Large Language Model Meets Graph Neural Network in Knowledge Distillation

Shengxiang Hu[1], Guobing Zou[1*], Song Yang[1], Shiyi Lin[1], Yanglan Gan[2], Bofeng Zhang[3], Yixin Chen[4]

[1]School of Computer Engineering and Science, Shanghai University, Shanghai, China
[2]School of Computer Science and Technology, Donghua University, Shanghai, China
[3]School of Computer and Information Engineering, Shanghai Polytechnic University, Shanghai, China
[4]Department of Computer Science and Engineering, Washington University in St. Louis, St. Louis, MO 63130 USA
{shengxianghu, gbzou, yangsong, sylin}@shu.edu.cn, ylgan@dhu.edu.cn, bfzhang@sspu.edu.cn, chen@cse.wustl.edu

Abstract

While Large Language Models (LLMs) show promise for Text-Attributed Graphs (TAGs) learning, their deployment is hindered by computational demands. Graph Neural Networks (GNNs) are efficient but struggle with TAGs' complex semantics. We propose **LinguGKD**, a novel LLM-to-GNN knowledge distillation framework that enables transferring both local semantic details and global structural information from LLMs to GNNs. First, it introduces TAG-oriented instruction tuning, enhancing LLMs with graph-specific knowledge through carefully designed prompts. Next, it develops a layer-adaptive multi-scale contrastive distillation strategy aligning LLM and GNN features at multiple granularities, from node-level to graph-level. Finally, the distilled GNNs combine the semantic richness of LLMs with the computational efficiency of traditional GNNs. Experiments demonstrate that LinguGKD outperforms existing graph distillation frameworks, the distilled simple GNNs achieve comparable or superior performance to more complex GNNs and teacher LLMs, while maintaining computational efficiency. This work bridges the gap between LLMs and GNNs, facilitating advanced graph learning in resource-constrained environments and providing a framework to leverage ongoing LLM advancements for GNN improvement.

Introduction

Text-Attributed Graphs (TAGs) integrate structured graph data with rich textual information, providing a comprehensive representation of complex systems across diverse domains (Li et al. 2022; Yang and Shi 2024). Graph Neural Networks (GNNs) (Veličković et al. 2018; Chen et al. 2020) excel at interpreting graph structures and offer efficient inference for various downstream tasks, making them a popular choice for graph-based learning. However, they struggle with semantic processing (Li et al. 2023), especially as the complexity and volume of associated textual data increase.

The advent of Large Language Models (LLMs) like ChatGPT (Ouyang et al. 2022) and Llama (Touvron et al. 2023) brings new opportunities for TAG processing. LLMs demonstrate exceptional capabilities in understanding complex semantics and capturing entity relationships, making them promising for graph learning tasks (Wang et al. 2024; Fatemi, Halcrow, and Perozzi 2023; Ye et al. 2024). Recent studies show that integrating knowledge graphs can further enhance LLMs' reasoning abilities (Pan et al. 2024b; Li et al. 2024), highlighting the potential synergy between LLMs and graph structures. However, applying LLMs to graph learning faces significant challenges, which include high computational and storage demands due to large parameter sizes (often billions[1]), extended latency during inference limiting practical applications.

To leverage the semantic understanding of LLMs while maintaining the efficiency of GNNs, a promising direction is to explore knowledge distillation (KD) techniques (Chen et al. 2022; Samy, T. Kefato, and Girdzijauskas 2023; Joshi et al. 2024), which offer the potential to transfer insights from complex LLMs to more compact GNNs, thereby potentially optimizing graph reasoning tasks for resource-constrained scenarios. However, the significant architectural differences between LLMs and GNNs pose substantial challenges for effective knowledge transfer, a problem that remains largely unexplored in the context of graph learning.

To bridge this gap between LLMs and GNNs, we propose **Lingu**istic **G**raph **K**nowledge **D**istillation (**LinguGKD**), a novel and versatile LLM-to-GNN knowledge distillation framework. LinguGKD operates in two key stages: First, it employs instruction tuning of pre-trained LLMs using carefully designed graph-oriented prompts, creating an effective teacher LLM (**LinguGraph LLM**) with enhanced graph understanding capabilities. This stage enables the LLM to better interpret and reason about graph structures and node attributes in natural language. Second, we develop a layer-adaptive multi-scale contrastive distillation strategy, which utilizes a contrastive learning framework to align LLM and GNN features at both node and graph levels. To further enhance knowledge transfer across different neural network depths, we introduce a layer-adaptive mechanism to dynamically adjust the importance of knowledge distillation at each layer, allowing for more flexible and effective transfer of LLM's hierarchical understanding to GNN's message-passing layers. By combining the downstream task loss with these distillation objectives, LinguGKD enables GNNs to effectively learn from LLMs while maintaining their compu-

*Corresponding author

[1]https://huggingface.co/spaces/HuggingFaceH4/open_llm_leaderboard

tational efficiency, thus optimizing graph reasoning tasks for resource-constrained environments.

Our main contributions are summarized as follows:

- We conceptualize the novel research problem of knowledge distillation from LLMs to GNNs and propose LinguGKD, an innovative graph knowledge distillation framework that leverages graph-oriented teacher LLMs to enrich student GNNs' feature learning capabilities while maintaining their high reasoning efficiency, offering significant potential for real-world applications.
- We design a unique layer-adaptive multi-scale contrastive distillation strategy within the LinguGKD framework. By employing a combination of local and global alignment losses with trainable layer-adaptive parameters, we ensure effective synchronization of hierarchical node features between the teacher LLM and the student GNN, thus guaranteeing the transfer of deep semantic knowledge and complex graph structural understanding.
- Through extensive experiments evaluations across diverse LLM-GNN combinations and multiple benchmark datasets, we demonstrate that LinguGKD significantly enhances GNN accuracy while maintaining a lightweight model structure. Our framework exhibits strong adaptability to different LLM architectures, enabling continuous improvement of GNN performance as LLM technology advances. This adaptability, combined with effective knowledge distillation, achieves an optimal balance between performance and efficiency, making it practical to deploy high-performing graph learning models in diverse resource-constrained real-world scenarios.

Preliminaries

Definition 1 (Text-Attributed Graph) *A Text-Attributed Graph (TAG) is a graph where each node is associated with textual data. Formally, a TAG is denoted as $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{X})$, where $\mathcal{V} = \{v_i\}_{i=1}^n$ represents the set of nodes, $\mathcal{E}$ is the set of edges, and $\mathcal{X} = \{x_i\}_{i=1}^n$ denotes the node attributes, where x_i represents the textual attribute of node v_i.*

Definition 2 (Graph Neural Network) *Graph Neural Networks (GNNs) are specialized for handling graph-structured data, primarily through a k-layer message-passing mechanism (Kipf and Welling 2017), enabling the capture and analysis of k-hop node relationships. The general form of message passing in GNNs can be defined as:*

$$\mathbf{h}_v^{(k)} = f\left(\mathbf{h}_v^{(k-1)}, AGG\left(\{\mathbf{h}_u^{(k-1)} : u \in \mathcal{N}(v)\}\right)\right) \quad (1)$$

where $\mathbf{h}_v^{(k)}$ is node v's feature at the k-th layer, $\mathcal{N}(v)$ includes v's neighboring nodes, $f(\cdot)$ is a node update function, and $AGG(\cdot)$ is a neighborhood aggregation function.

