# Heterogeneous Graph In-Context Learning with Large Language Models

1st Jian Jia
*School of Information and Electrical Engineering*
*Hebei University of Engineering*
Handan, China

2nd Di Jin
*College of Intelligence and Computing*
*Tianjin University*
Tianjin, China

***Abstract*—Heterogeneous graph learning is crucial for modeling multi-typed entities and their interactions in real-world complex systems. However, existing methods face significant limitations: Heterogeneous Graph Neural Networks (HGNNs) underutilize textual attributes, while Large Language Models (LLMs) struggle to bridge the semantic gap between textual data and graph structures. To address these limitations, we propose HGICL, a novel framework that facilitates deep collaborative optimization between LLMs and HGNNs. The framework leverages a small set of actively selected exemplar nodes, fine-tunes HGNNs with LLM-generated scores, and applies the optimized HGNNs to node classification. This integration raises two key technical challenges. First, selecting appropriate nodes as in-context learning (ICL) examples is critical, since LLMs cannot inherently model heterogeneous graph structures. Second, bridging the semantic gap between textual and graph-structural information is essential, given the inherent disparity between graph topology and textual modalities that demands explicit alignment. To overcome these challenges, we propose an optimization strategy that integrates HGNNs and LLMs. Specifically, we design a meta-path-based node selection mechanism to construct high-information-density ICL examples, use LLMs to assign confidence scores to different exemplar nodes, and leverage these scores as supervisory signals to fine-tune the HGNNs. Comparative experiments involving three HGNN architectures, two homogeneous graph architectures, and a variety of LLMs demonstrate HGICL's superior performance, establishing a new paradigm for collaborative optimization in heterogeneous graph representation learning.***

***Index Terms*—Heterogeneous Graph, Large Language Models, In-Context Learning**

## I. Introduction

Heterogeneous graphs have become a typical modeling method for describing complex systems, characterized by their heterogeneous topology consisting of multiple types of nodes and edges with diverse semantics. In recent years, the field of heterogeneous graph representation learning has seen numerous methods that primarily leverage graph topology to learn high-quality node representations [1]. Although these methods can learn effective node representations without relying on labels, their utilization of textual attributes remains largely at the level of feature engineering, and they lack deeper semantic understanding and integration [2].

Thus, the advent of LLMs opens new avenues for addressing these limitations. Their core strength lies in being capable of few-shot reasoning and ICL to achieve a deep semantic understanding of node text attributes.

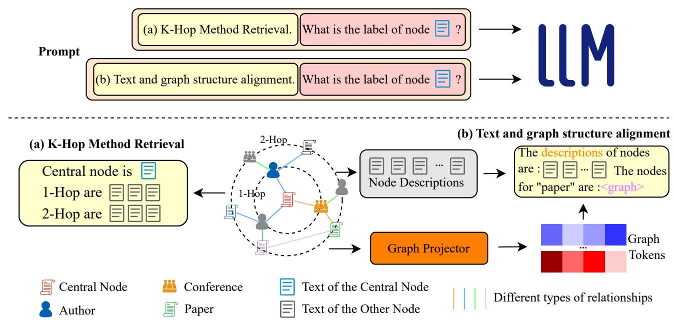


Fig. 1: Two methods to leverage ICL examples in graph structures are: (a) selecting $K$-hop domains as ICL examples, (b) ICL examples after aligning graph vectors with text vectors.

Two primary methodologies exist to leverage ICL in graph structures. The first retrieves strongly relevant example nodes from the graph, typically by selecting the $K$-hop neighborhood of a central node, as in GNN-RAG [3], InstructGLM [4], and GraphAdapter [5] (Fig.1 (a)). However, this approach suffers from limited receptive field (ignoring nodes beyond $K$-hops) and potential noise introduction [6]. The second method employs contrastive learning to align graph-encoder-generated "graph vectors" with LLM-friendly "text vectors" into a unified semantic space, which then serves as ICL, exemplified by TEA-GLM [7] and HiGPT [8] (Fig.1 (b)). Nevertheless, the inherent modality gap means such alignment reduces rather than eliminates the discrepancy [9].

