HiTuner: Hierarchical Semantic Fusion Model Fine-Tuning on Text-Attributed Graphs

Zihan Fang[1,2], **Zhiling Cai**[3], **Yuxuan Zheng**[1,2], **Shide Du**[1,2], **Yanchao Tan**[1,2], **Shiping Wang**[1,2*]
[1]College of Computer and Data Science, Fuzhou University, Fuzhou, China
[2]Key Laboratory of Intelligent Metro, Fujian Province University, Fuzhou, China
[3]College of Computer and Information Science, Fujian Agriculture and Forestry University, Fuzhou, China
fzihan11.com@163.com, zhilingcai@126.com, allison_zyx@163.com, dushidems@gmail.com, yctan@fzu.edu.cn, shipingwangphd@163.com

Abstract

Text-Attributed Graphs (TAGs) are vital for modeling entity relationships across various domains. Graph Neural Networks have become cornerstone for processing graph structures, while the integration of text attributes remains a prominent research. The development of Large Language Models (LLMs) provides new opportunities for advancing textual encoding in TAGs. However, LLMs face challenges in specialized domains due to their limited task-specific knowledge, and fine-tuning them for specific tasks demands significant resources. To cope with the above challenges, we propose *HiTuner*, a novel framework that leverages fine-tuned Pre-trained Language Models (PLMs) with domain expertise as tuner to enhance the hierarchical LLM contextualized representations for modeling TAGs. Specifically, we first strategically select hierarchical hidden states of LLM to form a set of diverse and complementary descriptions as input for the sparse projection operator. Concurrently, a hybrid representation learning is developed to amalgamate the broad linguistic comprehension of LLMs with task-specific insights of the fine-tuned PLMs. Finally, *HiTuner* employs a confidence network to adaptively fuse the semantically-augmented representations. Empirical results across benchmark datasets spanning various domains validate the effectiveness of the proposed framework. Our codes are available at: https://github.com/ZihanFang11/HiTuner

1 Introduction

Many graphs in the real-world possess textual features, and can be termed as Text-Attributed Graphs (TAGs). In TAGs, nodes typically represent textual entities like documents or sentences, while edges capture the relationships between different entities. Graph Neural Networks (GNNs) have emerged as a promising paradigm for TAG-related tasks due to their ability to propagate messages between neighboring nodes [Kipf and Welling, 2017; Luo *et al.*, 2024]. This capability allows them to effectively utilize a small number of labeled examples and generalize well to large-scale graph tasks [Hamilton *et al.*, 2017]. Typically, the text associated with each node must first be transformed into vector compatible with GNNs. Traditional shallow text encoders, such as skip-gram or bag-of-words models, are often employed for this purpose by converting text into numerical node features. However, these non-contextualized embeddings limit the ability of TAG-related models to learn rich semantic knowledge, as they lack contextual awareness to exploit textual attributions.

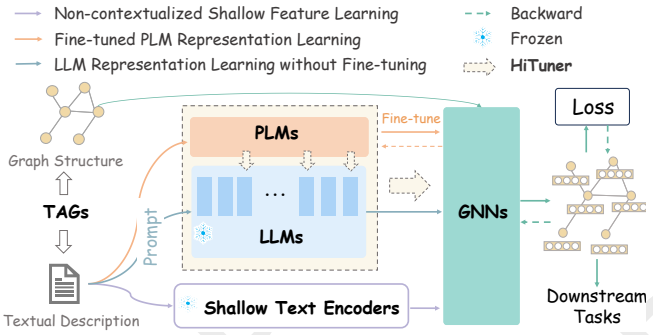


Figure 1: Three training pipelines and *HiTuner* on TAGs.

Recently, Large Language Models (LLMs), with cutting-edge developments such as ChatGPT and LLaMA2 [Touvron *et al.*, 2023] excel in general language understanding, have introduced new possibilities for processing TAGs. LLMs leverage attention-based transformers to adeptly capture the contextual semantics of text, demonstrating superior generalization and zero-shot learning capabilities across diverse linguistic scenarios [Wu *et al.*, 2024; Pan *et al.*, 2024]. However, when used for specialized domain-specific tasks in TAGs, *e.g.*, citation graph categorization of papers in some specialized areas, these models may exhibit limitations due to limited task-specific knowledge, in some cases, even produce erroneous or hallucinatory responses [Shen *et al.*, 2024]. In order to narrow the semantic gap between pretrain and downstream tasks, "pretrain-then-finetune" has become the mainstream paradigm. Fine-tuning on task-specific labeled