Definition 3 (LLM-to-GNN Knowledge Distillation) *LLM-to-GNN Knowledge Distillation (KD) aims to transfer semantic understanding from a large teacher LLM to a compact student GNN for processing TAGs. It enhances GNN's capability in capturing complex semantic relationships while maintaining its structural learning advantages and computational efficiency. The ultimate goal is to achieve a balance between model performance, inference speed, and memory footprint, enabling the deployment of semantically-rich graph learning models in resource-constrained environments.*

Approach

Figure 1 illustrates the LinguGKD framework for TAG-oriented graph knowledge distillation. Our approach adopts a modular architecture with three decoupled stages to ensure both effectiveness and efficiency. We first fine-tune a pre-trained language model through carefully designed instruction prompts to enhance its graph understanding capabilities, which serves as a one-time process and can leverage existing graph-specific LLMs as they become available. The fine-tuned LLM then acts as a teacher model to extract rich semantic features from graph nodes, which are cached to enable efficient reuse across multiple distillation experiments. Finally, we train a lightweight student GNN through our proposed layer-adaptive multi-scale contrastive distillation mechanism, utilizing the cached LLM features while keeping the teacher model frozen. This decoupled design not only ensures computational efficiency but also provides flexibility to incorporate advances in both LLM and GNN architectures, making our framework adaptable to future developments in both domains.

TAG Instruction Tuning of Pre-trained LLM

To address the lack of out-of-the-box graph-specific LLMs, we propose a tailored instruction tuning approach for graph tasks, which builds upon recent work in graph-oriented language models (Ye et al. 2024; Chen et al. 2024) and advanced LLM instruction tuning strategies (Wei et al. 2022; Zhang et al. 2023). This approach aims to create a teacher LLM capable of effectively processing and understanding graph-structured data, leveraging instruction tuning's proven efficacy in enhancing LLM capabilities for specialized tasks.

For a given center node v_i and a maximum neighbor hop k, we construct a series of subgraphs $\{\mathcal{G}_i^{(l)}\} = \{(v_i, \mathcal{N}^{(l)}(v_i), \mathcal{E}_i^{(l)}, \mathcal{X}_i^{(l)})\}_{l=0}^k$, where $\mathcal{N}^{(l)}(v_i)$ is v_i's l-hop neighbors, $\mathcal{E}_i^{(l)}$ is the set of edges between these nodes, and $\mathcal{X}_i^{(l)}$ is the set of corresponding node text attributes. For each subgraph, we design a three-part instruction prompt:

- Task-specific instruction ($\mathcal{I}$): Defines the graph task, e.g., "Classify the node in [graph type of $\mathcal{G}_i^{(l)}$], represented as (node ID, degree, attributes). Categorize the node into [classification categories] based on [task criterion]."
- Structural description (τ_l): Converts the subgraph structure into natural language, e.g., "(node [id], [degree], [attributes]) is connected within [l] hops to [l-hop neighbors in $\mathcal{N}^{(l)}(v_i)$] through [intermediate paths in $\mathcal{E}_i^{(l)}$]."
- Task-relevant query ($\mathcal{Q}$): Poses a task-specific question, e.g., "What category should the node ([id], [degree], [attributes]) be classified as?"

Following the instruction-following model design in (Taori et al. 2023), we then construct the instruction prompt

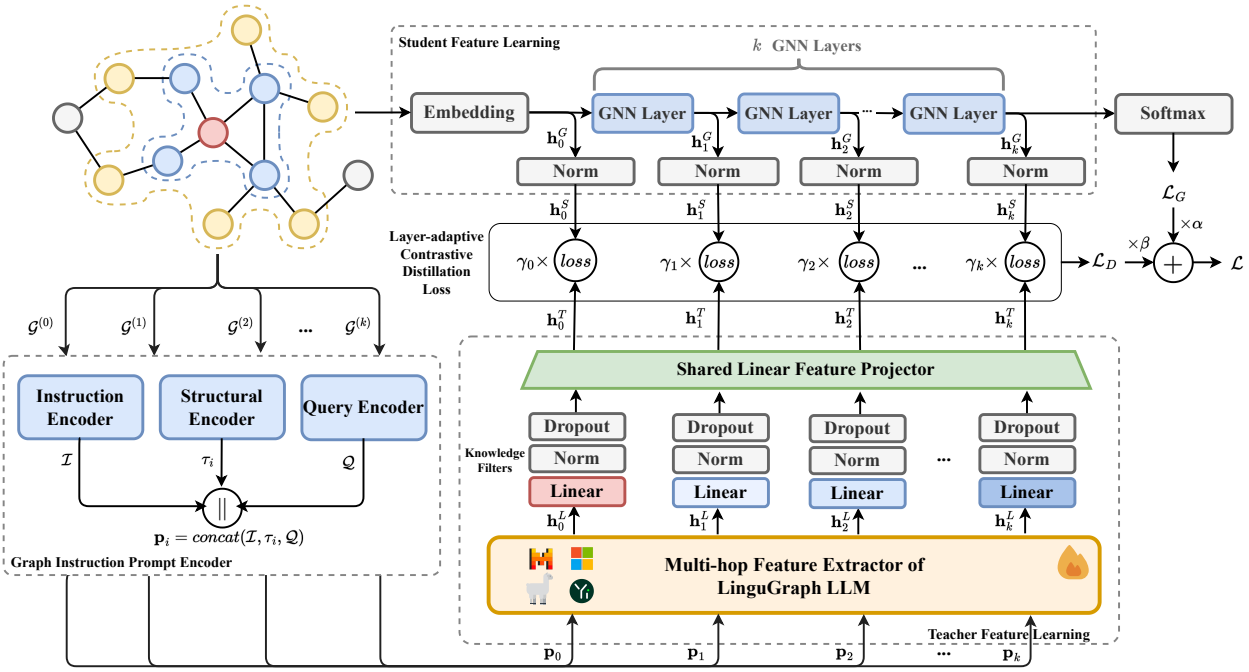


Figure 1: The LinguGKD framework for TAG-oriented LLM-to-GNN knowledge distillation.

$\mathbf{p}_l$ for the l-th order subgraph by concatenating the task-specific instruction, structural description, and task-relevant query through a predefined template:

$$\mathbf{p}_l = \text{concat}(\mathcal{I}, \tau_l, \mathcal{Q}) \tag{2}$$

To fine-tune the pre-trained LLM, we employ the negative log-likelihood loss as the objective function:

$$\mathcal{L}_{\text{NLL}}(\mathcal{P}) = -\sum_{l=0}^{k} \sum_{\mathbf{p}_l \in \mathcal{P}_l} \sum_{j=1}^{|y|} \log p(\hat{y}_j | \mathbf{p}_l, \hat{y}_{<j}) \tag{3}$$

where $\hat{y}_j$ is the j-th token generated by the LLM for the node label, $\mathcal{P}_l$ is the set of all instruction prompts for the l-th order subgraph, and $\mathcal{P} = \bigcup_{l=0}^{k} \mathcal{P}_l$ is the set of all instruction prompts across all orders.