To bridge the semantic gap between text and graph structures [10], AskGNN [11] proposes to employ Graph Neural Networks (GNNs) as information translators. Its core idea is to encode the graph structure with a GNN, select a set of the most informative ICL examples, thereby "translating" complex topological information into a form that is comprehensible to LLMs. These structure-infused examples are then provided to the LLM to stimulate deeper ICL reasoning. However, although AskGNN incorporates structural information, its retrieval strategy fails to adequately capture the heterogeneous semantics inherent in different node and relation types of a graph. This limitation hinders the full integration of heterogeneous semantic information, consequently restricting the model's expressive and reasoning power on such graphs.

Hence, the core challenge addressed in this paper is: *how to deeply understand the semantic relationships between different types in a heterogeneous graph and, based on this understanding, select the most informative nodes as ICL examples?* To this end, we propose a novel framework named HGICL, which deeply integrates the structural advantages of HGNNs with the semantic understanding capabilities of LLMs. In this framework, the HGNN acts as a structure-enhanced retriever. Utilizing meta-paths as the semantic core, it retrieves ICL nodes from the graph that are most semantically relevant to the target node. The framework then constructs high-quality ICL examples by forming structured node-text pairs, which combine the rich structural semantics embodied by the meta-paths with the nodes' own textual information. Finally, by designing task-appropriate prompt templates, the framework provides precise ICL input to the LLM, fully leveraging its capabilities in few-shot learning and semantic reasoning to achieve efficient collaboration between heterogeneous graph representation and semantic understanding [12].

To address the modal misalignment between structural information in heterogeneous graphs and textual semantic information, and to precisely select ICL examples combinations that maximize the reasoning performance of LLMs, we propose an optimization method for the retriever. The core of this method involves the introduction of an LLM-feedback-based optimization mechanism. Specifically, the method first uses the LLM to independently assess the confidence level of each candidate ICL example, quantifying its reliability as a basis for reasoning. This metric then serves as a supervisory signal to guide the HGNN in refining its retrieval strategy. This process forms an enhancement loop called "retrieval-evaluation-optimization", ultimately leading to continuous improvement in the predictive performance of the LLM. The three core contributions of this paper are as follows:

(1) We propose HGICL, a novel framework for few-shot learning on heterogeneous graphs.

(2) We design an HGNN-guided mechanism to construct high-quality ICL examples. By modeling complex heterogeneous relations, it selects informative nodes to enhance the reasoning of LLMs for node labeling.

(3) Extensive experiments on multiple public datasets demonstrate that HGICL outperforms baseline methods in node classification tasks, and its robustness and effectiveness.

## II. Related Work

### A. Heterogeneous Graph Neural Networks

HGNNs aim to capture the complex relationships between entities in heterogeneous graphs. Some methods generate multiple isomorphic subgraphs based on meta-paths and then apply GNNs. For example, HAN [13] applies node-level and semantic-level attention on the meta-path-based graph. MAGNN [1] also aggregates the representations of all nodes in the path on this basis. In contrast, HAHE [14] employs cosine similarity to replace non-attention mechanisms for calculating the importance of meta-path neighbors. Other methods do not use meta-paths. For example, HetSANN [15] and HGT [16] adopt a node-type-specific design, evaluating the importance of interactions with various types of neighbor nodes. HetGNN [17] samples heterogeneous neighbors via random walks.

### B. In-Context Learning

ICL is an important reasoning paradigm that enables pre-trained LLMs to adapt to new tasks using only task instructions and a few examples provided in the input prompt [18]. The optimization of ICL typically revolves around three key aspects: example selection, retrieval mechanisms, and long-context modeling. In terms of example selection, FLARE [19] proposes an iterative retrieval method capable of dynamically fetching relevant information during text generation. Tai et al. [20] emphasizes that systematically selecting and ordering ICL examples is crucial for maximizing classification performance. Regarding long-context modeling, a primary challenge is that models may suffer from issues such as attention dilution and information decay. To mitigate these, SELF-RAG [21] enhanced language models by introducing adaptive retrieval and self-reflection mechanisms. Similarly, OpenICL [22] introduces a framework that trains a unified retriever using a ranking-based approach to select relevant examples.

