

---

# One Prompt Fits All: Universal Graph Adaptation for Pretrained Models (Appendix)

---

Anonymous Author(s)

Affiliation

Address

email

## 1 A Proofs

### 2 A.1 Proof for Proposition 4.1

3 *Proof.*  $(C \circ T)(\mathbf{h}) = \mathbf{W}_C^\top (\mathbf{W}_T \mathbf{h} + \mathbf{b}_T) = (\mathbf{W}_T^\top \mathbf{W}_C)^\top \mathbf{h} + \mathbf{W}_C^\top \mathbf{b}_T$ . Then, we let  $\mathbf{W}_{C'} =$   
 4  $\mathbf{W}_T^\top \mathbf{W}_C$ ,  $\mathbf{b}_{C'} = \mathbf{W}_C^\top \mathbf{b}_T$ , we can get  $(C \circ T)(\mathbf{h}) = \mathbf{W}_{C'}^\top \mathbf{h} + \mathbf{b}_{C'} \equiv C'(\mathbf{h})$ . Therefore, we conclude  
 5 that any linear prompt combination can be represented as a linear classifier with a bias term.  $\square$

### 6 A.2 Proof for Proposition 4.2

7 *Proof.* For any parameters of objective function  $\mathbf{W}_{C'}$  and  $\mathbf{b}_{C'}$ , there exists  $\mathbf{W}_T$  and  $\mathbf{W}_C$ :

$$\mathbf{W}_C = (\mathbf{W}_T^\top)^{-1} \mathbf{W}_{C'}, \quad \mathbf{b}_T = \mathbf{W}_C^\dagger \mathbf{b}_{C'}, \quad (1)$$

8 where  $\mathbf{W}_C^\dagger$  is the pseudo-inverse matrix of  $\mathbf{W}_C$ . The mapping above is unique when  $\mathbf{W}_C^\dagger =$   
 9  $(\mathbf{W}_C^\top \mathbf{W}_C)^{-1} \mathbf{W}_C^\top$  and  $\mathbf{W}_C$  has full column rank. We calculate the gradient update paths of the two  
 10 optimization methods respectively. For the original gradient of  $\mathbf{W}_C$  and  $\mathbf{W}_T$ , we have the following:

$$\frac{\partial L}{\partial \mathbf{W}_C} = \frac{\partial L}{\partial C'} \frac{\partial C'}{\partial \mathbf{W}_C} = (\mathbf{W}_T \mathbf{h} + \mathbf{b}_T) \left( \frac{\partial L}{\partial C'} \right)^\top, \quad \frac{\partial L}{\partial \mathbf{W}_T} = \frac{\partial L}{\partial C'} \frac{\partial C'}{\partial \mathbf{W}_T} = \mathbf{W}_C \frac{\partial L}{\partial C'} \mathbf{h}^\top. \quad (2)$$

11 The detailed derivation of these two equations is provided in the Appendix. For ease of understanding,  
 12 the matrix of two equations are  $\nabla_{\mathbf{W}_C} L = (\mathbf{W}_T \mathbf{h} + \mathbf{b}_T) \cdot (\nabla_{C'} L)^\top$  and  $\nabla_{\mathbf{W}_T} L = \mathbf{W}_C \cdot \nabla_{C'} L \cdot \mathbf{h}^\top$ .  
 13 Then, the gradient of  $\mathbf{b}_T$  is  $\nabla_{\mathbf{b}_T} L = \mathbf{W}_C \cdot \nabla_{C'} L$ . For the classifier  $C'$ , we calculate the gradient of  
 14  $\mathbf{W}_{C'}$  and  $\mathbf{b}_{C'}$  using  $\nabla_{\mathbf{W}_{C'}} L = \mathbf{h} \cdot (\nabla_{C'} L)^\top$  and  $\nabla_{\mathbf{b}_{C'}} L = \nabla_{C'} L$ . According to  $\mathbf{W}_{C'} = \mathbf{W}_T^\top \mathbf{W}_C$   
 15 and  $\mathbf{b}_{C'} = \mathbf{W}_C^\top \mathbf{b}_T$ , we analyze the gradient propagation using the chain rule:

$$\begin{aligned} \Delta \mathbf{W}_{C'} &= \mathbf{W}_T^\top \Delta \mathbf{W}_C + (\Delta \mathbf{W}_T)^\top \mathbf{W}_C \\ &= \mathbf{W}_T^\top \left( \eta (\mathbf{W}_T \mathbf{h} + \mathbf{b}_T) \cdot (\nabla_{C'} L)^\top \right) + \eta \mathbf{h} (\nabla_{C'} L) \mathbf{W}_C^\top \mathbf{W}_C \\ &= \eta \mathbf{h} \cdot (\nabla_{C'} L)^\top, \quad \text{when } \mathbf{W}_C^\top \mathbf{W}_C = \mathbf{I}_k \text{ and } \mathbf{W}_T^\top \mathbf{W}_T = \mathbf{I}_d. \end{aligned} \quad (3)$$

16 For  $\mathbf{b}_{C'}$ , we have the following:

$$\begin{aligned} \Delta \mathbf{b}_{C'} &= \mathbf{W}_C^\top \Delta \mathbf{b}_T + (\Delta \mathbf{W}_C)^\top \mathbf{b}_T \\ &= \eta \mathbf{W}_C^\top \mathbf{W}_C \nabla_{C'} L + \eta (\mathbf{W}_T \mathbf{h} + \mathbf{b}_T)^\top \mathbf{b}_T \nabla_{C'} L, \end{aligned} \quad (4)$$

17 when  $\mathbf{W}_C$  has full column rank and  $\mathbf{b}_T$  is orthogonal to  $\mathbf{W}_T \mathbf{h} + \mathbf{b}_T$ , we can obtain  $\Delta \mathbf{b}_{C'} = \eta \nabla_{C'} L$ ,  
 18 which is consistent with the gradient of single linear classifier  $C'$ .  $\square$

## 19 B Other Experiments and Detail Settings

### 20 B.1 Experimental Setup

21 **Implementation details.** In our experiments, we use 2-layer GCN backbones for DGI and GRACE,  
22 and 2-layer GAT backbone for GraphMAE. For downstream prompt tuning, all classifiers employ  
23 2-layer MLPs. We fine-tune all GPL baselines across all pretrained models. We train for 2000 epochs  
24 with early stopping (patience=20). Following the ProG [1] benchmark settings, we conduct k-shot  
25 sampling evaluations under both in-domain and cross-domain settings with  $k \in \{1, 3, 5\}$ . To ensure  
26 performance reliability, we perform 20 repeated runs for each of 5 fixed random seeds (42, 12345,  
27 23344, 38108, 39788), reporting averaged results over 100 total trials. All of the experiments are  
28 conducted on a server with Xeon(R) Platinum 8352V CPU, 90GB of memory, an RTX 4090 graphics  
29 card, and 24GB of video memory.

