GraphICL: Unlocking Graph Learning Potential in LLMs through Structured Prompt Design

Anonymous ACL submission

Abstract

The growing importance of textual and relational systems has driven interest in enhancing large language models (LLMs) for graph-structured data, particularly Text-Attributed Graphs (TAGs), where samples are represented by textual descriptions interconnected by edges. While research has largely focused on developing specialized graph LLMs through task-specific instruction tuning, *a comprehensive benchmark for evaluating LLMs solely through prompt design* remains surprisingly absent. Without such a carefully crafted evaluation benchmark, most if not all, tailored graph LLMs are compared against general LLMs using simplistic queries (e.g., zero-shot reasoning with LLaMA), which can potentially camouflage many advantages as well as unexpected predicaments of them. To achieve more general evaluations and unveil the true potential of LLMs for graph tasks, we introduce **Graph In-context Learning (GraphICL) Benchmark**, a comprehensive benchmark comprising novel prompt templates designed to capture graph structure and handle limited label knowledge. Our systematic evaluation shows that general-purpose LLMs equipped with our GraphICL outperform state-of-the-art specialized graph LLMs and graph neural network models in resource-constrained settings and out-of-domain tasks. These findings highlight the significant potential of prompt engineering to enhance LLM performance on graph learning tasks without training and offer a strong baseline for advancing research in graph LLMs.

1 Introduction

Text-Attributed Graphs (TAGs), which integrate textual descriptions as node attributes with relational edges, form the foundation for understanding modern complex systems and relationships (Kipf and Welling, 2016a; Hamilton et al., 2017). Deep learning-based graph reasoning (GR) approaches, exemplified by graph neural networks (GNNs) (Li et al., 2021; Sun et al., 2021; Zhou et al., 2020; Wu et al., 2022; Reiser et al., 2022), have achieved remarkable success in many TAG-related reasoning tasks, such as node classification (Fan et al., 2019; Wu et al., 2020) and link prediction (Wu et al., 2022; Reiser et al., 2022; Tan et al., 2023; Kipf and Welling, 2016b).

However, most GNN-based approaches face two major hurdles: ❶ **Limited generalization across different graphs, particularly in cross-domain scenarios**. GNN models are typically tailored to specific graph structures they were originally trained on, and when applied to novel or cross-domain reasoning tasks, they exhibit a marked decline in performance (Zhao et al., 2023; Xu et al., 2024). Resolving this often requires fine-tuning or full retraining, resulting in substantial computational overhead and deployment efforts. ❷ **Performance depends heavily on labeled training graphs**. While GNNs perform well in supervised settings, their efficacy drastically diminishes in limited-label scenarios. Although graph few-shot learning (Garcia and Bruna, 2017) has been introduced to mitigate this issue, it still requires a significant number of related learning tasks to adequately train the model for transfer to unseen tasks.

To address these challenges, recent research has shifted from GNNs to graph LLMs (Tang et al., 2023; Chen et al., 2024b; Zhang et al., 2024; Liu et al., 2024b; He and Hooi, 2024), most of them leverage LLMs' strong generalization capabilities for graph-related tasks through in-context learning (ICL) (Dong et al., 2022). The key challenge for graph LLMs is incorporating graph structures into queries. Current approaches tackle this by either heuristically converting graphs into node sequences (Chen et al., 2024b; Ye et al., 2023) or embedding graph structures into hidden tokens via an auxiliary GNN (Tang et al., 2023; Zhang et al., 2024; Liu et al., 2024b; He and Hooi, 2024), which are then integrated into query templates for

graph reasoning. By fine-tuning additional neural components or the general LLM backbones using graph-specific instruction tuning, these specialized methods have demonstrated superior zero-shot ICL capabilities compared to standard GNN studies.

Despite the promising advances in specialized graph LLMs, their evaluation often relies on overly simplistic LLM baselines (Tang et al., 2023; Chen et al., 2024b), such as zero-shot reasoning with models like LLaMA or ChatGPT. Moreover, these models are typically assessed in in-domain scenarios and struggle to fully utilize limited labeled data (i.e., few-shot ICL), a capability that general-purpose LLM can readily support through effective prompt design. In the absence of a well-constructed evaluation benchmark, comparisons between specialized graph LLMs and general LLMs remain underexplored and poorly designed, potentially camouflaging many advantages as well unexpected predicaments of graph LLMs. This raises important questions: *Can general-purpose LLMs be effectively adapted to tackle real-world graph reasoning tasks (e.g., node classification and link prediction) solely through in-context learning? Have we truly made progress in the development of graph LLMs?*

To address these questions, we propose **GraphICL**: Benchmarking Large Language Models for **Graph** Reasoning via **I**n-**C**ontext **L**earning. In our framework, GraphICL refers to the design of task-specific prompts following a unified GraphICL template across diverse graph reasoning tasks. GraphICL facilitates graph reasoning in LLMs by leveraging four core components: task description, anchor node text, structure-aware information, and labeled demonstrations. By incorporating anchor nodes and their k-hop neighbors, we enable zero-shot graph reasoning, utilizing the inherent relationships between proximate nodes. Through strategical selection of neighbors and demonstrations, such as the top M most similar or influential nodes, we optimize few-shot reasoning, releasing the potential of LLMs. GraphICL pushes the boundaries of LLMs' capabilities in graph tasks, enabling performance that was previously unattainable. Our key contributions are summarized as follows:

- **Novel Research Problem.** We investigate whether better graph reasoning (GR) results can be achieved by simply prompting LLMs through GraphICL, without additional training, and whether this approach can outperform both supervised GNNs and specialized Graph LLMs in both in- and cross-domain scenarios.

- **A Comprehensive Prompt Benchmark for LLM in Graph Reasoning.** Previous comparisons between general LLMs and specialized graph models have been biased by underdeveloped prompts, which fail to harness the full potential of LLMs. We propose GraphICL, a comprehensive prompt set that encompasses graph structure, labeled demonstration, and diverse evaluation tasks.

- **Systematic Evaluation.** We conducted extensive experiments on 9 datasets, encompassing both in-domain and cross-domain scenarios, and benchmarked our approach against state-of-the-art graph LLMs as well as traditional supervised GNN models. Additionally, we performed comprehensive ablation studies to assess the impact of various prompt configurations within the GraphICL framework.

- **Promising Observations.** Our extensive evaluation yielded several valuable insights that can inform the future application of LLMs in graph reasoning, particularly through in-context learning. These findings also establish a solid foundation for advancing research in graph LLMs.

2 Related Work

Specialized Graph LLMs. Building on the success of large language models (LLMs), the application of LLMs to graph reasoning tasks has gained considerable attention. The core idea is to incorporate graph structures into queries and then instruction-tune the LLMs using graph-related tasks. Based on graph transformation strategies, existing efforts can be broadly categorized into two approaches: heuristic and learnable. The heuristic approach (Chen et al., 2024b; Liu et al., 2024a; Ye et al., 2023) converts graphs into node sequences using manually designed transformation rules (Ye et al., 2023) or random walks (Chen et al., 2024b). In contrast, the learnable approach (Tang et al., 2023; Zhang et al., 2024; Liu et al., 2024b; He and Hooi, 2024) encodes graph structures into hidden sequences through additional GNN encoders, which are either pre-trained (He and Hooi, 2024; Fang et al., 2024a; Tang et al., 2023) or jointly fine-tuned (Zhang et al., 2024; Liu et al., 2024b) with the LLM backbone during instruction tuning. While these specialized graph LLMs inherit the zero-shot in-context learning (ICL) capabilities of general LLMs, they struggle to effectively leverage

few-shot demonstrations for performing few-shot ICL on graphs.

General-purpose LLM for Graph Reasoning. In parallel, another line of research represents graph structures using natural language descriptions, combining them with task-specific templates to query general-purpose LLMs. Notable works such as (Huang et al., 2023; Zhao et al., 2023; Guo et al., 2023; Chen et al., 2024c; Li et al., 2024; Fang et al., 2024b) have advanced this area, primarily focusing on using LLMs for graph augmentations (He et al., 2023a; Chen et al., 2024c; Fang et al., 2024b). While some efforts (Huang et al., 2023) have explored graph structure's role in LLM inference through both zero-shot and few-shot ICL, they remain limited in terms of prompt template diversity, neighborhood and labeled demonstration selection, evaluation scenarios, and the breadth of GR tasks.

In contrast, we introduce a comprehensive prompt template design for graph reasoning tasks, where the prompts in (Huang et al., 2023; Li et al., 2024) can be seen as a subset of our approach. More importantly, we benchmark the performance of specialized graph LLMs and general-purpose LLMs equipped with our prompt suite, offering a timely and fair comparison of recent specialized graph LLM studies while providing insights into their strengths and weaknesses relative to general LLMs utilizing prompt design.

