LLMs as Zero-shot Graph Learners: Alignment of GNN Representations with LLM Token Embeddings

Duo Wang, Yuan Zuo*, Fengzhi Li, Junjie Wu

MIIT Key Lab of Data Intelligence and Management, Beihang University

*Corresponding author.

Research Background: Graph Neural Network

- **Graph Neural Network (GNN)**
  - GNNs leverage the inherent structure of the graph, consisting of nodes and edges, to learn expressive node representations through iterative message propagation and aggregation operations, presented as follows

$$
\begin{aligned}
m_v^{(l)} &= \text{Propagate}^{(l)}(\{h_u^{(l-1)} : u \in \mathcal{N}(v)\}), \\
h_v^{(l)} &= \text{Aggregate}^{(l)}(h_v^{(l-1)}, m_v^{(l)})
\end{aligned}
$$

  - The $\text{Propagate}^{(l)}$ function performs message passing by aggregating information from the neighboring nodes of v at the l-th layer
  - The $\text{Aggregate}^{(l)}$ function then combines the aggregated information with the previous layer's representation of node v to generate the updated representation $h_v^{(l)}$
  - By encoding graph structural information with the learned representations, GNNs can be customized for various downstream graph learning tasks, such as node classification and link prediction

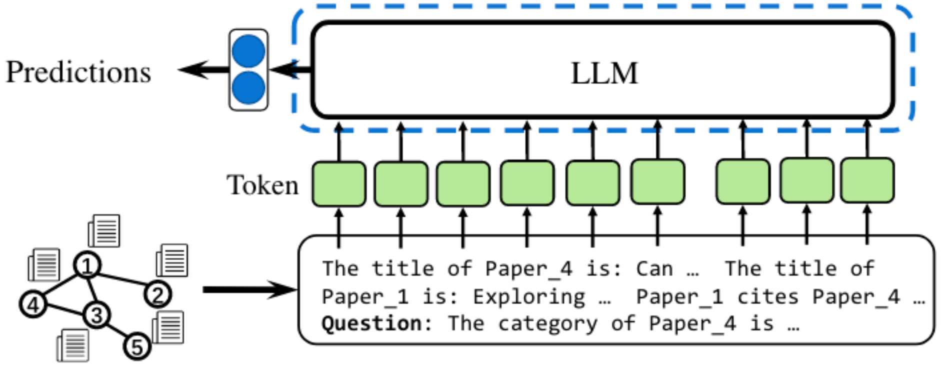
Research Background: Graph Neural Network

- **Graph Neural Network (GNN)**
  - The SOTA model architecture in graph machine learning
  - GNNs can effectively capture and model the complex relationships and dependencies present in graphs

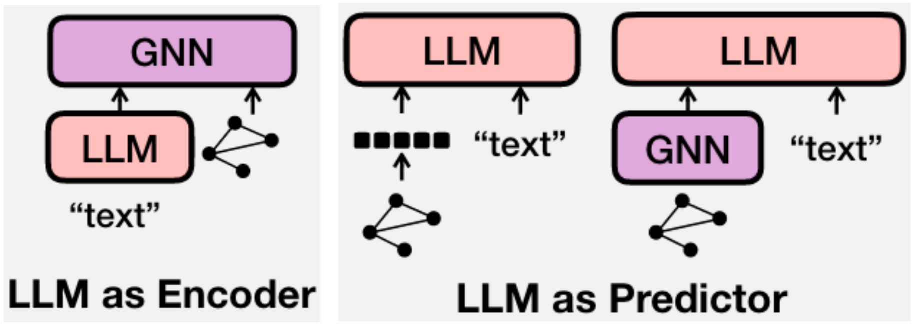
- **Weaknesses of GNN**
  - Limited generalization capabilities when transitioning across different datasets or downstream tasks (Mingxuan Ju et al. 2023)
  - The existing self-supervised learning and graph prompt learning methods often require extensive fine-tuning
  - Combining LLMs and utilizing the generalization capabilities of LLMs to address this issue is a viable approach

Related Work

- **Graph to Text**
  - Represent graph structure information as text input to LLMs
  - Since LLMs cannot understand graph structures, this often leads to suboptimal solutions (Jin Huang et al. 2023)



- **LLM as Encoders**
  - GNNs are the final components and adopt LLM as the initial text encoder
  - Limit the model's transferability since GNNs are ultimately used for prediction
- **LLM as Predictors**
  - Serve LLM as the final component to output representations or predictions
  - The existing methods do not perform well



Problem Definition

- Predict tasks at the node/edge/graph level based on graph structure information and textual information
- Capable of making predictions across datasets and tasks.