Through this process, we obtain an LLM adapted for graph tasks, which we term LinguGraph LLM. It serves as the teacher model in the subsequent knowledge distillation process, providing rich semantic and structural knowledge to the student GNN model.

Knowledge Distillation from LinguGraph LLM to GNN

Teacher Feature Learning via LinguGraph LLM The LinguGraph LLM, fine-tuned with graph-specific instruction prompts, serves as our teacher model for extracting semantically-rich node features. Inspired by (Xiao et al. 2024), we observe that tailored instructions significantly enhance the LLM's proficiency in generating semantic features. Consequently, we leverage the entire instruction prompt set $\mathcal{P}$ for the extraction of node semantic features, rather than limiting it to the structural prompt set $\mathcal{T}$.

For an instruction prompt $\mathbf{p}_l \in \mathcal{P}$, we extract the l-th order node latent feature through the LLM's transformer (Vaswani et al. 2017) architecture:

$$\mathbf{h}_l^L = \text{Transformer}(\text{Embedding}^L(\mathbf{p}_l); W_{\text{tr}})_{|\mathbf{p}_l|,:} \tag{4}$$

where $\text{Embedding}^L(\cdot)$ is the embedding layer of the LLM, W_{tr} denotes the parameters of the transformer layers. The output $\mathbf{h}_l^L \in \mathbb{R}^{d_L}$ represents the feature vector corresponding to the last token of the processed instruction prompt after self-attention, d_L denotes the dimension of the LLM's hidden state.

To prepare these LLM-extracted features for knowledge distillation, we introduce a two-step processing mechanism:

$$\mathbf{h}_l^T = \mathcal{M}_p(\text{LayerNorm}(\mathcal{M}_f^l(\mathbf{h}_l^L)); W_p, b_p) \tag{5}$$

where $\mathcal{M}_f^l$ is a hop-specific neural knowledge filter with learned parameters, $\text{LayerNorm}(\cdot)$ is a layer normalization operation, and $\mathcal{M}_p$ is a shared linear projector with parameters W_p and b_p. This design distills pertinent layer-wise information and aligns features from different hops into a unified distillation space $\mathbf{h}_l^T \in \mathbb{R}^{d_k}$, where d_k is the distillation space dimension. The process yields a set of hierarchical teacher node features $\mathcal{F}_T = \{\mathbf{h}_l^T\}_{l=0}^k$, capturing semantic information at multiple neighborhood levels.

Student Feature Learning via GNN The student GNN $\mathcal{M}_S$ extracts multi-hop node features through a message-passing mechanism (Kipf and Welling 2017), capturing the graph's structural information. While various GNN architectures exist (Veličković et al. 2018; Hamilton, Ying, and Leskovec 2017; Xu et al. 2018), they share a common principle of aggregating information from neighboring nodes.

Our framework is compatible with any off-the-shelf GNN variant, allowing flexibility in model choice.

For a given k-hop neighbor subgraph $\mathcal{G}_i^{(k)}$ of a central node v_i, the k-order message aggregation proceeds as:

$$\mathbf{h}_j^{(0)} = \text{Embedding}^G(x_j), \quad \forall x_j \in \{x_i\} \cup \mathcal{X}_i^{(k)} \tag{6}$$

$$\mathbf{m}_{i \leftarrow j}^{(l)} = \mathcal{M}_{\text{msg}}^{(l)}(\mathbf{h}_i^{(l-1)}, \mathbf{h}_j^{(l-1)}, e_{ij}; W_{\text{msg}}^{(l)}) \tag{7}$$

$$\mathbf{h}_l^G = \mathcal{M}_{\text{update}}^{(l)}(\mathbf{h}_i^{(l-1)}, \bigoplus_{v_j \in \mathcal{N}(v_i)} \mathbf{m}_{i \leftarrow j}^{(l)}; W_{\text{update}}^{(l)}) \tag{8}$$

where $\text{Embedding}^G(\cdot)$ initializes node features from textual attributes x_j. For $0 < l \leq k$, $\mathbf{m}_{i \leftarrow j}^{(l)}$ represents the message from node j to node i at the l-th layer. $\mathcal{M}_{\text{msg}}^{(l)}(\cdot)$ constructs messages, $\bigoplus$ denotes a differentiable, permutation-invariant aggregation function, and $\mathcal{M}_{\text{update}}^{(l)}(\cdot)$ updates node features. $W_{\text{msg}}^{(l)}$, and $W_{\text{update}}^{(l)}$ are learnable parameters. To align with the teacher's feature space, we apply a normalization layer:

$$\mathbf{h}_l^S = \text{Norm}(\mathbf{h}_l^G), \quad 0 \leq l \leq k \tag{9}$$

where $\mathbf{h}_l^S \in \mathbb{R}^{d_k}$ denotes the student knowledge, and $\text{Norm}(\cdot)$ is typically batch or layer normalization. The process yields a set of hierarchical student node features $\mathcal{F}_S = \{\mathbf{h}_l^S\}_{l=0}^k$, capturing structural information at multiple scales.

Layer-Adaptive Multi-scale Contrastive Distillation Traditional knowledge distillation methods face significant challenges in transferring knowledge from LLMs to GNNs due to their inherent heterogeneity. To address this, we propose an approach that aims for the GNN to learn both the feature distribution of the LLM and the relationships between these features, rather than directly fitting feature values. This allows for a more flexible knowledge transfer that respects graph structures while incorporating LLM's semantic understanding. We achieve this through a combination of local and global distillation losses.

The local distillation loss aims to align individual node representations between the GNN and LLM. However, directly minimizing the distance between these representations is suboptimal due to the fundamentally different architectures and objectives of GNNs and LLMs. To address this challenge, we leverage contrastive learning (Tian, Krishnan, and Isola 2020), which offers several key advantages: first, it preserves the LLM's semantic structure while adapting to the GNN's feature space by learning relative relationships rather than absolute values; second, it enhances the discriminative power of learned representations through hard negative mining. Thus, we formally define the local contrastive distillation loss as:

$$\mathcal{L}_{D_l}^l = -\mathbb{E}\left[\log \frac{e^{s(\mathbf{h}_l^S, \mathbf{h}_l^T)/\tau}}{e^{s(\mathbf{h}_l^S, \mathbf{h}_l^T)/\tau} + \sum_{j=1}^m e^{s(\mathbf{h}_l^S, \mathbf{h}_{l,j}^-)/\tau}}\right] \tag{10}$$

where $s(\cdot, \cdot)$ denotes a similarity function, τ is the temperature, $\mathbf{h}_l^S$ and $\mathbf{h}_l^T$ are l-th order features from the GNN and LLM, $\mathbf{h}_{l,j}^-$ represents teacher features from different categories serving as hard negative samples (Robinson et al. 2021), and m is the number of hard negative samples selected for each positive pair.