### 30 Hyperparameters Settings.

### 31 B.2 Real-world datasets

32 We introduce the details of the 9 commonly used real-world datasets, including homophily and  
33 heterophily graphs as follows, and the statistics of these datasets are shown in Table 1.

- 34 • *Cora* [2], *CiteSeer* [2] and *PubMed* [2] are citation datasets, nodes represent papers, edges  
35 represent citation relationships. Each dimension in the feature corresponds to a word. Labels  
36 are the categories into which the paper is divided.
- 37 • *Cornell* [3], *Texas* [3], and *Wisconsin* [3] are sub-datasets of WebKB [4], which is a webpage  
38 dataset collected from Carnegie Mellon University. Nodes represent web pages, and edges  
39 represent hyperlinks between web pages.
- 40 • *Chameleon* [3] and *Squirrel* [3] are page to page networks on specific topic collected from  
41 Wikipedia [5], nodes represent web pages and edges represent links between web pages.  
42 The average monthly traffic of the web page is converted into five categories to predict.
- 43 • *Actor* [3] is the actor-only induced subgraph of the film-director-actor-writer network. Each  
44 node corresponds to an actor, and the edge between two nodes denotes co-occurrence on the  
45 same Wikipedia page. Node features correspond to some keywords in the Wikipedia pages.  
46 The task is to classify the nodes into five categories in term of words of actor’s Wikipedia.

Table 1: Statistics of real-world datasets.

Dataset	#Nodes	#Edges	#Features	#Classes	#Homophily
Cora	2,708	5,278	1,433	7	0.81
CiteSeer	3,327	4,552	3,703	6	0.74
PubMed	19,717	44,324	500	3	0.80
Cornell	183	298	1,703	5	0.31
Texas	183	325	1,703	5	0.11
Wisconsin	251	515	1,703	5	0.20
Chameleon	2,277	36,101	2,277	5	0.24
Actor	7,600	30,019	932	5	0.22
Squirrel	5,201	217,073	2,089	5	0.22

### 47 B.3 Descriptions of Various Baselines

#### 48 Graph Semi-Supervised Baselines.

- 49 • **GCN** [6]: GCN introduces a spectral graph convolution framework based on localized first-  
50 order Chebyshev filters, utilizing mean-pooling for neighborhood aggregation. It recursively  
51 updates node representations by averaging the features of neighbors and uses learnable  
52 parameters to control the transformation process.

Table 2: In-domain node classification. Accuracy on 3-shot node classification tasks over three pretrained models and nine datasets. The best results in each pretrain strategy are highlighted in **bold**, and the runner-up with an underline.