## III. Preliminaries

A heterogeneous graph is a special type of information network that contains multiple types of objects (nodes) or links (edges). It is defined as $G = (V, E, \Lambda, \mathcal{T}, \mathcal{R}, \mathcal{X})$, which consists of: a node set $V$, an edge set $E$, and an adjacency matrix $\Lambda \in \{0,1\}^{|V|\times|V|}$, where $\Lambda_{ij} = 1$ indicates that there exists a direct edge from node $i$ to node $j$. $\mathcal{T}$ and $\mathcal{R}$ denote the sets of node types and edge, respectively, with the condition that $|\mathcal{T}| + |\mathcal{R}| > 2$. $\mathcal{X} = \{x_i\}_{i=1}^{|V|}$ represents the textual information $x_i$ associated with each node $i$.

Given this structured but complex data, this study aims to address the challenge of generating effective prompts for LLMs based on only a limited set of labeled nodes $D = \{(x_i, y_i)\}$. To fully leverage the heterogeneous information, our core methodology involves, for a given query node $v$, semantically selecting the most relevant ICL examples $D_v \subseteq D$ by considering the meta-paths and multi-typed neighborhoods in $G$. We then integrate these examples with the node-specific textual information $X_v$ and its heterogeneous context (the types and relations of its connected nodes) to construct a composite, structure-aware prompt $T(v) = (D_v, X_v, \mathcal{C}_v)$, where $\mathcal{C}_v$ encapsulates the relevant structural and relational semantics. This enriched prompt is then used as ICL input to guide the LLMs in performing more accurate and context-aware inference tasks related to node $v$.

## IV. Methodology

We propose the HGICL framework to address node classification in heterogeneous graphs under low-label-rate scenarios. It organically integrates the few-shot semantic understanding of LLMs with the structural modeling of HGNNs. Our HGICL framework, inspired by AskGNN [11], operates in three tightly coupled stages (Fig.2): (1) meta-path guided node

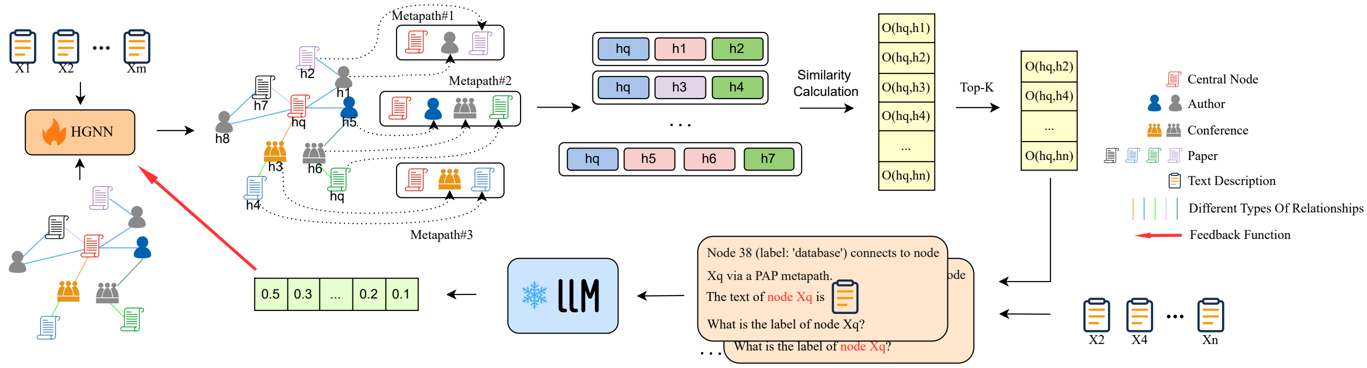


Fig. 2: The core of HGICL is to use feedback from LLMs to optimize HGNN. The process is as follows: node encoding and meta-path selection based on HGNN obtain relevant ICL examples, then use LLMs to analyze and provide feedback on these examples, continuously optimizing the HGNN model, thereby improving its performance on node classification tasks.