*Corresponding author

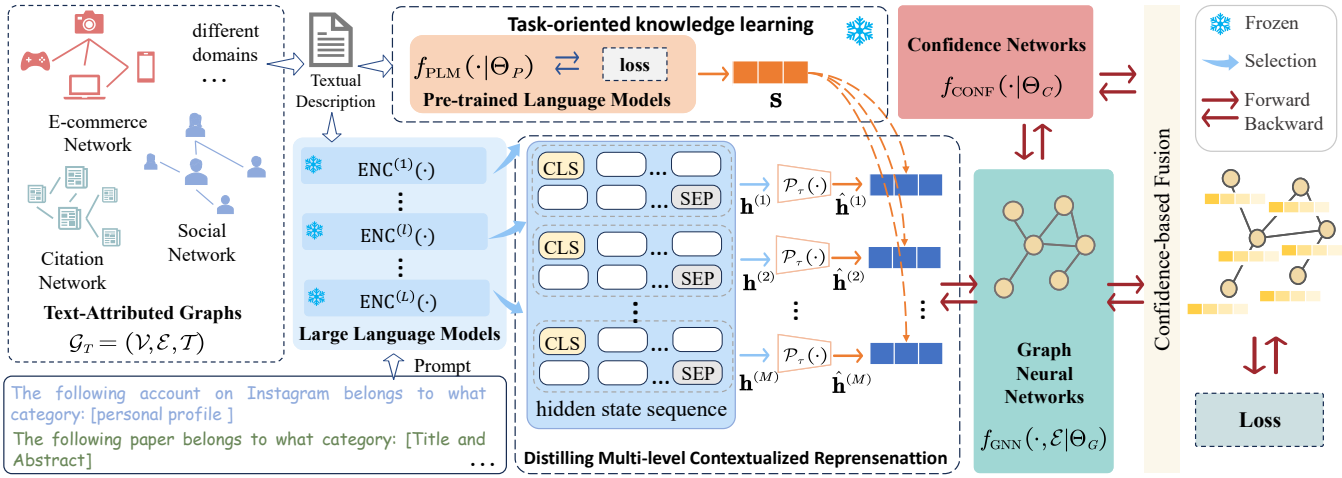


Figure 2: The *HiTuner* framework inputs text descriptions of TAGs into PLMs that leverage task-oriented knowledge and into LLMs to generate context-specific hidden states. Training modules consist of threshold-based filters $\mathcal{P}_\tau$ for distilling multi-level contextualized representation, confidence network f_{CONF} for contribution assessment, and graph neural networks f_{GNN} for incorporating topological information.

TAGs data theoretically enhances performance significantly, while the vast size of LLMs, which often includes billions of parameters, necessitates considerable computational resources [Tang *et al.*, 2024; Perin *et al.*, 2024]. In contrast, prompting offers a more resource-efficient alternative by adjusting the inputs to tailor outputs for specific tasks [Yu *et al.*, 2024]. Nevertheless, crafting effective prompts still requires extensive validation data to refine. A pertinent question arises: How to design efficient fine-tuning methods to address the resource-consumption issues of LLMs in specific tasks within TAGs?

Conversely, smaller-scale Pre-trained Language Models (PLMs), such as BERT [Devlin *et al.*, 2019] and RoBERTa [Liu *et al.*, 2019], can be flexibly fine-tuned for downstream tasks to leverage the rich semantics encoded in domain-specific textual features. In this work, we use PLMs to refer to relatively small language models that can be trained and fine-tuned under limited resources in the academic research labs. Fine-tuning involves using labeled samples as additional training signals in a task-specific manner, thus fine-tuned PLMs can be adapted to domain-specific terminology. Despite these advancements, the scarcity of labeled data in specialized domains complicates the training and fine-tuning of PLMs within TAGs, often leading to poor generalization of unseen data. Thus, the primary challenge remains: How to develop strategies that allow small-scale PLMs learn sufficiently generalized representation from limited labeled samples within TAGs.

While initial LM-based training frameworks excel at capturing semantic information, relying solely on textual attributes may hinder the representation learning in TAGs due to issues such as noise or incompleteness in the original text. Modeling node relationships to enable message passing offers a strategic solution, with a common approach being to cascade LM-generated text representations into GNNs [Chen *et al.*, 2024]. To illustrate this concept further, the three training pipelines, as previously mentioned, are described in Figure 1. Non-contextualized shallow feature learning involves using shallow text encoders to process text data without semantic understanding. LLMs leverage the linguistic capabilities to learn representation without task-specific optimization. Fine-tuned PLMs for specific tasks allow better adaptation to particular requirements, improving performance. As a result, those context-aware representations are more flexible and robust than static, shallow features [Apidianaki, 2023].

Based on the above observations, the objective of this research is to address the challenges of empowering text-attributed graph learning by developing resource-efficient methods for fine-tuning LLM-based representations. Toward this goal, we propose *HiTuner* to enhance both performance and scalability while addressing resource limitations. Initially, PLMs are fine-tuned on domain-specific datasets to acquire detailed knowledge pertinent to the field. Then, we extract different hidden states of LLM to aggregate versatile hierarchical text representations, harnessing their capacity to integrate extensive knowledge. To ensure diversity in the learned representations, hidden representations of different transformer layers in LLMs are selected for further process. Subsequently, we propose a mixed approach to tailor text representation processes to better align with the specific needs of the domain represented in the TAGs. This targeted approach can optimize computational efficiency by minimizing the computational overhead associated with LLMs. Finally, *HiTuner* employs a confidence network to adaptively fuse these diverse representations, enhancing the accuracy and reliability for subsequent analyses. An illustration of *HiTuner* is shown in Figure 2. The key contributions of this paper are as follows.

- **Distilling Hierarchical Contextualized Representations for Resource Efficiency**: To harness the complementary semantic information at different levels, we strategically select and integrate hidden states from LLMs without prohibitive computational demands.

- **Task-oriented Knowledge Infusion for Semantic Comprehension**: We develop a hybrid representation that amalgamates the extensive knowledge of LLMs with the targeted insights of fine-tuned PLMs, incorporating structure-related semantics related to TAGs.
- **SOTA performance and Great Scalability**: Extensive experiments across various TAG benchmarks show that *HiTuner* significantly boosts baseline performance, while also demonstrating adaptability and generalization capabilities.

2 Preliminaries

In this section, we establish some relevant notations and formalize the research problem of this work. Primarily, we introduce the definition of TAGs and three popular learning paradigms associated with them.

2.1 Text-Attributed Graphs

A general graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ consists of a set of nodes $\mathcal{V}$, and a set of edges $\mathcal{E}$ connecting these nodes. While TAG can be formally represented as $\mathcal{G}_T = (\mathcal{V}, \mathcal{E}, \mathcal{T})$, where $\mathcal{T}$ is the document set. Each node $v \in \mathcal{V}$ is associated with a sequential text $\mathbf{t}_v \in \mathcal{T}$ and the corresponding label $\mathbf{y}_v$. In this work, we focus on node classification, aiming to predict labels for unlabeled nodes $\mathcal{V}^U$ based on a given set of labeled nodes $\mathcal{V}^L$.

