GML Final Project

CSIE 113 陳宥橋

Info

- Drive : https://drive.google.com/drive/u/0/folders/1YuJvVqoef-LY1Ln0S4A2bABwk8AdWT8D
- google drive video linke:
  - Full ppt video (with all paper discussion)(over 15 min): https://drive.google.com/file/d/1igc28zJt4aRj__0T3c2Brmn0QDzwh463/view?usp=share_link
  - Part of ppt video: https://drive.google.com/file/d/1n_lW8oIRtYzAQJHzgYIQ6Un4LxqbUTCP/view?usp=share_link
- slide : https://docs.google.com/presentation/d/17OwGeC-KwttZikeY2mEP_KQRouBZaQeff77j6Rbq33k/edit?usp=sharing

Content

Introduction

Background

In most graph problems today, only one problem can be handled at a time, and if tasks need to be performed on different graphs, fine-tuning is required. Is there a chance to handle different graphs and tasks without fine-tuning?

Motivations

In the paper I read, the main achievement is that this model can accept other graphs as input without fine-tuning, and still achieve high performance. I am wondering if it's possible to modify it to not only handle multiple graphs, but also process multiple different tasks simultaneously.

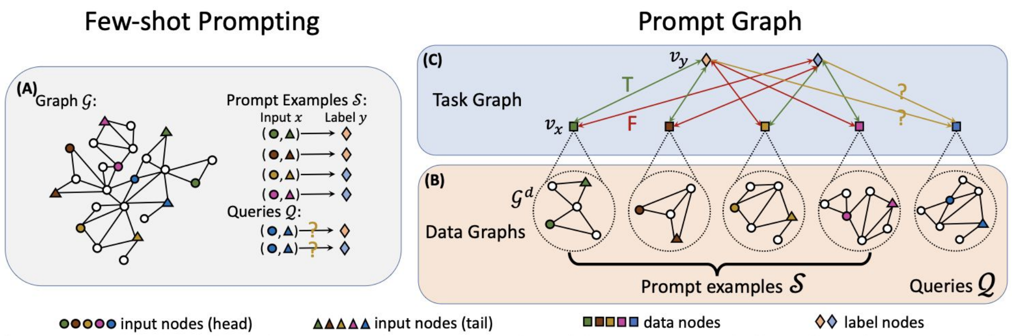


Figure 1: In-context few-shot prompting over graphs with prompt graph for *edge classification* in

Technical Challenges

Because the graph requires different processing for different tasks, and each structure of the graph itself is different, it is difficult to have a unified model that can handle different tasks and different structures simultaneously. So If this way works, we can save a lot of time and money to do multiple graph with a variety of tasks.

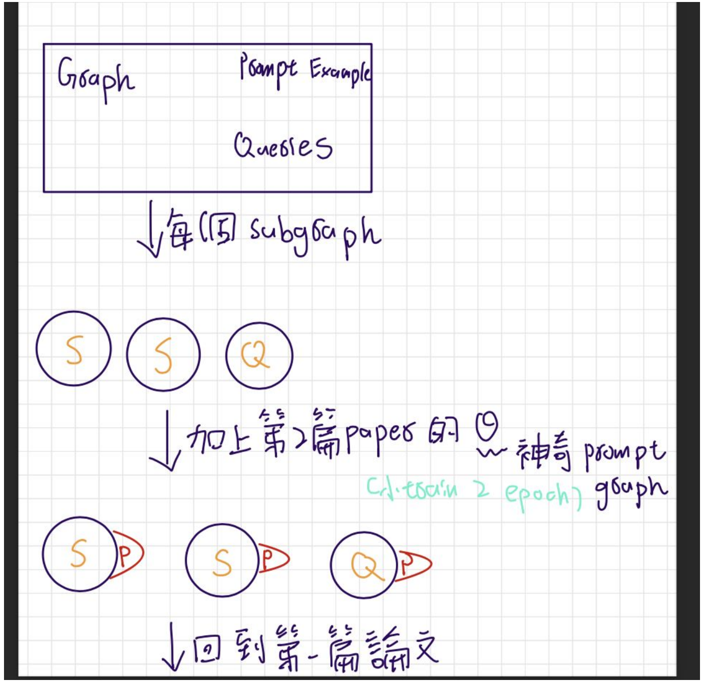
Problem Statement

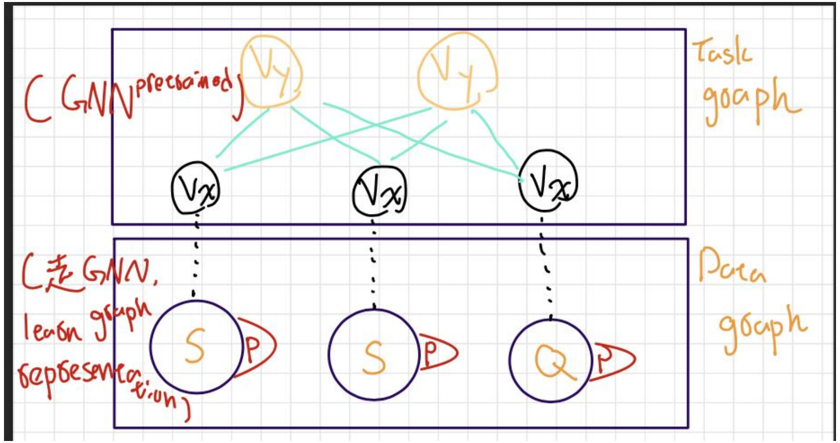
The purpose of this method is to make the model can predict node classification or link prediction on unseen graph.