Pretrain	Methods	Cora	CiteSeer	PubMed	Cornell	Texas	Wisconsin	Chameleon	Actor	Squirrel
DGI	GPPT	52.75 $\pm$ 6.52	45.07 $\pm$ 5.43	59.83 $\pm$ 4.92	37.55 $\pm$ 5.48	34.02 $\pm$ 9.71	35.86 $\pm$ 6.43	22.71 $\pm$ 2.40	19.70 $\pm$ 1.23	21.51 $\pm$ 1.37
	GraphPrompt	64.29 $\pm$ 47.8	<u>57.37</u> $\pm$ 5.26	60.56 $\pm$ 5.37	26.06 $\pm$ 6.24	36.89 $\pm$ 7.56	25.96 $\pm$ 9.75	25.71 $\pm$ 2.68	20.02 $\pm$ 1.39	<u>22.16</u> $\pm$ 2.42
	GPF	65.73 $\pm$ 5.53	55.82 $\pm$ 5.79	<u>62.94</u> $\pm$ 7.71	37.70 $\pm$ 7.22	39.66 $\pm$ 8.29	37.34 $\pm$ 5.70	<b>26.16</b> $\pm$ 3.15	21.84 $\pm$ 2.08	22.01 $\pm$ 2.30
	GPF+	<u>68.17</u> $\pm$ 4.28	54.52 $\pm$ 5.91	<b>64.58</b> $\pm$ 7.07	37.90 $\pm$ 7.53	34.49 $\pm$ 8.99	<u>38.12</u> $\pm$ 6.79	25.88 $\pm$ 2.65	21.81 $\pm$ 2.09	21.83 $\pm$ 2.17
	EdgePrompt	62.65 $\pm$ 3.39	49.49 $\pm$ 4.97	59.56 $\pm$ 3.43	<u>40.26</u> $\pm$ 7.81	41.88 $\pm$ 8.76	36.59 $\pm$ 7.37	25.07 $\pm$ 4.07	21.63 $\pm$ 2.45	21.45 $\pm$ 1.83
	EdgePrompt+	59.30 $\pm$ 4.10	49.25 $\pm$ 5.12	59.60 $\pm$ 3.36	39.98 $\pm$ 6.70	<u>42.55</u> $\pm$ 9.04	37.01 $\pm$ 7.58	25.45 $\pm$ 3.77	<u>22.39</u> $\pm$ 2.80	21.96 $\pm$ 1.87
	Ours	<b>69.07</b> $\pm$ 4.37	<b>61.73</b> $\pm$ 4.15	60.94 $\pm$ 6.46	<b>59.63</b> $\pm$ 5.84	<b>61.44</b> $\pm$ 14.55	<b>68.70</b> $\pm$ 6.99	<u>25.90</u> $\pm$ 3.08	<b>27.32</b> $\pm$ 3.26	<b>24.19</b> $\pm$ 2.35
GRACE	GPPT	54.24 $\pm$ 7.29	51.71 $\pm$ 5.39	57.13 $\pm$ 4.60	36.05 $\pm$ 8.71	33.55 $\pm$ 3.66	37.69 $\pm$ 5.65	31.45 $\pm$ 3.69	20.78 $\pm$ 1.21	24.17 $\pm$ 2.68
	GraphPrompt	<u>67.60</u> $\pm$ 4.90	<u>53.84</u> $\pm$ 8.22	56.60 $\pm$ 7.00	29.80 $\pm$ 7.56	34.82 $\pm$ 8.84	29.66 $\pm$ 8.12	<b>32.46</b> $\pm$ 3.60	20.98 $\pm$ 1.85	<u>24.41</u> $\pm$ 3.18
	GPF	64.09 $\pm$ 4.04	52.45 $\pm$ 4.90	61.93 $\pm$ 6.95	38.75 $\pm$ 7.05	41.76 $\pm$ 9.79	36.11 $\pm$ 3.65	30.23 $\pm$ 3.41	21.61 $\pm$ 2.07	21.28 $\pm$ 3.88
	GPF+	63.91 $\pm$ 5.08	53.24 $\pm$ 6.88	55.88 $\pm$ 5.54	38.40 $\pm$ 5.00	41.37 $\pm$ 8.39	36.63 $\pm$ 5.07	<u>32.07</u> $\pm$ 3.59	19.67 $\pm$ 3.47	23.32 $\pm$ 2.66
	EdgePrompt	60.45 $\pm$ 4.39	48.65 $\pm$ 4.08	57.33 $\pm$ 4.67	<u>42.58</u> $\pm$ 10.88	42.97 $\pm$ 6.22	36.46 $\pm$ 8.73	27.42 $\pm$ 3.19	<u>21.63</u> $\pm$ 1.33	23.14 $\pm$ 2.02
	EdgePrompt+	61.60 $\pm$ 2.95	45.12 $\pm$ 4.53	<u>62.38</u> $\pm$ 6.11	42.11 $\pm$ 9.13	<u>43.83</u> $\pm$ 7.29	39.26 $\pm$ 5.16	27.82 $\pm$ 3.19	21.41 $\pm$ 0.98	23.37 $\pm$ 1.53
	Ours	<b>67.71</b> $\pm$ 5.24	<b>61.93</b> $\pm$ 3.73	<b>66.83</b> $\pm$ 6.14	<b>60.86</b> $\pm$ 8.37	<b>64.22</b> $\pm$ 3.84	<b>67.60</b> $\pm$ 8.57	27.71 $\pm$ 3.66	<b>25.56</b> $\pm$ 1.37	<b>25.22</b> $\pm$ 2.47
GraphMAE	GPPT	57.64 $\pm$ 5.74	40.14 $\pm$ 6.89	56.63 $\pm$ 8.23	35.12 $\pm$ 9.70	38.28 $\pm$ 9.54	<u>40.94</u> $\pm$ 6.23	<u>27.46</u> $\pm$ 2.27	20.06 $\pm$ 2.56	20.58 $\pm$ 1.13
	GraphPrompt	<b>67.49</b> $\pm$ 3.01	57.51 $\pm$ 6.52	62.78 $\pm$ 4.76	23.79 $\pm$ 7.02	29.76 $\pm$ 10.51	27.90 $\pm$ 8.98	23.02 $\pm$ 3.37	21.50 $\pm$ 2.05	<b>26.29</b> $\pm$ 2.34
	GPF	57.91 $\pm$ 4.28	43.44 $\pm$ 12.02	<u>64.32</u> $\pm$ 7.22	36.33 $\pm$ 6.82	38.79 $\pm$ 9.89	36.86 $\pm$ 5.95	27.09 $\pm$ 2.88	21.30 $\pm$ 2.51	20.82 $\pm$ 1.81
	GPF+	56.55 $\pm$ 6.74	44.71 $\pm$ 6.36	60.60 $\pm$ 7.87	<u>38.59</u> $\pm$ 7.84	37.27 $\pm$ 8.10	38.06 $\pm$ 9.06	26.87 $\pm$ 3.29	20.56 $\pm$ 3.21	20.95 $\pm$ 0.95
	EdgePrompt	64.18 $\pm$ 4.20	<u>57.56</u> $\pm$ 6.66	54.32 $\pm$ 7.07	35.42 $\pm$ 6.17	40.95 $\pm$ 8.73	37.31 $\pm$ 5.69	26.60 $\pm$ 4.02	19.66 $\pm$ 4.94	22.05 $\pm$ 1.27
	EdgePrompt+	64.36 $\pm$ 3.89	53.46 $\pm$ 6.12	63.05 $\pm$ 6.35	37.20 $\pm$ 6.09	<u>41.00</u> $\pm$ 8.92	38.80 $\pm$ 5.81	22.10 $\pm$ 2.67	20.59 $\pm$ 4.43	21.72 $\pm$ 0.90
	Ours	<u>66.16</u> $\pm$ 6.69	<b>61.90</b> $\pm$ 2.95 $\pi$	<b>64.62</b> $\pm$ 5.71	<b>59.92</b> $\pm$ 5.06	<b>65.62</b> $\pm$ 2.75	<b>71.60</b> $\pm$ 2.88	<b>27.78</b> $\pm$ 2.17	<b>24.77</b> $\pm$ 1.83	<u>22.82</u> $\pm$ 1.19

- **GAT [7]:** GAT proposes multi-head attention mechanisms to dynamically compute node-specific weights during message passing. It adopts a learnable attention coefficient to quantify the importance of neighbors, thereby achieving adaptive aggregation.

**Graph Pretraining Models.** We introduce the classic graph pretraining models as follows.

- **DGI [8]:** Deep Graph Infomax (DGI) learns node embeddings by maximizing the mutual information (MI) between local node representations and graph representation. It utilizes GCNs to generate node representations, and aggregates node representations into a graph representation. DGI treats the corrupted graph as a negative example and train by identifying the relationship between nodes and graphs, thereby maximizing MI between them.
- **GRACE [9]:** GRACE learns node embeddings by maximizing mutual information between node representations in two augmented views. It generates different views through edge removal and feature masking. It use InfoNCE as loss function, which maximizes the similarity of two augmented nodes generated by the same node and minimizes the similarity of other nodes to train the model.
- **GraphCL [10]:** GraphCL learns graph-level representations by maximizing mutual information between augmented views of graphs. It introduces four graph augmentation types (node dropping, edge perturbation, attribute masking, subgraph sampling) to generate augmented views. The InfoNCE loss maximizes similarities between positive pairs (augmented views of the same graph) while contrasting against negative pairs (other graphs in the batch), corresponding to mutual information maximization between augmented representations and unifies diverse contrastive learning frameworks.
- **GraphMAE [11]:** GraphMAE is a generative self-supervised graph autoencoder that learns robust representations through masked feature reconstruction. It employs a two-stage framework: (1) A GNN-based encoder learns node embeddings from input graphs with randomly masked node features; (2) A GNN decoder reconstructs the masked features using a re-mask decoding strategy, optimized by a scaled cosine error loss that emphasizes directional alignment over magnitude.