While local loss aligns individual node representations, it may not fully capture the global feature space structure of the LLM. To address this, we introduce a global alignment loss to ensure that the GNN's overall feature distribution and inter-feature relationships mirror those of the LLM. Specifically, we employ KL divergence to measure the discrepancy between feature distributions of the student GNN and teacher LLM at layer l:

$$\mathcal{L}_{D_g}^l = \text{KL}(\text{P}_l^S || \text{P}_l^T) \tag{11}$$

where P_l^S and P_l^T represent the distributions of pairwise feature similarities. Let H_l^S and $\text{H}_l^T \in \mathbb{R}^{n \times d_k}$ denote the node feature matrices at layer l for the GNN and LLM respectively. The similarity distributions are computed using cosine similarity:

$$\text{P}_l^S[i,j] = \frac{\text{H}_l^S[i,:] \cdot \text{H}_l^S[j,:]^\top}{\|\text{H}_l^S[i,:]\|_2 \cdot \|\text{H}_l^S[j,:]\|_2} \tag{12}$$

$$\text{P}_l^T[i,j] = \frac{\text{H}_l^T[i,:] \cdot \text{H}_l^T[j,:]^\top}{\|\text{H}_l^T[i,:]\|_2 \cdot \|\text{H}_l^T[j,:]\|_2} \tag{13}$$

The combination of local and global losses ensures comprehensive knowledge transfer. While the local loss focuses on node-level alignment, the global loss preserves the overall structure of the feature space, capturing higher-order relationships that might be missed by pairwise comparisons alone. Furthermore, to account for the varying importance of knowledge at different graph depths (Kipf and Welling 2017; Huang, Wang, and Chao 2019), we introduce a layer-adaptive mechanism:

$$\mathcal{L}_D = \sum_{l=0}^k \gamma_l (\mathcal{L}_{D_l}^l + \mathcal{L}_{D_g}^l) \tag{14}$$

where γ_l are trainable layer-adaptive parameters. This mechanism allows the model to dynamically adjust the contribution of each layer to the overall distillation process, adapting to different graph structures and tasks by focusing on the most informative layers for effective knowledge transfer.

Model Training

Training the student GNN involves both knowledge distillation from the teacher LLM and optimization for the specific downstream task. We formulate this as a multi-task joint optimization problem. Taking node classification as an example, we first define the task-specific prediction as:

$$\hat{\mathbf{y}} = \text{softmax}(W_G \mathbf{h}_k^S + \mathbf{b}_G) \tag{15}$$

where $\mathbf{h}_k^S$ is the final layer output of the GNN, and $W_G, \mathbf{b}_G$ are learnable parameters. The classification loss is then computed using cross-entropy:

$$\mathcal{L}_G = -\sum_{i=1}^{|\mathcal{D}_{\text{tr}}|} \mathbf{y}_i \log(\hat{\mathbf{y}}_i) \tag{16}$$

where $\mathbf{y}_i$ and $\hat{\mathbf{y}}_i$ are the true and predicted label distributions respectively, and $\mathcal{D}_{\text{tr}}$ is the training set. To balance knowledge distillation and task-specific performance, we define a joint loss function:

$$\mathcal{L} = \alpha\mathcal{L}_G + \beta\mathcal{L}_D \tag{17}$$

Here, α, β are tunable factors. The student GNN is trained end-to-end using mini-batch AdamW optimization, effectively balancing the transfer of rich semantic knowledge from the teacher LLM with the structural learning capabilities inherent to GNNs for the specific downstream task.

Complexity Analysis

Our proposed LinguGKD framework is structured into three modular stages, each orchestrated to manage computational complexity while maximizing efficiency and performance. Initially, the fine-tuning of a pre-trained LLM involves a complexity of $O(|\mathcal{D}_{tune}| \cdot L \cdot d_L^2)$, where $|\mathcal{D}_{tune}|$ represents the tuning dataset size, L is the prompt sequence length, and d_L denotes the hidden dimension of the LLM. This process is executed once and sets the foundation for subsequent stages, allowing pre-tuned graph-specific LLMs to be utilized as they become available.

In the feature extraction phase, the task's complexity is bounded by $O(|\mathcal{V}| \cdot k \cdot L \cdot d_L^2)$, where $|\mathcal{V}|$ is the total number of nodes and k is the maximum hop count around each node. The extracted features are systematically cached, thus enabling their reutilization across various knowledge distillation tasks, effectively amortizing the computational cost over time.

The final stage involves training the student GNN enhanced by our layer-adaptive multi-scale contrastive distillation mechanism. The training complexity for each epoch involves the GNN's message-passing steps, approximated by $O(|\mathcal{E}| \cdot d_k)$, where $|\mathcal{E}|$ signifies the edge count and d_k the distillation space dimension. Additionally, the local and global distillation processes contribute to this computational framework, with complexities of $O(|\mathcal{V}| \cdot k \cdot m \cdot d_k)$ and $O(|\mathcal{V}|^2 \cdot d_k)$ respectively—though the latter can be alleviated through efficient sampling strategies.

Overall, the modular design and strategic feature caching inherent in our framework ensure that, in practical applications, the resource demands of conducting knowledge distillation via LinguGKD remain comparable to those required for conventional GNN training. This efficiency is achieved while facilitating enhanced performance, thus reflecting the judicious integration of LLM-derived semantic insights into the GNN paradigm.

Experiments

Experimental Setup

Datasets and Model Selection We evaluated our LinguGKD framework on three widely-used benchmark datasets for node classification: Cora, PubMed (Yang, Cohen, and Salakhudinov 2016), and Arxiv (Hu et al. 2020). These datasets represent academic papers as nodes and citations as edges. Table 1 summarizes the key statistics of these datasets. Due to the lack of initial text attributes for each node in the original datasets, we reconstructed titles, abstracts, and other text attributes for each node following the method described in (He et al. 2024) for graph instruction tuning of the teacher LLM.

	Cora	**PubMed**	**Arxiv**
# Node	2,708	19,717	169,343
# Edge	5,429	44,338	1,166,243
# Class	7	3	40
# Features	1433	500	128
Embedding Tech.	BoW	TF-IDF	Skip-gram
$\lvert\mathcal{D}_{tr}\rvert : \lvert\mathcal{D}_{val}\rvert : \lvert\mathcal{D}_{test}\rvert$	6:2:2	6:2:2	5.4:1.8:2.8

Table 1: Dataset Statistics

For teacher LLMs, we selected Llama2-7B (Touvron et al. 2023) and Llama3-8B (Dubey et al. 2024). Our student GNNs include GCN (Kipf and Welling 2017), GAT (Veličković et al. 2018), GraphSAGE (Hamilton, Ying, and Leskovec 2017), and GIN (Xu et al. 2018).

Experimental Results and Analyses

Our experimental analysis aims to validate the effectiveness and efficiency of the proposed LinguGKD framework. We focus on addressing the following key research questions:

RQ1: How do LinguGraph LLMs compare to existing graph learning baselines?

RQ2: What performance gains do distilled GNNs achieve?

RQ3: How does LinguGKD compare to other graph knowledge distillation frameworks?

RQ4: What are the trade-offs between LinguGraph LLMs and distilled GNNs?

RQ5: How do different hyperparameters affect the performance of LinguGKD?

Performance of LinguGraph LLMs (RQ1) We compared LinguGraph LLMs against representative single-model graph learning baselines from three authoritative leaderboards[2] [3] [4]. Table 2 shows the node classification results, where LinguGraph LLMs consistently outperform existing baselines across all datasets. Notably, LinguGraph-Llama3 (8B) achieves state-of-the-art results with accuracies of 91.51%, 95.59%, and 79.73% on Cora, PubMed, and Arxiv, respectively. Performance improves with larger model sizes and pre-training corpora, supporting the potential of LLMs as foundational models for graph learning (Ye et al. 2024). These results establish LinguGraph LLMs as strong teacher models for our KD framework.