Input : An unseen graph (Adjecent Matrix, Nodes) with target node or link

Output : the label of node or the result of link prediction

Preliminary Methods



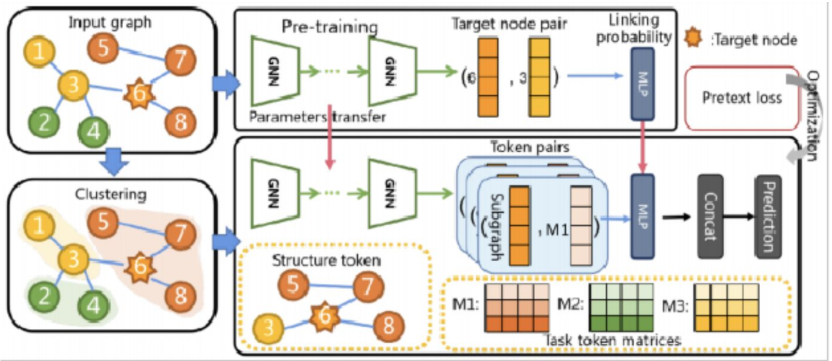


Contribution

Although the data obtained in this experiment are not very impressive, I believe my idea can serve as a stepping stone for others' experiments, allowing the problems I've raised to be solved.

Related Works

Related Work



1. GPPT: Graph Pre-training and Prompt Tuning to Generalize Graph Neural Networks
   a. 2022 KDD
   b. The first paper use prompt in GNN to realize self-supervised learning
   c. the inherent training objective gap between the pretext and downstream tasks
   d. Method: novel transfer learning paradigm to generalize GNNs
      i. Pre-train model : link prediction
      ii. downstream model : node prediction

Related Work

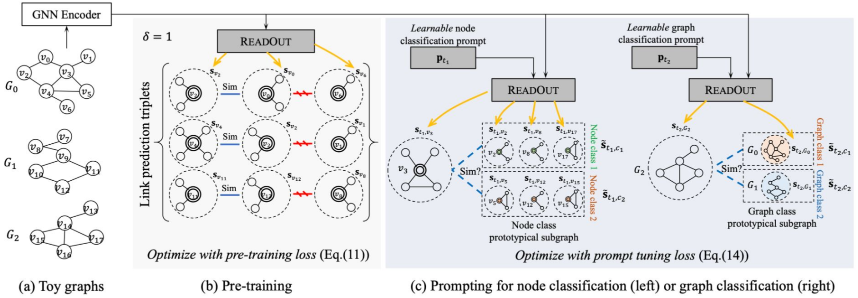
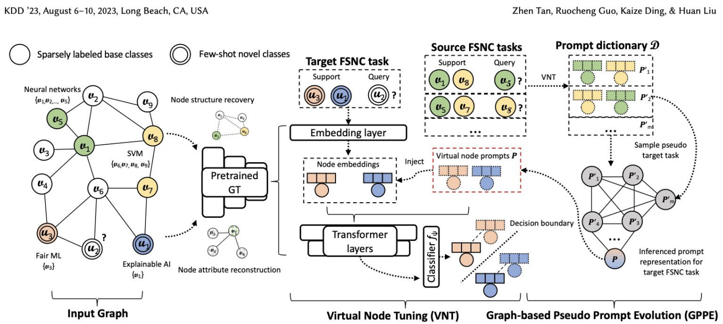


Figure 2: Overall framework of GraphPrompt.

2. GraphPrompt: Unifying Pre-Training and Downstream Tasks for Graph Neural Networks

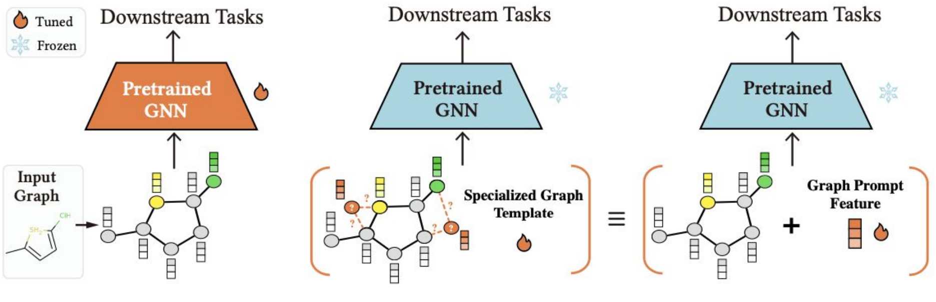
- Target
  - deal with the gap between downstream tasks and pre-train model
  - Can distinguish between different downstream tasks and adjust the model automatically.
- model:
  - Pretrain : link prediction

Related Work



- Virtual Node Tuning for Few-shot Node Classification
  - Target
    - In graph representation learning, a few labeled nodes per class are available for training and base classes have no or limited labeled nodes is a big challenge.
  - Method :
    - utilizes a pretrained graph transformer as the encoder and injects virtual nodes as soft prompts in the embedding space
    - Feed them back to GT and can get the predicted label. We can also use this method to train the last classification layer.

Related Work



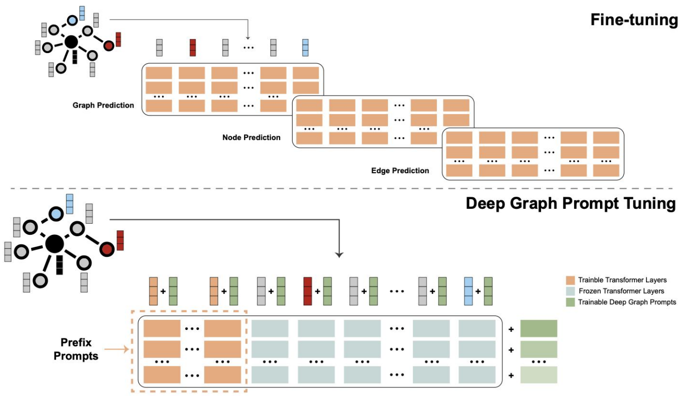
(a) Fine-tuning (b) Specialized Graph Prompt Tuning (c) Universal Graph Prompt Tuning