**Graph Prompt Learning Baselines.**

- **GPPT [12]:** GPPT introduces a "pre-train, prompt, fine-tune" framework for GNNs, enabling parameter-efficient adaptation through learnable prompt vectors without retraining GNN weights. It pre-trains a GNN encoder with self-supervised objectives, then designs task-specific prompts into the input layer during fine-tuning. The graph prompting function generates token pairs: task tokens are trainable class embeddings, and structure tokens are node representations derived from clustered subgraphs. Node classification is reformulated as linking scores between token pairs, decoupling pretraining from downstream tasks.

Table 3: In-domain node classification. Accuracy on 5-shot node classification tasks over three pretrained models and nine datasets. The best results in each pretrain strategy are highlighted in **bold**, and the runner-up with an underline.

Pretrain	Methods	Cora	CiteSeer	PubMed	Cornell	Texas	Wisconsin	Chameleon	Actor	Squirrel
DGI	GPPT	57.78 $\pm$ 4.46	51.64 $\pm$ 5.06	64.59 $\pm$ 3.68	41.95 $\pm$ 4.57	42.19 $\pm$ 6.56	41.37 $\pm$ 5.85	23.47 $\pm$ 2.98	20.87 $\pm$ 1.24	21.80 $\pm$ 1.47
	GraphPrompt	65.36 $\pm$ 4.72	<u>62.33</u> $\pm$ 2.60	66.83 $\pm$ 6.05	27.94 $\pm$ 6.51	40.91 $\pm$ 7.12	31.20 $\pm$ 7.22	25.98 $\pm$ 3.38	20.38 $\pm$ 1.04	22.82 $\pm$ 2.18
	GPF	66.57 $\pm$ 7.50	60.99 $\pm$ 3.73	<b>68.33</b> $\pm$ 5.03	42.96 $\pm$ 6.01	42.61 $\pm$ 8.83	43.68 $\pm$ 6.29	27.10 $\pm$ 2.94	22.79 $\pm$ 1.56	<u>23.38</u> $\pm$ 2.37
	GPF+	<u>69.10</u> $\pm$ 3.70	57.84 $\pm$ 4.22	68.81 $\pm$ 4.57	43.63 $\pm$ 6.62	43.21 $\pm$ 8.64	45.11 $\pm$ 6.42	<u>27.86</u> $\pm$ 2.74	22.39 $\pm$ 2.01	21.48 $\pm$ 3.01
	EdgePrompt	66.82 $\pm$ 3.62	56.99 $\pm$ 4.12	64.08 $\pm$ 6.27	<u>45.14</u> $\pm$ 5.71	<u>49.10</u> $\pm$ 11.77	47.61 $\pm$ 6.32	25.05 $\pm$ 3.76	<u>23.82</u> $\pm$ 1.87	21.62 $\pm$ 1.53
	EdgePrompt+	67.10 $\pm$ 3.94	56.12 $\pm$ 3.88	62.95 $\pm$ 6.15	43.05 $\pm$ 4.58	46.88 $\pm$ 9.06	<u>50.40</u> $\pm$ 5.49	24.96 $\pm$ 3.74	23.49 $\pm$ 1.99	21.53 $\pm$ 2.15
	Ours	<b>70.58</b> $\pm$ 3.01	<b>65.10</b> $\pm$ 3.15	<b>70.97</b> $\pm$ 4.33	<b>68.02</b> $\pm$ 4.32	<b>67.86</b> $\pm$ 8.36	<b>70.43</b> $\pm$ 9.34	<b>28.04</b> $\pm$ 2.68	<b>28.20</b> $\pm$ 2.66	<b>23.88</b> $\pm$ 2.19
GRACE	GPPT	56.51 $\pm$ 7.10	50.88 $\pm$ 5.62	65.97 $\pm$ 5.75	44.36 $\pm$ 4.88	41.15 $\pm$ 7.09	41.98 $\pm$ 7.60	<u>33.10</u> $\pm$ 3.47	21.36 $\pm$ 2.18	<u>24.70</u> $\pm$ 2.14
	GraphPrompt	68.58 $\pm$ 4.30	52.65 $\pm$ 3.84	65.49 $\pm$ 6.66	35.28 $\pm$ 5.94	38.65 $\pm$ 7.75	33.76 $\pm$ 7.48	32.68 $\pm$ 3.32	21.17 $\pm$ 1.14	22.55 $\pm$ 1.87
	GPF	68.56 $\pm$ 3.98	59.53 $\pm$ 3.91	68.20 $\pm$ 4.59	46.01 $\pm$ 6.72	44.17 $\pm$ 7.43	41.66 $\pm$ 4.84	28.62 $\pm$ 3.26	22.91 $\pm$ 1.49	21.29 $\pm$ 2.57
	GPF+	<u>68.86</u> $\pm$ 3.95	<u>61.51</u> $\pm$ 3.90	68.30 $\pm$ 3.99	45.89 $\pm$ 6.10	40.99 $\pm$ 8.35	45.65 $\pm$ 5.22	29.46 $\pm$ 3.33	22.64 $\pm$ 1.45	24.61 $\pm$ 2.24
	EdgePrompt	63.76 $\pm$ 3.49	51.81 $\pm$ 6.08	68.23 $\pm$ 3.16	48.17 $\pm$ 8.16	<u>54.45</u> $\pm$ 7.50	<u>47.14</u> $\pm$ 6.15	32.04 $\pm$ 3.80	23.17 $\pm$ 1.55	24.22 $\pm$ 1.26
	EdgePrompt+	66.16 $\pm$ 3.38	53.90 $\pm$ 3.03	<u>71.03</u> $\pm$ 2.40	47.66 $\pm$ 8.64	53.98 $\pm$ 6.93	46.46 $\pm$ 5.98	32.44 $\pm$ 3.41	<u>24.46</u> $\pm$ 1.64	24.35 $\pm$ 1.41
	Ours	<b>72.99</b> $\pm$ 3.48	<b>63.64</b> $\pm$ 3.80	<b>74.21</b> $\pm$ 2.81	<b>68.13</b> $\pm$ 4.35	<b>68.36</b> $\pm$ 4.92	<b>71.43</b> $\pm$ 4.58	<b>33.66</b> $\pm$ 1.54	<b>26.68</b> $\pm$ 1.87	<b>26.07</b> $\pm$ 0.84
GraphMAE	GPPT	64.66 $\pm$ 5.47	46.87 $\pm$ 6.52	62.50 $\pm$ 7.81	<u>46.25</u> $\pm$ 4.64	41.04 $\pm$ 6.81	<u>46.10</u> $\pm$ 5.06	26.49 $\pm$ 3.32	20.14 $\pm$ 2.49	20.97 $\pm$ 1.15
	GraphPrompt	69.80 $\pm$ 4.65	49.17 $\pm$ 4.19	<u>67.51</u> $\pm$ 6.93	25.83 $\pm$ 5.77	38.54 $\pm$ 9.35	30.60 $\pm$ 7.55	23.65 $\pm$ 2.67	20.08 $\pm$ 1.65	<b>24.71</b> $\pm$ 2.50
	GPF	72.09 $\pm$ 3.98	52.73 $\pm$ 4.94	65.47 $\pm$ 4.85	41.55 $\pm$ 6.06	43.08 $\pm$ 8.13	42.30 $\pm$ 5.95	28.51 $\pm$ 2.63	22.62 $\pm$ 3.36	21.04 $\pm$ 1.39
	GPF+	63.28 $\pm$ 6.20	55.60 $\pm$ 6.03	65.96 $\pm$ 5.03	44.53 $\pm$ 7.08	39.61 $\pm$ 6.69	42.38 $\pm$ 6.35	26.86 $\pm$ 3.34	20.89 $\pm$ 2.70	21.12 $\pm$ 0.81
	EdgePrompt	67.74 $\pm$ 3.60	<u>62.29</u> $\pm$ 3.24	58.66 $\pm$ 6.36	44.37 $\pm$ 6.27	44.02 $\pm$ 7.88	43.53 $\pm$ 5.40	29.35 $\pm$ 2.20	22.00 $\pm$ 2.67	21.68 $\pm$ 1.32
	EdgePrompt+	<u>73.81</u> $\pm$ 2.00	48.29 $\pm$ 4.39	65.60 $\pm$ 3.68	44.02 $\pm$ 7.48	<u>44.56</u> $\pm$ 7.51	43.14 $\pm$ 5.61	<b>29.84</b> $\pm$ 2.28	21.74 $\pm$ 2.57	21.35 $\pm$ 1.15
	Ours	<b>74.77</b> $\pm$ 2.26	<b>65.74</b> $\pm$ 2.80	<b>70.49</b> $\pm$ 4.77	<b>67.73</b> $\pm$ 3.71	<b>71.02</b> $\pm$ 5.21	<b>73.89</b> $\pm$ 6.62	<u>29.77</u> $\pm$ 2.26	<b>24.96</b> $\pm$ 1.65	<u>23.23</u> $\pm$ 1.21

