopulos, Xifeng Yan, Ravi Kumar, Fatma Ozcan, and Jieping Ye (eds.), *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2023, Long Beach, CA, USA, August 6-10, 2023*, pp. 2177–2188. ACM, 2023. doi: 10.1145/3580305.3599541. URL https://doi.org/10.1145/3580305.3599541. 11, 24

[76] Ross Taylor, Marcin Kardas, Guillem Cucurull, Thomas Scialom, Anthony Hartshorn, Elvis Saravia, Andrew Poulton, Viktor Kerkez, and Robert Stojnic. Galactica: A large language model for science. *CoRR*, abs/2211.09085, 2022. doi: 10.48550/ARXIV.2211.09085. URL https://doi.org/10.48550/arXiv.2211.09085. 10

[77] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 5998–6008, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html. 5

[78] Zhihao Wen and Yuan Fang. Augmenting low-resource text classification with graph-grounded pre-training and prompting. In Hsin-Hsi Chen, Wei-Jou (Edward) Duh, Hen-Hsen Huang, Makoto P. Kato, Josiane Mothe, and Barbara Poblete (eds.), *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2023, Taipei, Taiwan, July 23-27, 2023*, pp. 506–516. ACM, 2023. doi: 10.1145/3539618.3591641. URL https://doi.org/10.1145/3539618.3591641. 2

[79] Ho-Hsiang Wu, Prem Seetharaman, Kundan Kumar, and Juan Pablo Bello. Wav2clip: Learning robust audio representations from clip. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2022, Virtual and Singapore, 23-27 May 2022*, pp. 4563–4567. IEEE, 2022. doi: 10.1109/ICASSP43922.2022.9747669. URL https://doi.org/10.1109/ICASSP43922.2022.9747669. 2

[80] Xuansheng Wu, Kaixiong Zhou, Mingchen Sun, Xin Wang, and Ninghao Liu. A survey of graph prompting methods: Techniques, applications, and challenges. *CoRR*, abs/2303.07275, 2023. doi: 10.48550/ARXIV.2303.07275. URL https://doi.org/10.48550/arXiv.2303.07275. 11

[81] Zhenqin Wu, Bharath Ramsundar, Evan N. Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S. Pappu, Karl Leswing, and Vijay S. Pande. Moleculenet: A benchmark for molecular machine learning. *CoRR*, abs/1703.00564, 2017. URL http://arxiv.org/abs/1703.00564. 10

[82] Jun Xia, Lirong Wu, Jintao Chen, Bozhen Hu, and Stan Z. Li. Simgrace: A simple framework for graph contrastive learning without data augmentation. In Frédérique Laforest, Raphaël

Troncy, Elena Simperl, Deepak Agarwal, Aristides Gionis, Ivan Herman, and Lionel Médini (eds.), *WWW '22: The ACM Web Conference 2022, Virtual Event, Lyon, France, April 25 - 29, 2022*, pp. 1070–1079. ACM, 2022. 6, 11

[83] Jun Xia, Yanqiao Zhu, Yuanqi Du, and Stan Z. Li. A survey of pretraining on graphs: Taxonomy, methods, and applications. *CoRR*, abs/2202.07893, 2022. URL https://arxiv.org/abs/2202.07893. 2

[84] Hu Xu, Gargi Ghosh, Po-Yao Huang, Dmytro Okhonko, Armen Aghajanyan, Florian Metze, Luke Zettlemoyer, and Christoph Feichtenhofer. Videoclip: Contrastive pre-training for zero-shot video-text understanding. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pp. 6787–6800. Association for Computational Linguistics, 2021. doi: 10.18653/V1/2021.EMNLP-MAIN.544. URL https://doi.org/10.18653/v1/2021.emnlp-main.544. 2

[85] Zhe Xu, Kaveh Hassani, Si Zhang, Hanqing Zeng, Michihiro Yasunaga, Limei Wang, Dongqi Fu, Ning Yao, Bo Long, and Hanghang Tong. Language models are graph learners. *CoRR*, abs/2410.02296, 2024. doi: 10.48550/ARXIV.2410.02296. URL https://doi.org/10.48550/arXiv.2410.02296. 2