selection, where the HGNN acts as a structure-aware retriever; (2) generation of textual prompts and confidence scores for the LLM; (3) confidence score driven training, where the LLM's semantic feedback refines the HGNN. This forms a closed-loop optimization cycle that enhances both components iteratively.

### A. Meta-path guided node selection

As the initial step of our framework, this module functions as a structure-aware retriever. Given the constraint of a limited labeling budget, it is crucial to efficiently identify the most informative nodes. Its objective is to automatically select the top-$K$ most representative nodes for constructing the ICL example set $\mathcal{D}$. The selection is based on the node representations learned by the HGNN and must balance two key factors: topological importance and semantic similarity. The detailed procedure is as follows:

*1) Node Embedding:* We process the heterogeneous graph G using a HGNN. The model first projects node features based on their types, then iteratively refines node representations through multiple layers of neighborhood aggregation. This process outputs a learned representation vector for each node.

$$\boldsymbol{h}_v^{(l)} = \text{Aggregate}^{(l)}\left(\text{Propagate}^{(l)}\left(\boldsymbol{h}_u^{(l-1)}, \boldsymbol{h}_v^{(l-1)}\right)\right). \tag{1}$$

*2) Candidate Set Generation:* After obtaining the node representations, we generate an initial candidate set by retrieving the set of nodes connected to the target node via meta-paths.

$$P_v = \{t \mid \exists \phi \in \Phi, \forall t \in \text{Path}_\phi(v)\}, \tag{2}$$

where $\Phi$ is the predefined set of meta-paths, and $\text{Path}_\phi(v)$ represents the set of all nodes on the meta-path.

*3) Computation of intermediate nodes in meta-paths:* To better capture semantic variations along meta-paths, we incorporate the representations of intermediate nodes into the semantic similarity computation. This design promotes both the quality and diversity of the selected examples. Specifically, when computing the meta-path similarity for node $v$ with respect to node $u$, we also account for the influence of intermediate nodes $m$ along the meta-path $\phi$. Their aggregated representation is obtained via average pooling:

$$\boldsymbol{h}_t'' = \frac{1}{|M|}\sum_{m \in M} \boldsymbol{h}_m', \tag{3}$$

where $M$ denotes the set of intermediate nodes (the number of nodes in the path instance excluding the endpoints).

This representation integrates the information from intermediate nodes along the meta-path, serving as a key input for subsequent semantic similarity calculation. It allows the model to jointly capture both direct relational signals and the broader indirect semantic context.

*4) Meta-path Semantic Similarity Calculation:* Following the aforementioned computations, the semantic similarity along meta-paths can be evaluated. Specifically, for a target node $v$, the meta-path semantic similarity aims to measure the similarity $S(v,u)$ between nodes connected via a set of meta-paths denoted by $\text{Path}_\phi(v,u)$.

Since multiple meta-paths may exist between $v$ and $u$, we employ a weighted fusion strategy. Concretely, for each meta-path $\phi \in \text{Path}_\phi(v,u)$, we compute the cosine similarity between the node representations. The contribution of each meta-path is then weighted by a factor $w_\phi$, which assigns higher weights to more frequent meta-paths in the network.

$$w_\phi = \frac{\text{count}_\phi(v,u)}{\sum_{u \in P_v} \text{count}_\phi(v,u)}, \tag{4}$$

where $\text{count}_\phi(v,u)$ is the number of instances of the meta-path $\phi$ connecting $v$ and $u$.