- Universal Prompt Tuning for Graph Neural Networks
  - Target
    - No prompt-based tuning method is available for models pre-trained using alternative strategies. Furthermore, existing prompt-based tuning methods for GNN models are predominantly designed based on intuition, lacking theoretical guarantees for their effectiveness.
    - To deal with the diversity of graph pre-training strategies, we propose a universal prompt-based tuning method that can be applied to the pre-trained GNN models that employ any pre-training strategy.
  - Method :
    - Given a frozen pre-trained GNN model f, a learnable projection head θ, and a downstream task dataset D, our target is to obtain a task-specific graph prompt. The graph prompt gφ(·) transforms the input graph G into a specific prompted graph gφ(G). And then gφ(G) will replace G as input to the pre-trained GNN model f.
    - During the evaluation stage, the test graph Gtest is first transformed by graph prompt gφ(·), and the resulting prompted graph gφ(Gtest) is processed through the frozen GNN model f.
    - Need to fine-tune specific head

$$p_1, p_2, \ldots p_N \in \mathbb{R}^F$$

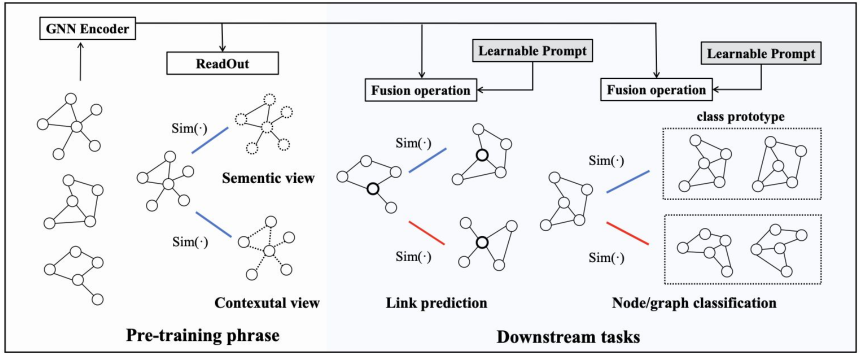
$$\mathbf{X} = \{x_1, x_2, \ldots, x_N\} \quad \mathbf{X}^* = \{x_1 + p_1, x_2 + p_2, \ldots, x_N + p_N\}$$

Related Work



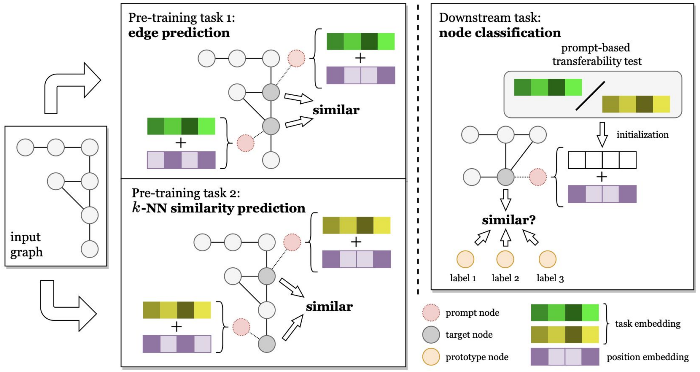
- Deep Prompt Tuning for Graph Transformers
  - Target
    - To deal with the graph transformer problems, the quadratic complexity of self-attention operations and the extensive layering in graph transformer architectures present challenges when ap- plying them to graph based prediction tasks. This challenge arises from the potential overfitting to small datasets, and even reduced-parameter versions may lack the necessary representational richness for complex graph datasets
    - a novel approach serves as an alternative to fine-tuning for leveraging large graph transformer models in downstream graph prediction tasks.
  - Method :
    - We introduce graph prompt tokens to the input graph representations and each transformer layer activation. Additionally, we pre-pend prefix tokens to all transformer layers and to direct the graph transformer in solving downstream node/graph classification tasks by fine-tuning these task-specific tokens.
    - By freezing most of the pre-trained model’s parameters and only updating the concatenated and added prompt tokens, we can fine-tune a large transformer on different downstream datasets.

Related Work



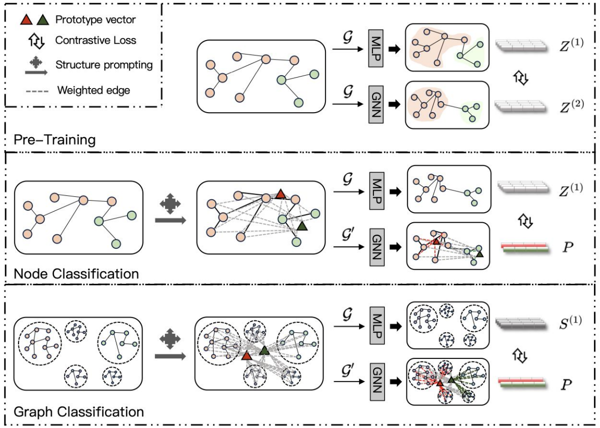
- Prompt Tuning for Multi-View Graph Contrastive Learning
  - Target
    - lacking a framework that can accommodate commonly used graph pre-training methods and downstream tasks.
    - that can accommodate various types of graphs and downstream tasks.
    - hinging on a fusion operation and a learnable prompt design to transfer the pre-trained knowledge to different downstream tasks for improving perfor- mance.
    - we propose a multi-view graph contrastive learning method as pretext and design a prompting tuning for it.
  - Method :
    - We first reformulate graph pre-training and downstream tasks into a common format. Second, we construct multi-view contrasts to capture relevant information of graphs by GNN. Third, we design a prompting tuning method for our multi-view graph contrastive learning method to bridge the gap between pretexts and downsteam tasks.

Related Work



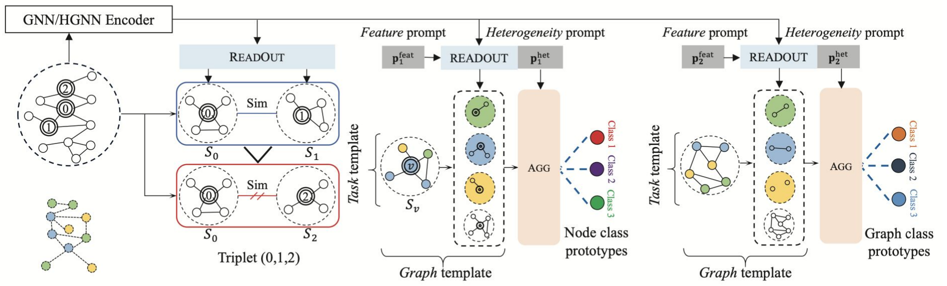
- ULTRA-DP: Unifying Graph Pre-training with Multi-task Graph Dual Prompt
  - Target
    - Traditional way cannot distinguish tasks and results in some transferable task-specific knowledge distortion by each other. Moreover, most GNNs cannot distinguish nodes located in different parts of the graph, making them fail to learn position-specific knowledge and lead to suboptimal performance.
    - we propose a unified framework for graph hybrid pre-training which injects the task identification and position identification into GNNs through a prompt mechanism, namely multi-task graph dual prompt (ULTRA-DP)
  - Method:
    - We will add a node which has information about target node's position and task. Then this prompt node will connect to target node. Finally, we can