[86] Hao Yan, Chaozhuo Li, Ruosong Long, Chao Yan, Jianan Zhao, Wenwen Zhuang, Jun Yin, Peiyan Zhang, Weihao Han, Hao Sun, Weiwei Deng, Qi Zhang, Lichao Sun, Xing Xie, and Senzhang Wang. A comprehensive study on text-attributed graphs: Benchmarking and rethinking. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/37d00f567a18b478065f1a91b95622a0-Abstract-Datasets_and_Benchmarks.html. 2

[87] Junhan Yang, Zheng Liu, Shitao Xiao, Chaozhuo Li, Defu Lian, Sanjay Agrawal, Amit Singh, Guangzhong Sun, and Xing Xie. Graphformers: Gnn-nested transformers for representation learning on textual graph. *Advances in Neural Information Processing Systems*, 34:28798–28810, 2021. 11

[88] Zhilin Yang, William W. Cohen, and Ruslan Salakhutdinov. Revisiting semi-supervised learning with graph embeddings. In Maria-Florina Balcan and Kilian Q. Weinberger (eds.), *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pp. 40–48. JMLR.org, 2016. URL http://proceedings.mlr.press/v48/yanga16.html. 4, 6

[89] Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. Graph contrastive learning with augmentations. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/3fe230348e9a12c13120749e3f9fa4cd-Abstract.html. 6

[90] Seongjun Yun, Minbyul Jeong, Raehyun Kim, Jaewoo Kang, and Hyunwoo J. Kim. Graph transformer networks. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 11960–11970, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/9d63484abb477c97640154d40595a3bb-Abstract.html. 6

[91] Zheni Zeng, Yuan Yao, Zhiyuan Liu, and Maosong Sun. A deep-learning system bridging molecule structure and biomedical text with comprehension comparable to human professionals. *Nature communications*, 13(1):862, 2022. 10

[92] Ningyu Zhang, Luoqiu Li, Xiang Chen, Shumin Deng, Zhen Bi, Chuanqi Tan, Fei Huang, and Huajun Chen. Differentiable prompt makes pre-trained language models better few-shot learners. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL https://openreview.net/forum?id=ek9a0qIafW. 2

[93] Renrui Zhang, Ziyu Guo, Wei Zhang, Kunchang Li, Xupeng Miao, Bin Cui, Yu Qiao, Peng Gao, and Hongsheng Li. Pointclip: Point cloud understanding by CLIP. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pp. 8542–8552. IEEE, 2022. doi: 10.1109/CVPR52688.2022.00836. URL https://doi.org/10.1109/CVPR52688.2022.00836. 2

[94] Xikun Zhang, Antoine Bosselut, Michihiro Yasunaga, Hongyu Ren, Percy Liang, Christopher D Manning, and Jure Leskovec. Greaselm: Graph reasoning enhanced language models for question answering. *arXiv preprint arXiv:2201.08860*, 2022. 11

[95] Haiteng Zhao, Shengchao Liu, Chang Ma, Hannan Xu, Jie Fu, Zhihong Deng, Lingpeng Kong, and Qi Liu. GIMLET: A unified graph-text model for instruction-based molecule zero-shot learning. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/129033c7c08be683059559e8d6bfd460-Abstract-Conference.html. 10

[96] Huanjing Zhao, Beining Yang, Yukuo Cen, Junyu Ren, Chenhui Zhang, Yuxiao Dong, Evgeny Kharlamov, Shu Zhao, and Jie Tang. Pre-training and prompting for few-shot node classification on text-attributed graphs. In Ricardo Baeza-Yates and Francesco Bonchi (eds.), *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2024, Barcelona, Spain, August 25-29, 2024*, pp. 4467–4478. ACM, 2024. doi: 10.1145/3637528.3671952. URL https://doi.org/10.1145/3637528.3671952. 4

[97] Jianan Zhao, Meng Qu, Chaozhuo Li, Hao Yan, Qian Liu, Rui Li, Xing Xie, and Jian Tang. Learning on large-scale text-attributed graphs via variational inference. *arXiv preprint arXiv:2210.14709*, 2022. 11

[98] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 2023. 11

[99] Lecheng Zheng, Jinjun Xiong, Yada Zhu, and Jingrui He. Contrastive learning with complex heterogeneity. In Aidong Zhang and Huzefa Rangwala (eds.), *KDD '22: The 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, August 14 - 18, 2022*, pp. 2594–2604. ACM, 2022. doi: 10.1145/3534678.3539311. URL https://doi.org/10.1145/3534678.3539311. 6