Performance Gains of Distilled GNNs (RQ2) As shown in Table 2, our LinguGKD framework significantly improves student GNN performance across datasets, with the distilled GNNs consistently outperforming their vanilla counterparts.

[2]https://paperswithcode.com/sota/node-classification-on-cora-60-20-20-random

[3]https://paperswithcode.com/sota/node-classification-on-pubmed-60-20-20-random

[4]https://ogb.stanford.edu/docs/leader_nodeprop/

Methods	Cora		PubMed		Methods	Arxiv	
	Acc.↑	F1↑	Acc.↑	F1↑		Acc.↑	F1↑
GCN	86.53±0.92	85.66±0.78	86.12±0.93	85.64±0.82	GCN	71.74±0.21	71.04±0.37
GAT	86.12±0.95	85.05±0.88	85.49±0.76	84.89±0.71	GAT	73.66±0.33	72.44±0.19
GraphSAGE	87.08±0.85	85.96±0.73	87.69±0.92	87.38±0.68	GraphSAGE	71.19±0.26	70.87±0.45
GIN	86.60±0.91	85.37±0.74	85.84±0.92	85.31±0.63	GIN	71.62±0.47	71.13±0.33
BernNet (He et al. 2021)	88.52±0.95	87.96±0.85	88.48±0.41	87.52±0.79	GTAN (Wu and Wang 2022)	72.97±0.17	71.77±0.22
FAGCN (Bo et al. 2021)	88.85±1.36	87.92±0.65	89.98±0.54	88.72±0.53	UniMP (Shi et al. 2021)	73.11±0.20	72.14±0.38
GCNII (Chen et al. 2020)	88.93±1.37	87.58±0.71	89.80±0.30	88.96±0.62	GCNII (Chen et al. 2020)	72.74±0.00	72.22±0.44
RevGAT (Li et al. 2021)	89.11±0.00	87.65±0.58	88.50±0.05	87.12±0.73	RevGAT (Li et al. 2021)	74.02±0.18	73.56±0.29
ACM-GCN+ (Luan et al. 2022)	89.75±1.16	88.94±0.54	90.96±0.62	89.77±0.51	E2EG (Dinh et al. 2023)	73.62±0.14	72.96±0.26
Graphormer (Ying et al. 2021)	80.41±0.30	79.98±0.56	88.24±1.50	87.52±0.71	SGFormer (Wu et al. 2024)	72.63±0.13	71.58±0.42
LinguGraph-Llama2 (7B)	88.19±0.83	88.12±0.73	94.09±0.78	93.55±0.61	LinguGraph-Llama2 (7B)	75.67±0.52	75.60±0.41
LinguGraph-Llama3 (8B)	91.51±0.46	**91.53±0.18**	**95.59±0.29**	**95.55±0.10**	LinguGraph-Llama3 (8B)	**79.73±0.18**	**79.29±0.56**
$GCN_{(Llama2)}$	90.59±0.71	89.62±0.66	88.97±0.82	88.56±0.71	$GCN_{(Llama2)}$	73.87±0.22	73.87±0.61
$GCN_{(Llama3)}$	90.77±0.28	90.35±0.37	89.76±0.44	89.46±0.37	$GCN_{(Llama3)}$	74.68±0.45	74.29±0.32
$GAT_{(Llama2)}$	90.33±0.67	89.72±0.59	87.93±0.28	87.42±0.36	$GAT_{(Llama2)}$	74.92±0.14	74.48±0.28
$GAT_{(Llama3)}$	91.51±0.35	91.45±0.58	88.31±0.76	87.93±0.65	$GAT_{(Llama3)}$	75.71±0.41	75.06±0.36
$GraphSAGE_{(Llama2)}$	90.22±0.77	89.89±0.19	89.96±0.50	89.67±0.34	$GraphSAGE_{(Llama2)}$	72.53±0.61	72.42±0.49
$GraphSAGE_{(Llama3)}$	**91.70±0.51**	91.08±0.62	90.14±0.56	89.96±0.48	$GraphSAGE_{(Llama3)}$	75.38±0.38	75.22±0.32
$GIN_{(Llama2)}$	90.26±0.67	89.20±0.48	87.73±0.29	87.20±0.30	$GIN_{(Llama2)}$	73.71±0.25	73.42±0.10
$GIN_{(Llama3)}$	91.33±0.28	91.05±0.53	89.22±0.79	88.87±0.61	$GIN_{(Llama3)}$	75.64±0.46	75.28±0.39
Avg. Dist. Gains	4.61%	5.22%	2.79%	2.84%	*Avg. Dist. Gains*	3.85%	4.22%

Table 2: Node classification performance of various graph learning models across selected datasets, with the highest-performing outcomes in **bold**, second-best scores highlighted in gray, and best baseline performances underlined. The prefix $LinguGraph-$ represents teacher LLMs obtained by fine-tuning different PLMs with graph instruction prompts, while GNNs with different subscripts represent student models distilled from the corresponding teacher LLMs indicated by the subscripts. The term *Avg. Dist. Gains.* refers to the average knowledge distillation gains obtained by different student GNNs.

Model	Cora	Arxiv
Teacher of baselines	88.93	73.91
Teacher of LinguGKD	**91.51**	**79.73**
KD (Hinton, Vinyals, and Dean 2015)	86.38	71.55
FitNet (Romero et al. 2015)	85.40	71.38
LSP (Yang et al. 2020)	84.92	71.52
GraphAKD (He et al. 2022)	86.39	-
G-CRD (Joshi et al. 2024)	-	71.64
LinguGKD	90.77	74.68

Table 3: Results (%) of different knowledge distillation methods. For baselines, GCNII is used as the teacher on Cora, and GAT on Arxiv. LinguGKD uses LinguGraph-Llama3 as the teacher. All methods use GCN as the student model. '-' means not available.

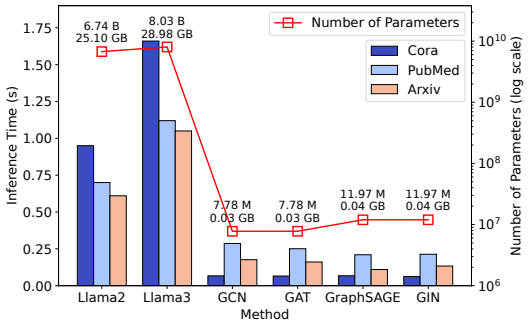
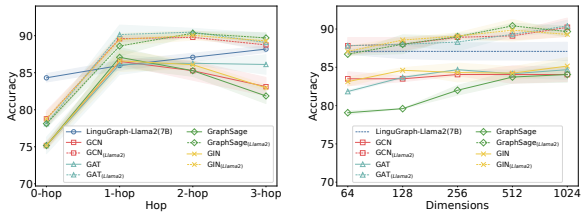


Figure 2: Model size and inference time of different models.

For instance, on Cora, $GCN_{(Llama3)}$ achieves 90.77% accuracy compared to 86.53% for vanilla GCN, representing a 4.24% improvement. The average distillation gains range from 2.79% (PubMed) to 4.61% (Cora). Notably, some distilled GNNs even surpass more complex models. For example, $GAT_{(Llama3)}$ on Cora (91.51% accuracy) outperforms RevGAT (89.11%) and ACM-GCN+ (89.75%).