Related Work



- Enhancing Graph Neural Networks with Structure-Based Prompt
  - Target
    - the structure information of graph is usually exploited during pre-training for learning node representations, while neglected in the prompt tuning stage for learning task-specific parameters.
  - Method:
    - In particular, SAP 1) employs a dual-view contrastive learning to align the latent semantic spaces of node attributes and graph structure, and 2) incorporates structure information in prompted graph to elicit more pre-trained knowledge in prompt tuning.
    - During the pre-training stage, we will utilize contrastive learning. The input graph will learn its embedding through MLP and GNN. After acquiring embeddings from each method, we will employ contrastive learning and a loss function to train the parameters
    - In the testing stage, we will first add prompt nodes to the graph, with each class having one prompt node. Then, we will process this 'prompt-enhanced' graph using GNN and process the original graph using MLP. Finally, we will evaluate the similarity of representations between the MLP-based and GNN-based views of the graph from different classes to determine the class to which a node or graph belongs

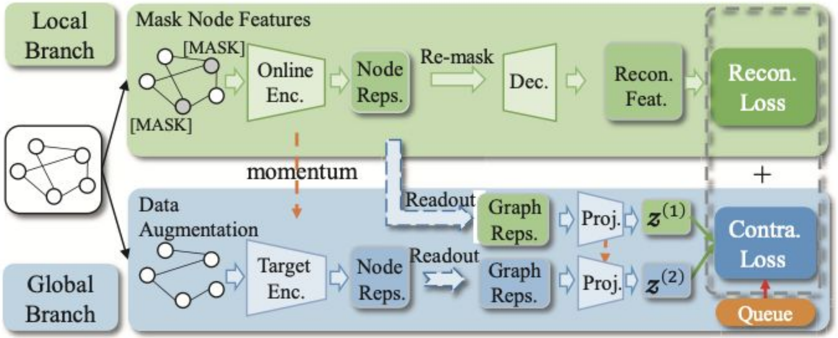
Related Work



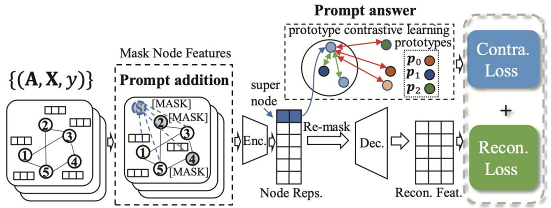
(a) Pre-training graphs (b) Pre-training (link prediction) (c) Downstream node classification (d) Downstream graph classification

- HGPROMPT: Bridging Homogeneous and Heterogeneous Graphs for Few-shot Prompt Learning
    - Target
        - While there has been some early exploration of prompt-based learning on graphs, they primarily deal with homogeneous graphs, ignoring the heterogeneous graphs that are prevalent in downstream applications.
        - a novel pre-training and prompting framework to unify not only pre-training and downstream tasks but also homogeneous and heterogeneous graphs. Moreover, we propose dual-prompt in HGPROMPT to assist a downstream task in locating the most relevant prior to bridge the gaps caused by not only feature variations but also heterogeneity differences across tasks.

Related Work



- SGL-PT: A Strong Graph Learner with Graph Prompt Tuning
  - Target
    - The first is the lack of a strong and universal pre-training task across sundry pre-training methods in graph domain.
    - The second challenge lies in the difficulty of designing a consistent training objective for both pre-training and downstream tasks due to the inherent abstraction of graph data.
  - Method:
    - For local branch, it focuses on learning intra-data relations via a graph masked autoencoder. For global branch, it empowers the pre-training model with instance-wise discriminative ability by graph contrastive learning. Then we propose a non-trivial solution to integrate these two branches effectively.
    - Prompt Addition: Re-formulate downstream task
    - Prompt Answer: Verbalizer-free Class Mapping