[100] Lecheng Zheng, Dongqi Fu, Ross Maciejewski, and Jingrui He. DrGNN: Deep residual graph neural network with contrastive learning. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=frb6sLbACS. 2

[101] Lecheng Zheng, Baoyu Jing, Zihao Li, Hanghang Tong, and Jingrui He. Heterogeneous contrastive learning for foundation models and beyond. In Ricardo Baeza-Yates and Francesco Bonchi (eds.), *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2024, Barcelona, Spain, August 25-29, 2024*, pp. 6666–6676. ACM, 2024. doi: 10.1145/3637528.3671454. URL https://doi.org/10.1145/3637528.3671454. 2

[102] Lecheng Zheng, Dawei Zhou, Hanghang Tong, Jiejun Xu, Yada Zhu, and Jingrui He. Fairgen: Towards fair graph generation. In *40th IEEE International Conference on Data Engineering, ICDE 2024, Utrecht, The Netherlands, May 13-16, 2024*, pp. 2285–2297. IEEE, 2024. doi: 10.1109/ICDE60146.2024.00181. URL https://doi.org/10.1109/ICDE60146.2024.00181. 2

[103] Dawei Zhou, Lecheng Zheng, Jiawei Han, and Jingrui He. A data-driven graph generative model for temporal interaction networks. In Rajesh Gupta, Yan Liu, Jiliang Tang, and B. Aditya Prakash (eds.), *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*, pp. 401–411. ACM, 2020. doi: 10.1145/3394486.3403082. URL https://doi.org/10.1145/3394486.3403082. 2

[104] Dawei Zhou, Lecheng Zheng, Dongqi Fu, Jiawei Han, and Jingrui He. Mentorgnn: Deriving curriculum for pre-training gnns. In Mohammad Al Hasan and Li Xiong (eds.), *Proceedings of the 31st ACM International Conference on Information & Knowledge Management, Atlanta, GA, USA, October 17-21, 2022*, pp. 2721–2731. ACM, 2022. 11

[105] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pp. 16795–16804. IEEE, 2022. doi: 10.1109/CVPR52688.2022.01631. URL https://doi.org/10.1109/CVPR52688.2022.01631. 2

[106] Jason Zhu, Yanling Cui, Yuming Liu, Hao Sun, Xue Li, Markus Pelger, Tianqi Yang, Liangjie Zhang, Ruofei Zhang, and Huasha Zhao. Textgnn: Improving text encoder via graph neural network in sponsored search. In *Proceedings of the Web Conference 2021*, pp. 2848–2857, 2021. 11

[107] Yun Zhu, Jianhao Guo, and Siliang Tang. SGL-PT: A strong graph learner with graph prompt tuning. *CoRR*, abs/2302.12449, 2023. doi: 10.48550/ARXIV.2302.12449. URL https://doi.org/10.48550/arXiv.2302.12449. 11, 24

[108] Jiaru Zou, Mengyu Zhou, Tao Li, Shi Han, and Dongmei Zhang. Promptintern: Saving inference costs by internalizing recurrent prompt during large language model fine-tuning. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2024, Miami, Florida, USA, November 12-16, 2024*, pp. 10288–10305. Association for Computational Linguistics, 2024. URL https://aclanthology.org/2024.findings-emnlp.602. 2

A Proof of Theorem 3.1

Proof. The cross-entropy loss between the true distribution $p(\cdot)$ and the predicted distribution $q(\cdot)$ is given by:

$$\text{CE}(p, q) = -\sum_y p(y) \log q(y)$$

where $q(y) = Pr(c(\mathbf{x}) = y)$.

To find the optimal classification, we minimize the cross-entropy loss subject to the constraint $\sum_y q(y) = 1$. We define the Lagrangian as:

$$\mathcal{L}(q, \lambda) = -\sum_y p(y) \log q(y) + \lambda \left(\sum_y q(y) - 1 \right)$$

For any $y \in \mathcal{Y}$, take the derivative of $\mathcal{L}$ with respect to $q(y)$ and λ and set them to zero, we get:

$$\frac{\partial \mathcal{L}}{\partial q(y)} = -\frac{p(y)}{q(y)} + \lambda = 0$$
$$\frac{\partial \mathcal{L}}{\partial \lambda} = \sum_y q(y) - 1 = 0$$

Solving these equations, we find:

$$q(y) = \frac{p(y)}{\lambda}$$
$$\sum_y q(y) = \sum_y \frac{p(y)}{\lambda} = \frac{1}{\lambda} \sum_y p(y) = 1$$

Therefore, $\lambda = 1$ and $q(y) = p(y)$.