2.2 GNN-based Paradigm

GNNs are message-passing frameworks for graph-structured data that aggregate information across neighboring nodes to learn the vectorized representation. Mathematically, the update process of GNNs for each node v can be expressed as:

$$\mathbf{x}_v^{(l)} = \text{UPD}\left(\mathbf{x}_v^{(l-1)}, \text{AGG}\left(\{\mathbf{x}_u^{(l-1)} : u \in \mathcal{N}(v)\}\right)\right), \quad (1)$$

where u is the neighbor of node v, $\mathcal{N}(v)$ denotes the set of neighbors of v, and $\text{AGG}(\cdot)$ and $\text{UPD}(\cdot)$ are aggregation and update functions, respectively. The initial node feature vector $\mathbf{x}_v^{(0)}$ is a numerical feature extracted from the text $\mathbf{t}_v$.

2.3 Pre-training of PLMs

PLMs encode the vector of a word by dynamically combining it with surrounding words. A text document $\mathbf{t}_v \in \mathcal{T}$ can be tokenized into a sequence of tokens $\mathcal{S}_v = \{\mathtt{s}_{v,i}\}_{i=0}^{|\mathcal{S}_v|}$, where $\mathtt{s}_{v,i}$ denotes a specific token-id. A PLM takes the sequence of tokens $\mathcal{S}_v$ as input, and produces the hidden states of each token:

$$\{\mathbf{s}_{v,0}, \ldots, \mathbf{s}_{v,|\mathcal{S}_v|}\} = \text{PLM}\left(\{\mathtt{s}_{v,0}, \ldots, \mathtt{s}_{v,|\mathcal{S}_v|}\}\right). \quad (2)$$

Generally, the BERT-like language model can be fine-tuned for various tasks by simply adding a classifier on top of the `[CLS]` token, a special classification token placed at the beginning of each sequence. This token utilizes the self-attention mechanism to capture the contextual information relevant to the specific task. Here, we denote the final hidden state of the `[CLS]` token $\mathbf{s}_{v,0}$ as semantic-aware representation $\mathbf{s}_v$:

$$\mathbf{s}_v = f_{\text{PLM}}(\mathbf{t}_v|\Theta_P) \in \mathbb{R}^{d_P}. \quad (3)$$

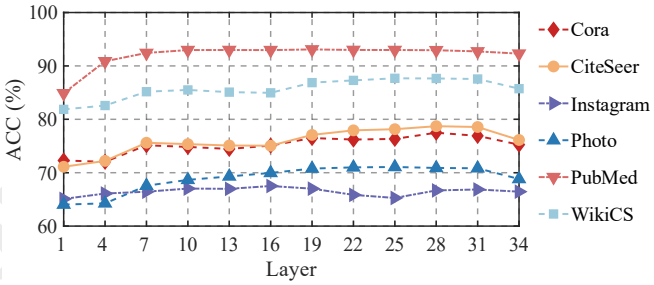


Figure 3: Test accuracy of the MLP applied to the hidden states from each transformer layer of LLaMA2.

To predict specific outputs for downstream tasks, one can simply adopt a multilayer perceptron (MLP) as a predictor and the model parameters are trained with

$$\mathcal{L}_{pre} = \sum_{v \in \mathcal{V}^L} \text{CrossEntropy}\left(\text{MLP}\left(\mathbf{s}_v\right), \mathbf{y}_v\right). \quad (4)$$

2.4 LLMs with Prompt

LLMs are first pre-trained on large-scale text datasets to learn common sense and then generate outputs based on a prompt `P` to process the input without modifying the model parameters, rather than being fine-tuned for specific downstream tasks. Specifically, prompting methods use templates containing unfilled slots, which modify the original text into a textual string prompt. For example, the task is to classify the topics to which papers belong based on their abstracts [`ABSTRACT`]. By inserting some prompts in the sentences, we can create a template with placeholders for specific information.

$$\begin{aligned} &\textit{Here is the abstract of a paper}: [\texttt{ABSTRACT}], \\ &\textit{This paper belongs to the } [\texttt{TOPIC}] \textit{ category}. \end{aligned} \quad (5)$$

We denote this new sentence with the prompt [`TOPIC`] inserted as $\mathcal{S}_{\mathtt{P}|v}$, where `P` represents the inserted prompt tokens.

LLMs are built with multi-layer transformer encoders that use attention-based mechanisms to aggregate information in the sequence of tokens $\mathcal{S}_{\mathtt{P}|v}$. Thus, these hidden states can be viewed as representations at different levels:

$$\{\mathbf{h}_{v,0}^{(l)}, \ldots, \mathbf{h}_{v,|\mathcal{S}_{\mathtt{P}|v}|}^{(l)}\} = \text{ENC}^{(l)}\left(\mathbf{h}_{v,0}^{(l-1)}, \ldots, \mathbf{h}_{v,|\mathcal{S}_{\mathtt{P}|v}|}^{(l-1)}\right), \quad (6)$$

where $\text{ENC}^{(l)}(\cdot)$ denotes the l-th layer of LLM, and $\{\mathbf{h}_{v,i}^{(l)}\}_{i=0}^{|\mathcal{S}_{\mathtt{P}|v}|}$ means hidden state sequence in l-th layer for target node v. Note that $|\mathcal{S}_{\mathtt{P}|v}|$ indicates the length of the sequence. A readout function [Reimers and Gurevych, 2019] can be applied to token-level representations:

$$\mathbf{h}_v^{(l)} = \text{POOL}\left(\mathbf{h}_{v,1}^{(l)}, \mathbf{h}_{v,2}^{(l)}, \ldots, \mathbf{h}_{v,|\mathcal{S}_{\mathtt{P}|v}|}^{(l)}\right) \in \mathbb{R}^{D_L}. \quad (7)$$

Here, $\mathbf{h}_v^{(l)}$ denotes the resulting hidden state sequence from the l-layer of LLM, and this work adopts mean pooling.