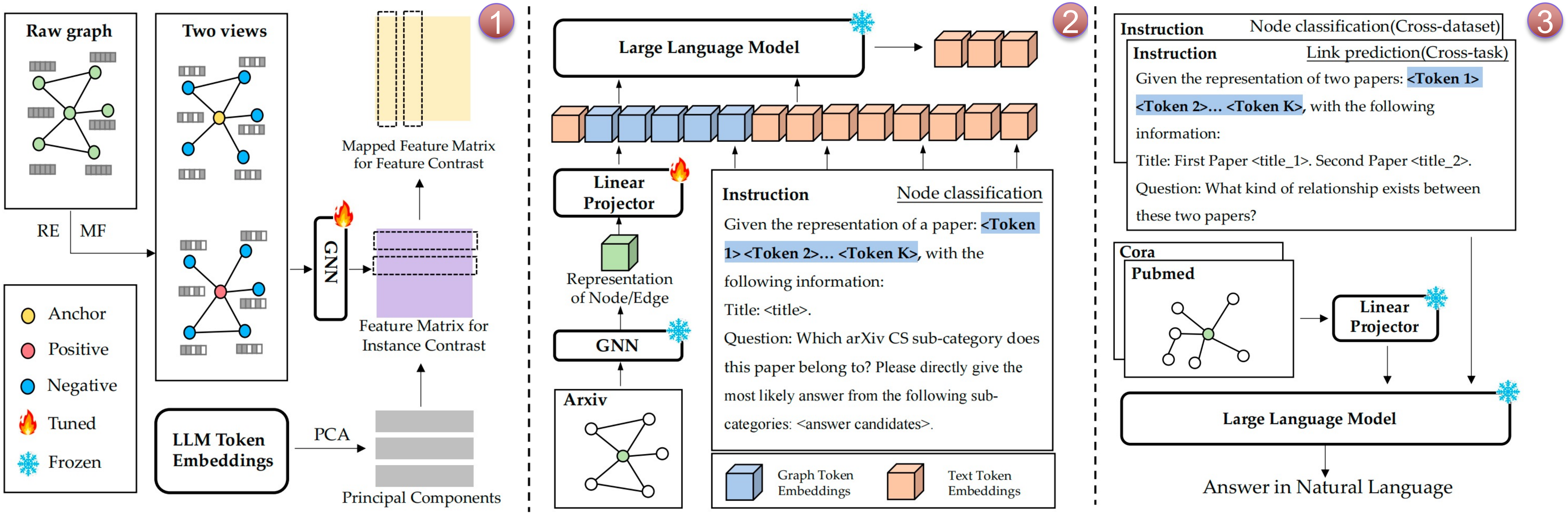
Formally, a graph is denoted as $\mathcal{G} = (\mathcal{V}, \mathcal{E}, A, X)$, where $\mathcal{V}$ indicating the total number of nodes and $\mathcal{E}$ representing the sets of nodes and edges, respectively. The adjacency matrix is denoted as $A \in \mathbb{R}^{N \times N}$. The feature matrix $X \in \mathbb{R}^{N \times F_N}$ contains the attribute or feature information associated with each node, where $x_i \in \mathbb{R}^{F_N}$ is the feature of v_i, and F_N represents the dimensionality of features.