3 Problem Statement

Given a Text-Attributed Graph $\mathcal{G} = (\mathcal{V}, \mathcal{A}, \mathcal{T})$, where $\mathcal{V}$ represents nodes, $\mathcal{A}$ is the adjacency matrix, and $\mathcal{T}$ contains the node texts, along with a LLM $f(\cdot)$, this paper aims to leverage Graph In-Context Learning (GICL) to generate relevant GICL-Prompts P, which are the outputs of $GP(\cdot)$, as inputs for the LLM $f(\cdot)$ to solve two classic graph reasoning tasks: node classification (NC) and link prediction (LP).

Node Classification via GICL. For node classification, we can use two different GICL methods to predict the label y_i of node $\mathcal{V}_i$ in Graph $\mathcal{G}$.

i) ***NC-Zero-shot:*** Use only the anchor node text $\mathcal{T}_i$, or include neighboring node texts $\mathcal{T}'$, as the main content to generate the GICL-Prompt, where $P = GP(\mathcal{T}_i, \mathcal{T}')$. This prompt is then fed into the LLM to obtain the prediction, $y_p = f(P)$.

ii) ***NC-Few-shot:*** Building upon the zero-shot template, we further incorporate neighboring nodes' texts $\mathcal{T}'$ and their labels $\mathcal{Y}'$, or additionally include demonstration texts $\mathcal{T}''$ and labels $\mathcal{Y}''$, to form a more informative GICL-Prompt, where $P = GP(\mathcal{T}_i, \mathcal{T}', \mathcal{Y}', \mathcal{T}'', \mathcal{Y}'')$. This enriched prompt is then input into the LLM to generate the final prediction, $y_p = f(P)$.

Link Prediction via GICL. For link prediction between nodes $\mathcal{V}_m$ and $\mathcal{V}_n$, we can also utilize these two approaches:

i) ***LP-Zero-shot:*** We begin by using the textual information of the two nodes, $\mathcal{T}_m$ and $\mathcal{T}_n$, and optionally incorporate neighboring node texts $\mathcal{T}'$ to construct a GICL-Prompt, $P = GP(\mathcal{T}_m, \mathcal{T}_n, \mathcal{T}')$. This prompt is then passed into the LLM to predict the existence of a link, $y_p = f(P)$.

ii) ***LP-Few-shot:*** To further improve performance, we introduce demonstration texts $\mathcal{T}''$ and corresponding link relationships to enrich the Prompt, $P = GP(\mathcal{T}_m, \mathcal{T}_n, \mathcal{T}', \mathcal{T}'')$. This more comprehensive prompt is then used by the LLM to generate a refined link prediction, $y_p = f(P)$.

4 Prompt Design Driven by Graph In-Context Learning

In this section, we will explain how each type of graph in-context learning method is implemented within our framework. Our GraphICL prompt template consists of 4 fundamental components: *anchor node text*, *task description*, *structure-aware information*, and *demonstrations*, as shown in Figure 1. By combining these 4 components, we can generate 55 different prompt templates. These templates are applied to 2 classic graph reasoning tasks on 9 datasets. By comparing with multiple models, we demonstrate the significant boost our GICL template provides to various of LLMs. Section 4.1 and 4.2 explain the generation and function of each component of the prompt respectively. Section 4.3 shows how different modules of the prompt are combined to form the final input for the LLMs.

4.1 Basic Content

The basic content primarily conveys the information specific to the anchor node, ensuring that the LLM comprehends the graph reasoning task it is expected to execute. It constitutes a critical component of the general prompt and serves as the foundational text in GraphICL.

Text of Anchor Nodes. The text associated with the anchor node can vary, such as the title and abstract of a paper (Shibata et al., 2012) or the description of a product (Hasan and Zaki, 2011). In

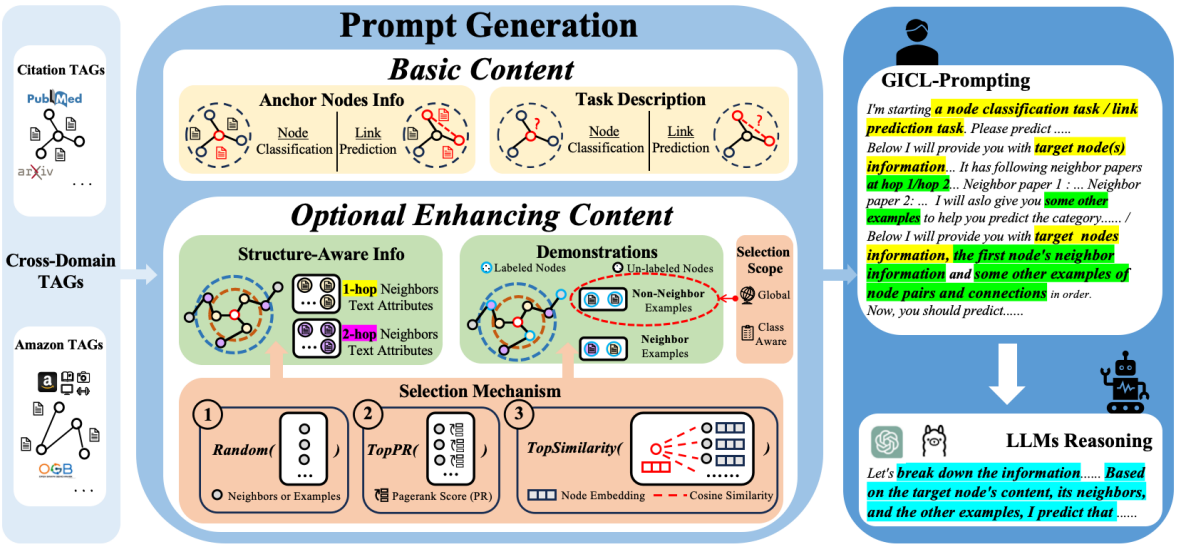

Figure 1: The overall framework of our GraphICL. We implement various graph in-context learning templates by combining basic content with optional enhancing content. These templates are then input as prompts into large language models to obtain relevant prediction results.

the context of link prediction, however, the anchor nodes refer to both the source and target nodes of the predicted edge. The corresponding text in this case is the concatenated text of these two nodes.

Task Description. For different graph reasoning tasks, it is crucial to explicitly define the task objectives for the LLMs. This guiding piece of text is referred to as the task description. In node classification tasks, for instance, LLMs may not inherently recognize the specific names of categories within the dataset. Therefore, we explicitly provide the names of all labels within the task description. Typically, the task description serves as the system prompt (Giray, 2023) for LLMs.

4.2 Optional Enhancing Content

While the basic content provides the essential information needed for LLMs to perform fundamental graph zero-shot learning, it alone is insufficient. To enhance the LLM's ability to reason effectively, additional structural information and other relevant data must be integrated. This supplementary layer of information, known as **Enhancing Content**, serves to deepen the LLMs' understanding and reasoning capabilities.

Structure-Aware Information. Graph structures exhibit complex dependencies, prompting GNNs to employ message passing for gathering and updating node information from neighbors, enriching node representations (Zhou et al., 2020). In graph in-context learning, we simulate this by providing textual information from an anchor node's neighbors for a well-established LLM, effectively enabling message passing at the textual level. We focus on 1-hop neighbors for capturing immediate, direct influences on the target node, representing short-term dependencies, and 2-hop neighbors due to their importance in graph reasoning, as GNNs typically utilize two layers (Chen et al., 2024c). By incorporating 1-hop or 2-hop neighbors' text, our framework enriches structure-aware information and enhances the flexibility of prompt design.

The strategy for selecting neighbors is also crucial, as there is typically no one-size-fits-all approach that achieves optimal results across all graph reasoning tasks. Below, we will introduce three selection strategies employed in our work:

(1) **Random Selection:** Randomly selecting M nodes from the k-hop neighbors of the anchor node, treating each neighbor as equally contributing to the reasoning process.

(2) **Similarity-based Selection:** Calculating cosine similarity between the anchor node and k-hop neighbors, selecting the Top M most similar neighbors, prioritizing those with higher textual relevance to the anchor node.

(3) **PageRank-based Selection:** Computing PageRank scores (Page et al., 1999) for each k-hop neighbor, selecting the Top M based on their importance within the graph structure.

