A Datasets

Since LLMs work with textual contents, we utilize popular text attribute graph version of commonly used datasets for node classification tasks: Cora, Citeseer, PubMed, WikiCS and Ogbn-arxiv (Chen et al., 2024). Detailed statistics are listed in Table 1.

Datasets	#Nodes	#Edges	#Feat	#Classes
Cora	2,708	5,429	1,433	7
Citeseer	3,186	3,703	4,277	6
PubMed	19,717	44,335	500	3
WikiCS	11,701	215,863	300	10
Ogbn-arxiv	169,343	1,166,243	128	40

Table 1: Statistics of datasets.

B Baselines

In this section, we present a more detailed introduction to related works, especially those baselines adopted in this paper.

(i) GCN (Kipf and Welling, 2017): The one of most widely used model for graph-oriented tasks, leveraging neighbor aggregation to propagate node features across edges. The main idea is to weighted average the information of each node's neighbors and itself, in order to obtain a result vector that can be fed into the neural network.

(ii) Graph-SAGE (Hamilton et al., 2017): The core idea is to generate the embedding vector of the target vertex by learning a function that aggregates neighboring vertices.

(iii) Confidence: The traditional FSGL method, leverage trained GNN on labeled nodes to generate pseudo-labels for unlabeled nodes, and use confidence score to filter out prediction noises.

(iv) The excellent few-shot pipeline (Li et al., 2023) designs multi-level data augmentation with consistency and contrastive regularization, diversifying the model predictions and enabling the confident labeling information to be propagated from the labeled nodes into more unlabeled nodes.

(v) GLIM (Hu et al., 2023) adopts self-training framework for selecting high-quality labels, and spectral-based graph matching for implementing the migration of label information between different label density regions.

(vi) MoDis-MS (Pei et al., 2024) uses the entropy of an accumulated prediction distribution to model the prediction uncertainty, incorporating the trajectory of softmax distribution to improve the discriminative ability.

(vii) LLM-GNN (Chen et al., 2024): the latest excellent label-free pipeline, which involves LLMs to generate labels for unlabeled nodes, and using cluster-based sampling and confidence-based post-filtering to further enhance the label quality.

C Detailed Introductions of Prompts

We refine the process as inputting, questioning, and answering. (i) Different from the normal inputs, APL-LLM adds the class prototypes with the expectation of achieving reliable and condensed 1-shot. Here the prototype uses a dynamic cache to record the average feature information of correctly predicted nodes in supervised phase, which can gradually near the true representation as iterations progress, helping make predictions for unlabeled nodes. It seems a summary of reliable information. (ii) The questioning strategy includes: zero-shot, few-shot, and prototype prompt, here are their templates displayed. (iii) The answer process combines top-k and generality, inspired by recent research on LLM-confidence (Detommaso et al., 2024; Chen et al., 2024). Specifically, multiple rounds of queries are conducted and the most common answer is taken as the final result. Each query requires LLM to generate top-k answers with high probability, and the most likely answer is selected as the query result to avoid interference from a large number of low value answers.

Zero-Shot

Paper:
"title": <title>, "abstract": <abstract>.
Question:
What's the category of this paper in the given categories: <list of classes>?
Give your 3 best guesses, with confidence scores representing their probability from most to least likely. Each confidence ranges from 0.0 to 100.0, and the sum should be 100.0. Please in the form of a list of python dicts like [{"answer": <answer_here>, "confidence": <confidence_here>}]

D Fine-tuning with LoRA

An inevitable question is what to do if LLM makes an error prediction. Fortunately, there is still

Few-Shot with Prototypes

I will first give you an example and you should complete task following it.
Example: <example_content>
Paper:
"title": <title>, "abstract": <abstract>.
Question:
What's the category of this paper in the given categories: <list of prototypes>?
Give your 3 best guesses, with confidence scores representing their probability from most to least likely. Each confidence ranges from 0.0 to 100.0, and the sum should be 100.0. Please in the form of a list of python dicts like [{"answer":<answer_here>, "confidence": <confidence_here>}]

Prototype generation

The following list records some papers related to the <category>.
Paper: <list of paper_contents>
Question:
Please summarize the information above with a short paragraph, find some common points which can reflect the <category>

a chance to make fine-tune in supervised phase in FSGL. The Parameter Efficient Fine Tuning (PEFT) approaches updates only a small number of additional parameters to adapt to new tasks, making it more efficient and flexible, and reducing the risk of over-fitting. Therefore, while training GCN on labeled data, LLM is also used for prediction. If there are prediction errors, utilizing PEFT to adjust the large model. Otherwise, it can be considered that the large model already has reliable knowledge and predictive ability related to the task. Among them, the Low Rank Adaptation (LoRA) technique was adopted, which achieved fine-tuning of the model by training only the low rank matrix and freezing most parameters in the large model, reducing computational requirements.

E Design Philosophy Behind APL-LLM

Graphs are widely applicated various domains, such as financial risk prediction. In actual credit scenarios, the physical information of many small and medium-sized enterprises is not sound, and the system often contains a small number of large-scale financial entities with comprehensive information and records. To reduce financial risks, different ratings and corresponding processes need to be applied to different users. The proposed APL-LLM approach can effectively adapt to this few-shot scenario with sparse labels, thus better serving credit risk prediction tasks.

E.1 Why FSGL?

In many practical applications, collecting labeled data is an expensive and time-consuming process. Few-Shot Graph Learning (FSGL) leverages a small amount of labeled data and a large amount of unlabeled data to train models. Compared with traditional supervised learning, FSGL can effectively utilize unlabeled data, thus improving the model's performance in data-scarce situations.