Key Challenges

- **Integration of models from different modalities**
  - There is a gap between the node representation space obtained by the GNN and the token embedding space of the LLM
  - Enabling the LLM to understand node representations is a key challenge

- **Generalization ability on unseen datasets and tasks**
  - Enhancing the model's generalization ability is also a challenge
  - The key lies in training the model to learn how to solve problems, rather than memorizing answers

Model Framework

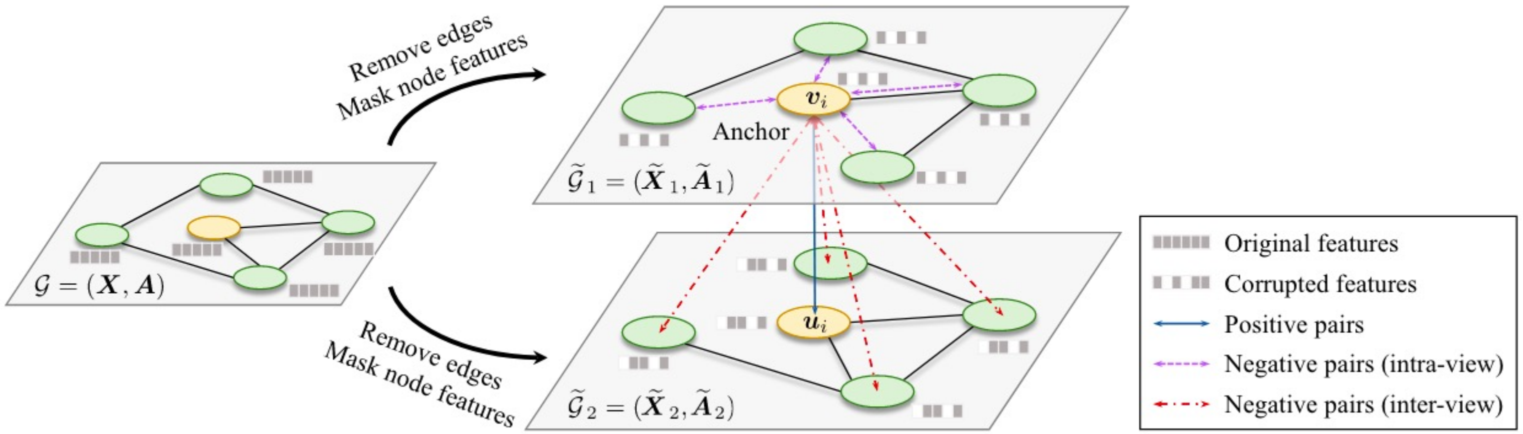


① **Contrastive learning of GNN:** Instance-wise and feature-wise contrastive learning to obtain general representations aligned with LLMs

② **Alignment tuning of projector:** Train a linear projector to map each node embedding into the token embedding

③ Zero-shot tasks: Perform zero-shot tasks on unseen datasets and tasks

Module I: Instance-wise contrastive learning of GNN



- **Removing Edges (RE) and Masking Node Features (MF)**
  - a random masking matrix $\widetilde{\mathbf{R}} \in \{0,1\}^{N \times N}$ to mask the raw adjacency matrix
  - a random mask vector $\widetilde{\boldsymbol{m}} \in \{0,1\}^{F}$ to mask the raw feature matrix

$$\widetilde{\mathbf{A}} = \mathbf{A} \circ \widetilde{\mathbf{R}},$$

$$\widetilde{\mathbf{X}} = [\boldsymbol{x}_1 \circ \widetilde{\boldsymbol{m}}; \boldsymbol{x}_2 \circ \widetilde{\boldsymbol{m}}; \ldots; \boldsymbol{x}_N \circ \widetilde{\boldsymbol{m}}].$$

- **Two views of raw graph**

$$\mathbf{U}_* = f_{\mathrm{GNN}}\left(\widetilde{\mathbf{X}}_*, \widetilde{\mathbf{A}}_*\right) \in \mathbb{R}^{N \times F_U},$$

Module I: Instance-wise contrastive learning of GNN

- **Instance-wise contrastive learning**

$$\ell\left(\boldsymbol{u_i}, \boldsymbol{u_i}'\right) = \log \frac{e^{\theta\left(\boldsymbol{u_i}, \boldsymbol{u_i}'\right)/\tau}}{\underbrace{e^{\theta\left(\boldsymbol{u_i}, \boldsymbol{u_i}'\right)/\tau}}_{\text{the positive pair}} + \underbrace{\sum_{j=1}^{N} 1_{[j \neq i]} e^{\theta\left(\boldsymbol{u_i}, \boldsymbol{u_j}\right)/\tau}}_{\text{intra-view negative pairs}} + \underbrace{\sum_{j=1}^{N} 1_{[j \neq i]} e^{\theta\left(\boldsymbol{u_i}, \boldsymbol{u_j}'\right)/\tau}}_{\text{inter-view negative pairs}}},$$