Few-shot Demonstrations. Demonstrations are a crucial component of few-shot learning for LLMs, with their design tailored to specific downstream tasks. These demonstrations are intended to aid LLMs in gaining a deeper understanding of the tasks and enhancing their reasoning abilities

(Brown et al., 2020). For tasks such as node classification and link prediction, additional text from other nodes, along with their corresponding labels, must be provided to facilitate the LLMs' inference.

When selecting demonstrations, we employ the same three strategies used for neighbor selection: random selection, similarity-based selection, and PageRank-based selection. The selection scope for demonstrations can be either "Global", where M demonstrations are chosen from the training set using these selection methods, or "Class-Aware", where one demonstration is provided for each class label. In the latter case, the selection method for each demonstration of each label follows one of the aforementioned three strategies.

4.3 Graph In-Context Learning Prompting

This section discusses how to integrate the four components of the graph in-context learning prompt for different graph reasoning tasks. Examples are provided in Figure 2 for further illustration.

NC-Graph Zero-shot Prompting: The zero-shot prompt includes the Basic Content, which consists of the anchor node's text and a description of the classification task. Structure-aware information can be optionally added as enhanced content but without including labels of k-hop neighbors.

NC-Graph Few-shot Prompting: Similar to zero-shot, the few-shot prompt also includes Basic Content. Additionally, it provides labeled demonstrations as enhanced content, including neighbor information with labels. Alternatively, labeled neighbor information can also be used as a demonstration for few-shot learning, where structure-aware information is omitted to avoid redundancy.

LP-Graph Zero-shot Prompting: Providing the textual information of both the start and end nodes of the target relationship, with the option to include neighbor information for one of the nodes.

LP-Graph Few-shot Prompting: Building on zero-shot template, it adds connection relationships between node pairs from the training set, along with their textual information. The selection of these relationships follows the three methods outlined in section 4.2. Here, the PageRank score of an edge is defined as the average PageRank of its two end nodes, while the edge embedding is computed as the average of their embeddings. Unlike in node classification, the relationships between a node and its neighbors cannot be directly used as demonstrations, as they are implicitly encoded within the structure-aware information.

5 Experiments

In this section, we present a rigorous evaluation of specialized graph LLMs and general-purpose LLMs equipped with our GraphICL. The experiments are divided into two key parts: a comparative analysis of GraphICL against state-of-the-art graph LLMs across various scenarios, and an exploration of how different GraphICL configurations impact the performance of general-purpose LLMs.

5.1 Experiment Configurations

Datasets. We conducted experiments on two major types of datasets: Citation Networks and Amazon Review Datasets. The Citation Networks include PubMed (Sen et al., 2008), Cora (McCallum et al., 2000), and OGB-Arxiv (Hu et al., 2020), while the Amazon review datasets include OGB-Products (Hu et al., 2020), Amazon-Photo, Amazon-History, Amazon-Computers, Amazon-Sports-Fitness and Amazon-Children-Book (Shchur et al., 2018). For specific data splits, please refer to Appendix A.3.1.

Large Language Models. We utilized a total of three language models for testing: LLaMA2-13B-Chat[1], LLaMA3-70B-Instruct[2], and GPT-4o[3]. Due to budget constraints, we did not conduct comprehensive experiments with GPT-4o.

Baselines. In our performance evaluation, we take into account a range of state-of-the-art methods for a thorough assessment. (i) The first category consists of MLP, which utilizes a Multi-layer Perceptron for prediction. (ii) The second category includes prominent GNN encoders, such as GraphSAGE (Li et al., 2021), GCN (Kipf and Welling, 2016a), RevGAT (Li et al., 2021). (iii) The third category encompasses influential specialized Graph LLMs, including LLaGA (ND, HO) (Chen et al., 2024a), GraphGPT (Tang et al., 2023), GraphTranslator (Zhang et al., 2024), and Graph-Prompter (Liu et al., 2024b). (iv) The last category consists of pure zero-shot LLMs, which can also be viewed as methods that input basic content into LLMs for reasoning. In our experiments, all the settings used for GraphICL are explained in detail in the Appendix A.2.

5.2 Multi-scenario Graph Reasoning Testing

We begin by analyzing the node classification results, focusing on two scenarios: **in-domain** and

[1]https://llama.meta.com/llama2/
[2]https://llama.meta.com/llama3/
[3]https://platform.openai.com/docs/models/gpt-4o

Table 1: In-domain node classification results: For MLP, GCN (Kipf and Welling, 2016a), RevGAT (Li et al., 2021), and SAGE (Sun et al., 2021), we uniformly use BERT embeddings (Devlin et al., 2018). Acc(%) is used as the evaluation metric, and we calculate the relative difference between the best results of our method and others. For each LLaMA model in GraphICL, S1 and S2 denote the first- and second-best GraphICL prompts. "NA" indicates that the result is unavailable. For further explanation and settings, please refer to the Appendix A.3.4.

Method	Computers	Sports	PubMed	Cora	Arxiv	Products
Semi-Supervised In-Domain Results						
MLP	44.56 (+96.07%)	58.74 (+55.58%)	59.38 (+56.92%)	47.23 (+76.96%)	37.10 (+98.60%)	65.36 (+24.66%)
GCN	59.12 (+47.78%)	70.24 (+30.11%)	74.25 (+25.49%)	**68.82 (+21.45%)**	**55.27 (+33.31%)**	74.47 (+9.41%)
SAGE	58.52 (+49.30%)	69.53 (+31.44%)	64.66 (+44.11%)	64.58 (+29.42%)	54.05 (+36.32%)	72.35 (+12.62%)
RevGAT	55.48 (+57.48%)	64.63 (+41.40%)	64.10 (+45.37%)	65.31 (+27.97%)	48.86 (+50.80%)	71.45 (+14.04%)
LLaGA-ND	49.48 (+76.58%)	52.19 (+75.11%)	39.96 (+133.18%)	48.52 (+72.26%)	54.26 (+35.79%)	73.32 (+11.13%)
LLaGA-HO	55.68 (+56.91%)	63.81 (+43.22%)	40.37 (+130.81%)	40.96 (+104.05%)	53.02 (+38.97%)	72.76 (+11.98%)
GraphGPT	NA	NA	NA	NA	NA	NA
GraphPrompter	**62.46 (+39.88%)**	**80.92 (+12.94%)**	**88.11 (+5.75%)**	51.11 (+63.53%)	54.12 (+36.14%)	**76.34 (+6.73%)**
GraphTranslator	38.95 (+124.31%)	22.88 (+299.43%)	60.46 (+54.12%)	35.59 (+134.84%)	28.48 (+158.71%)	41.32 (+97.19%)
Supervised In-Domain Results						
MLP	61.74 (+41.51%)	85.06 (+7.44%)	82.55 (+12.88%)	63.12 (+32.41%)	69.20 (+6.47%)	67.56 (+20.60%)
GCN	74.35 (+17.51%)	88.14 (+3.69%)	86.43 (+7.81%)	75.08 (+11.32%)	72.63 (+1.45%)	75.56 (+7.83%)
SAGE	73.54 (+18.81%)	89.00 (+2.69%)	86.26 (+8.02%)	74.94 (+11.53%)	73.33 (+0.48%)	73.44 (+10.94%)
RevGAT	73.16 (+19.42%)	87.18 (+4.83%)	86.70 (+7.47%)	74.21 (+12.63%)	72.88 (+1.10%)	73.62 (+10.67%)
LLaGA-ND	**86.99 (+0.44%)**	90.91 (+0.53%)	88.89 (+4.83%)	88.19 (-5.23%)	73.04 (+0.88%)	73.62 (+10.68%)
LLaGA-HO	78.78 (+10.90%)	88.35 (+3.44%)	88.77 (+4.97%)	**88.56 (-5.62%)**	**74.02 (-0.46%)**	73.64 (+10.65%)
GraphGPT	NA	NA	84.68 (+10.04%)	NA	62.18 (+18.49%)	NA
GraphPrompter	78.38 (+11.47%)	**91.85 (-0.50%)**	**94.32 (-1.21%)**	70.11 (+19.21%)	72.38 (+1.80%)	**79.04 (+3.09%)**
GraphTranslator	38.95 (+124.31%)	22.88 (+299.43%)	60.46 (+54.12%)	35.59 (+134.84%)	28.48 (+158.71%)	41.32 (+97.19%)
Results of Zero-shot LLMs						
LLaMA3-70b-Instruct	57.42 (+52.16%)	67.45 (+35.49%)	91.94 (+1.35%)	66.24 (+26.18%)	62.72 (+17.47%)	57.62 (+41.41%)
LLaMA2-13b-Chat	47.48 (+84.01%)	35.39 (+158.24%)	74.04 (+25.85%)	52.58 (+58.96%)	44.04 (+67.30%)	57.34 (+42.10%)
Results of GraphICL (Ours)						
GraphICL-LLaMA3-S1	**87.37**	**91.39**	**93.18**	**83.58**	**73.68**	**81.48**
GraphICL-LLaMA3-S2	87.37	91.12	93.05	83.21	73.54	81.04
GraphICL-LLaMA2-S1	87.06	85.52	82.56	77.49	70.20	78.84
GraphICL-LLaMA2-S2	85.11	83.19	79.54	76.94	69.82	78.27

cross-domain. In the in-domain scenario, testing is performed on datasets used during training, whereas in the cross-domain scenario, the test datasets have no overlap with the training data.