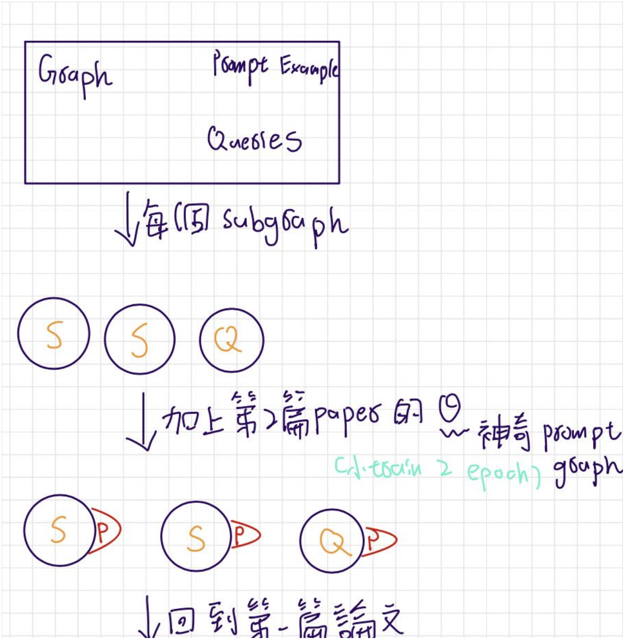
Problem Statement

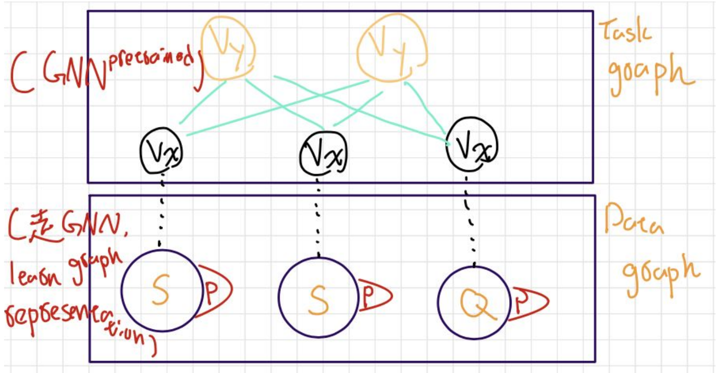
Problem Statement

- The purpose of this method is to make the model can predict node classification or link prediction on unseen graph.
- Input : (G, S, Q)
    - G : Source Graph
            - (V , E , R)
            - V : set of nodes
            - E : set of edges
            - R : set of relations
    - E = Edge
        - E = (u , v , r)
        - u,v $\in$ V , r $\in$ R
    - S = Prompt Example
        - $S = \{(x_i, y_i)\}^{m \cdot k}_{i=1}$
    - Q = Query
        - $Q = \{ x_i \}^{n}_{i=1}$
    - X = (V,E,R)
        - Node Classification : the input node that we aim to make predictions
            - $|V_i| = 1$ and $|E_i| = 0$
        - Edge Classification : consists of (subject, object) pair
            - $|V_i| = 2$ and $|E_i| = 0$
    - y : a set of classes

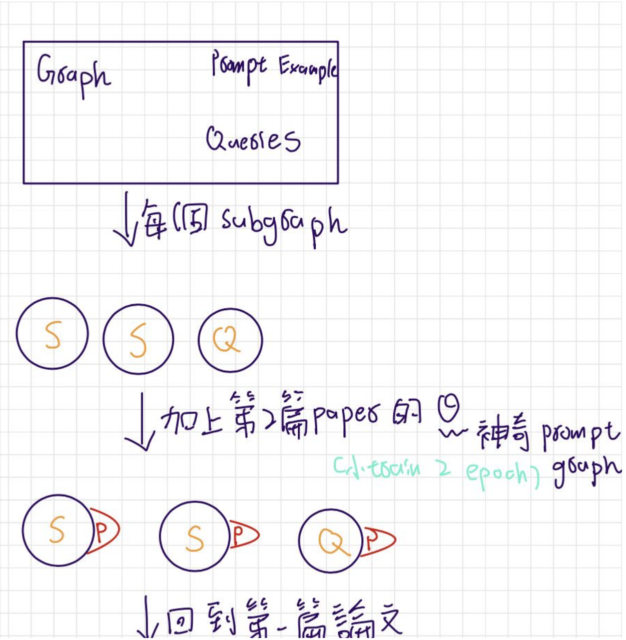
The Proposed Method

Preliminary Methods





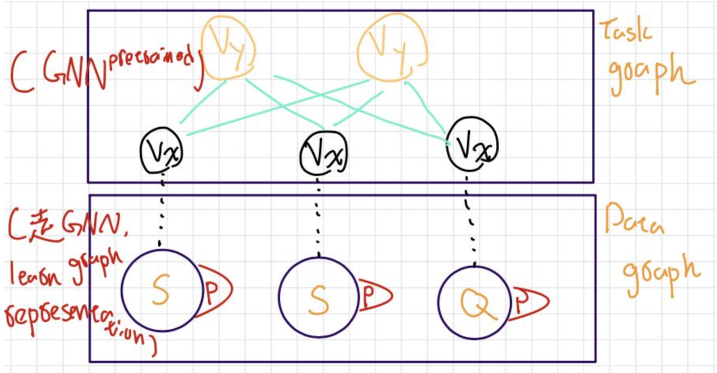
Preliminary Methods



In the Prog paper, the results of my research indicate that his primary approach involves utilizing a pre-trained graph label prediction model. Subsequently, under the condition of maintaining the overall framework unchanged, problems at the edge and node levels are elevated to the level of graph labels using prompts. This allows the model to meet the requirements of the original pre-training and enables it to accomplish diverse tasks.

Therefore, in my approach, after extracting the graph, prompts can be added to elevate it to the level of graph label problems.

Preliminary Methods



Then, using the framework from the original paper, in the data graph section, we employ GNN to learn graph embeddings and utilize the task graph to determine the corresponding label.

Therefore, we need to:

- Utilize meta-learning to learn suitable prompts.
- Train a model capable of extracting graph embeddings from a graph based on query and prompt.

Experiments

Dataset

Pretrain MetaLearning

1. CiteSeer
   - The CiteSeer dataset consists of 3312 scientific publications classified into one of six classes. The citation network consists of 4732 links.

Dataset

- Pretraining :
  - MAG240M :
    - a large scale citation network
    - 122 million nodes and 1.3 billion edges
    - Mainly use to pretrained node classification
  - Wiki :
    - a knowledge graph (KG)
    - 4.8 million nodes and 5.9 million edges
    - Mainly use to pretrained edge classification

Dataset

- Evaluate :
  - arXiv
    - the downstream task is an m-ways node classification task that predicts the paper category.
  - KG dataset :
    - ConceptNet
      - designed to help computers understand the meanings of words that people use.
    - FB15K-237
    - NELL
    - downstream task is an m-ways relation type classification task that predicts the relationship connecting the two input nodes.

Evaluation Settings

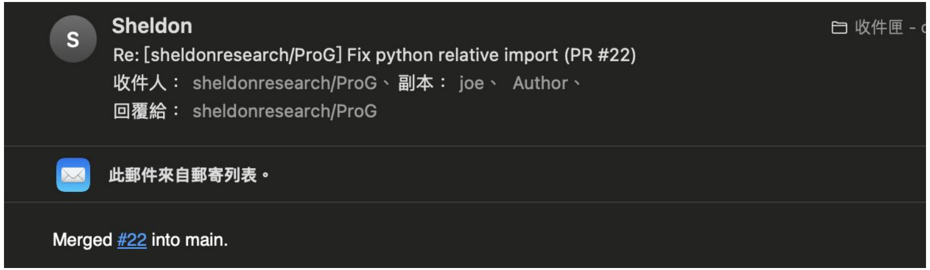
- Meta learning:
  - embedding dimension : 768
  - num class = 2
  - token_num = 10,30,50
  - cross_prune=0.1
  - inner_prune=0.3
  - adapt_lr = 0.01
  - meta_lr = 0.001
  - GNN
    - Gnn type : TransformerConv
    - two layers
    - 768 -> 768 -> 768