- **Instance-wise loss**

$$\mathcal{L}_{ins} = \frac{1}{2N} \sum_{i=1}^{N} \left[\ell\left(\boldsymbol{u_i}, \boldsymbol{u_i}'\right) + \ell\left(\boldsymbol{u_i}', \boldsymbol{u_i}\right)\right].$$

Module II : Feature-wise contrastive learning of GNN

- **Feature-wise loss**
  - For the feature matrix U_*, denote the columns in different views as $m_i \in \mathrm{U}_1^{\mathrm{T}}$ and $n_i \in \mathrm{U}_2^{\mathrm{T}}$

$$\mathcal{L}_{fea} = \frac{1}{F_U} \sum_{i=1}^{F_U} \log \frac{e^{\theta(\boldsymbol{m}_i, \boldsymbol{n}_i)/\tau}}{\sum_{j=1}^{F_U} \left[e^{\theta(\boldsymbol{m}_i, \boldsymbol{m}_j)/\tau} + e^{\theta(\boldsymbol{m}_i, \boldsymbol{n}_j)/\tau} \right]}.$$

- **Principal components projection**
  - Using the principal components of the token embeddings of LLMs as coordinate axes
  - This approach ensures that the representations of similar instances are closely aligned in the textual embedding space

Final loss: $\mathcal{L} = \frac{1}{2}(\mathcal{L}_{ins} + \mathcal{L}_{fea})$

Module III: Linear projector

- **Multiple graph tokens**
  - Train a linear projector to map each node embedding into **multiple** token embeddings
  - Due to the complex information in graph structures, which cannot be captured by a single graph token, we aim to adequately convey the graph information through multiple graph tokens

$$\mathbf{H}_{token} = f_{\text{Linear}}\left(\boldsymbol{u_i}\right)$$

  - where $u_i \in \mathrm{U},\ \mathrm{H}_{token} \in \ \mathbb{R}^{K \times F_L}$

Module IV: Unified instructions

- **Graph information provision**

> Given the representation of a paper/two papers/a paper set: ⟨graph⟩, with the following information:
> Title: First Paper: {title1} ...,

- **Task description**
  - To achieve cross-dataset capability, the instruction is designed to include not only the task description itself but also the set of alternative answers

> Question: Which arXiv CS sub-category does this paper belong to? Please directly give the most likely answer from the following sub-categories: {answer candidates}

- **Fixed number of tokens**
  - To achieve cross-task capability, we use a readout operation to obtain representations at the edge/graph level, thus the number of graph token embeddings is fixed regardless of the task type

Training and evaluation strategy

- **The first stage**
  - Train the GNN model with the loss function

$$\mathcal{L} = \frac{1}{2}(\mathcal{L}_{ins} + \mathcal{L}_{fea})$$

- **The second stage**
  - Fix the parameters of the GNN model and the LLM, and train the linear projector

- **Evaluation**
  - Test on unseen datasets and tasks

Evaluations

- **Data**
  - Citation datasets: Arxiv, Pubmed, Cora (with more categories)
  - E- commerce datasets (Hao Yan et al. 2023): Computer, Photo, Children, History, Sports
  - Source datasets: Arxiv & Computer

- **Baseline methods**
  - Non-graph neural network approach: **MLP**
  - Supervised methods: **GCN** (Thomas et al. 2017), **GraphSAGE** (Will 2017), **GAT** (Petar et al. 2018)
  - Self-supervised methods: **DGI** (Petar et al. 2019)
  - Graph knowledge distillation frameworks: **GKD** (Chenxiao Yang et al. 2022), **GLNN** (Shichang Zhang et al. 2022)
  - Graph transformer networks: **NodeFormer** (Qitian Wu et al. 2022), **DIFFormer** (Qitian Wu et al. 2023)
  - Large language models: **Vicuna-7B-v1.5**
  - Latest models equipped with transfer and zero-shot capabilities: **OFA** (Liu Hao et al. 2024), **GraphGPT** (Jiabin Tang et al. 2023), **LLaGA** (Runjin Chen et al. 2024)

Evaluations

- **Tasks**
  - Cross-dataset: Pretraining on source datasets and evaluate on target datasets
  - Cross-task: Pretraining on node-level task and evaluate on edge-level task