- **GraphPrompt** [13]: GraphPrompt proposes a unified pretraining and prompting framework for GNNs, bridging the gap between pretraining and downstream tasks through a subgraph similarity-based template. It introduces learnable task-specific prompts that guide the ReadOut operation to dynamically emphasize task-relevant features during subgraph representation aggregation. By mapping both link prediction (pretraining) and node/graph classification (downstream) tasks to subgraph similarity learning, GraphPrompt enables parameter-efficient adaptation via prompt tuning—freezing pre-trained GNN weights while optimizing lightweight prompts.
- **GPF/GPF-plus** [14]: GPF introduces a universal prompt tuning framework for pre-trained GNNs, enabling adaptation to downstream tasks by modifying input feature space rather than model parameters. It injects learnable feature prompts into node attributes. Based on that, GPF-plus trains several independent basis vectors and utilizes attentive aggregation of these basis vectors with the assistance of several learnable linear projections. Theoretically, GPF bridges pretraining and downstream objectives by optimizing prompts to reconstruct optimal graph representations, while maintaining parameter efficiency and compatibility across diverse pretraining strategies.
- **EdgePrompt/EdgePrompt-plus** [15]: EdgePrompt introduces a graph prompt tuning framework for pre-trained GNNs by injecting learnable edge-wise prompts into adjacency matrices. It designs edge-specific trainable vectors to customize message aggregation patterns between nodes. This structural adaptation bridges the objective gap between pretraining and downstream tasks while preserving GNN parameters. EdgePrompt+ enables each edge to learn its customized prompt vectors, which is similar to GPF and GPF-plus.

#### Multi-Domain Graph Pretrain Baselines.

- **GCOPE** [16]: GCOPE proposes a cross-domain graph pretraining framework that unifies diverse graph structures by introducing learnable "coordinators" to align various datasets. These coordinators interconnect isolated source datasets into a unified large-scale graph, enabling joint pretraining with objectives. During pretraining, GCOPE learns transferable representations by balancing shared multi-domain knowledge and domain-specific features through latent alignment strategies. The framework supports flexible transfer via fine-tuning or graph prompting while maintaining parameter efficiency.
- **MDGPT** [17]: MDGPT introduces a dual-prompt framework for downstream adaptation: a unifying prompt transfers broadly learned cross-domain knowledge by aligning target domains with the pre-trained prior, and a mixing prompt enables fine-grained domain-specific alignment through learnable projections. MDGPT bridges pretraining and downstream tasks by optimizing domain-invariant representations via self-supervised objectives on multi-domain data.

- **MDGFM** [18]: MDGFM integrates multiple source domains during pretraining, leveraging contrastive learning to maximize mutual information between multi-view graph augmentations. The topology-aware refinement process, which aligns different graph topologies into a unified semantic space via meta-prompts (for global knowledge transfer) and task-specific prompts (for domain adaptation).

#### B.4 3/5-shot node classification on different pretrained models

We further conduct 3-shot and 5-shot node classification experiments on the same nine datasets, based on three different pretrained models, as shown in Table 2 and Table 3. Consistent with the 1-shot results, our method outperforms existing GPL approaches across most datasets under different pretrained model settings. However, the baselines become more competitive in these scenarios, with each achieving second-best performance on certain datasets. Another observation is that, as the number of shots increases, the performance discrepancies among models under different pretrained settings become more pronounced, especially on datasets such as *CiteSeer* and *Chameleon*. In comparison to the GPL baselines, our method demonstrates significantly more stable performance across all settings.