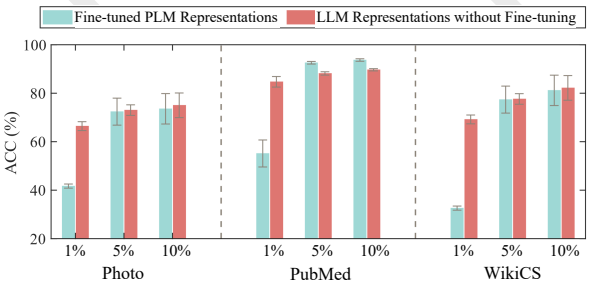


Figure 4: Comparing node classification using different initial node features generated from PLMs/LLMs, with a MLP classifier trained under various labeling ratios (1%, 5%, 10%).

3 Proposed Framework

In this section, we begin by illustrating our motivation through two experiments. Building upon these insights, we then introduce the proposed framework, referred to as *HiTuner*, which uses fine-tuned PLMs with domain expertise as tuners for hierarchical LLM contextualized representations.

3.1 Motivation

Hierarchical Insight from Contextual Layer

While the attention mechanism is applied uniformly across all transformer layers of LLMs, previous research has shown that different contextual layers retain lexical meaning at varying levels of abstraction [Yuan *et al.*, 2023]. We revisit the learning capabilities at varying levels of LLMs in Figure 3. In this setup, the hidden state sequences from various layers of LLMs are then directly fed into an MLP trained for classification tasks. The analysis indicates that deeper layers capture richer semantics but may degrade beyond a certain depth, and the optimal layer varies across datasets. This suggests that the traditional method of extracting the last layer of the LLMs as a representation may not be the optimal strategy, because the information captured by different contextual layers can be complementary. *This observation inspires us to exploit multiple hidden states from LLMs to better understand hierarchical semantic information in text.*

Leveraging Domain Expertise in Fine-Tuning PLMs

To validate the previously discussed concepts, Figure 4 illustrates experimental evidence comparing the performance of fine-tuned PLMs with unfine-tuned LLMs across three distinct datasets from different domains. Remarkably, LLMs excel across diverse domains, even with a minimal labeling rate of 1%, while the performance gains between 1% to 10% supervision are modest. In contrast, fine-tuned PLMs fail at low labeling rate settings, but show a significant improvement in ACC from 1% to 5%, and maintain a slight edge over LLMs at 10% supervision. These results not only confirm the limitations of unfine-tuned LLMs in specific domains but also highlight that fine-tuning may fail to transfer sufficient targeted knowledge for downstream tasks under label-scarce scenarios. *This phenomenon inspires us to leverage task-specific knowledge in fine-tuned PLMs to compensate for the domain semantic gaps, while utilizing the linguistic capabilities of LLMs to overcome label scarcity.*

3.2 Distilling Hierarchical Contextualized Representations

From various contextual layers of the frozen LLM, hidden state sequences $\{\mathbf{h}^{(m)} \in \mathbb{R}^{D_L}\}_{m=1}^{M}$ are pre-computed in advance, where $M<L$. Each vector $\mathbf{h}^{(m)}$ captures unique information pertinent to its respective context or level. Generally, handling such high-dimensional data often requires dimensionality reduction techniques to focus on the most informative features. We define a denoised projector $\mathcal{P} : \mathbb{R}^{D_L} \rightarrow \mathbb{R}^{D_P}$ to filter out noise and redundant information. $\mathbf{h}^{(m)}$ is first linearly transformed by specific weight matrix $\mathbf{W}^{(m)} \in \mathbb{R}^{D_L \times D_P}$, which maps the high-dimensional data into a new space. The transformed data is then subjected to a threshold-based filter, where higher values of τ result in a greater proportion of the input being filtered out,

$$\hat{\mathbf{h}}^{(m)} \leftarrow \mathcal{P}_\tau\left(\mathbf{h}^{(m)}\mathbf{W}^{(m)}\right), \tag{8}$$

where $\mathcal{P}_\tau(\cdot)$ creates a differential response to distinguish between different types of feature activation, potentially adding robustness against noise and providing additional contextual information.

3.3 Semantic and Structure-aware Augmentation

Enhancing Task-oriented Semantic Understanding. The unsupervised representations learned by LLMs capture the language regularities in large-scale corpus, but may lack sensitivity for specific tasks. Integrating fine-tuning processes in deploying LLMs, such as updating parameters, requires a lot of resources and time, so we choose to incorporate the learned task-oriented knowledge by small-scale PLMs into the semantic representation at a lower cost. Denoting $\mathbf{s}$ as the fine-tuned PLM representation, we simply design the following mixed strategy:

$$\mathbf{m}^{(m)} = \lambda\mathbf{s} + (1-\lambda)\,\hat{\mathbf{h}}^{(m)}, \tag{9}$$

where λ is a trade-off parameter to balance the influence of the fine-tuned and unsupervised representations. Overall, $\mathbf{m}^{(m)}$ encodes both general knowledge and task-specific refinements, enabling efficient task adaptation with lower fine-tuning overhead.

Incorporating Global Structural Information. Incorporating topological information significantly boosts the capability of models to interpret structure-related semantics within TAGs. This is beneficial in specialized domains like academic research, where analyzing citation patterns can help categorize papers. To effectively leverage the information from a few labeled samples and extend their influence to more unlabeled samples, *HiTuner* inputs the mixture representation $\mathbf{m}^{(m)}$ to GNNs for accurate comprehension of node semantics from a global perspective. Each layer progressively encodes complex structural and contextual information into the node representations. Formally, we denote the GNN-based paradigm as

$$\mathbf{z}^{(m)} = f_{\text{GNN}}\left(\mathbf{m}^{(m)}, \mathcal{E} | \Theta_G\right) \in \mathbb{R}^C, \tag{10}$$

where Θ_G is the trainable parameters, and $\mathcal{E}$ provides the structural context. The final node output $\mathbf{z}^{(m)}$ encoded by GNNs is better to meet downstream graph-related tasks such as node classification and link prediction.