- **Implementation details**
  - Data split: Follow the methodology outlined in GraphGPT (Jiabin Tang et al. 2023) and TAG benchmark (Hao Yan et al. 2023)
  - Evaluation metrics: Accuracy node classification and AUC for link prediction

Cross-dataset zero-shot ability

Table 1: Zero-shot accuracy on citation and e-commerce datasets (**bold** highlights the best result across all methods, while <u>underline</u> highlights the second-best results)

Model type	Model	Citation		E-commerce			
		Pubmed	Cora	Children	History	Photo	Sports
	MLP	0.323±0.027	0.021±0.006	0.029±0.037	0.080±0.041	0.110±0.070	0.042±0.021
GNN as predictor	GCN	0.288±0.092	0.017±0.004	0.030±0.018	0.063±0.042	0.103±0.047	0.042±0.025
	GraphSAGE	0.316±0.058	0.014±0.007	0.008±0.007	0.195±0.206	0.056±0.055	0.051±0.015
	GAT	0.343±0.064	0.016±0.004	0.086±0.084	0.172±0.098	0.050±0.027	0.142±0.138
	DGI	0.329±0.103	0.026±0.009	0.082±0.035	0.218±0.168	0.224±0.127	0.049±0.017
	GKD	0.399±0.033	0.042±0.008	0.202±0.064	0.339±0.138	0.166±0.086	0.208±0.077
	GLNN	0.390±0.011	0.031±0.006	0.187±0.012	0.283±0.021	<u>0.403±0.019</u>	0.317±0.048
	NodeFormer	0.308±0.093	0.016±0.007	0.048±0.028	0.168±0.127	0.073±0.015	0.165±0.057
	DIFFormer	0.361±0.071	0.029±0.014	0.129±0.030	0.275±0.171	0.321±0.055	0.306±0.131
	OFA	0.314±0.059	0.130±0.019	0.064±0.086	0.052±0.049	0.340±0.026	0.101±0.071
LLM as predictor	Vicuna-7B-v1.5	0.719±0.010	0.156±0.001	<u>0.270±0.001</u>	<u>0.363±0.001</u>	0.378±0.004	<u>0.370±0.001</u>
	GraphGPT-std	0.701	0.126	-	-	-	-
	GraphGPT-cot	0.521	<u>0.181</u>	-	-	-	-
	LLaGA	<u>0.793±0.036</u>	0.168±0.032	0.199±0.007	0.146±0.067	0.276±0.069	0.352±0.033
	TEA-GLM	**0.848±0.010**	**0.202±0.014**	**0.271±0.010**	**0.528±0.058**	**0.497±0.027**	**0.404±0.010**

- For models that use GNN as a predictor, we utilize the GNN backbone trained on the source dataset along with a classifier trained with target data
- For Vicuna, we use the version without fine-tuning, relying solely on text information for prediction