The performance improvements across different datasets and GNN architectures demonstrate the effectiveness and generalizability of our LinguGKD framework. By transferring semantic knowledge and structural understanding from LLMs to GNNs, we enable simpler GNN models to achieve competitive or superior performance compared to more complex graph learning approaches.

Comparison with Other Graph Knowledge Distillation Frameworks (RQ3) To evaluate LinguGKD's effectiveness, we compared it with state-of-the-art graph knowledge distillation frameworks. Table 3 shows the performance of different approaches on Cora and Arxiv datasets. LinguGKD consistently outperforms existing methods, achieving accuracy improvements of 3.06 and 3.04 percentage points on Cora and Arxiv, respectively. This significant gain can be attributed to: (1) using LLMs as teacher models,



(a) (b)

Figure 3: The results of performance impacts as the variations of neighbor orders and hidden feature dimensions.

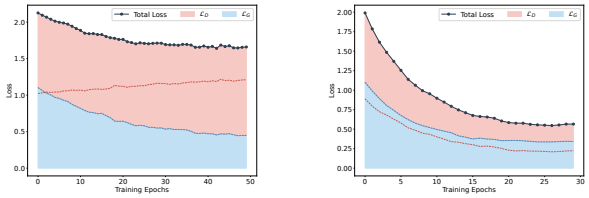
Methods	Cora		PubMed		Arxiv	
	T.	S.	T.	S.	T.	S.
Llama2	9.23	85.21	25.04	85.88	10.72	70.94
LinguGraph-Llama2	88.19	90.59	94.09	88.97	75.67	73.87
Llama3	14.92	85.48	30.17	86.06	15.31	70.58
LinguGraph-Llama3	**91.51**	**90.77**	**95.59**	**89.76**	**79.73**	**74.68**

Table 4: Node classification accuracy (%) of teacher LLMs (T.) and their distilled student GNNs (S.) before and after graph instruction tuning across different datasets.

providing richer semantic information, and (2) our novel layer-adaptive multi-scale contrastive distillation strategy that combines LLM's semantic understanding with GNN's structural capabilities. These results highlight the potential of leveraging LLMs for graph knowledge distillation and demonstrate our approach's effectiveness in transferring semantic and structural knowledge to lightweight GNNs.

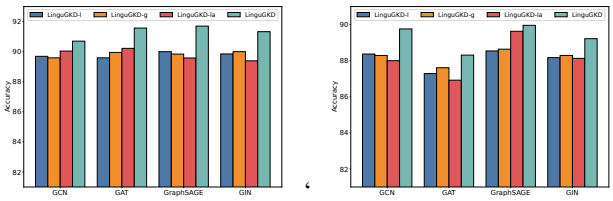
Trade-offs between LinguGraph LLMs and Distilled GNNs (RQ4) Figure 2 illustrates the trade-offs between model size and inference time for LinguGraph LLMs and distilled GNNs. LinguGraph LLMs offer superior performance but at the cost of significantly larger model sizes (6.74B-8.03B parameters, >25GB storage) and longer inference times (>0.5s). In contrast, distilled GNNs achieve comparable accuracy with much smaller footprints (few million parameters, 0.03-0.04GB) and faster inference. On Cora, $\text{GAT}_{(Llama3)}$ matches LinguGraph-Llama3's 91.51% accuracy while being orders of magnitude smaller and faster. This makes distilled GNNs ideal for resource-constrained or real-time applications, while LinguGraph LLMs are preferable when computational resources are ample and maximum accuracy is crucial. These trade-offs highlight the versatility of our LinguGKD framework, offering flexible solutions for various operational contexts in graph learning tasks.

Impact of Hyperparameters on LinguGKD Performance (RQ5) We conducted an extensive analysis of two critical hyperparameters: neighbor orders (k) and hidden feature dimensions (d_G) of GNNs, using the Cora dataset as our benchmark. As show in Figure 3a, increasing neighbor orders improves performance up to 2-hop, beyond which over-smoothing occurs in vanilla GNNs but is mitigated in distilled GNNs. Interestingly, Figure 3b reveals that while



(a) Loss Convergence Before LLM Fine-tuning (b) Loss Convergence After LLM Fine-tuning

Figure 4: Comparison of loss convergence before and after LLM fine-tuning. (a) Before fine-tuning, the distillation loss ($\mathcal{L}_D$) and task-specific loss ($\mathcal{L}_G$) show conflicting trends. (b) After fine-tuning, both losses decrease collaboratively, indicating better alignment of transferred knowledge.



(a) Accuracy on Cora (b) Accuracy on PubMed

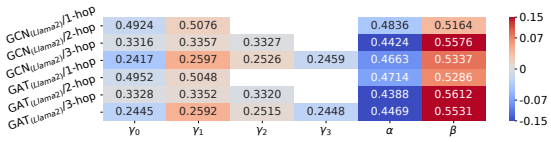
Figure 5: Performance comparison of GNNs distilled with LinguGKD and the ablated variants.

vanilla GNNs' performance plateaus at 128 dimensions, distilled GNNs show continuous improvement up to 1024 dimensions, indicating better utilization of semantic information. Based on these findings, we identified the 2-hop setting and 1024-dimensional hidden features as optimal, balancing performance and efficiency for LinguGKD. These results demonstrate LinguGKD's robustness across different graph structures and its ability to effectively leverage high-dimensional semantic spaces inherited from LLMs.

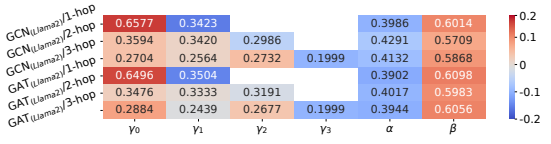
Ablation Study

Necessity of Graph Instruction Tuning In this study, we underscore the pivotal role of graph instruction tuning in significantly enhancing the performance of LLMs and the effectiveness of their knowledge distillation into GNNs. As detailed in Table 4, the initial performance of pre-tuned LLMs on graph-based node classification tasks was suboptimal, with accuracy ranging from 8% to 30%. Following graph instruction tuning, a substantial improvement was observed, with accuracy soaring to 75-95% across various datasets. This improvement is largely attributed to the LLMs' improved ability to align pre-trained knowledge with graph-specific tasks, thereby mitigating issues such as repetitive outputs commonly encountered with LLMs.

Furthermore, graph instruction tuning enhances the knowledge distillation process, as shown in the increased performance of distilled GNNs across all datasets (Table 4).



(a) Cora



(b) PubMed

Figure 6: Heatmap visualization of layer-adaptive factors (γ_l) and classification-distillation loss weights (α, β) during training on Cora and PubMed datasets. Darker colors indicate higher values relative to the average.

The collaborative decrease in distillation and task-specific losses post-tuning is depicted in Figure 4, illustrating better alignment of transferred knowledge with the target tasks. This aligns optimization objectives, leading to more effective and efficient learning processes. Our approach not only facilitates increased model accuracy but also demonstrates robustness and scalability across datasets of varying sizes and complexities, reinforcing its applicability in diverse graph learning scenarios.