Evaluation Settings

- prodigy
  - dataset_len_cap : 50010
  - val_len_cap : 100
  - test_len_cap : 100
  - checkpoint_step : 1000
  - way : 30
  - shot : 3
  - query : 4
  - eval_step : 1000
  - batch_size : 1

Training Procedure

- Fix Bug in two repository
- Put prompt graph into first paper
- Ajust meta learning
- Train new Gnn for meta learning
    - Original (choose target node)
    - graph embedding + pooling
- Train new meta learning
- Put new meta learning in Prodigy
- Adjust Prodigy model
- Train Prodigy


Sheldon

收件匣 -

Re: [sheldonresearch/ProG] Fix python relative import (PR #22)

收件人： sheldonresearch/ProG、副本： joe、 Author、

回覆給： sheldonresearch/ProG

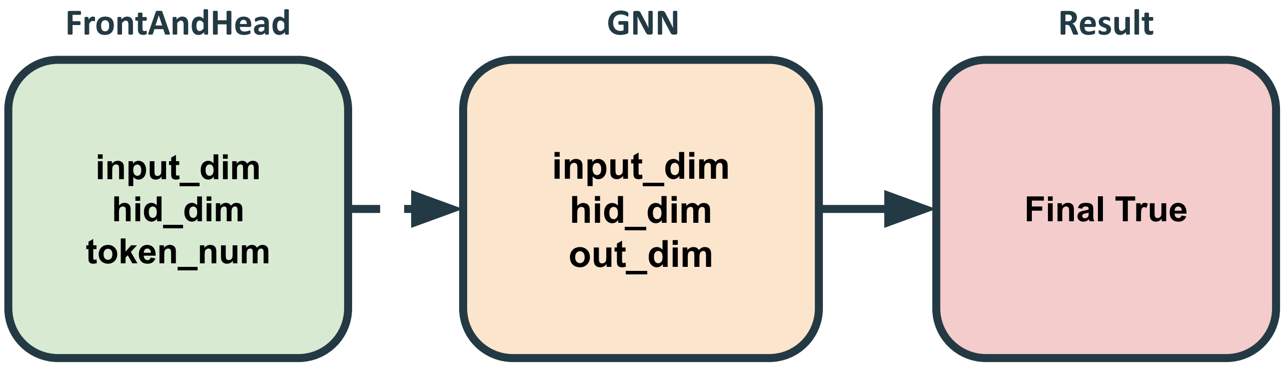
此郵件來自郵寄列表。

Merged #22 into main.