5.2.1 In-Domain Node Classification

RQ 1. *Can GICL-prompted LLMs outperform state-of-the-art GNNs and specialized GraphLLMs in the in-domain scenario?*

Experiment Settings. We used six datasets (see Table 1) to evaluate our GraphICL method combined with LLaMA2 and 3 (as described in Section 5.1), comparing it against various GNNs, Graph LLMs, and LLM methods. For LLM-based methods, only the single most likely label was predicted, and accuracy was calculated accordingly.

Observation 1. ***Equipped with GraphICL, general-purpose LLMs can achieve competitive or even superior performance compared to specialized graph LLMs in both semi-supervised and supervised settings in the in-domain scenario.*** Specifically, in the semi-supervised setting, GraphICL achieves an average relative improvement of around 20% across datasets, with a significant 39.88% increase on the Computers dataset compared to GraphPrompter. Even in the supervised setting, GraphICL continues to outperform most Graph LLMs and all GNNs. Moreover, it demonstrates a marked improvement in reasoning capabilities over zero-shot LLMs, further solidifying its effectiveness across different learning paradigms.

5.2.2 Cross-Domain Node Classification

RQ 2. *Can GICL-prompted LLMs excel over top GNNs and GraphLLMs in Cross-Domain tasks with mismatched training and testing data?*

Experiment Settings. We used the same five datasets as the in-domain testing. Given that GNNs lack cross-domain capabilities, this experiment focused solely on comparing GraphICL with tailored GraphLLMs.

Observation 2. ***In the cross-domain scenario, GraphICL enables LLaMA to outperform tailored***

Table 2: Cross-Domain results of node classification. In the table, "NA" indicates "Not Applicable," meaning the corresponding dataset is part of the training set.

Method	Computers	Sports	PubMed	Cora	Arxiv	Products
Supervised Cross-Domain Results (GraphLLMs)						
LLaGA-ND	14.88 (+487.16%)	3.57 (+2459.94%)	NA	NA	NA	NA
LLaGA-HO	14.71 (+493.95%)	4.84 (+1788.22%)	NA	NA	NA	NA
GraphGPT	14.61 (+498.02%)	8.24 (+1009.10%)	NA	41.14 (+103.16%)	NA	31.67 (+157.28%)
GraphPrompter	26.40 (+230.95%)	9.26 (+886.93%)	NA	NA	3.62 (+1935.36%)	15.42 (+428.40%)
GraphTranslator	32.85 (+165.97%)	12.9 (+608.44%)	46.17 (+101.82%)	34.06 (+145.39%)	NA	18.31 (+345.00%)
Results of Zero-shot LLMs						
LLaMA3-70b-Instruct	**57.42 (+52.16%)**	**67.45 (+35.49%)**	**91.94 (+1.35%)**	**66.24 (+26.18%)**	**62.72 (+17.47%)**	**57.62 (+41.41%)**
LLaMA2-13b-Chat	47.48 (+84.01%)	35.39 (+158.24%)	74.04 (+25.85%)	52.58 (+58.96%)	44.04 (+67.30%)	57.34 (+42.10%)
Results of GraphICL (Ours)						
GraphICL-LLaMA3-S1	**87.37**	**91.39**	**93.18**	**83.58**	**73.68**	**81.48**
GraphICL-LLaMA3-S2	87.37	91.12	93.05	83.21	73.54	81.04
GraphICL-LLaMA2-S1	87.06	85.52	82.56	77.49	70.20	78.84
GraphICL-LLaMA2-S2	85.11	83.19	79.54	76.94	69.82	78.27

Graph LLMs without requiring additional training, demonstrating a significant advantage. For the Graph LLM methods, we employed a diverse combination of mixed training sets to enhance their cross-domain capabilities. However, despite these efforts, both Graph LLM and zero-shot LLM methods fall considerably short, with the former showing a relative performance gap exceeding 101%, showcasing its potential to adapt LLMs to unseen graph data and broader applications.

5.2.3 Link Prediction Testing

For link prediction, the substantial increase in text data significantly extends testing time. Therefore, we selected Cora for multi-scenario testing, similar to node classification. As shown in Table 3, our GraphICL method consistently achieves the best performance compared to other models, especially in the supervised setting, where it outperforms the best result from the remaining methods, including LLaGA-HO, by 1.26%, further confirming the observations made in the node classification task.

Table 3: Link prediction results in Cora. For GCN and GraphSAGE, we use sbert embeddings (Reimers, 2019).

Train → Test	Method	Accuracy
Cora (Semi-Supervised) ↓ Cora	GCN	58.97
	GraphSAGE	67.68
	GraphGPT	-
	LLaGA-ND	58.38
	LLaGA-HO	59.12
	LLaMA2	75.00
	LLaMA3	84.11
	GraphICL (Ours)	**88.08**
Cora (Supervised) ↓ Cora	GCN	81.59
	GraphSAGE	79.15
	GraphGPT	80.26
	LLaGA-ND	83.79
	LLaGA-HO	86.82
	LLaMA2	75.00
	LLaMA3	84.11
	GraphICL (Ours)	**88.08**
Arxiv+PubMed ↓ Cora	GCN	56.73
	GraphSAGE	58.92
	GraphGPT	50.74
	LLaGA-ND	86.47
	LLaGA-HO	87.35
	LLaMA2	75.00
	LLaMA3	84.11
	GraphICL (Ours)	**88.08**

5.3 Impact of GraphICL Configuration

Factors such as the type of LLMs using the GICL method and the inclusion of structural information can affect performance. In this section, we will explore these influencing factors.

5.3.1 LLMs Comparison with GraphICL

RQ 3. *How does the performance vary when different LLMs are paired with the same GICL method across various diverse datasets or tasks?*

Experiment Settings. We selected the Cora and Sports datasets to compare the results of three different LLMs presented in Table 4 for node classification. To reduce testing costs, we randomly sampled 1,000 data points from the test set of each dataset, and for each dataset, we chose one GICL method for evaluation.

Observation 3. *Based on the results, it is reasonable to infer that* ***more capable LLMs tend to perform better when integrated with GICL for graph reasoning (GR)****. We also anticipate that future large language models will be incorporated into our GICL benchmark, enabling a deeper investigation of their potential in GR tasks.* These differences in LLMs' capabilities are reflected in the consistent ranking of results across both datasets in Table 4, where GPT-4o outperforms the other models by 1%-16% on both datasets.

Table 4: The node classification accuracies of different LLMs under several different GraphICL methods on two dataset. $S-A$ represents specific neighborhood information in structure-aware, S_{SM} denotes the neighbor selection method, $Demo$ denotes the type of demonstrations, and D_{SM} denotes the method of selecting demonstrations.

Dataset	LLM	GraphICL				Acc(%)
		$S-A$	S_{SM}	$Demo$	D_{SM}	
Cora	GPT-4o	1-hop	Similarity	global	Pagerank	76.60
	LLaMA3-70B-Instruct	1-hop	Similarity	global	Pagerank	75.40
	LLaMA2-13B-Chat	1-hop	Similarity	global	Pagerank	70.60
Sports	GPT-4o	1-hop	Random	global	Random	91.00
	LLaMA3-70B-Instruct	1-hop	Pagerank	global	Random	84.80
	LLaMA2-13B-Chat	1-hop	Pagerank	global	Random	75.90

5.3.2 The Impact of Structural Information

RQ 4. *How critical is structural information in graph reasoning tasks?*

Experiment Settings. We conducted evaluations on both node classification and link prediction tasks, selecting the most popular dataset for each task as shown in Table 5. The test sets for both datasets were the same as those described in Section 5.1. We employed LLaMA3-70b-Instruct as the backbone for our GraphICL framework.

Observation 4. ***Structural information via GraphICL significantly boosts performance****, particularly in datasets where neighbor relationships play a crucial role, such as citation networks.* As illustrated in Table 5, graph prompt incorporating structure-aware neighbor information consistently yield better results than those omitting this information. This is true for both graph reasoning tasks, with different degrees of improvement observed when 1-hop or 2-hop neighbors are selected.