Table 4: 1-Shot Hyperparameter Settings of UniPrompt for Different Pretrained Models

Pretrained Model	Dataset	hidden_dim	up_lr	up_wd	down_lr	down_wd	$k$	$\tau$
DGI-Pretrained	Cora	256	0.001	0.00001	0.05	0.00005	50	0.99999
	CiteSeer	256	0.0005	0.00001	0.05	0.00005	50	0.9999
	PubMed	256	0.0005	0.00001	0.05	0.00005	1	0.9999
	Cornell	256	0.0001	0.00001	0.005	0.00005	50	0.9999
	Texas	256	0.01	0.00001	0.005	0.00005	50	0.9999
	Wisconsin	256	0.0001	0.00001	0.0005	0.00005	50	0.9999
	Chameleon	256	0.01	0.00001	0.001	0.00005	50	0.9999
	Actor	256	0.00005	0.00001	0.005	0.00005	50	0.9999
	Squirrel	256	0.0005	0.00001	0.05	0.00005	50	0.9999
GRACE-Pretrained	Cora	256	0.001	0.00001	0.005	0.00005	50	0.9999
	CiteSeer	256	0.05	0.00001	0.001	0.00005	50	0.9999
	PubMed	256	0.01	0.00001	0.05	0.00005	1	0.9999
	Cornell	256	0.0001	0.00001	0.01	0.00005	50	0.9999
	Texas	256	0.0001	0.00001	0.00005	0.00005	50	0.9999
	Wisconsin	256	0.001	0.00001	0.00005	0.00005	50	0.9999
	Chameleon	256	0.01	0.00001	0.0005	0.00005	1	0.99999
	Actor	256	0.00001	0.00001	0.01	0.00005	50	0.9999
	Squirrel	256	0.01	0.00001	0.05	0.00005	50	0.9999
GraphMAE-Pretrained	Cora	256	0.0005	0.00001	0.0005	0.00005	50	0.9999
	CiteSeer	256	0.001	0.00001	0.0005	0.00005	50	0.9999
	PubMed	256	0.001	0.00001	0.0001	0.00005	1	0.999
	Cornell	256	0.00005	0.00001	0.05	0.00005	50	0.9999
	Texas	256	0.00001	0.00001	0.0005	0.00005	50	0.9999
	Wisconsin	256	0.00005	0.00001	0.01	0.00005	50	0.9999
	Chameleon	256	0.0005	0.00001	0.001	0.00005	50	0.99999
	Actor	256	0.005	0.00001	0.05	0.00005	50	0.9999
	Squirrel	256	0.005	0.00001	0.05	0.00005	50	0.9999

#### B.5 Hyperparameter Settings

We also conducted extensive experiments to explore the impact of various hyperparameters on the performance of our model, as shown in Table 4, Table 5, and Table 6, ensuring that our approach achieves robust and consistent results across diverse settings.

## C Related Works

**Graph Pretraining.** Graph pretraining has emerged as a powerful paradigm for learning generalizable and transferable representations from large-scale unlabeled graph data, aiming to mitigate the dependency on labeled data in downstream tasks. Unlike traditional supervised methods that require extensive manual annotations [6, 7], graph pretraining leverages self-supervised strategies

to capture structural and semantic patterns in graphs. Graph Self-Supervised Learning (GSSL) currently has attracted widespread attention in the academic community, which mainly designs self-supervised objective functions to train the model based on maximizing Mutual Information (MI). As a classic paradigm, DGI [8] maximizes mutual information between node representations and the global summary of the graph. GGD [19] further explores the DGI, summarizing it as a group classification task, greatly reducing the computational time overhead. GRACE [9] introduces Graph Contrastive Learning (GCL) by augmenting the input graph and aligning node embeddings across different views, and optimizes using InfoNCE [20] loss. Based on this loss function, a large amount of instance-discrimination-based GCLs have proposed [21, 22, 23, 24], improving the sampling strategy and achieving improvement. BGRL [25] adopts BYOL [26], which trains the online encoder by predicting the target encoder to generate efficient node representations. This backbone is followed by some recent works, such as AFGRL [27], SGCL [28].

Table 5: 3-Shot Hyperparameter Settings of UniPrompt for Different Pretrained Models

Pretrained Model	Dataset	hidden_dim	up_lr	up_wd	down_lr	down_wd	k	$\tau$
DGI-Pretrained	Cora	256	0.0005	0.00001	0.05	0.00005	10	0.9999
	CiteSeer	256	0.0005	0.00001	0.05	0.00005	10	0.9999
	PubMed	256	0.0001	0.00001	0.05	0.00005	50	0.9999
	Cornell	256	0.001	0.00001	0.01	0.00005	50	0.9999
	Texas	256	0.0001	0.00001	0.00005	0.00005	50	0.9999
	Wisconsin	256	0.005	0.00001	0.0001	0.00005	50	0.9999
	Chameleon	256	0.00001	0.00001	0.05	0.00005	10	0.999
	Actor	256	0.00001	0.00001	0.01	0.00005	50	0.9999
	Squirrel	256	0.0005	0.00001	0.01	0.00005	50	0.99
GRACE-Pretrained	Cora	256	0.001	0.00001	0.05	0.00005	50	0.9999
	CiteSeer	256	0.00001	0.00001	0.05	0.00005	50	0.9999
	PubMed	256	0.01	0.00001	0.05	0.00005	1	0.9999
	Cornell	256	0.00001	0.00001	0.0001	0.00005	50	0.9999
	Texas	256	0.00005	0.00001	0.0001	0.00005	50	0.9999
	Wisconsin	256	0.0001	0.00001	0.0005	0.00005	50	0.999
	Chameleon	256	0.001	0.00001	0.001	0.00005	50	0.9999
	Actor	256	0.005	0.00001	0.01	0.00005	50	0.9999
	Squirrel	256	0.005	0.00001	0.05	0.00005	50	0.99999
GraphMAE-Pretrained	Cora	256	0.0005	0.00001	0.05	0.00005	1	0.99999
	CiteSeer	256	0.001	0.00001	0.05	0.00005	50	0.9999
	PubMed	256	0.0001	0.00001	0.05	0.00005	10	0.9999
	Cornell	256	0.00005	0.00001	0.005	0.00005	50	0.9999
	Texas	256	0.00005	0.00001	0.0005	0.00005	50	0.9999
	Wisconsin	256	0.00001	0.00001	0.00005	0.00005	50	0.9999
	Chameleon	256	0.001	0.00001	0.001	0.00005	50	0.9999
	Actor	256	0.001	0.00001	0.05	0.00005	50	0.9999
	Squirrel	256	0.001	0.00001	0.05	0.00005	5	0.99999