3.4 Confidence-based Fusion

Subsequently, *HiTuner* integrates the enhanced representations $\{\mathbf{z}^{(m)}\}_{m=1}^{M}$ to infer final predictions. To assess the reliability of each representation, we compute its true class probability, *i.e.* the likelihood assigned to the correct label. Therefore, higher true class probabilities indicate greater model reliability, thereby serving as a critical measure for assessing confidence. In situations where only a limited subset of samples is labeled, confidence network is employed to estimate the certainty of these predictions:

$$t_v^{(m)} = f_{\text{CONF}}\left(\mathbf{z}_v^{(m)} | \Theta_C\right). \tag{11}$$

Here, $t_v^{(m)}$ quantifies the predictive confidence of node v of the m-th level contextualized representation and $f_{\text{CONF}}(\cdot)$ is achieved by MLP combined with a sigmoid function to normalize the output between 0 and 1.

Furthermore, a relative confidence metric ψ is introduced to dynamically adjust the influence of the m-th level by comparing to the aggregate confidence of other levels:

$$\psi_v^{(m)} = \frac{\log\left(\prod_{j\neq m}^{M} t_v^{(j)}\right)}{\log\left(\prod_{i=1}^{M} t_v^{(i)}\right)}. \tag{12}$$

Then, fusion weights are derived using the softmax function to ensure that contribution of each level is proportional to both its assessed and relative confidence:

$$\mathbf{w}_v = \text{softmax}([\psi_v^{(1)} \cdot t_v^{(1)}, \ldots, \psi_v^{(M)} \cdot t_v^{(M)}]) \in \mathbb{R}^M. \tag{13}$$

Therefore, this weight can dynamically adjust according to the confidence, preventing low-confidence representation from affecting the final results, expressed as

$$\hat{\mathbf{z}}_v = [\mathbf{z}_v^{(1)}, \cdots, \mathbf{z}_v^{(M)}]\mathbf{w}_v^T. \tag{14}$$

Finally, we define the loss function used to train the model as the cross-entropy loss between predictions and true labels:

$$\mathcal{L} = \textstyle\sum_{v\in\mathcal{V}^L} \text{CrossEntropy}\left(\hat{\mathbf{z}}_v, \mathbf{y}_v\right). \tag{15}$$

All trainable parameters in the proposed method are denoted by $\Theta = \{\Theta_G, \Theta_C, \tau\}$. The entire training process only requires training Θ, and the subsequent learning does not involve the updating of LLM parameters. By leveraging the knowledge from smaller PLMs, we can efficiently enhance the semantic representation without incurring the high resource expenses caused by full-scale fine-tuning of LLMs.

4 Experiment

Our experimentation focuses on the following questions:
RQ1: Can *HiTuner* enhance the performance in modeling TAGs compared with baselines?
RQ2: Are all components comprising of *HiTuner* valid?
RQ3: Is the *HiTuner* sensitive to the fusion of two kind of semantic knowledge in different domains?
RQ4: Does *HiTuner* have excellent scalability?

Dataset	#Nodes	#Edges	#Clusters	Domain
CiteSeer	3,186	4,277	6	Academic
Cora	2,708	5,429	7	Academic
Instagram	11,339	144,010	2	Social
Photo	48,362	500,928	12	E-commerce
PubMed	19,717	44,338	3	Academic
WikiCS	11,701	216,123	10	Wikipedia

Table 1: Statistics of the TAG datasets.

4.1 Implementation Details

In this section, we perform node classification with low labeling ratio to valid the effectiveness of *HiTuner*. All experiments are conducted on NVIDIA A100 GPUs with 80GB memory. By default, we employ LLaMA2-7B [Touvron *et al.*, 2023] and BERT as the backbones, and SAGE as an instance of GNNs. For a fair comparison, we run 5 times and report the mean result and the standard deviation.

Datasets

We evaluate the proposed method on various types of datasets, with the relevant statistics summarized in Table 1. These datasets encompass three citation networks, including **CiteSeer**, **Cora** and **PubMed**. Along with a social network dataset (**Instagram**), an E-commerce dataset from Amazon (Electronics-Photography, namely **Photo**) and a Wikipedia-based dataset (**WikiCS**). Specifically, the ratio of nodes used for the train/valid/test stage is 10%/10%/80%.

Comparison Methods

Following [Chen *et al.*, 2024], we compare *HiTuner* with three types of comparison methods:

(1) Benchmark models: For GNN-based method, we select GCN [Kipf and Welling, 2017] and SAGE [Hamilton *et al.*, 2017], while MLP as a baseline. We test 3-layer architectures with a hidden dimension of 256, and each layer is accompanied by a batch operation. We also consider initial node features with three textual feature extraction methods:

- **Non-contextualized Shallow Embeddings** ($\boldsymbol{F}_S$): Text embeddings extracted using shallow methods, such as BoW, TF-IDF, and word2vec.
- **LLM Embeddings without Fine-tuning** ($\boldsymbol{F}_{LLM}$): Text embeddings taken directly from a LLM without the application of specific task labels. Additionally, the hidden states based on prompts.
- **Fine-tuned PLM Embeddings** ($\boldsymbol{F}_{FLM}$): Text embeddings extracted from BERT-like PLM that has been fine-tuned on a specific dataset.

(2) Standard fine-tuned PLMs: We select three popular LMs: BERT (bert-base-uncased) [Devlin *et al.*, 2019], DeBERTa (deberta-base) and SentenceBERT (bert-base-nli-mean-tokens) [Reimers and Gurevych, 2019]. In this setting, we employ a simple cascading structure, where PLMs are first fine-tuned on the downstream datasets, and the generated text embeddings are input as the initial node features for the downstream classifier.