Table 5: Comparison of results with and without structure-aware information. For GraphICL, we adopted the abbreviation format as presented in Table 4, with further details available in the Appendix A.2.

Dataset	Task	GraphICL	Acc(%)	$\uparrow_{1-hop}$	$\uparrow_{2-hop}$
Arxiv	NC	XXGR	30.36	+107.83%	+111.85%
		1SGR	63.10		
		2SGR	64.32		
Cora	LP	XXCR	71.91	+4.29%	+10.83%
		1RCR	75.00		
		2RCR	79.70		

5.3.3 Further Analysis

Similar Neighbors Boosts Node Classification. GraphICL offers three neighbor selection methods: random, pagerank-based, and similarity-based. Using LLaMA3-70B-Instruct across four datasets, we calculated the average accuracy for each method. As shown in Table 6, the similarity-based method consistently achieved the highest accuracy, likely due to its focus on text similarity, where neighbors with similar content often share labels, enabling LLMs to make more accurate predictions.

Chain-of-Thought: Not Always Beneficial. We tested Chain-of-Thought (CoT) prompting in GraphICL by adding "Let's think step by step" to the prompt (Wei et al., 2022). However, CoT sometimes does not improve reasoning. Table 7 shows results from LLaMA2-13B-Chat on Cora, where despite some improvements in certain GraphICL methods, the average accuracy slightly declined, indicating CoT is not always effective.

Table 6: The average accuracy of different neighbor selection methods across four datasets.

Selection Mechanism	Cora	PubMed	Photo	History
Random	68.45	67.25	50.30	40.13
Pagerank	68.40	67.16	46.66	38.10
Similarity	**68.76**	**67.63**	**59.06**	**41.97**

Table 7: Accuracy comparison of different GraphICL Mmthods on Cora with and without CoT. "Average" represents the mean accuracy of all 55 settings.

GraphICL	CoT	Acc(%)	$\uparrow_{CoT}$
1RGR	No	75.46	+2.95%
1RGR	Yes	78.41	
1SCR	No	70.85	-3.69%
1SCR	Yes	67.16	
Average	No	65.53	-0.43%
Average	Yes	65.10	

6 Conclusions

We introduce GraphICL, a comprehensive prompt benchmark for graph in-context learning utilizing LLMs across a wide range of graph inference tasks. Extensive experimental evaluations demonstrate that GraphICL enables LLMs to achieve outstanding performance on multiple datasets, in some cases even outperforming state-of-the-art supervised GNNs and specialized Graph LLMs. Moving forward, we hope to expand our benchmark to incorporate additional LLMs, with the objective of advancing the frontiers of LLM capabilities in addressing complex graph-related challenges.

7 Limitation

We introduce GraphICL, which leverages graph in-context learning to enhance the performance of LLMs in graph reasoning. In terms of breadth, we acknowledge the need to test our template on more classic graph tasks. Additionally, to expand our benchmark, incorporating more large language models is essential for further enrichment. As for depth, given the complexity of graph structures, we need to explore how to better integrate structural information with demonstrations in the prompts, especially for text graphs of varying natures, such as molecular graphs, to achieve better results.

References

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Runjin Chen, Tong Zhao, Ajay Jaiswal, Neil Shah, and Zhangyang Wang. 2024a. Llaga: Large language and graph assistant. *arXiv preprint arXiv:2402.08170*.

Runjin Chen, Tong Zhao, AJAY KUMAR JAISWAL, Neil Shah, and Zhangyang Wang. 2024b. Llaga: Large language and graph assistant. In *Forty-first International Conference on Machine Learning*.

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

Eli Chien, Wei-Cheng Chang, Cho-Jui Hsieh, Hsiang-Fu Yu, Jiong Zhang, Olgica Milenkovic, and Inderjit S Dhillon. 2021. Node feature extraction by self-supervised multi-scale neighborhood prediction. *arXiv preprint arXiv:2111.00064*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, and Zhifang Sui. 2022. A survey on in-context learning. *arXiv preprint arXiv:2301.00234*.

Wenqi Fan, Yao Ma, Qing Li, Yuan He, Eric Zhao, Jiliang Tang, and Dawei Yin. 2019. Graph neural networks for social recommendation. In *The world wide web conference*, pages 417–426.

Yi Fang, Dongzhe Fan, Sirui Ding, Ninghao Liu, and Qiaoyu Tan. 2024a. Uniglm: Training one unified language model for text-attributed graphs. *arXiv preprint arXiv:2406.12052*.

Yi Fang, Dongzhe Fan, Daochen Zha, and Qiaoyu Tan. 2024b. Gaugllm: Improving graph contrastive learning for text-attributed graphs with large language models. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 747–758.

Victor Garcia and Joan Bruna. 2017. Few-shot learning with graph neural networks. *arXiv preprint arXiv:1711.04043*.

Louie Giray. 2023. Prompt engineering with chatgpt: a guide for academic writers. *Annals of biomedical engineering*, 51(12):2629–2633.

Jiayan Guo, Lun Du, Hengyu Liu, Mengyu Zhou, Xinyi He, and Shi Han. 2023. Gpt4graph: Can large language models understand graph structured data? an empirical evaluation and benchmarking. *arXiv preprint arXiv:2305.15066*.

Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30.

ZS Harris. 1954. Distributional structure.

Mohammad Al Hasan and Mohammed J Zaki. 2011. A survey of link prediction in social networks. *Social network data analytics*, pages 243–275.

Xiaoxin He, Xavier Bresson, Thomas Laurent, Bryan Hooi, et al. 2023a. Explanations as features: Llm-based features for text-attributed graphs. *arXiv preprint arXiv:2305.19523*, 2(4):8.

Xiaoxin He, Xavier Bresson, Thomas Laurent, Adam Perold, Yann LeCun, and Bryan Hooi. 2023b. Harnessing explanations: Llm-to-lm interpreter for enhanced text-attributed graph representation learning. In *The Twelfth International Conference on Learning Representations*.

Yufei He and Bryan Hooi. 2024. Unigraph: Learning a cross-domain graph foundation model from natural language. *arXiv preprint arXiv:2402.13630*.

Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. 2020. Open graph benchmark: Datasets for machine learning on graphs. *Advances in neural information processing systems*, 33:22118–22133.

Jin Huang, Xingjian Zhang, Qiaozhu Mei, and Jiaqi Ma. 2023. Can llms effectively leverage graph structural information: when and why. *arXiv preprint arXiv:2309.16595*.

Thomas N Kipf and Max Welling. 2016a. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

Thomas N Kipf and Max Welling. 2016b. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308*.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. *Preprint*, arXiv:2309.06180.

Guohao Li, Matthias Müller, Bernard Ghanem, and Vladlen Koltun. 2021. Training graph neural networks with 1000 layers. In *International conference on machine learning*, pages 6437–6449. PMLR.

Rui Li, Jiwei Li, Jiawei Han, and Guoyin Wang. 2024. Similarity-based neighbor selection for graph llms. *arXiv preprint arXiv:2402.03720*.

Yuyan Liu, Sirui Ding, Sheng Zhou, Wenqi Fan, and Qiaoyu Tan. 2024a. Moleculargpt: Open large language model (llm) for few-shot molecular property prediction. *arXiv preprint arXiv:2406.12950*.

Zheyuan Liu, Xiaoxin He, Yijun Tian, and Nitesh V Chawla. 2024b. Can we soft prompt llms for graph learning tasks? In *Companion Proceedings of the ACM on Web Conference 2024*, pages 481–484.

Andrew Kachites McCallum, Kamal Nigam, Jason Rennie, and Kristie Seymore. 2000. Automating the construction of internet portals with machine learning. *Information Retrieval*, 3:127–163.

Lawrence Page, Sergey Brin, Rajeev Motwani, Terry Winograd, et al. 1999. The pagerank citation ranking: Bringing order to the web.

N Reimers. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Patrick Reiser, Marlen Neubert, André Eberhard, Luca Torresi, Chen Zhou, Chen Shao, Houssam Metni, Clint van Hoesel, Henrik Schopmans, Timo Sommer, et al. 2022. Graph neural networks for materials science and chemistry. *Communications Materials*, 3(1):93.

Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. 2008. Collective classification in network data. *AI magazine*, 29(3):93–93.

Oleksandr Shchur, Maximilian Mumme, Aleksandar Bojchevski, and Stephan Günnemann. 2018. Pitfalls of graph neural network evaluation. *arXiv preprint arXiv:1811.05868*.