Thus, the optimal classification is $Pr(c(\mathbf{x}) = y) = p(y)$.

□

B Dataset Details

B.1 Dataset Statistics

Table 6 summarizes the statistics of the public real-world datasets, which we used in the few-shot experiments. For our synthetic datasets in the zero-shot prototype, we summarize their statistics in Table 7. As discussed in Section 5.4, the connections of our synthetic datasets are real, and we only replace the node feature by $[1, 0]$ and $[0, 1]$. The code to download the public data and the code to create synthetic data are provided in the supplementary materials.

B.2 Text Labels

When created, real-world graph datasets are usually coupled with textual meanings, but a common practice is to convert the textual meanings into numbers to create labels, which weakens the supervision of the graph data. For each real-world dataset, we convert the numerical labels back to text labels and feed into Morpher Language encoder through "[learnable text prompt] + [text label]". The mapping from the numbers to text labels for each dataset are provided as follows:

MUTAG. MUTAG is a dataset of nitroaromatic compounds, aiming to predict their mutagenicity on Salmonella typhimurium. Therefore, the mapping from numerical labels to text labels is: {0: non-mutagenic on Salmonella typhimurium, 1: mutagenic on Salmonella typhimurium}.

Table 6: Dataset statistics

Dataset	task level	# graphs	average # nodes	average # edges	# feature dimension	# classes	# shots per class	feature characteristic
MUTAG	graph	188	17.9	39.6	7	2	10	one-hot, sparse
ENZYMES	graph	600	32.6	124.3	3	6	10	one-hot, sparse
PROTEINS	graph	1113	39.1	145.6	3	2	10	one-hot, sparse
MSRC_21C	graph	209	40.28	96.60	22	17	1	one-hot, sparse
Cora	node, edge	1	2708	10556	1433	7	2 (node), 20 (edge)	sum 1, sparse
CiteSeer	node, edge	1	3327	9104	3703	6	2 (node), 20 (edge)	sum 1, sparse
PubMed	node	1	19,717	88648	500	3	10	TF-IDF value, dense

Table 7: Synthetic Zero-shot Class Generalization Dataset statistics

Dataset	# graphs	average # nodes	average # edges	#feature dimension	# classes	# shots per class
ZERO-Cora	120	8.41	10.38	2	2	10
ZERO-CiteSeer	120	10.03	21.31	2	2	10
ZERO-PubMed	120	20.33	41.75	2	2	10

ENZYMES. ENZYMES aims to predict which subcategory each enzyme belongs to. The subcategories are: 0: oxidoreductases, 1: transferases, 2: hydrolases, 3: lyases, 4: isomerases, 5: ligases.

PROTEINS. PROTEINS is a dataset comprising proteins classified as either enzymes or non-enzymes. Therefore, the mapping is: 0: 'enzyme', 1: 'non-enzyme'.

MSRC_21C. Each graph in MSRC is constructed according to an image. The graph label is the image label. MSRC_21C contains 20 classes in MSRC, and "C" here means "Challenging" as the graphs(images) that are easy to classify has been filtered. The mapping from the numerical labels to text labels is: {0: building, 1: grass, 2: tree, 3: cow, 4: sheep, 5: sky, 6: airplane, 7: water, 8: face, 9: car, 10: bicycle, 11: flower, 12: sign, 13: bird, 14: book, 15: chair, 16: road}.

Cora. Cora is a citation network of papers in seven research areas. Each paper is labeled according to its corresponding research area. The mapping from the numerical labels to text labels is: {0: case based, 1: genetic algorithms, 2: neural networks, 3: probabilistic methods, 4: reinforcement learning, 5: rule learning, 6: theory}.

CiteSeer. CiteSeer is a citation network of papers, each labeled according to one of six research areas. The mapping from the numerical labels to text labels is: {0: Agents, 1: AI, 2: DB, 3: IR, 4: ML, 5: HCI}. We note that using abbreviations of the research area is not an issue because these abbreviations frequently appear, and the LLM tends to tokenize each of them as one token.