(3) LLM-enhanced methods: We further compare the following SOTA methods that combine LLMs and GNNs:

Methods		CiteSeer	Cora	Instagram	Photo	PubMed	WikiCS	*Average*
Original Textual Attributes								
	BERT	67.02 ± 1.75	62.36 ± 2.29	62.42 ± 3.03	73.57 ± 0.48	92.87 ± 0.44	81.18 ± 1.00	*73.27*
	DeBERTa	45.83 ± 17.2	36.46 ± 9.55	64.90 ± 1.52	74.42 ± 0.38	93.06 ± 0.64	79.56 ± 2.90	*65.80*
	SentenceBERT	67.08 ± 3.05	62.07 ± 3.92	62.75 ± 0.90	73.38 ± 0.54	92.74 ± 0.58	80.61 ± 1.35	*73.17*
Non-contextualized Shallow Embeddings								
	MLP	70.40 ± 1.08	56.47 ± 0.97	64.67 ± 0.31	62.47 ± 0.25	81.79 ± 0.09	73.63 ± 0.35	*68.24*
	GCN	69.78 ± 0.27	79.53 ± 1.36	64.54 ± 0.30	83.01 ± 0.28	82.57 ± 0.52	79.51 ± 0.40	*76.49*
	SAGE	71.22 ± 0.24	81.50 ± 0.69	64.76 ± 0.45	83.11 ± 0.29	83.91 ± 0.08	81.85 ± 0.29	*77.73*
LLM Embeddings without Fine-tuning								
LLaMA2	MLP	66.55 ± 3.42	70.30 ± 2.12	66.74 ± 1.10	74.73 ± 0.46	89.55 ± 0.29	82.19 ± 0.76	*75.01*
	GCN	67.40 ± 2.97	82.55 ± 2.01	67.86 ± 0.65	85.36 ± 0.47	86.67 ± 0.50	81.53 ± 0.62	*78.56*
	SAGE	69.37 ± 3.17	82.40 ± 3.12	66.83 ± 1.20	85.80 ± 0.25	87.95 ± 0.34	82.44 ± 0.66	*79.13*
LLaMA2⋆	MLP	67.02 ± 3.37	70.15 ± 1.92	67.15 ± 0.67	74.75 ± 0.42	89.60 ± 0.19	81.67 ± 1.08	*75.06*
	GCN	67.62 ± 2.61	82.44 ± 1.93	67.14 ± 0.41	85.40 ± 0.48	86.55 ± 0.34	80.83 ± 0.60	*78.33*
	SAGE	69.40 ± 3.57	82.32 ± 3.20	67.39 ± 0.96	85.88 ± 0.30	87.69 ± 0.53	82.27 ± 1.32	*79.16*
Fine-tuned PLM Embeddings								
BERT	GCN	67.55 ± 3.09	80.44 ± 2.03	65.99 ± 1.03	85.59 ± 0.16	91.35 ± 0.47	82.64 ± 0.88	*78.93*
	SAGE	68.81 ± 3.09	80.55 ± 3.20	66.62 ± 0.69	83.97 ± 0.47	93.00 ± 0.42	83.08 ± 1.06	*79.34*
DeBERTa	GCN	61.00 ± 4.89	69.74 ± 7.18	66.60 ± 1.27	85.73 ± 0.26	92.04 ± 0.41	82.31 ± 0.60	*76.24*
	SAGE	60.22 ± 8.91	70.59 ± 6.06	66.94 ± 0.68	84.59 ± 0.40	93.31 ± 0.35	82.26 ± 1.02	*76.39*
SentenceBERT	GCN	67.49 ± 1.87	79.93 ± 3.65	66.22 ± 0.90	85.59 ± 0.28	91.17 ± 0.40	82.30 ± 0.48	*78.83*
	SAGE	69.06 ± 2.31	80.00 ± 2.25	66.51 ± 0.56	84.04 ± 0.51	93.21 ± 0.53	82.27 ± 1.00	*79.02*
TAPE (GPT3.5)		-	**84.91 ± 0.90**	-	-	92.84 ± 0.12	-	-
ENGINE (LLaMA2)		**72.32 ± 0.81**	81.75 ± 1.06	66.93 ± 0.51	84.64 ± 0.14	87.46 ± 0.74	84.04 ± 0.47	*79.52*
GraphAdapter (LLaMA2)		70.50 ± 0.90	78.03 ± 1.46	67.45 ± 0.51	-	88.30 ± 0.38	-	-
HiTuner		71.35 ± 3.28	84.58 ± 2.69	**68.10 ± 0.18**	**86.09 ± 0.58**	**93.50 ± 0.42**	**85.20 ± 1.19**	***81.47***

Table 2: Accuracy (%) of all compared methods across six datasets, where the best and runner-up performance are highlighted in bold and underlined respectively (mean% ± standard deviation%). ⋆ denotes prompt tuning.

- **TAPE** [He *et al.*, 2024] leverages LLMs to generate pseudo labels and explanations as enhancements, and concatenates them with original text for PLMs.
- **ENGINE** [Zhu *et al.*, 2024] combines LLMs and GNNs via a tunable side structure to adopt message passing at each layer of LLMs to integrate structural information.
- **GraphAdapter** [Huang *et al.*, 2024] introduces a residual learning procedure to pre-train GNNs, which serve as efficient adapters when integrated with LLMs.

4.2 Overall Comparison (RQ1)

We present the comparative analysis in Table 2. From the experimental results, we derive the following observations:

Observation 1: Incorporating semantic information into features significantly improves performance. A consistent performance hierarchy is evident among different feature types: $\boldsymbol{F}_S$ may suffice for basic tasks, but it lacks the depth to grasp semantic nuances. In contrast, LLM-generate message greatly improves the model's ability to handle ambiguity and context changes. On one hand, $\boldsymbol{F}_{LLM}$ effectively captures language regularities from extensive corpora, but the improvement offered by LLMs remains limitations without a well-designed prompt. On the other hand, $\boldsymbol{F}_{FLM}$ trained with supervised information demonstrates superior performance with large variations on several datasets.

Observation 2: Contextualized representation cascaded GNNs provide a strong baseline. The standard PLM pipeline, which focuses primarily on tasks like text classification, neglects the inherent correlations within global graph structures. PLMs with structure-aware cues obtain relative improvements ranging from 4% to 8%, indicating that natural language understanding is important on these datasets. On Cora and Photo datasets, the advantages of using GCN and SAGE are more pronounced, demonstrating the role of graph structures for enhanced predictive performance. However, the inclusion of neighboring information occasionally leads to performance degradation, as in PubMed, which may be caused by "heterophily" problem in graph.