$S(v,u)$ is then computed as:

$$S(v,u) = \frac{\sum_{\phi \in \text{Path}_\phi(v,u)} w_\phi \cdot \cos\left(\mathbf{h}_v', \frac{\mathbf{h}_u' + \boldsymbol{h}_t''}{2}\right)}{\sum_{\phi \in P_v} w_\phi}, \tag{5}$$

where $\phi \in \text{Path}_\phi(v,u)$ denotes the meta-paths that exist between node $v$ and node $u$. $P_v$ denotes the meta-paths that contain node $v$.

TABLE I: Components of a Template for Converting ICL Examples into Text Prompts

| System Prompt | ICL Examples | Query Node | Answer |
|---|---|---|---|
| You are an intelligent academic literature classification system. I will provide an abstract of a target paper and several examples of similar papers with labels. Please infer the most likely label for the target paper based on semantic similarity, referring to the labels of the examples, and output it in the format '(Label:_,Probability:_)'. | Paper854 (Label:Database) semantic similarity: 0.98, and there exists a meta-path: paper-field-paper. Abstract: Query by output It has recently been asserted... | Abstract: CORDS: automatic generation of... Query: Please infer the label. | Label: Database, Probability: 88% |

*5) Select The Top K Nodes:* In this section, we select the top-$K$ nodes most relevant to the target node $v$ based on the meta-path semantic similarity $S(v,u)$, thereby forming the ICL examples set $\mathcal{D}_v$. To identify the most representative samples, we first define a scoring function $\Psi(v,u) = S(v,u) \cdot T(u)$, where $T(u)$ is a graph node centrality metric. The specific calculation is as follows:

$$\mathcal{D}_v = \underset{u \in P_v}{\mathrm{TopK}} \left( \Psi(v,u) \right). \tag{6}$$

### B. Generation of prompt text and confidence score

To render the $K$ selected example nodes suitable for LLMs, they are converted into prompt text, and calibrated confidence signals are generated. The core idea is to transform the graph structure information into semantic text descriptions by parsing the nodes and relationships in the graph. As shown in Table I, the confidence signal is calculated as:

$$\sigma(u,v) = \mathrm{LLM}(T(u,v)) \in [0,1], \tag{7}$$

where $T(u,v)$ represents the descriptions for nodes $u$ and $v$.

### C. Confidence score driven training

We design a loss function that leverages the calibrated confidence $\sigma(u,v)$ (obtained from LLMs) to optimize the HGNN. This aims to bridge the semantic gap between the structural information of the graph and the textual semantics.

$$r_{u,v} = \mathrm{Rank}\left( \sigma(u,v), \Psi(v,u) \right), \tag{8}$$

where $r_{u,v}$ denotes the importance ranking of the node in $D_v$ with respect to the target node $u$.

$$a_v = \frac{1}{r_{u,v}}, \tag{9}$$

where $a_v$ represents the weight of node $v$ in the loss function.

$$\mathcal{L}_{\mathrm{feedback}} = -\sum_{u \in N} \sum_{k \in K} \alpha_k \log \frac{e^{\sin(\mathbf{h}_u, \mathbf{h}_k)}}{\sum_{j=0}^{K} e^{\sin(\mathbf{h}_u, \mathbf{h}_j)}}. \tag{10}$$

This effectively incorporates the confidence from the LLMs into $\mathcal{L}_{\mathrm{feedback}}$ to guide the optimization of the HGNN.

$$\mathcal{L}_{\mathrm{clf}} = -\sum_{i \in [N]} \mathrm{CrossEntropy}(x_i, y_i), \tag{11}$$

where $N$ is the number of nodes, $x_i$ and $y_i$ denote the feature vector and the ground-truth label of node $i$, respectively. $\mathcal{L}_{\mathrm{clf}}$ is the node classification loss, which enables the model to capture both local and global structural patterns for learning effective node representations.