Effectiveness of Layer-Adaptive Multi-scale Contrastive Distillation To evaluate our layer-adaptive multi-scale contrastive distillation strategy, we conducted ablation studies through three variants: LinguGKD-l (without local distillation loss), LinguGKD-g (without global alignment loss), and LinguGKD-la (without layer-adaptive mechanism). Results in Figure 5 show that the complete LinguGKD framework consistently outperforms across diverse GNN architectures and datasets. The ablation of local or global distillation components leads to suboptimal knowledge transfer, while removing the layer-adaptive mechanism significantly impairs the model's ability to capture multi-scale graph representations.

The efficacy of our adaptive mechanism is further substantiated through quantitative analysis presented in Figure 6, which reveals the distribution of layer-adaptive factors (γ_l) and loss weights (α, β). The dataset-specific adaptation patterns evidence the framework's ability to capture graph characteristics: higher first-order neighbor factors (γ_1) in Cora indicate dominance of local topology, while elevated structure-free features (γ_0) in PubMed suggest stronger dependence on node attributes. This automatic adaptation, combined with balanced optimization of distillation (β) and task-specific (α) objectives, demonstrates LinguGKD's advantage in maintaining knowledge fidelity across heterogeneous graphs.

Related Work

Recent advancements in graph learning have been significantly enriched by the integration of LLMs. Research in this domain primarily follows three approaches: LLM as Enhancer (LaE), LLM as Predictor (LaP), and LLM as Teacher (LaT). He et al. (He et al. 2024) proposed TAPE, which generates interpretive explanations and pseudo-labels to enrich graph's textual attributes, while Chen et al. (Chen et al. 2024) introduced the Knowledge Entity Augmentation (KEA) strategy, employing LLMs to generate knowledge entities with textual descriptions. Ye et al. (Ye et al. 2024) developed scalable prompting techniques that create direct relational links between nodes through natural language, outperforming traditional GNNs in node classification tasks. The emerging LaT approach focuses on transferring LLM knowledge to graph models. Pan et al. (Pan et al. 2024a) leverage explicit knowledge transfer through prompting and interpretation, extracting rationales from LLMs to guide graph models, showing particular effectiveness on datasets with complex domain knowledge like PubMed.

Parallel to these developments, graph knowledge distillation (He and Ma 2022; Wu et al. 2023; Joshi et al. 2024) has emerged as a crucial technique for enhancing GNNs' effectiveness and efficiency. He et al. (He and Ma 2022) proposed SGKD, which focuses on transferring final output representations, while Joshi et al. (Joshi et al. 2024) introduced G-CRD, aligning intermediate node embeddings between teacher and student models.

Our LinguGKD framework introduces a novel perspective within the LaT paradigm by directly aligning feature spaces to preserve implicit semantic relationships from LLM's attention mechanism, which differs from existing explicit knowledge extraction methods, showing superior performance on datasets with straightforward relationships while maintaining computational efficiency. The complementary strengths of explicit rationale-based methods and our feature alignment approach suggest promising directions for future research in LLM-to-GNN knowledge transfer.

Conclusion

In this paper, we introduced LinguGKD, a novel LLM-to-GNN knowledge distillation framework that effectively bridges the semantic understanding of LLMs with the efficiency of GNNs for TAGs. It combines TAG-oriented instruction tuning for LLMs with a layer-adaptive multi-scale contrastive distillation strategy, enabling efficient transfer of complex semantic knowledge. Extensive experiments demonstrated significant improvements in GNNs' predictive accuracy, while achieving superior inference speed and reduced resource requirements compared to LLMs. LinguGKD not only advances graph learning but also provides a practical solution for deploying advanced models in resource-constrained environments, establishing a promising direction for leveraging LLM advancements to enhance GNN performance. Our future work will explore extending LinguGKD to dynamic and heterogeneous graphs, further broadening its applicability in real-world scenarios.

Acknowledgments

This work was supported by National Natural Science Foundation of China (No. 62272290, 62172088), and Shanghai Natural Science Foundation (No. 21ZR1400400).

References

Bo, D.; Wang, X.; Shi, C.; and Shen, H. 2021. Beyond Low-Frequency Information in Graph Convolutional Networks. In *AAAI Conference on Artificial Intelligence (AAAI)*, volume 35, 3950–3957.

Chen, J.; Chen, S.; Bai, M.; Gao, J.; Zhang, J.; and Pu, J. 2022. SA-MLP: Distilling Graph Knowledge From GNNs Into Structure-Aware MLP. *arXiv Preprint arXiv:2210.09609*.

Chen, M.; Wei, Z.; Huang, Z.; Ding, B.; and Li, Y. 2020. Simple and Deep Graph Convolutional Networks. In *International Conference on Machine Learning (ICML)*, 1725–1735.

Chen, Z.; Mao, H.; Li, H.; Jin, W.; Wen, H.; Wei, X.; Wang, S.; Yin, D.; Fan, W.; and Liu, H. e. a. 2024. Exploring The Potential Of Large Language Models (LLMs) In Learning On Graphs. *ACM SIGKDD Explorations Newsletter*, 25(2): 42–61.

Dinh, T. A.; den Boef, J.; Cornelisse, J.; and Groth, P. 2023. E2EG: End-to-End Node Classification Using Graph Topology and Text-Based Node Attributes. In *IEEE International Conference on Data Mining Workshops (ICDMW)*, 1084–1091.

Dubey, A.; Jauhri, A.; Pandey, A.; et al. 2024. The Llama 3 Herd of Models. *arXiv preprint arXiv:2407.21783*.

Fatemi, B.; Halcrow, J.; and Perozzi, B. 2023. Talk Like A Graph: Encoding Graphs For Large Language Models. In *International Conference on Learning Representations (ICLR)*.

Hamilton, W.; Ying, Z.; and Leskovec, J. 2017. Inductive Representation Learning on Large Graphs. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30.

He, H.; Wang, J.; Zhang, Z.; and Wu, F. 2022. Compressing Deep Graph Neural Networks Via Adversarial Knowledge Distillation. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, 534–544.

He, M.; Wei, Z.; Xu, H.; and Others. 2021. Bernnet: Learning Arbitrary Graph Spectral Filters via Bernstein Approximation. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, 14239–14251.

He, X.; Bresson, X.; Laurent, T.; Perold; and Hooi, B. 2024. Harnessing Explanations: LLM-to-LM Interpreter for Enhanced Text-Attributed Graph Representation Learning. In *International Conference on Learning Representations (ICLR)*.

He, Y.; and Ma, Y. 2022. SGKD: A Scalable and Effective Knowledge Distillation Framework for Graph Representation Learning. In *IEEE International Conference on Data Mining Workshops (ICDMW)*, 666–673.

Hinton, G.; Vinyals, O.; and Dean, J. 2015. Distilling The Knowledge In A Neural Network. *arXiv Preprint arXiv:1503.02531*.

Hu, W.; Fey, M.; Zitnik, M.; Dong, Y.; Ren, H.; Liu, B.; Catasta, M.; and Leskovec, J. 2020. Open Graph Benchmark: Datasets for Machine Learning on Graphs. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, 22118–22133.

Huang, L.; Wang, C.-D.; and Chao, H.-Y. 2019. Higher-Order Multi-Layer Community Detection. In *The AAAI Conference on Artificial Intelligence (AAAI)*, volume 33, 9945–9946.