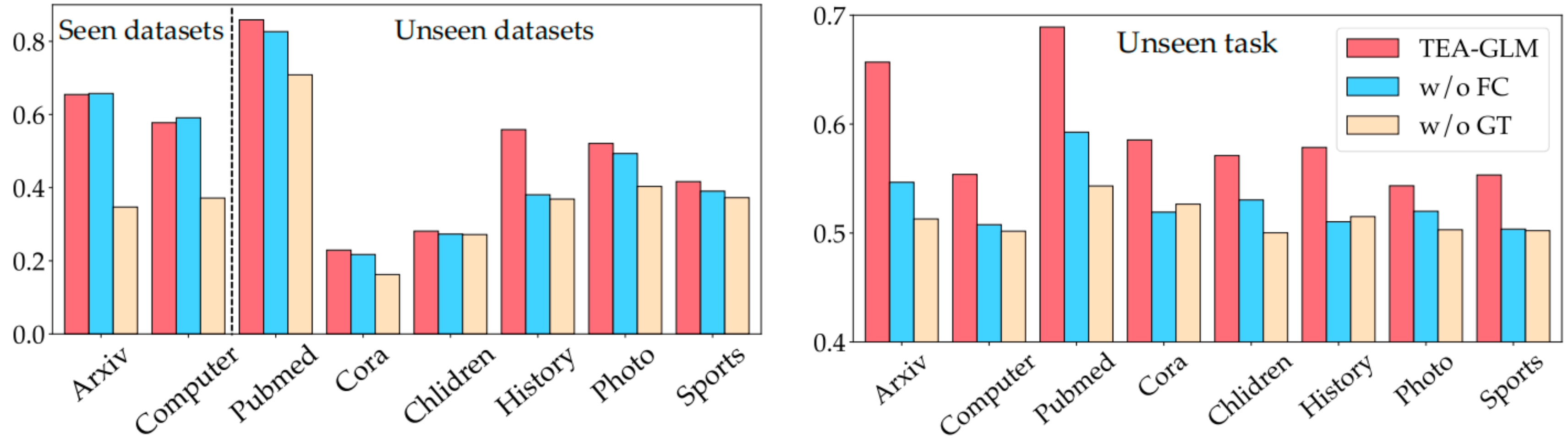
Cross-task zero-shot ability

Table 2: AUC of link prediction (Cross-task)

Model	Citation			E-commerce				
	Arxiv	Pubmed	Cora	Children	History	Computer	Photo	Sports
OFA	0.469	0.481	0.492	0.484	0.431	0.461	0.459	0.517
Vicuna-7B-v1.5	0.513	0.543	0.527	<u>0.500</u>	<u>0.515</u>	<u>0.502</u>	<u>0.501</u>	0.502
GraphGPT-std	<u>0.649</u>	0.501	0.520	-	-	-	-	-
LLaGA	0.570	<u>0.569</u>	<u>0.537</u>	0.422	0.449	0.479	0.478	**0.597**
TEA-GLM	**0.657**	**0.689**	**0.586**	**0.571**	**0.579**	**0.554**	**0.545**	<u>0.553</u>

- The proposed method significantly outperforms the baseline methods

Ablation Study



(a) Accuracy of node classification

(b) AUC of link prediction

Figure 1: Ablation study results ("Seen datasets" are used to train the GNN and linear projector, while "unseen datasets" are not. "Unseen task" means the model wasn't trained for link prediction.)

- "w/o FC" means that we pretrain the GNN without feature-wise contrastive learning, while "w/o GT" means predicting without graph token embeddings
- Graph tokens provide graph information to LLMs, aiding in making more accurate predictions
- FC further aligns node representations with LLMs, resulting in more general representations that are easier for LLMs to understand

Evaluation of Legality rate

- **Legality rate** (Mengmei Zhang et al. 2024)
  – Since training on specific datasets or tasks can lead LLMs to produce incorrect answers, it is crucial to evaluate the training's impact on their performance
  – The proportion of valid answers produced by the model

Table 3: Legality rate of LLM-backbone model (The worst results are marked in gray)

Dataset	Arxiv	Computer	Pubmed	Cora	Children	History	Photo	Sports
Model	Legality rate(%)							
Vicuna-7B-v1.5	99.3	96.7	100.0	95.8	99.2	98.9	94.1	99.6
LLaGA	100.0	100.0	98.9	79.9	93.1	92.4	77.8	94.3
TEA-GLM	100.0	100.0	100.0	92.6	97.0	99.6	99.2	98.5

- Compared to existing methods, our training process has a lesser impact on LLMs, , attributable to the alignment of node representations with LLMs