Naoki Shibata, Yuya Kajikawa, and Ichiro Sakata. 2012. Link prediction in citation networks. *Journal of the American society for information science and technology*, 63(1):78–85.

Chuxiong Sun, Hongming Gu, and Jie Hu. 2021. Scalable and adaptive graph neural networks with self-label-enhanced training. *arXiv preprint arXiv:2104.09376*.

Qiaoyu Tan, Ninghao Liu, Xiao Huang, Soo-Hyun Choi, Li Li, Rui Chen, and Xia Hu. 2023. S2gae: Self-supervised graph autoencoders are generalizable learners with graph masking. In *Proceedings of the sixteenth ACM international conference on web search and data mining*, pages 787–795.

Jiabin Tang, Yuhao Yang, Wei Wei, Lei Shi, Lixin Su, Suqi Cheng, Dawei Yin, and Chao Huang. 2023. Graphgpt: Graph instruction tuning for large language models. *arXiv preprint arXiv:2310.13023*.

Zheng Wang, Hongming Ding, Li Pan, Jianhua Li, Zhiguo Gong, and S Yu Philip. 2024. From cluster assumption to graph convolution: Graph-based semi-supervised learning revisited. *IEEE Transactions on Neural Networks and Learning Systems*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Shiwen Wu, Fei Sun, Wentao Zhang, Xu Xie, and Bin Cui. 2022. Graph neural networks in recommender systems: a survey. *ACM Computing Surveys*, 55(5):1–37.

Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. 2020. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, 32(1):4–24.

Yuhao Xu, Xinqi Liu, Keyu Duan, Yi Fang, Yu-Neng Chuang, Daochen Zha, and Qiaoyu Tan. 2024. Graphfm: A comprehensive benchmark for graph foundation model. *arXiv preprint arXiv:2406.08310*.

Ruosong Ye, Caiqi Zhang, Runhui Wang, Shuyuan Xu, and Yongfeng Zhang. 2023. Natural language is all a graph needs. *arXiv preprint arXiv:2308.07134*.

Mengmei Zhang, Mingwei Sun, Peng Wang, Shen Fan, Yanhu Mo, Xiaoxiao Xu, Hong Liu, Cheng Yang, and Chuan Shi. 2024. Graphtranslator: Aligning graph model to large language model for open-ended tasks. In *Proceedings of the ACM on Web Conference 2024*, pages 1003–1014.

Jianan Zhao, Le Zhuo, Yikang Shen, Meng Qu, Kai Liu, Michael Bronstein, Zhaocheng Zhu, and Jian Tang. 2023. Graphtext: Graph reasoning in text space. *arXiv preprint arXiv:2310.01089*.

Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. 2020. Graph neural networks: A review of methods and applications. *AI open*, 1:57–81.

A Appendix

A.1 General Prompt Template

The prompt inputted into LLMs consists of a system prompt, user content, and some special characters. In this section, we will showcase the system prompts and user content we designed for various tasks and datasets.

System Prompt Design. The system prompt is often used to make the LLMs aware of the task they are about to perform. Table 13 presents the system prompts used for node classification tasks across different datasets, while Table 14 shows the system prompts for link prediction tasks in two major types of datasets.

User Content Design. In GraphICL, user content is used to record information other than the task description, such as structure-aware text information, anchor node text information, and demonstrations. The specific templates are shown in Table 16 and Table 15.

A.2 Methodology

For the zero-shot setting, we first provide the text information of the anchor nodes and implement seven different structure-aware methods: "XX," "1R," "1P," "1S," "2R," "2P," and "2S." In this context, "1" and "2" represent one-hop and two-hop neighbor information, respectively, which is incorporated into the structure-aware content.

In the few-shot setting, there are multiple approaches to implementing demonstrations. In the structure-aware configuration, six methods are used: "1'R," "1'P," "1'S," "2'R," "2'P," and "2'S." Here, "1'" and "2'" indicate that one-hop and two-hop neighbors are used as demonstrations. In the non-structure-aware configuration, seven methods are applied: "GR," "GP," "GS," "CR," "CP," "CS," and "XX."

The absence of "XX" in the structure-aware category is due to the fact that "XX" in the structure-aware context is equivalent to "XX" in the non-structure-aware context. Therefore, "XX" is counted only in the non-structure-aware group. Additionally, "G" and "C" refer to the demonstration selection scope: "G" indicates that demonstrations are selected without regard to labels, while "C" ensures that one demonstration is selected per label from the training set. The letters "R," "P," and "S" indicate the selection mechanisms—random, PageRank, and similarity, respectively.

In total, there are 55 possible combinations: 7 structure-aware methods combined with 7 demonstrations, plus 1 structure-aware "XX" combined with 6 non-structure-aware demonstrations, resulting in $7 \times 7 + 1 \times 6 = 55$ methods.

For the GICL settings we used in Table 1 and Table 2, please refer to Table 8. The GICL setting we used in link prediction testing (Table 3) is "1SXX".

Table 9: Statistics of the TAG datasets.

Dataset	#Nodes	#Edges	#Classes
Cora	2,708	5,429	7
PubMed	19,717	44,338	3
OGB-Arxiv	169,343	1,166,245	40
OGB-Products (subset)	54,025	74,420	47
Amazon-Sports	173,055	1,946,555	13
Amazon-Computers	87,229	721,107	10
Amazon-Photo	48,362	500,939	12
Amazon-Children	76,875	1,631,453	24
Amazon-History	41,551	358,574	13

A.3 Experiments

A.3.1 Evaluation Datasets

The statistics for all TAG datasets used in this study can be found in Table 9. In our node classification experiments, data splitting was rigorously conducted according to established protocols to ensure consistency and comparability of the results. For the Cora, PubMed and OGB-Products datasets, we followed the splits specified by TAPE (He et al., 2023b). For OGB-Arxiv dataset, we used the standard split provided by the OGB framework (Hu et al., 2020), ensuring strict compliance with the benchmark's guidelines. For the other Amazon datasets, we applied a 6:2:2 ratio for training, validation, and testing sets.

In the supervised setting, the splits for Cora and PubMed were based on TAPE's guidelines. For OGB-Products, we sampled 5000 instances from the testing set based on the TAPE split. Similarly, we also sampled 5000 instances from the standard testing set. For other Amazon datasets, we followed the 6:2:2 spliting strategy.

In the semi-supervised setting, for Cora and PubMed, we adopted the standard semi-supervised splits (Wang et al., 2024), while for OGB-Products, we applied a 20-shot split. For all Amazon datasets, a 300-shot split was used. Additionally, we ensured that the testing sets in the semi-supervised setting were consistent with those in the supervised setting.

For the link prediction evaluation, in the supervised setting, we followed the same splits as used in LLaGA (Chen et al., 2024a). In the semi-supervised setting, we randomly sampled 5% of the examples from the supervised training set, ensuring an equal number of positive and negative samples,

Figure 2: Examples of graph in-context learning prompting in different graph reasoning tasks.

Task Description | Anchor Node Info | Structure-Aware Info | Demonstrations

Human Question(Node Classification):

Basic Content

I'm starting a node classification task. Using the provided [paper/product] description, categorize the target node ([paper/product]) into one of the following categories: …. Respond in this format: ….
Target node content: ….

Optional Enhancing Content

Below I will provide you with target node information, target node neighbor information and some other examples in order. You need to use target node neighbor information and some other examples to help you predict the category of target node.
It has following neighbor products(co-purchase) at hop k. …… Product n Description: … Label n: …. ……
I will give you some other examples to help you predict the category. …… Other example n's content: …. Another example n's category:…. ……

LLMs Response:

Based on the provided information, I would classify the [target node] in the [category]. The reason for this categorization is that ….

Human Question(Link Prediction):

Basic Content

I'm starting a link prediction task. Please predict whether there's a link between the following 2 nodes. In this graph, links between nodes represent [citation relationships between papers/2 products are frequently purchased or browsed together]. Your answer should be '0' or '1'. '0' means there's no link and '1' means there's a link.
The 2 target nodes content: ….

Optional Enhancing Content

For the first node:
It has following neighbor products(co-purchase) at hop k. …… Product n Description: … Label n: …. ……
I will give you some other examples to help you predict the category. …… Other example n's content: …. Another example n's category:…. ……

LLMs Response:

My prediction: [1/0].

Table 8: The GICL settings of in-domain node classification results.

Model	Computers	Sports	PubMed	Cora	Arxiv	Products
GraphICL-LLaMA3-S1	1RCP	1RGP	2SCS	1RGR	1RCP	1RGS
GraphICL-LLaMA3-S2	1RGP	1RGS	2SCR	1SCP	1RGR	1RGR
GraphICL-LLaMA2-S1	1'SXX	2SXX	2'SXX	2SGR	2SCP	2SCP
GraphICL-LLaMA2-S2	1'RXX	1SGS	2'RXX	1SGR	1SCP	1'SXX

while keeping the test set unchanged.