PubMed. PubMed is a collection of scientific publications from the PubMed database related to diabetes, classified into one of three categories. The mapping from the numerical labels to text labels is: {0: Diabetes Mellitus Experimental, 1: Diabetes Mellitus Type 1, 2: Diabetes Mellitus Type 2}.

Edge-level tasks. Cora, CiteSeer and PubMed can also be used as link prediction datasets. For link prediction, the mapping from the numerical labels to text labels is: {0: not connected, 1: connected}.

Synthetic Zero-shot Class Generalization Datasets. For ZERO-Cora, we synthetic three classes of ego-graph in a citation network. The first and second classes, respectively, have text labels "machine learning" and "theory", and the third (novel) class to generalize is "machine learning theory". For ZERO-CiteSeer, we synthetic three classes of ego-graph in a citation network. The first and second classes, respectively, have text labels "biology" and "informatics", and the third (novel) class to generalize is "bioinformatics". For ZERO-PubMed, we synthetic three classes of ego-graph in a citation network in the medical domain. The first and second classes, respectively, have text labels "cardiology" and "neurology", and the third (novel) class to generalize is "neurocardiology".

Table 8: Comparison of graph prompts.

Method	prompt level	level of supported downstream tasks			learnable prompt	semantic
		node-level	edge-level	graph-level		
GPF-Plus [11]	token-level	√	×	×	√	×
Gprompt [55]	token-level	√	×	√	√	×
VNT [75]	token-level	×	×	√	√	×
ULTRA-DP [5]	token-level	√	×	×	√	×
GPPT [72]	token-level	√	×	×	√	×
SGL-PT [107]	token-level	√	×	×	√	×
SAP [19]	graph-level	√	×	√	√	×
PRODIGY [28]	graph-level	√	√	√	×	×
All-in-one (AIO) [73]	graph-level	√	√	√	√	×
ImprovedAIO (ours)	graph-level	√	√	√	√	×
Morpher (ours)	graph-level	√	√	√	√	√

C Experiment Details

C.1 Reproducibility

Code. The code for the experiments is provided in the supplementary material with a well-written README file. We also provide the commands and instructions to run the code. The datasets used will be automatically downloaded when the code is executed.

Environment. We run all our experiments on a Windows 11 machine with a 13th Gen Intel(R) Core(TM) i9-13900H CPU, 64GB RAM, and an NVIDIA RTX A4500 GPU. We have also tested the code on a Linux machine with NVIDIA TITAN RTX GPU. All the code of our algorithms is written in Python. The Python version in our environment is 3.9.18. In order to run our code, one has to install some other common libraries, including PyTorch, PyTorch Geometric, pandas, numpy, scipy, etc. Please refer to our README in the code directory for downloading instructions.

We have optimized our code and tested that the space cost of **the CPU memory is less than 16 GB, and the space cost of the graphics card is less than 6 GB**. The execution time to run an experiment is less than 20 minutes on our machine.

C.2 Implementation Details

We provide the configuration files for the experiments to reproduce the results. We initialize the graph prompt using kaiming_initialization, and we initialize the text prompts through real token embeddings. We have tested multiple initializations, and they would not affect the overall results. Specifically, we initialize the text prompt for each dataset as follows.

MUTAG: “a graph with property”; ENZYMES: “this enzyme is”; PROTEINS: “this protein is”; MSRC_21C: “an image of”; Cora: “a paper of”; CiteSeer: “a paper of”; PubMed: “a paper of”; Edge tasks: “central nodes are”.

In our few-shot setting, we split the labeled data into training samples and validation samples at approximately 1:1. For all the parameters, we used the Adam optimizer, whose learning rate and weight decay are provided in the configuration files.

C.3 Experiment with ELECTRA and DistilBERT

On the LLM pre-training side, RoBERTa is one of the most advanced encoder-only LLMs until now, and we have demonstrated the effectiveness with RoBERTa serving on the LLM side in the Morpher paradigm. Additionally, we conducted experiments with ELECTRA [8] and DistilBERT [67]. Using these two LLMs, Morpher can also achieve comparable performances to RoBERTa. The results are shown as follows.

Table 9: Few-shot graph classification performance (%) of Morpher with ELECTRA [8] as language encoder. Other experiment settings are identical to the main experiment.