Parameter Sensitivity—Number of Graph Tokens

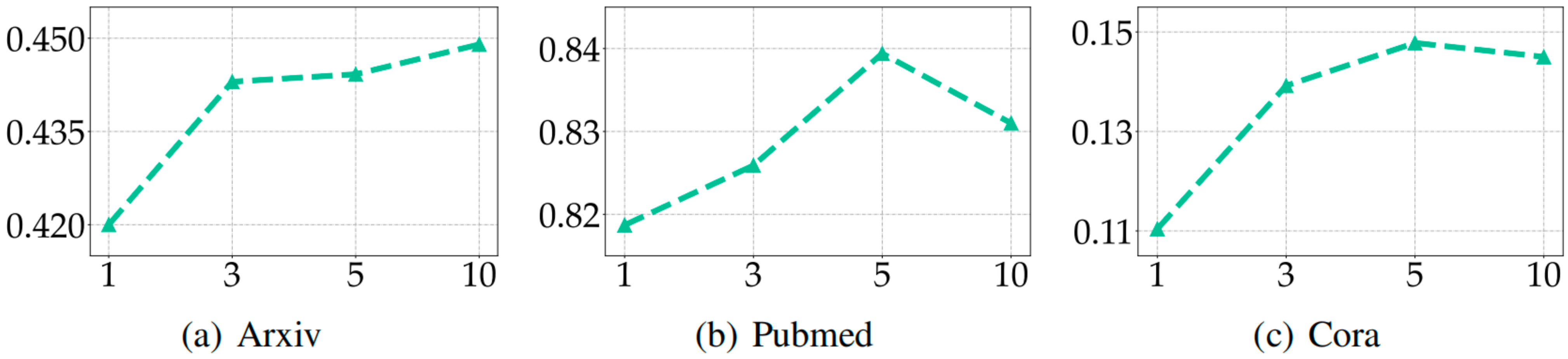


Figure 2: Impact of number of graph token embeddings

- Supervised learning: Enhancing the model's performance can be achieved by increasing the quantity of graph token embeddings
- Zero-shot: Only a minimal number of graph tokens is required to achieve satisfactory performance, indicating that the number of parameters in our model is significantly less than concurrent works

Parameter Sensitivity—Number of Principal Components

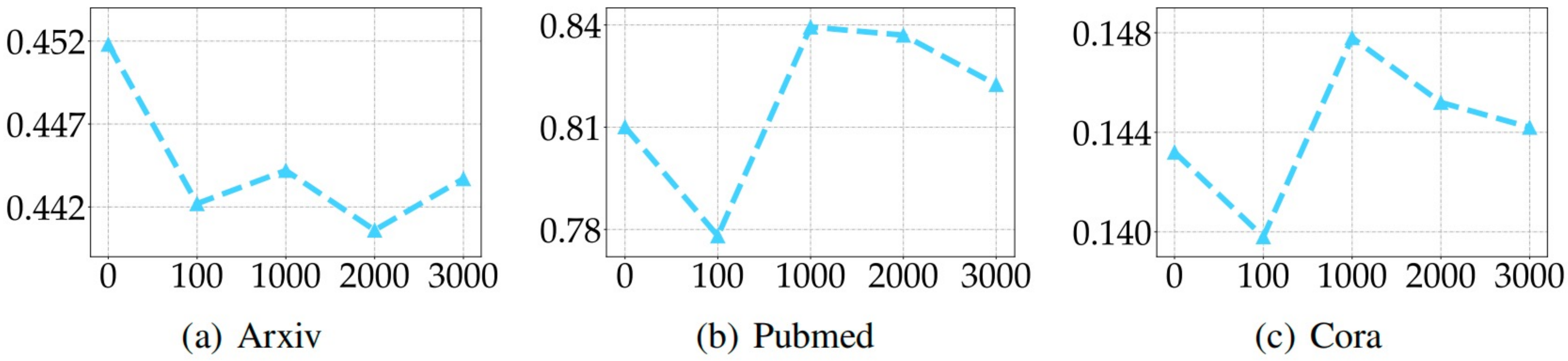


Figure 3: Impact of number of principal components

- In supervised learning scenarios, omitting contrastive learning with principal components can lead to a slight increase in accuracy. However, this makes the model overfitting on training datasets
- When the number of principal components is too small, it adversely affects the model's learning capability. Remarkably, when P = 1000, the model demonstrates satisfactory performance. At this level, the principal components capture 50% of the variance of LLM's token embeddings

Concluding Remarks

- **Technical Contributions**
  - We introduce a novel framework that aligns GNN representations with LLM token embeddings, enabling cross-dataset and cross-task zero-shot learning for graph machine learning
  - We propose a linear projector that maps graph representations into a fixed number of graph token embeddings. These embeddings are incorporated into a unified instruction designed for various graph tasks at different levels, enhancing the model's generalization capabilities
  - Our extensive experiments demonstrate that our framework significantly outperforms state-of-the-art methods on unseen datasets and tasks

THANK YOU FOR YOUR TIME!

Duo Wang, Yuan Zuo*, Fengzhi Li, Junjie Wu
MIIT Key Lab of Data Intelligence and Management, Beihang University