A.3.2 Computing Environment and Resources

We leveraged the vLLM package (Kwon et al., 2023) for large language models inference. Locally, we deployed the LLaMA2-13B-Chat model on a single NVIDIA A800 80GB GPU and the LLaMA3-70b-Instruct model on two of these GPUs to accommodate its greater computational requirements. For GPT-4o inference, we utilized the OpenAI API.

A.3.3 Number of Neighbors and Examples

Based on preliminary exploratory experiments, we observed that generally, the more neighbors or examples provided in the prompt, the higher the prediction accuracy of the LLMs. However, there is a length constraint on the LLMs' input window. Within this constraint, we determined that a maximum of 6 neighbors or examples can be included. We utilize GIA (Chien et al., 2021) embeddings to compute similarity.

A.3.4 Node Classification Results

In Table 1, we report results for the supervised setting of GraphGPT, as the available checkpoints only support joint supervised training on the PubMed and OGB-Arxiv datasets, preventing us from evaluating its semi-supervised performance. And for Table 3, the checkpoint for semi-supervised training on cora is also missing. Similarly, GraphTranslator's self-supervised training does not involve label information from the dataset, making it unsuitable for division into supervised or semi-supervised categories. Therefore, we include its results both under the semi-supervised and supervised setting. For all datasets listed in Table 1, in addition to using BERT embeddings for MLP and GNN models, we also employed BoW (Harris, 1954) and GIA (Chien et al., 2021) embeddings in both semi-supervised and supervised settings, as shown in Table 10.

For LLaGA, GraphPrompter, GraphTranslator, and GraphGPT, we utilized Vicuna-7b-v1.5-16k, LLaMA2-13b-Chat-HF, ChatGLM6B, and Vicuna-7B-v1.5 as their respective LLM back-

bones. Additionally, for LLaGA, GraphTranslator, and GraphGPT, we used the same types of embeddings as in the original works, while for GraphPrompter, we tested using GIA embeddings (Chen et al., 2024c). In the in-domain experiments, we adopted a 'single focus' training approach for all models except GraphGPT, meaning that each model was trained on an individual training set and tested on its corresponding test set. For GraphGPT, we directly loaded the model parameters from its mixed training setup for testing. In the cross-domain experiments, all models were trained on multiple training sets jointly. Apart from LLaGA, which followed its original mixed training strategy, other models combined training sets at a 1:1 ratio.

For the three Amazon review datasets (Amazon-Photo, Amazon-Children, and Amazon-History), which are not included in Table 1, we provide both in-domain and cross-domain results in Table 11 and Table 12, respectively.

We selected PubMed to showcase the best results from the 55 configurations tested with our GraphICL combined with LLaMA3-70B-Instruct, comparing them to supervised GNN methods. Detailed comparisons can be found in Figure 3.

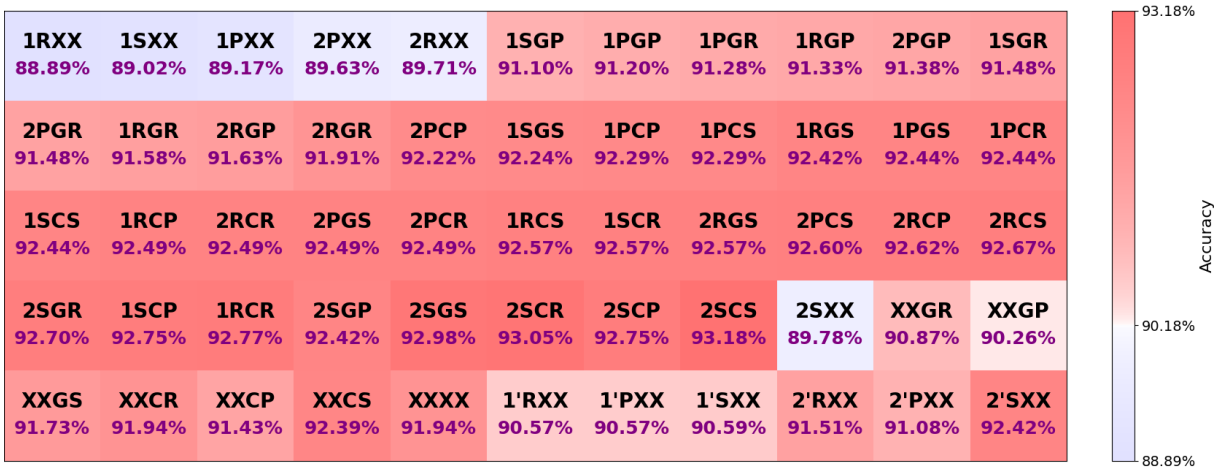

Figure 3: The comparison heat map between the GraphICL method based on LLaMA3-70B-Instruct and the best Supervised GNNs results (SAGE-90.18%) in the table 10 in PubMed. The results of the GNNs are used as the baseline, with higher performance relative to the baseline shown in red and lower performance shown in blue.

Method	Embedding	Computers	Sports	PubMed	Cora	Arxiv	Products
Semi-Supervised In-Domain Results							
MLP	BoW	49.69	67.17	67.14	52.95	27.38	56.80
	BERT	44.56	58.74	59.38	47.23	37.10	65.36
	GIA	66.80	81.80	74.82	64.02	48.39	70.23
GCN	BoW	72.58	64.76	80.07	74.58	50.31	71.65
	BERT	70.24	59.12	59.38	68.82	55.27	74.47
	GIA	81.40	76.80	77.62	69.45	51.36	74.95
SAGE	BoW	73.16	63.23	77.72	67.23	45.73	69.21
	BERT	69.53	58.52	64.66	64.58	54.05	72.35
	GIA	83.18	76.36	76.06	70.85	55.20	73.66
RevGAT	BoW	69.05	59.41	73.28	71.40	39.41	67.99
	BERT	64.63	55.48	64.10	65.31	48.86	71.45
	GIA	81.55	74.78	78.09	72.88	50.94	73.78
Supervised In-Domain Results							
MLP	BoW	64.90	84.12	71.88	74.72	55.59	58.83
	BERT	61.74	85.58	82.28	60.89	66.07	67.56
	GIA	75.72	90.97	90.04	77.12	71.64	70.91
GCN	BoW	77.99	88.87	86.76	88.19	71.31	72.06
	BERT	74.84	88.33	85.51	86.90	72.82	75.56
	GIA	82.74	91.97	88.82	88.39	73.56	75.36
SAGE	BoW	76.86	88.81	85.46	88.93	70.43	70.25
	BERT	73.62	89.21	85.66	83.39	72.54	73.44
	GIA	82.98	92.64	90.18	82.95	74.07	74.38
RevGAT	BoW	77.91	89.76	89.56	86.90	70.59	70.93
	BERT	72.87	88.70	86.01	82.66	73.21	73.62
	GIA	83.43	92.94	88.92	82.47	74.74	74.88
Results of GraphICL							
GraphICL-LLaMA3-S1	-	**87.37**	**91.39**	**93.18**	**83.58**	**73.68**	**81.48**
GraphICL-LLaMA3-S2	-	87.37	91.12	93.05	83.21	73.54	81.04

Table 10: Extended in-domain Results from **Table 1** using different embedding types for MLP and GNN models.

Method	Embedding	Photo	Children	History
Semi-Supervised In-Domain Results				
MLP	BoW	51.07	25.96	58.47
	BERT	42.08	31.54	69.41
	GIA	66.70	36.70	74.21
GCN	BoW	63.05	30.77	64.60
	BERT	59.78	35.06	69.51
	GIA	69.80	34.13	71.67
SAGE	BoW	63.20	30.84	68.09
	BERT	59.75	35.61	73.79
	GIA	71.44	40.01	75.75
RevGAT	BoW	60.03	29.97	61.41
	BERT	54.07	34.60	70.99
	GIA	70.08	36.96	73.22
Supervised In-Domain Results				
MLP	BoW	68.50	49.71	77.37
	BERT	67.93	51.46	82.41
	GIA	79.73	55.96	84.13
GCN	BoW	77.05	53.56	81.12
	BERT	77.08	54.53	83.45
	GIA	82.62	55.23	84.27
SAGE	BoW	77.41	54.86	80.82
	BERT	76.40	55.27	84.06
	GIA	83.28	58.41	85.12
RevGAT	BoW	77.84	52.96	80.97
	BERT	75.87	53.10	83.09
	GIA	83.33	55.73	84.38
Results of GraphICL				
GraphICL-LLaMA3-S1	-	**79.35**	**47.96**	**80.89**
GraphICL-LLaMA3-S2	-	77.78	47.63	79.18

Table 11: In-domain results of amazon-photo, amazon-children, and amazon-history. For the Amazon-Photo dataset, S1 is "1RGS" and S2 is "1RCS". For Amazon-History, S1 is "1SGS" and S2 is "1PGS". For Amazon-Children, S1 is "1RGP" and S2 is "1RGS".