GNN pretraining	MUTAG		ENZYMES		PROTEINS		MSRC_21C	
	Acc	F1	Acc	F1	Acc	F1	Acc	F1
GraphCL + GCN	78.00	78.17	20.41	15.79	67.38	65.66	43.42	47.19
GraphCL + GAT	76.67	75.75	20.41	11.37	66.26	65.66	44.57	49.01
GraphCL + GT	76.67	77.04	19.16	14.68	73.06	72.70	42.28	44.09
SimGRACE + GCN	70.00	70.99	19.79	12.41	68.96	67.77	45.71	48.44
SimGRACE + GAT	77.33	77.51	18.12	13.31	68.96	67.78	44.00	49.43
SimGRACE + GT	72.67	73.55	18.33	15.76	70.18	70.28	41.14	44.50

Table 10: Few-shot graph classification performance (%) of Morpher with DistilBERT [67] as language encoder. Other experiment settings are identical to the main experiment.

GNN pretraining	MUTAG		ENZYMES		PROTEINS		MSRC_21C	
	Acc	F1	Acc	F1	Acc	F1	Acc	F1
GraphCL + GCN	78.00	78.61	20.62	10.00	66.44	65.54	43.42	47.98
GraphCL + GAT	77.33	75.64	21.25	15.87	70.59	68.25	45.14	48.82
GraphCL + GT	74.67	75.20	19.58	14.96	70.27	70.55	44.57	47.28
SimGRACE + GCN	69.33	70.36	20.62	18.82	66.91	66.41	45.14	47.77
SimGRACE + GAT	77.33	76.90	18.54	14.44	67.56	65.08	45.71	44.36
SimGRACE + GT	72.67	73.52	17.91	11.06	70.55	70.36	45.14	44.01

In general, using ELECTRA and DistilBERT results in similar performance compared to using RoBERTa, showing the robustness of Morpher with respect to the language encoder.

C.4 Experiment with GNNs trained using GraphMAE and MVGRL

In the main pages, we used GraphCL and SimGRACE to show that Morpher achieves better performance given a pre-trained GNN. Additionally, to further verify the robustness of Morpher over the pre-train method, we conducted experiments on the pre-trained GNNs using GraphMAE [24] and MVGRL [21]. We use GCN as the GNN backbone and RoBERTa as the LLM encoder, and the results are reported as follows.

Table 11: Few-shot graph classification performance (%) of Morpher with the GNN pre-trained by GraphMAE [24]. Other experiment settings are identical to the main experiment.

GNN pretraining	MUTAG		ENZYMES		PROTEINS		MSRC_21C	
	Acc	F1	Acc	F1	Acc	F1	Acc	F1
Pre-train + Fine-tune	71.33	71.41	16.04	12.14	65.86	65.22	39.42	40.20
ImprovedAIO	76.67	76.95	19.58	12.59	66.36	65.30	42.28	46.81
Morpher	78.67	78.67	20.20	16.95	67.38	65.66	45.71	48.49

Using GraphMAE or MVGRL to pre-train the GNN, the trend of performance is similar to that when using GraphCL or SimGRACE. Also, ImprovedAIO and Morpher's performance is similar to that of pre-trained GNNs from GraphCL or SimGRACE and can still significantly outperform the pre-train + fine-tune baseline, showing the robustness of Morpher with respect to the pre-training strategy.

D Limitations

Graph prompt learning assumes the "pre-train + prompt" framework to build graph foundation models, yet there could be other paths to achieve graph-related foundation models. Also, graph

Table 12: Few-shot graph classification performance (%) of Morpher with the GNN pre-trained by MVGRL [21]. Other experiment settings are identical to the main experiment.

GNN pretraining	MUTAG		ENZYMES		PROTEINS		MSRC_21C	
	Acc	F1	Acc	F1	Acc	F1	Acc	F1
Pre-train + Fine-tune	68.67	69.46	16.45	10.16	65.15	64.71	38.85	40.56
ImprovedAIO	74.67	74.00	18.13	15.57	66.54	65.90	42.85	46.66
Morpher	78.00	77.81	18.96	14.97	67.56	66.79	44.57	48.67

prompt learning only works on the graph neural network architecture, and might not work for other architectures that are proposed in the future. Another limitation of this work is the requirement of language encoder. While RoBERTa is one of the most advanced encoder-only language models and can be considered an LLM with over 0.1B parameters, more recent LLMs such as Llama or Mistral cannot be used in Morpher because they are decoder-only LLMs and do not explicitly have an encoder. Yet it is possible to retrieve the hidden representation before the decoder layer. We leave this direction as future work.