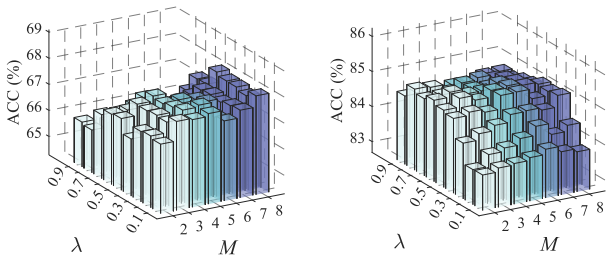
Observation 3: With enhancing semantic information, *HiTuner* achieves state-of-the-art performance. By distilling hierarchical contextualized representations from LLMs and leveraging semantics of category labels from fine-tuned PLMs, *HiTuner* strikes a balanced synergy between these two approaches, surpassing the performance of each method when applied independently. *HiTuner* can effectively model TAGs with outstanding performance across a wide range of datasets, highlighted by the best average performance (81.47%), demonstrates its effectiveness in combining semantic understanding with graph-based learning.

4.3 Ablation Study (RQ2)

We evaluate all components by removing each item from *HiTuner* in turn: (a) *w/o* structural information: replacing GNN with MLP; (b) *w/o* hierarchical representations: only take the last layer of LLMs as input. (c) *w/o* task-specific knowledge: λ is set to 0, training the *HiTuner* without any task-specific knowledge. (d) *w/o* confidence-based fusion: simple average fusion is used to instead. **Observation 4: *HiTuner* benefits**

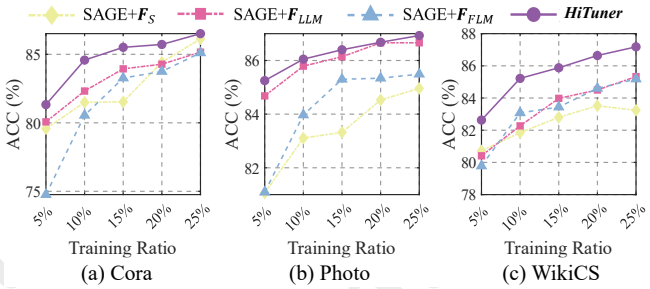
	CiteSeer	Cora	Instagram	Photo	PubMed	WikiCS	*Average*
w/o structural information	68.81 ± 3.46	71.14 ± 3.72	65.34 ± 4.29	74.69 ± 0.31	93.43 ± 0.47	83.15 ± 1.00	*76.09*
w/o hierarchical reprensenattions	71.00 ± 3.07	83.95 ± 2.18	65.15 ± 1.94	84.87 ± 0.31	93.36 ± 0.52	85.14 ± 0.89	*80.58*
w/o task-specific knowledge	69.97 ± 3.10	82.66 ± 5.80	64.59 ± 1.25	85.36 ± 0.52	87.70 ± 1.20	83.65 ± 1.30	*78.99*
w/o confidence-based fusion	70.41 ± 2.42	82.69 ± 5.98	64.49 ± 1.23	85.64 ± 0.25	93.24 ± 0.34	85.08 ± 1.26	*80.26*
HiTuner	**71.35 ± 3.28**	**84.58 ± 2.69**	**68.10 ± 0.18**	**86.09 ± 0.58**	**93.50 ± 0.42**	**85.20 ± 1.19**	***81.47***

Table 3: Ablation study of *HiTuner* where *w/o* denotes the removal of a specific component.



(a) Instagram (b) WikiCS

Figure 5: Parameter sensitivity analysis of λ and M in *HiTuner* on two types of datasets.



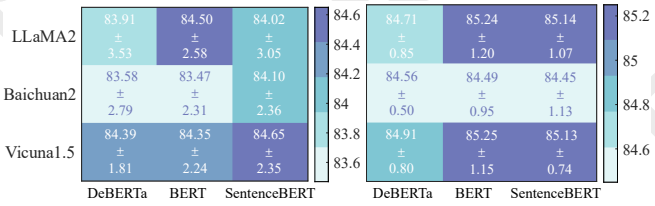
(a) Cora (b) Photo (c) WikiCS

Figure 6: Test accuracy of various baselines on three datasets (Cora, Photo, WikiCS) under different training data ratios.

from integrating complementary textual and structural semantic. From Table 3, we observe that removing any components leads to decrease performance. Furthermore, the importance of these components across different datasets. Overall, removing structural information in the most significant performance drop across all datasets, aligning with the nature of TAGs that structure-related information plays a more pivotal role than pure text. Excluding task-specific knowledge also impacts performance, confirming that fine-tuning complements general LLM-derived features by adapting them to dataset-specific nuances.

4.4 Parameter Sensitivity (RQ3)

The trade-off parameter λ is chosen from 0.1 to 0.9 with step size 0.1, and the number of layers M varies from 2 to 8. As demonstrated in Figure 5, we arrive at **Observation 5: *HiTuner* effectively balances effectiveness and model complexity across diverse datasets.** λ regulates the mixing ratio of general language knowledge and task-specific knowledge, where higher λ values increase the influence of task-specific knowledge but may impact the model's capacity for generalization to unseen data. Performance fluctuates noticeably as



(a) Cora (b) WikiCS

Figure 7: Test accuracy of *HiTuner* based on different fine-tuned PLMs and LLMs across two datasets (Cora, WikiCS).

M changes, suggesting that adding more layers does not consistently improve results. Overall, moderate λ values combined with moderate M values achieve desired performance.