The total loss function is defined as follows:

$$L_{\mathrm{total}} = \lambda \mathcal{L}_{\mathrm{feedback}} + (1-\lambda) \mathcal{L}_{\mathrm{clf}}. \tag{12}$$

## V. EXPERIMENTS

### A. Experimental Settings

*1) Datasets:* We select three public heterogeneous graph datasets for experiments: ACM [23], DBLP [24], and IMDB [24]. The training set consists of 1%, 5%, and 10% of labeled nodes randomly selected from each category, while the remaining nodes are equally divided into a validation set and a test set. ACM: We adopt the meta-paths $\{PAP, PSP\}$, DBLP: We adopt the meta-paths $\{APA, APCPA, APTPA\}$, IMDB: We adopt the meta-paths $\{MAM, MDM\}$.

*2) Experimental Details:* Our framework employs label classification and probability inference for target nodes by employing LLMs. To select the most suitable LLM, we evaluate a series of models including SparkX1, Spark Lite, Spark Pro, Spark Max, Spark Ultra, Hunyuan TI. Based on the evaluation, we adopt Deepseek R1 [25]. The model's overall performance is evaluated using three metrics: Macro-F1, Micro-F1, and AUC.

*3) Baseline Methods:* To systematically verify the effectiveness of the proposed framework, we compare it with the following three categories of baseline methods: Homogeneous graph-based models: GCN [26]; Heterogeneous graph-based models: HAN [13], HGT [16], RGCN [27]; GNNs integrated with LLMs: GNN-RAG [3], AskGNN [11].

### B. Main Results

We evaluate HGICL on three heterogeneous graph datasets: ACM, DBLP, and IMDB. As shown in Table II, HGICL consistently outperforms all baseline methods, achieving state-of-the-art performance, which validates its ability to select high-quality ICL examples. Specifically, its variants (e.g., HAN+LLM, HGT+LLM) consistently outperform baseline methods such as AskGNN and GNN-RAG.

Effectiveness of Example Selection. Experiments demonstrate that correctly leveraging heterogeneous graph structural information for ICL examples screening can substantially enhance the reasoning performance of LLMs. Our framework,

TABLE II: Performance evaluation of node classification on the ACM, DBLP and IMDB datasets with 1%, 5% and 10% of the data used as training sets, where the optimal results are marked in bold.