**Graph Prompt Learning.** Graph prompt learning aims to address the gap between pretrained models and downstream tasks by introducing tunable components into the inputs, model parameters, or outputs of pretrained models. This approach facilitates the alignment of the pretraining domain with the target domain, thereby improving performance in downstream tasks, particularly in few-shot fine-tuning scenarios. GPPT [12] introduces task and structure tokens to reformulate node classification as link prediction, leveraging pre-trained edge-level models. It pioneers prompt tuning for graph tasks but is limited to binary edge prediction. GPF [14] proposes feature-based prompts by adding learnable vectors to node features, enabling lightweight adaptation for graph classification. Later extended to GPF-Plus, which employs multiple basis vectors for richer expressiveness. All in One [29] unifies node-, edge-, and graph-level tasks into graph-level contrastive learning via learnable subgraph prompts. It employs meta-learning for cross-task generalization and highlights prompt-as-operation for structural flexibility. GraphPrompt [13] aligns downstream tasks with graph similarity objectives through task-specific prompts. It integrates prompts into graph pooling layers, enabling compatibility with diverse pre-training strategies. PRODIGY [30] introduces task graphs to unify pre-training and downstream tasks via in-context learning. It avoids parameter tuning by reformulating tasks as link prediction between data and label tokens. GraphControl [31] aligns cross-domain graphs via conditional prompts inspired by ControlNet [32], enabling semantic consistency in transfer learning. Although various graph prompt strategies have advanced the field, there remains

no unified understanding of how these prompts interact with pretrained models, which is the problem our work tries to explain and solve.

Table 6: 5-Shot Hyperparameter Settings of UniPrompt for Different Pretrained Models

Pretrained Model	Dataset	hidden_dim	up_lr	up_wd	down_lr	down_wd	$k$	$\tau$
DGI-Pretrained	Cora	256	0.0001	0.00001	0.05	0.00005	10	0.99999
	CiteSeer	256	0.0001	0.00001	0.05	0.00005	10	0.9999
	PubMed	256	0.0005	0.00001	0.05	0.00005	10	0.99999
	Cornell	256	0.00005	0.00001	0.001	0.00005	50	0.9999
	Texas	256	0.00001	0.00001	0.0001	0.00005	50	0.9999
	Wisconsin	256	0.00001	0.00001	0.0001	0.00005	50	0.9999
	Chameleon	256	0.00001	0.00001	0.05	0.00005	10	0.999
	Actor	256	0.0005	0.00001	0.005	0.00005	50	0.9999
	Squirrel	256	0.0001	0.00001	0.0001	0.00005	50	0.9999
GRACE-Pretrained	Cora	256	0.001	0.00001	0.05	0.00005	50	0.9999
	CiteSeer	256	0.00001	0.00001	0.05	0.00005	50	0.9999
	PubMed	256	0.01	0.00001	0.0001	0.00005	1	0.9999
	Cornell	256	0.00001	0.00001	0.0005	0.00005	50	0.9999
	Texas	256	0.00005	0.00001	0.0005	0.00005	50	0.9999
	Wisconsin	256	0.0001	0.00001	0.0001	0.00005	50	0.9999
	Chameleon	256	0.005	0.00001	0.05	0.00005	50	0.99999
	Actor	256	0.005	0.00001	0.05	0.00005	50	0.9999
	Squirrel	256	0.005	0.00001	0.05	0.00005	50	0.99999
GraphMAE-Pretrained	Cora	256	0.005	0.00001	0.0005	0.00005	1	0.9999
	CiteSeer	256	0.001	0.00001	0.05	0.00005	10	0.9999
	PubMed	256	0.0001	0.00001	0.05	0.00005	1	0.9999
	Cornell	256	0.00005	0.00001	0.0005	0.00005	50	0.9999
	Texas	256	0.00005	0.00001	0.0005	0.00005	50	0.9999
	Wisconsin	256	0.00001	0.00001	0.0005	0.00005	50	0.9999
	Chameleon	256	0.001	0.00001	0.05	0.00005	50	0.9999
	Actor	256	0.01	0.00001	0.05	0.00005	50	0.9999
	Squirrel	256	0.005	0.00001	0.0001	0.00005	50	0.9999

## D Limitations

Despite the excellent results of our proposed UniPrompt, there are several limitations to consider:

- **Limited Integration with LLMs:** Our proposed method currently lacks consideration of LLMs as encoders, limiting the exploration of their effectiveness.
- **Hyperparameter Dependency:** Although the method is simple and effective, it still requires tuning two hyperparameters,  $\tau$  and  $k$ , across different datasets.
- **Limited Task Coverage:** The downstream evaluation focuses only on node classification, without exploring other tasks such as graph classification.

## E Broader Impacts

The introduction of UniPrompt represents a significant advancement in the field of graph prompt learning. The broader impact of this work includes:

- **Further Integration with LLMs:** Our proposed method and theory can be combined with LLMs to further validate the effectiveness of our approach.
- **Extension to Multiple Tasks:** Based on our method, we can further explore the performance of UniPrompt on graph classification tasks.
- **Expansion to New Scenarios:** Incorporating LLMs into our model enables effective unified task formulations and makes efficient zero-shot classification tasks feasible.