4.5 Scalability Study (RQ4)

We conduct a scalability study focusing on the performance across varying proportions of training data, and different LLMs (Vicuna1.5-7B [Chiang *et al.*, 2023] and Baichuan2-7B [Yang *et al.*, 2023]). As depicted in Figure 6, *HiTuner* consistently outperforms several baseline models as the training ratio increases, highlighting its ability to effectively integrate features for enhancing generalization. Furthermore, as shown in Figure 7, the choice of LLMs does not have a significant impact on the performance, demonstrating the favorable compatibility of the proposed framework. It can derive **Observation 6: *HiTuner* is a generalizable and scalable framework, proficient in adapting to diverse training environments and model architectures.**

5 Conclusion

To enhance the task-specific understanding and reasoning capabilities of LLMs, we proposed *HiTuner*, a framework designed to bridge the gap between pre-training and downstream objectives. Focusing on fully unleashing the potential of LLMs for TAGs, *HiTuner* efficiently distilled hierarchical contextualized representations while minimizing unnecessary computational overhead. Specifically, we fine-tuned small-scale PLMs, infusing task-oriented knowledge at a lower cost and making the framework more adaptable to diverse contexts. Furthermore, to enhance the quality of node representations in TAGs, we integrate these refined representations into GNNs, enabling them to capture and interpret structure-related semantics within TAGs. Extensive experiments across various TAG benchmarks reveal that *HiTuner* significantly boosts baseline performance.

Acknowledgments

This work is in part supported by the National Natural Science Foundation of China under Grants U21A20472 and 62276065, the Fujian Provincial Natural Science Foundation of China under Grant 2024J01510026, and the Fujian Provincial Department of Education Youth Project of China under Grant JZ230011.

References

[Apidianaki, 2023] Marianna Apidianaki. From word types to tokens and back: A survey of approaches to word meaning representation and interpretation. *Computational Linguistics*, 49(2):465–523, 2023.

[Chen *et al.*, 2024] Zhikai Chen, Haitao Mao, Hang Li, Wei Jin, Hongzhi Wen, Xiaochi Wei, Shuaiqiang Wang, Dawei Yin, Wenqi Fan, Hui Liu, et al. Exploring the potential of large language models s (LLMs) in learning on graphs. *ACM SIGKDD Explorations Newsletter*, 25(2):42–61, 2024.

[Chiang *et al.*, 2023] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, march 2023. *URL https://lmsys. org/blog/2023-03-30-vicuna*, 3(5), 2023.

[Devlin *et al.*, 2019] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, pages 4171–4186, 2019.

[Hamilton *et al.*, 2017] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. volume 30, pages 1024–1034, 2017.

[He *et al.*, 2024] Xiaoxin He, Xavier Bresson, Thomas Laurent, Adam Perold, Yann LeCun, and Bryan Hooi. Harnessing explanations: LLM-to-LM interpreter for enhanced text-attributed graph representation learning. In *ICLR*, pages 1–22, 2024.

[Huang *et al.*, 2024] Xuanwen Huang, Kaiqiao Han, Yang Yang, Dezheng Bao, Quanjin Tao, Ziwei Chai, and Qi Zhu. Can gnn be good adapter for llms? In *WWW*, pages 893–904, 2024.

[Kipf and Welling, 2017] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017.

[Liu *et al.*, 2019] 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*, arXiv/1907.11692, 2019.

[Luo *et al.*, 2024] Renqiang Luo, Huafei Huang, Shuo Yu, Zhuoyang Han, Estrid He, Xiuzhen Zhang, and Feng Xia. FUGNN: Harmonizing fairness and utility graph neural networks. In *SIGKDD*, pages 2072–2081, 2024.

[Pan *et al.*, 2024] Bo Pan, Zheng Zhang, Yifei Zhang, Yuntong Hu, and Liang Zhao. Distilling large language models for text-attributed graph learning. In *CIKM*, page 1836–1845, 2024.

[Perin *et al.*, 2024] Gabriel Perin, Xuxi Chen, Shusen Liu, Bhavya Kailkhura, Zhangyang Wang, and Brian Gallagher. RankMean: Module-level importance score for merging fine-tuned LLM models. In *ACL*, pages 1776–1782, 2024.

[Reimers and Gurevych, 2019] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *EMNLP*, pages 3980–3990, 2019.

[Shen *et al.*, 2024] Junhong Shen, Neil A. Tenenholtz, James Brian Hall, David Alvarez-Melis, and Nicolò Fusi. Tag-llm: Repurposing general-purpose llms for specialized domains. In *ICML*, 2024.

[Tang *et al.*, 2024] Jiabin Tang, Yuhao Yang, Wei Wei, Lei Shi, Lixin Su, Suqi Cheng, Dawei Yin, and Chao Huang. Graphgpt: Graph instruction tuning for large language models. In *SIGIR*, pages 491–500, 2024.

[Touvron *et al.*, 2023] Hugo Touvron, Louis Martin, and et al Kevin Stone. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288, 2023.

[Wu *et al.*, 2024] Songhao Wu, Quan Tu, Hong Liu, Jia Xu, Zhongyi Liu, Guannan Zhang, Ran Wang, Xiuying Chen, and Rui Yan. Unify graph learning with text: Unleashing LLM potentials for session search. In *WWW*, pages 1509–1518, 2024.

[Yang *et al.*, 2023] Aiyuan Yang, Bin Xiao, Bingning Wang, Borong Zhang, Ce Bian, Chao Yin, Chenxu Lv, Da Pan, Dian Wang, Dong Yan, et al. Baichuan 2: Open large-scale language models. *arXiv preprint arXiv:2309.10305*, 2023.

[Yu *et al.*, 2024] Xingtong Yu, Zhenghao Liu, Yuan Fang, Zemin Liu, Sihong Chen, and Xinming Zhang. Generalized graph prompt: Toward a unification of pre-training and downstream tasks on graphs. *IEEE Transactions on Knowledge and Data Engineering*, 36(11):6237–6250, 2024.

[Yuan *et al.*, 2023] Hongyi Yuan, Zheng Yuan, Chuanqi Tan, Fei Huang, and Songfang Huang. Hype: Better pre-trained language model fine-tuning with hidden representation perturbation. In *ACL*, 2023.

[Zhu *et al.*, 2024] Yun Zhu, Yaoke Wang, Haizhou Shi, and Siliang Tang. Efficient tuning and inference for large language models on textual graphs. In *IJCAI*, pages 5734–5742, 2024.