| Method | ACM | | | DBLP | | | IMDB | | |
|---|---|---|---|---|---|---|---|---|---|
| | Mi-F1 | Ma-F1 | AUC | Mi-F1 | Ma-F1 | AUC | Mi-F1 | Ma-F1 | AUC |
| **Training Ratio: 1%** | | | | | | | | | |
| GCN | 75.52±2.32 | 75.84±1.12 | 85.31±0.17 | 75.85±1.15 | 75.52±0.24 | 85.86±0.16 | 40.71±0.12 | 38.64±0.15 | 43.25±0.10 |
| RGCN | 78.25±1.83 | 76.38±0.78 | 86.51±0.30 | 78.59±1.28 | 76.56±1.29 | 86.27±0.46 | 45.38±0.15 | 40.02±0.13 | 47.50±0.26 |
| HAN | 80.58±0.69 | 79.15±0.75 | 91.56±0.32 | 80.22±1.65 | 80.94±0.55 | 91.21±0.39 | 45.35±0.11 | 40.06±0.26 | 48.53±0.31 |
| HGT | 80.50±0.32 | 78.89±0.12 | 92.81±0.28 | 80.69±0.27 | 79.83±0.36 | 92.32±0.06 | 45.79±0.36 | 40.39±0.26 | 48.22±0.17 |
| GNN-RAG | 78.87±0.17 | 78.32±0.19 | 85.24±0.28 | 78.38±0.15 | 77.59±0.48 | 85.14±0.69 | 45.56±0.29 | 40.41±0.16 | 46.23±0.59 |
| AskGNN | 84.63±0.13 | 84.31±0.65 | 93.78±0.37 | 83.83±0.42 | 83.51±0.32 | 92.16±0.31 | 45.53±0.21 | 40.36±0.23 | 48.81±0.16 |
| GCN+LLM | 79.21±0.17 | 79.52±0.45 | 88.96±0.16 | 77.23±0.69 | 79.82±0.11 | 88.65±0.10 | 41.02±0.26 | 38.99±0.17 | 44.01±0.15 |
| RGCN+LLM | 83.94±1.04 | 83.24±0.95 | 93.57±0.33 | 83.59±1.25 | 83.68±0.24 | 93.95±0.10 | 45.87±0.11 | 40.53±0.12 | 48.32±0.20 |
| HAN+LLM | 86.23±0.16 | **86.72±0.46** | 94.88±0.68 | 86.25±0.74 | **86.93±0.72** | 94.92±0.35 | 45.98±0.23 | 40.47±0.15 | 49.42±0.31 |
| HGT+LLM | **86.51±0.92** | 83.68±0.18 | **95.84±0.44** | **86.80±1.96** | 84.76±0.82 | **95.08±0.61** | **46.03±0.22** | **41.61±0.10** | **49.74±0.24** |
| **Training Ratio: 5%** | | | | | | | | | |
| GCN | 85.45±0.28 | 84.95±0.15 | 93.94±0.09 | 85.17±0.16 | 85.08±0.29 | 93.98±0.47 | 46.52±0.21 | 44.67±0.12 | 49.83±0.08 |
| RGCN | 86.62±0.95 | 86.17±0.22 | 93.12±0.23 | 86.59±1.52 | 86.38±0.29 | 93.31±0.57 | 51.58±0.24 | 46.59±0.26 | 54.89±0.10 |
| HAN | 88.59±0.17 | 88.25±0.38 | 97.23±0.12 | 88.42±0.36 | 88.31±0.27 | 96.25±0.51 | 51.65±0.17 | 46.52±0.12 | 54.92±0.23 |
| HGT | 89.58±0.34 | 89.60±0.31 | 96.36±0.25 | 89.45±0.40 | 89.79±0.22 | 96.08±0.20 | 52.02±0.13 | 46.97±0.23 | 55.14±0.37 |
| GNN-RAG | 87.16±0.26 | 86.92±0.42 | 94.74±0.11 | 87.29±0.17 | 86.46±0.24 | 94.37±0.08 | 51.63±0.43 | 46.28±0.14 | 54.53±0.31 |
| AskGNN | 88.85±0.13 | 88.82±0.48 | 96.45±0.18 | 88.41±0.23 | 88.19±0.56 | 97.45±0.48 | 51.83±0.19 | 46.56±0.18 | 55.32±0.08 |
| GCN+LLM | 86.48±0.35 | 86.55±0.28 | 94.31±0.27 | 86.36±0.35 | 86.53±0.23 | 94.63±0.12 | 47.02±0.15 | 45.10±0.17 | 50.36±0.37 |
| RGCN+LLM | 87.38±0.25 | 86.95±0.14 | 95.36±0.29 | 87.78±0.74 | 87.15±0.30 | 95.45±0.22 | 52.09±0.11 | 46.77±0.24 | 55.21±0.13 |
| HAN+LLM | 89.63±0.48 | 89.67±0.61 | 97.28±0.24 | 89.32±0.64 | 90.01±0.16 | **97.69±0.16** | 52.04±0.10 | 46.82±0.17 | 55.16±0.21 |
| HGT+LLM | **90.56±0.30** | **90.55±0.39** | **97.33±0.35** | **90.68±0.59** | **91.77±0.29** | 97.63±0.36 | **52.49±0.10** | **47.65±0.12** | **55.63±0.23** |
| **Training Ratio: 10%** | | | | | | | | | |
| GCN | 87.58±0.25 | 87.31±0.18 | 95.35±0.21 | 87.52±0.48 | 87.23±0.39 | 95.47±0.10 | 48.46±0.33 | 45.39±0.08 | 51.75±0.12 |
| RGCN | 87.83±0.29 | 87.66±0.36 | 96.19±0.11 | 87.26±0.52 | 87.57±0.69 | 96.53±0.24 | 53.37±0.31 | 48.25±0.10 | 56.48±0.11 |
| HAN | 90.87±0.21 | 91.05±0.13 | 97.86±0.10 | 90.53±0.35 | 91.24±0.39 | 97.37±0.42 | 53.36±0.11 | 48.12±0.26 | 56.40±0.06 |
| HGT | 90.17±0.47 | 90.19±0.23 | 96.51±0.12 | 90.64±0.64 | 91.38±0.27 | 97.43±0.13 | 53.52±0.08 | 49.56±0.15 | 56.58±0.14 |
| GNN-RAG | 88.28±0.48 | 88.03±0.34 | 96.24±0.19 | 88.52±0.15 | 88.31±0.64 | 96.12±0.16 | 54.36±0.41 | 48.38±0.29 | 57.05±0.53 |
| AskGNN | 91.08±0.12 | 90.95±0.23 | 97.77±0.15 | 90.92±0.23 | 90.58±0.33 | 97.42±0.11 | 53.37±0.11 | 48.38±0.09 | 56.85±0.05 |
| GCN+LLM | 88.63±1.96 | 88.36±1.32 | 96.49±0.14 | 88.53±0.48 | 88.27±0.13 | 96.41±0.29 | 48.56±0.16 | 45.51±0.15 | 51.88±0.19 |
| RGCN+LLM | 89.07±0.32 | 89.37±0.22 | 97.57±0.19 | 89.28±0.32 | 88.82±0.64 | 96.97±0.12 | 53.84±0.10 | 49.53±0.12 | 56.83±0.18 |
| HAN+LLM | **91.16±0.29** | 91.29±0.49 | 97.68±0.11 | **91.06±0.31** | 91.71±0.15 | 98.33±0.38 | 53.93±0.13 | 49.46±0.08 | 57.06±0.27 |
| HGT+LLM | 91.03±0.35 | **91.54±0.23** | **98.61±0.26** | 90.95±0.76 | **91.88±0.18** | **98.59±0.23** | **54.93±0.17** | **49.86±0.15** | **57.39±0.12** |