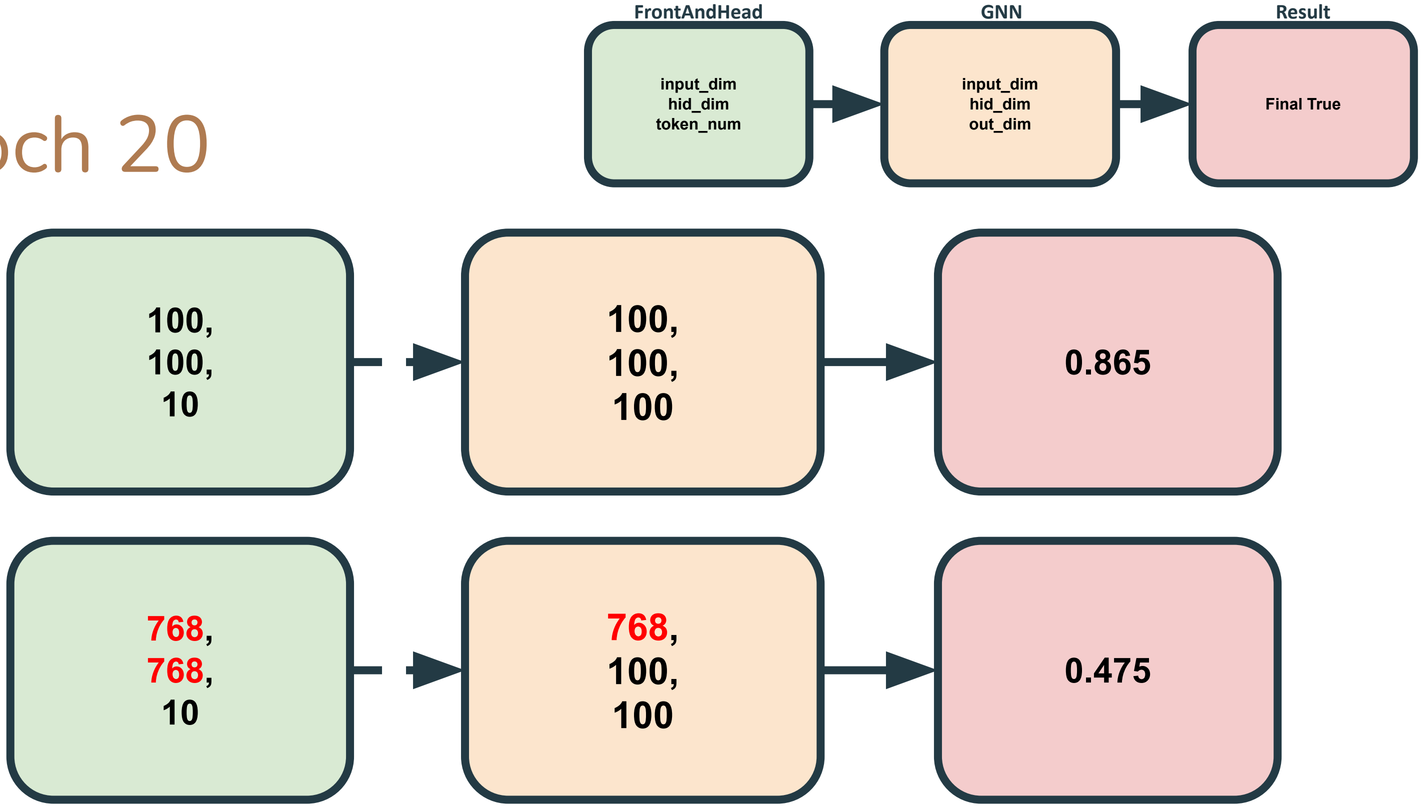
Evaluation Results

- Time
  - Train prodigy all framework : 1d 23 hr
  - Train meta learning : 5 hr

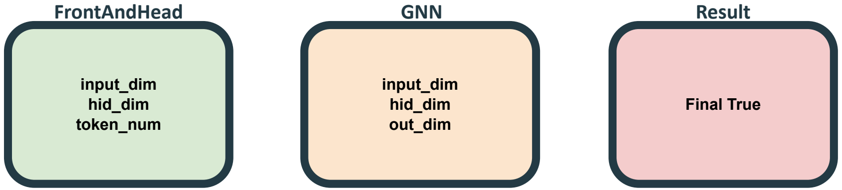
Training result

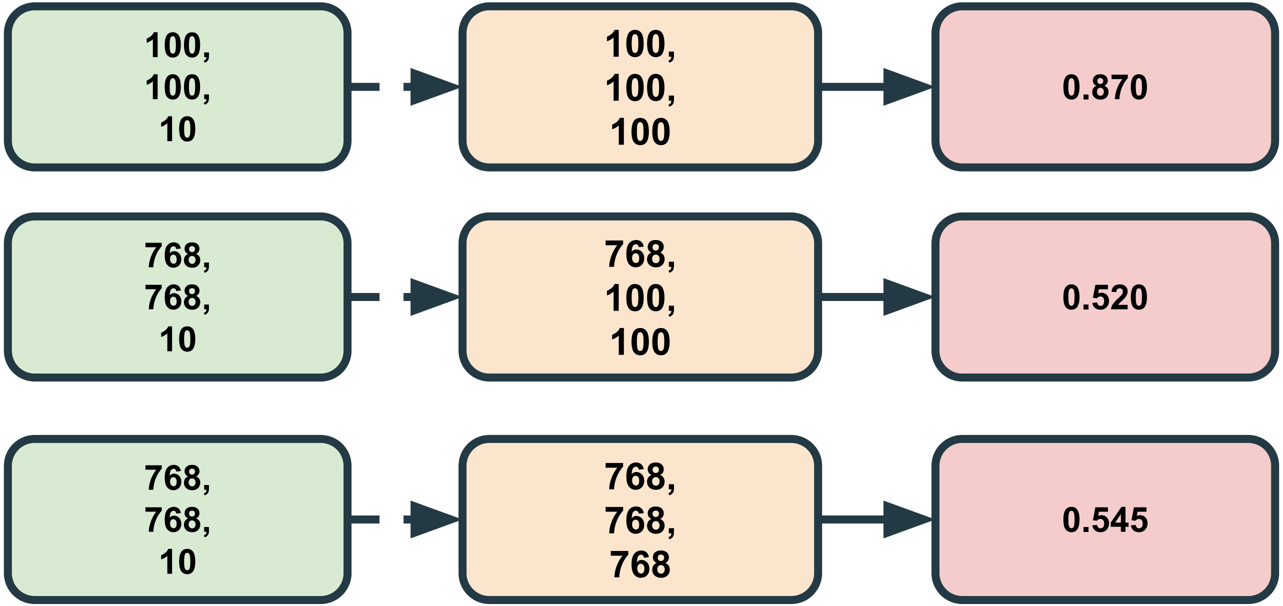


Epoch 20

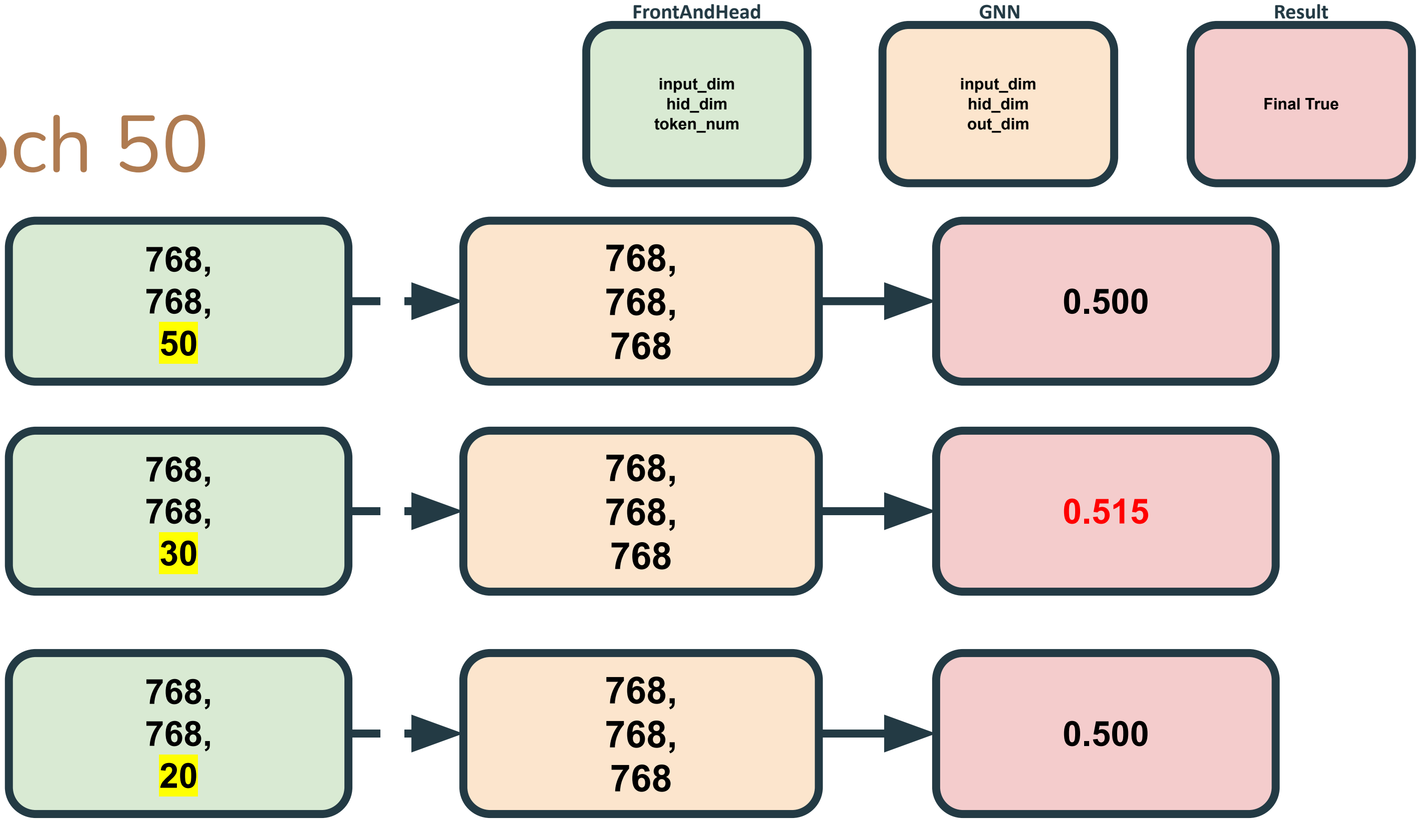

FrontAndHead
input_dim
hid_dim
token_num
GNN
input_dim
hid_dim
out_dim
Result
Final True
100,
100,
10
100,
100,
100
0.865
768,
768,
10
768,
100,
100
0.475

Epoch 50


FrontAndHead
input_dim
hid_dim
token_num
GNN
input_dim
hid_dim
out_dim
Result
Final True


100,
100,
10
100,
100,
100
0.870
768,
768,
10
768,
100,
100
0.520
768,
768,
10
768,
768,
768
0.545

Epoch 50


FrontAndHead
input_dim
hid_dim
token_num
GNN
input_dim
hid_dim
out_dim
Result
Final True
768,
768,
50
768,
768,
768
0.500
768,
768,
30
768,
768,
768
0.515
768,
768,
20
768,
768,
768
0.500

Evaluation Results

- Prodigy Outcome node classification

	prompt	no
original	0.72889	0.72778
original + prompt	0.71267	0.70302
ux + prompt	0.6908	0.72036

- Prodigy Outcome link classification

	prompt	no
original	0.35618	0.418
original + prompt	0.16193	0.23952
ux + prompt	0.2549	0.3006

Evaluation Results

- Prodigy Outcome node classification

	train prompt	no
original	0.68418	0.72778
original + train prompt	0.65889	0.66858
ux + train prompt	0.65724	0.70369

- Prodigy Outcome edge prediction

	train prompt	no
original	0.3213	0.418
original + train prompt	0.09358	0.18502
ux + train prompt	0.26575	0.33037

Conclusions and Discussion

Discussion

1. Meta Learning:
    a. The dimension is too large, failing to achieve the best results
        i. The amount of data is too small to fully learn the knowledge.
        ii. Too many epochs lead to overfitting.
        iii. Adjust the training objectives and strategies.
    b. Use a different GNN
        i. Adjust the GNN
        ii. Directly train using the model we will ultimately use.
    c. The graph is too small
        i. The graph could be replaced with a larger one.

Discussion

- Prodigy plus’ dicussiong :
  - At the very beginning, it can be seen that having a prompt indeed helps to significantly improve scores, proving that prompts are indeed useful.
  - It is observed during training that under the conditions of 'original + prompt' and 'ux + prompt', the 'ux + prompt' performs better. This indicates that a new embedding method is indeed necessary.
  - The performance without training the prompt is better than with training the prompt. The inferred reason is similar to what was discussed earlier: our current prompt, after being trained, contains more information that is unsuitable for our model, leading to poorer results.