## References

- [1] Chenyi Zi, Haihong Zhao, Xiangguo Sun, Yiqing Lin, Hong Cheng, and Jia Li. Prog: A graph prompt learning benchmark. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024.
- [2] Zhilin Yang, William W. Cohen, and Ruslan Salakhutdinov. Revisiting semi-supervised learning with graph embeddings. In *Proceedings of the 33rd International Conference on Machine Learning, ICML 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 40–48. JMLR.org, 2016.
- [3] Hongbin Pei, Bingzhe Wei, Kevin Chen-Chuan Chang, Yu Lei, and Bo Yang. Geom-gcn: Geometric graph convolutional networks. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.
- [4] Alberto P García-Plaza, Víctor Fresno, Raquel Martínez Unanue, and Arkaitz Zubiaga. Using fuzzy logic to leverage html markup for web page representation. *IEEE Transactions on Fuzzy Systems*, 25(4):919–933, 2016.
- [5] Benedek Rozemberczki, Carl Allen, and Rik Sarkar. Multi-scale attributed node embedding. *J. Complex Networks*, 9(2), 2021.
- [6] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017.
- [7] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph Attention Networks. *International Conference on Learning Representations*, 2018.
- [8] Petar Velickovic, William Fedus, William L. Hamilton, Pietro Liò, Yoshua Bengio, and R. Devon Hjelm. Deep graph infomax. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.
- [9] Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. Deep graph contrastive representation learning. *CoRR*, abs/2006.04131, 2020.
- [10] Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. Graph contrastive learning with augmentations. In *Advances in Neural Information Processing Systems, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.
- [11] Zhenyu Hou, Xiao Liu, Yukuo Cen, Yuxiao Dong, Hongxia Yang, Chunjie Wang, and Jie Tang. Graphmae: Self-supervised masked graph autoencoders. In *KDD '22: The 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, August 14 - 18, 2022*, pages 594–604. ACM, 2022.
- [12] Mingchen Sun, Kaixiong Zhou, Xin He, Ying Wang, and Xin Wang. GPPT: graph pre-training and prompt tuning to generalize graph neural networks. In *KDD '22: The 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, August 14 - 18, 2022*, pages 1717–1727. ACM, 2022.
- [13] Zemin Liu, Xingtong Yu, Yuan Fang, and Xinming Zhang. Graphprompt: Unifying pre-training and downstream tasks for graph neural networks. In *Proceedings of the ACM Web Conference 2023, WWW 2023, Austin, TX, USA, 30 April 2023 - 4 May 2023*, pages 417–428. ACM, 2023.
- [14] Taoran Fang, Yunchao Zhang, Yang Yang, Chunping Wang, and Lei Chen. Universal prompt tuning for graph neural networks. In *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.
- [15] Xingbo Fu, Yinhan He, and Jundong Li. Edge prompt tuning for graph neural networks. In *The Thirteenth International Conference on Learning Representations*, 2025.



- [16] Haihong Zhao, Aochuan Chen, Xiangguo Sun, Hong Cheng, and Jia Li. All in one and one for all: A simple yet effective method towards cross-domain graph pretraining. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2024, Barcelona, Spain, August 25-29, 2024*, pages 4443–4454. ACM, 2024.
- [17] Xingtong Yu, Chang Zhou, Yuan Fang, and Xinming Zhang. Text-free multi-domain graph pre-training: Toward graph foundation models. *arXiv preprint arXiv:2405.13934*, 2024.
- [18] Shuo Wang, Bokui Wang, Zhixiang Shen, Boyan Deng, and Zhao Kang. Multi-domain graph foundation models: Robust knowledge transfer via topology alignment. *arXiv preprint arXiv:2502.02017*, 2025. Accepted for ICML 2025.
- [19] Yizhen Zheng, Shirui Pan, Vincent C. S. Lee, Yu Zheng, and Philip S. Yu. Rethinking and scaling up graph contrastive learning: An extremely efficient approach with group discrimination. In *Advances in Neural Information Processing Systems, NeurIPS 2022*, 2022.
- [20] Aäron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *CoRR*, abs/1807.03748, 2018.
- [21] Jun Xia, Lirong Wu, Ge Wang, Jintao Chen, and Stan Z. Li. Progc1: Rethinking hard negative mining in graph contrastive learning. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 24332–24346. PMLR, 2022.
- [22] Wen-Zhi Li, Chang-Dong Wang, Hui Xiong, and Jian-Huang Lai. Homogc1: Rethinking homophily in graph contrastive learning. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2023, Long Beach, CA, USA, August 6-10, 2023*, pages 1341–1352. ACM, 2023.
- [23] Hengrui Zhang, Qitian Wu, Yu Wang, Shaofeng Zhang, Junchi Yan, and Philip S. Yu. Localized contrastive learning on graphs. *CoRR*, abs/2212.04604, 2022.
- [24] Mengyue Liu, Yun Lin, Jun Liu, Bohao Liu, Qinghua Zheng, and Jin Song Dong. B<sup>2</sup>-sampling: Fusing balanced and biased sampling for graph contrastive learning. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2023*, pages 1489–1500. ACM, 2023.
- [25] Shantanu Thakoor, Corentin Tallec, Mohammad Gheshlaghi Azar, Rémi Munos, Petar Veličković, and Michal Valko. Bootstrapped representation learning on graphs. In *ICLR 2021 Workshop on Geometrical and Topological Representation Learning*, 2021.
- [26] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Ávila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent - A new approach to self-supervised learning. In *Advances in Neural Information Processing Systems*, 2020.
- [27] Namkyeong Lee, Junseok Lee, and Chanyoung Park. Augmentation-free self-supervised learning on graphs. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022*, pages 7372–7380. AAAI Press, 2022.
- [28] Wangbin Sun, Jintang Li, Liang Chen, Bingzhe Wu, Yatao Bian, and Zibin Zheng. Rethinking and simplifying bootstrapped graph latents. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining, WSDM 2024, Merida, Mexico, March 4-8, 2024*, pages 665–673. ACM, 2024.
- [29] Xiangguo Sun, Hong Cheng, Jia Li, Bo Liu, and Jihong Guan. All in one: Multi-task prompting for graph neural networks. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2023, Long Beach, CA, USA, August 6-10, 2023*, pages 2120–2131. Long Beach, CA, USA, 2023. ACM.

- 291 [30] Qian Huang, Hongyu Ren, Peng Chen, Gregor Krzmar, Daniel Zeng, Percy Liang, and Jure  
292 Leskovec. PRODIGY: enabling in-context learning over graphs. In Alice Oh, Tristan Naumann,  
293 Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine, editors, *Advances in Neural  
294 Information Processing Systems 36: Annual Conference on Neural Information Processing  
295 Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.
- 296 [31] Yun Zhu, Yaoke Wang, Haizhou Shi, Zhenshuo Zhang, Dian Jiao, and Siliang Tang. Graph-  
297 control: Adding conditional control to universal graph pre-trained models for graph domain  
298 transfer learning. In *Proceedings of the ACM on Web Conference 2024, WWW 2024, Singapore,  
299 May 13-17, 2024*, pages 539–550. ACM, 2024.
- 300 [32] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image  
301 diffusion models. In *Proceedings of the IEEE/CVF international conference on computer vision*,  
302 pages 3836–3847, 2023.