Joshi, C. K.; Liu, F.; Xun, X.; Lin, J.; and Foo, C. S. 2024. On Representation Knowledge Distillation for Graph Neural Networks. *IEEE Transactions on Neural Networks and Learning Systems*, 35(4): 4656–4667.

Kipf, T. N.; and Welling, M. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *International Conference on Learning Representations (ICLR)*.

Li, G.; Müller, M.; Ghanem, B.; and Koltun, V. 2021. Training Graph Neural Networks with 1000 Layers. In *International Conference on Machine Learning (ICML)*, 6437–6449.

Li, J.; Peng, H.; Cao, Y.; Dou, Y.; Zhang, H.; Yu, P. S.; and He, L. 2023. Higher-Order Attribute-Enhancing Heterogeneous Graph Neural Networks. *IEEE Transactions on Knowledge and Data Engineering*, 35(1): 560–574.

Li, Y.; Wang, X.; Ning, Y.; and Wang, H. 2022. Fairlp: Towards Fair Link Prediction on Social Network Graphs. In *International AAAI Conference on Web and Social Media (ICWSM)*, volume 16, 628–639.

Li, Y.; Zhang, R.; Liu, J.; and Liu, G. 2024. An Enhanced Prompt-Based LLM Reasoning Scheme via Knowledge Graph-Integrated Collaboration. *arXiv Preprint arXiv:2402.04978*.

Luan, S.; Hua, C.; Lu, Q.; Zhu, J.; Zhao, M.; Zhang, S.; Chang, X.-W.; and Precup, D. 2022. Revisiting Heterophily for Graph Neural Networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, 1362–1375.

Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A.; Schulman, J.; Hilton, J.; Kelton, F.; Miller, L.; Simens, M.; Askell, A.; Welinder, P.; Christiano, P. F.; Jan, L.; and Ryan, L. 2022. Training Language Models to Follow Instructions with Human Feedback. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, 27730–27744.

Pan, B.; Zhang, Z.; Zhang, Y.; Hu, Y.; and Zhao, L. 2024a. Distilling Large Language Models for Text-Attributed Graph Learning. In *ACM International Conference on Information and Knowledge Management (CIKM)*, 1836–1845.

Pan, S.; Luo, L.; Wang, Y.; Chen, C.; Wang, J.; and Wu, X. 2024b. Unifying Large Language Models and Knowledge Graphs: A Roadmap. *IEEE Transactions on Knowledge and Data Engineering*.

Robinson, J. D.; Chuang, C.-Y.; Sra, S.; and Jegelka, S. 2021. Contrastive Learning With Hard Negative Samples. In *International Conference on Learning Representations (ICLR)*.

Romero, A.; Ballas, N.; Ebrahimi Kahou, S.; Chassang, A.; Gatta, C.; and Bengio, Y. 2015. FitNets: Hints for Thin Deep Nets. In *International Conference on Learning Representations (ICLR)*.

Samy, A. E.; T. Kefato, Z.; and Girdzijauskas, S. 2023. Graph2Feat: Inductive Link Prediction via Knowledge Distillation. In *ACM Web Conference (WWW)*, 805–812.

Shi, Y.; Huang, Z.; Feng, S.; Zhong, H.; Wang, W.; and Sun, Y. 2021. Masked Label Prediction: Unified Message Passing Model for Semi-Supervised Classification. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 1548–1554.

Taori, R.; Gulrajani, I.; Zhang, T.; Dubois, Y.; Li, X.; Guestrin, C.; Liang, P.; and Hashimoto, T. B. 2023. Alpaca: A Strong, Replicable Instruction-Following Model. *Stanford Center for Research on Foundation Models*, 3(6): 7.

Tian, Y.; Krishnan, D.; and Isola, P. 2020. Contrastive Representation Distillation. In *International Conference on Learning Representations (ICLR)*.

Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, M.-A.; Lacroix, T.; Rozière, B.; Goyal, N.; Hambro, E.; and Azhar, F. 2023. Llama: Open And Efficient Foundation Language Models. *arXiv Preprint arXiv:2302.13971*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit; Gomez, A. N.; Kaiser, L. u.; and Polosukhin, I. 2017. Attention Is All You Need. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30.

Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; and Bengio, Y. 2018. Graph Attention Networks. In *International Conference on Learning Representations (ICLR)*.

Wang, H.; Feng, S.; He, T.; Tan, Z.; Han, X.; and Tsvetkov, Y. 2024. Can Language Models Solve Graph Problems in Natural Language? In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36.

Wei, J.; Bosma, M.; Zhao, V.; Guu, K.; Yu, A. W.; Lester, B.; Du, N.; Dai, A. M.; and Le, Q. V. 2022. Finetuned Language Models Are Zero-Shot Learners. In *International Conference on Learning Representations (ICLR)*.

Wu, N.; and Wang, C. 2022. Gtnet: A Tree-based Deep Graph Learning Architecture. *arXiv preprint arXiv:2204.12802*.

Wu, Q.; Zhao, W.; Yang, C.; Zhang, H.; Nie, F.; Jiang, H.; Bian, Y.; and Yan, J. 2024. Simplifying and Empowering Transformers for Large-Graph Representations. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36.

Wu, T.; Zhao, Z.; Wang, J.; Bai, X.; Wang, L.; Wong, N.; and Yang, Y. 2023. Edge-Free But Structure-Aware: Prototype-Guided Knowledge Distillation From GNNs To MLPs. *arXiv Preprint arXiv:2303.13763*.

Xiao, S.; Liu, Z.; Zhang, P.; Muennighoff, N.; Lian, D.; and Nie, J.-Y. 2024. C-Pack: Packed Resources For General Chinese Embeddings. In *International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*.

Xu, K.; Hu, W.; Leskovec, J.; and Jegelka, S. 2018. How Powerful Are Graph Neural Networks? *arXiv Preprint arXiv:1810.00826*.

Yang, R.; and Shi, J. 2024. Efficient High-Quality Clustering for Large Bipartite Graphs. *Proceedings of the ACM on Management of Data*, 2(1): 1–27.

Yang, Y.; Qiu, J.; Song, M.; Tao, D.; and Wang, X. 2020. Distilling Knowledge From Graph Convolutional Networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 7074–7083.

Yang, Z.; Cohen, W.; and Salakhudinov, R. 2016. Revisiting Semi-Supervised Learning with Graph Embeddings. In *International Conference on Machine Learning (ICML)*, volume 48, 40–48.

Ye, R.; Zhang, C.; Wang, R.; Xu, S.; and Zhang, Y. 2024. Language Is All A Graph Needs. In *Findings of the Association for Computational Linguistics (EACL)*, 1955–1973.

Ying, C.; Cai, T.; Luo, S.; Zheng, S.; Ke, G.; He, D.; Shen, Y.; and Liu, T.-Y. 2021. Do Transformers Really Perform Badly for Graph Representation? In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, 28877–28888.

Zhang, S.; Dong, L.; Li, X.; Zhang, S.; Sun, X.; Wang, S.; Li, J.; Hu, R.; Zhang, T.; Wu, F.; et al. 2023. Instruction Tuning for Large Language Models: A Survey. *arXiv preprint arXiv:2308.10792*.