  - However, from the later part, we can see that 'ux + train prompt' still has a very good effect. It has the highest score among the changes made to the model and the prompt, which indicates that altering both simultaneously indeed has a synergistic effect, even though it still falls short of the original in some aspects.
  - The performance in linking is not very good overall. I speculate the reason is that our prompt is not very effective, failing to directly transform the task into a graph generation task. Therefore, a better prompt graph is needed.

Discussion

- How to improve rodigy plus' accuracy:
  a. Prompt graph location
     i. Adjust prompt graph location
     ii. Adjust meta learning token number
  b. We can find a more suitable way to transform data-graph into graph embeddings.

Conclusion

- I believe my idea is feasible. Although there hasn't been significant progress in the experimental data, I am confident that if we implement the improvements mentioned in the previous pages, we should be able to achieve our goal

Future Work

Future Work

- Meta Learning
  - Adjust the training objectives and strategies.
  - Adjust the Pretrain GNN
  - Directly train using the model we will ultimately use.
  - The graph could be replaced with a larger one.
- Prodigy Plus
  - Adjust prompt graph location
  - Adjust meta learning token number
  - We can find a more suitable way to transform data-graph into graph embeddings.

References

Refernces

- PRODIGY: Enabling In-context Learning Over Graphs
  https://github.com/snap-stanford/prodigy?tab=readme-ov-file
- Prog : https://github.com/sheldonresearch/ProG
- GPPT: Graph Pre-training and Prompt Tuning to Generalize Graph Neural Networks
  https://dl.acm.org/doi/10.1145/3534678.3539249
- SGL-PT: A Strong Graph Learner with Graph Prompt Tuning
  https://arxiv.org/abs/2302.12449
- GraphPrompt: Unifying Pre-Training and Downstream Tasks for Graph Neural Networks
  https://dl.acm.org/doi/10.1145/3543507.3583386
- Virtual Node Tuning for Few-shot Node Classification
  https://arxiv.org/abs/2306.06063

Refernces

- Deep Prompt Tuning for Graph Transformers
  https://arxiv.org/abs/2309.10131
- Prompt Tuning for Multi-View Graph Contrastive Learning
  https://arxiv.org/abs/2310.10362
- ULTRA-DP:Unifying Graph Pre-training with Multi-task Graph Dual Prompt
  https://arxiv.org/abs/2310.14845
- Enhancing Graph Neural Networks with Structure-Based Prompt
  https://arxiv.org/abs/2310.17394
- HGPROMPT: Bridging Homogeneous and Heterogeneous Graphs for Few-shot Prompt Learning
  https://arxiv.org/abs/2312.01878
- Universal Prompt Tuning for Graph Neural Networks
  https://arxiv.org/abs/2209.15240

Thanks for Listening