Its main advantages include: (i) It can utilize a large amount of unlabeled data for training, thereby enhancing the model's generalization ability; (ii) It can reduce the cost of labeling data and improve learning efficiency; (iii) It can achieve better prediction results with limited labeled data. Meanwhile, its main disadvantages include: (i) Due to incomplete data labeling, the model's prediction performance may be suboptimal; (ii) Appropriate algorithms need to be designed to enable effective learning with limited labeled data.

This paper aims to enable the proposed algorithm to fully explore the certain knowledge of labeled data and the potential knowledge of unlabeled data as much as possible.

E.2 Why use LLM for FSGL?

Compared with traditional models, the advantages of Large Language Models (LLMs) mainly contains: (i) **Stronger learning ability:** LLMs possess a larger number of parameters and more complex architectures, enabling them to fit data better and demonstrating great applicability across multiple domains. (ii) **Superior generalization ability:** The knowledge learned by LLMs during the training process is more general. They can better generalize to unseen data, reducing the reliance on a large amount of labeled data. They can learn more powerful feature representations and pattern recognition capabilities from a vast amount of data, thus effectively withstanding the impacts of noise and interference. (iii) **Higher efficiency computing ability:** By adopting technologies such as hierar-

chical design and distributed training, LLMs can be trained efficiently on existing hardware devices.

Traditional data annotation methods are usually carried out manually, which is not only costly but also inefficient. In contrast, LLMs can quickly adapt to specific tasks through techniques such as transfer learning and fine-tuning, greatly improving the efficiency and accuracy of data annotation. The advantages of LLM-based annotation are as follows: (i) **High efficiency:** LLMs have powerful feature extraction and generalization capabilities, enabling them to complete the annotation of a large amount of data in a short time and thus enhancing the annotation efficiency. (ii) **High accuracy:** Through transfer learning and fine-tuning, LLMs can better adapt to specific tasks, reducing annotation errors and improving annotation accuracy. (iii) **Automation:** LLMs have a certain degree of self-adaptability and can, to some extent, automate the annotation process, alleviating the burden on human annotators.

Existing pseudo-labeling methods generate pseudo-labels based on the predictions of unlabeled data by GNNs trained on labeled data. These methods suffer from issues such as unstable predictions, data distribution biases, and low data utilization rates. By incorporating the rich knowledge of LLMs, it is expected to generate stable and high-quality pseudo-labels.

E.3 Why Dynamic Threshold?

Corresponding content shown in Figure 1.(b) and Section 4.2. In FSGL,maintaining pseudo-label quality requires addressing dynamic sample difficulty, a dynamic difficulty indicator to discriminate transient noise from stable structural signals is needed. Therefore, we introduce a dynamic pseudo-label threshold method, which filters unstable samples (high-confidence noise) with strict thresholds while relaxing for time-stable samples (consistent predictions), balancing precision-recall tradeoffs. By analyzing historical model states, it adaptively includes pseudo-labels during optimal time windows, blocking noisy low-confidence moments and enhancing semi-difficult sample recall. Such method ensures high-quality pseudo-labels by discriminating transient noise from stable structural signals, improving few-shot learning tasks.

E.4 Why Consistency Learning?

Deep learning models are prone to overfitting. When subjected to small perturbations (noise), the prediction results can be significantly affected. To mitigate over-fitting, in supervised learning, new loss terms are added, while in FSGL, consistency regularization is employed. Its core idea is to constrain the features learned by the model by comparing the similarity between pairs of unlabeled data with the same label. The advantages include: improving model performance, reducing the dependence on labeled data, and being easy to implement.

Specific Description: Based on the smoothness assumption and cluster assumption in semi-supervised learning, data points with distinct labels are separated by low-density regions, while similar data points exhibit consistent outputs. When practical perturbations are applied to unlabeled data instances, the underlying principle dictates that their predictions should not exhibit significant deviations, thereby ensuring output consistency. This approach typically leverages prediction vectors from model outputs rather than explicit labels, making it inherently suitable for FSGL frameworks. By constructing an unsupervised regularization loss term that measures the discrepancy between perturbed predictions $\hat{y}$ and original predictions y on unlabeled data, the model's generalization capabilities are systematically enhanced.

References

Zhikai Chen, Haitao Mao, Hongzhi Wen, Haoyu Han, Wei Jin, Haiyang Zhang, Hui Liu, and Jiliang Tang. 2024. Label-free node classification on graphs with large language models (llms). In *The Twelfth International Conference on Learning Representations, ICLR*.

Gianluca Detommaso, Martin Bertran Lopez, Riccardo Fogliato, and Aaron Roth. 2024. Multicalibration for confidence scoring in llms. In *Forty-first International Conference on Machine Learning, ICML*.

William L. Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems 30, NeurIPS*, pages 1024–1034.

Zhihui Hu, Yao Fu, Hong Zhao, Xiaoyu Cai, Weihao Jiang, and Shiliang Pu. 2023. Liberate pseudo labels from over-dependence: Label information migration on sparsely labeled graphs. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management, CIKM*, pages 833–842.

Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations, ICLR*.

Quan Li, Lingwei Chen, Shixiong Jing, and Dinghao Wu. 2023. Pseudo-labeling with graph active learning for few-shot node classification. In *IEEE International Conference on Data Mining, ICDM*, pages 1115–1120.

Hongbin Pei, Yuheng Xiong, Pinghui Wang, Jing Tao, Jialun Liu, Huiqi Deng, Jie Ma, and Xiaohong Guan. 2024. Memory disagreement: A pseudo-labeling measure from training dynamics for semi-supervised graph learning. In *Proceedings of the ACM on Web Conference 2024, WWW*, pages 434–445.