Method	Photo	Children	History
Supervised Cross-Domain Results (GraphLLMs)			
LLaGA-ND	19.83	7.49	6.45
LLaGA-HO	6.16	11.14	7.94
GraphGPT	6.18	14.56	10.94
GraphPrompter	25.01	10.35	14.62
GraphTranslator	38.96	16.13	6.64
Results of GraphICL			
GraphICL-LLaMA3-S1	**79.35**	**47.96**	**80.89**
GraphICL-LLaMA3-S2	77.78	47.63	79.18

Table 12: Cross-Domain Results of Amazon-Photo, Amazon-Children, and Amazon-History. S1 and S2 are the same as Table 11

Dataset	System Prompt Content
Cora	I'm starting a node classification task. Please predict the most appropriate category for the target node (paper). Choose from the following categories: \n Rule Learning \n Neural Networks \n Case Based \n Genetic Algorithms \n Theory \n Reinforcement Learning \n Probabilistic Methods.
PubMed	I'm starting a node classification task. Please predict the most likely type of the target node (paper). Your answer should be chosen from: \n Type 1 diabetes. \n Type 2 diabetes. \n Experimentally induced diabetes.
OGB-Arxiv	I'm starting a node classification task. Please predict the most appropriate Arxiv Computer Science (CS) sub-category for the target node (paper). The predicted sub-category should be in the format 'cs.XX'.
Amazon-History	I'm starting a node classification task. Using the provided history-related book's title and description, categorize the target node (book) into one of the following categories: ['Americas', 'Asia', 'Australia & Oceania', 'World', 'Europe', 'Middle East', 'Historical Study & Educational Resources', 'Arctic & Antarctica', 'Ancient Civilizations', 'Africa', 'Russia', 'Military']. Respond in this format: The book belongs to the [Category] category due to [evidence from the book product descriptions].
Amazon-Computers	I'm starting a node classification task. Given the product review provided, please categorize the target node (product) into one of the following categories: ['Tablet Replacement Parts', 'Monitors', 'Networking Products', 'Computers & Tablets', 'Computer Accessories & Peripherals', 'Tablet Accessories', 'Laptop Accessories', 'Computer Components', 'Data Storage', 'Servers']. Your classification should be based on the content of the review. Please support your answer with evidence from the review. Response Format: The product falls under the category of [Category]. This determination is based on the product review, where [specific details from the review supporting the classification].
Amazon-Photo	I'm starting a node classification task. Given the product review provided, please categorize the target node (product) into one of the following categories: ['Flashes', 'Film Photography', 'Accessories', 'Lighting & Studio', 'Video Surveillance', 'Underwater Photography', 'Digital Cameras', 'Tripods & Monopods', 'Lenses', 'Video', 'Binoculars & Scopes', 'Bags & Cases'] Your classification should be based on the content of the review. Please support your answer with evidence from the review. Response Format: The product falls under the category of [Category]. This determination is based on the product review, where [specific details from the review supporting the classification].
Amazon-Book	I'm starting a node classification task. Using the provided children book's title and description, categorize the target node (book) into one of the following categories: ['Literature & Fiction', 'Animals', 'Growing Up & Facts of Life', 'Humor', 'Cars, Trains & Things That Go', 'Fairy Tales, Folk Tales & Myths', 'Activities, Crafts & Games', 'Science Fiction & Fantasy', 'Classics', 'Mysteries & Detectives', 'Action & Adventure', 'Geography & Cultures', 'Education & Reference', 'Arts, Music & Photography', 'Holidays & Celebrations', 'Science, Nature & How It Works', 'Early Learning', 'Biographies', 'History', 'Children's Cookbooks', 'Religions', 'Sports & Outdoors', 'Comics & Graphic Novels', 'Computers & Technology']. Please provide your reasoning. Respond in this format: The book belongs to the [Category] category due to [evidence from the book product descriptions].
Amazon-Sports	I'm starting a node classification task. Using the provided item's title in the Sports & Fitness category, categorize the target node (item) into one of the following categories: ['Other Sports', 'Exercise & Fitness', 'Hunting & Fishing', 'Accessories', 'Leisure Sports & Game Room', 'Team Sports', 'Boating & Sailing', 'Swimming', 'Tennis & Racquet Sports', 'Golf', 'Airsoft & Paintball', 'Clothing', 'Sports Medicine']. Please provide your reasoning. Respond in this format: The item belongs to the [Category] category due to [evidence from the item descriptions].
OGB-Products	I'm starting a node classification task. Using the provided amazon product's title and description, please predict the most likely category of this node (product) from Amazon. Your answer should be chosen from the following categories: (Categories omitted due to length)

Table 13: System prompts for node classification tasks across various datasets.

Dataset Type	System Prompt Content
Citation Network	I'm starting a link prediction task. Please predict whether there's a link between the following 2 nodes. In this graph, links between nodes represent the citation relationships between papers. Your answer should be '0' or '1'. '0' means there's no link and '1' means there's a link.
Amazon Review Dataset	I'm starting a link prediction task. Please predict whether there's a link between the following 2 nodes. In this graph, links between nodes represent that 2 <specific type> products are frequently purchased or browsed together. Your answer should be '0' or '1'. '0' means there's no link and '1' means there's a link.

Table 14: System prompts for link prediction tasks across two types of datasets (Citation Networks and Amazon Datasets).

GraphICL	User Content
Zero-Shot without Structure-Aware Information	Below I will provide you with target node information. Please reason step by step. \n Target node content: <Target Node Text>.
Zero-Shot with Structure-Aware Information	Below I will provide you with target node information and target node neighbor information. You need to use target node neighbor information to help you predict the category of target node. Please reason step by step. \n Target node content: <Target Node Text>. \n It has following neighbor <products(co-purchase) / books / papers> at hop <Number of Hops>: [Neighbors' Text>.
Few-Shot without Structure-Aware Information	Below I will provide you with target node information and some other examples in order. You need to use examples to help you predict the category of target node. Please reason step by step. \n Target node content: <Target Node Text>. \n I will give you some other examples to help you predict the category: <Example's Text, Example's Label>.
Few-Shot with Structure-Aware Information	Below I will provide you with target node information, target node neighbor information and some other examples in order. You need to use target node neighbor information and some other examples to help you predict the category of target node. Please reason step by step. \n Target node content: <Target Node Text>. \n It has following neighbor <products(co-purchase) / books / papers> at hop <Number of Hops>: [Neighbors' Text>. \n I will give you some other examples to help you predict the category: <Example's Text, Example's Label>.

Table 15: User Content for node classification tasks across two types of datasets (Citation Networks and Amazon Datasets).

GraphICL	User Content
Zero-Shot without Structure-Aware Information	Below I will provide you with target 2 nodes information. \n The 2 target nodes content: <2 Target Nodes Text>
Zero-Shot with Structure-Aware Information	Below I will provide you with target 2 nodes information and the first node's neighbor information in order. You need to use the first node's neighbor information to help you predict the link between the 2 target nodes. \n The 2 target nodes content: <2 Target Nodes Text> \n For the first node: It has following neighbor papers at hop <Number of Hops>: <Neighbors' Text>.
Few-Shot without Structure-Aware Information	Below I will provide you with target 2 nodes information and some other examples of node pairs and connections in order. You need to use the other examples to help you predict the link between the 2 target nodes. \n The 2 target nodes content: <2 Target Nodes Text>. \n The following are the some other examples of node pairs and connections: <Examples of Node Pairs' Text, Connected: Yes/No>
Few-Shot with Structure-Aware Information	Below I will provide you with target 2 nodes information, the first node's neighbor information and some other examples of node pairs and connections in order. You need to use the first node's neighbor information and other examples to help you predict the link between the 2 target nodes. \n The 2 target nodes content: <2 Target Nodes Text>. \n For the first node: It has following neighbor papers at hop <Number of Hops>: <Neighbors' Text>. \n The following are the some other examples of node pairs and connections: <Examples of Node Pairs' Text, Connected: Yes/No>

Table 16: User Content for link prediction tasks across two types of datasets (Citation Networks and Amazon Datasets).