which encodes structures via HGNNs, outperforms homogeneous methods by better capturing richer structural semantics. For instance, on ACM with 1% labeled data, HGICL (HAN+LLM) achieves an absolute gain of 5.6% in Micro-F1 over the pure HAN, whereas the GCN-based variant improves by only 3.6%. This validates that effectively leveraging heterogeneity is key to selecting high-quality examples.

Robustness and Adaptability. Experimental results show that HGICL obtains significant performance improvements over the baseline models even under low-label scenarios. This shows that the few-shot reasoning ability of LLMs effectively enhances applicability in low-resource settings. As the proportion of training data increases, the performance advantage of HGICL remains stable. Furthermore, it generalizes well across different heterogeneous datasets, confirming the framework's strong adaptability and practical viability.

### C. *Different versions of LLMs*

This section investigates how the scale and architecture of various LLMs influence the performance of the HGICL framework in integrating semantic and structural information from heterogeneous graphs.

TABLE III: Performance Comparison of Large Models.

| Metric | X1 | Lite | Pro | Max | Ultra | T1 | R1 |
|---|---|---|---|---|---|---|---|
| Mi-F1 | 85.75 | 86.09 | 86.12 | 86.21 | 86.04 | 86.17 | 86.23 |
| Ma-F1 | 85.95 | 86.52 | 86.29 | 86.44 | 86.32 | 86.63 | 86.72 |
| AUC | 94.88 | 94.31 | 94.05 | 94.67 | 94.29 | 94.54 | 94.88 |

We evaluate our framework on the ACM dataset using HAN combined with multiple LLM variants (including Spark X1, Spark Lite, Spark Pro, Spark Max, Spark Ultra, Hunyuan T1, and Deepseek R1). The experimental results reveal a strong correlation between the performance gain of the framework and the inherent capability of the underlying LLMs. Specifically, more powerful LLMs yield significantly higher performance improvements compared to their weaker counterparts, as illustrated in Table III. This phenomenon is likely due to the former having undergone extensive pre-training on

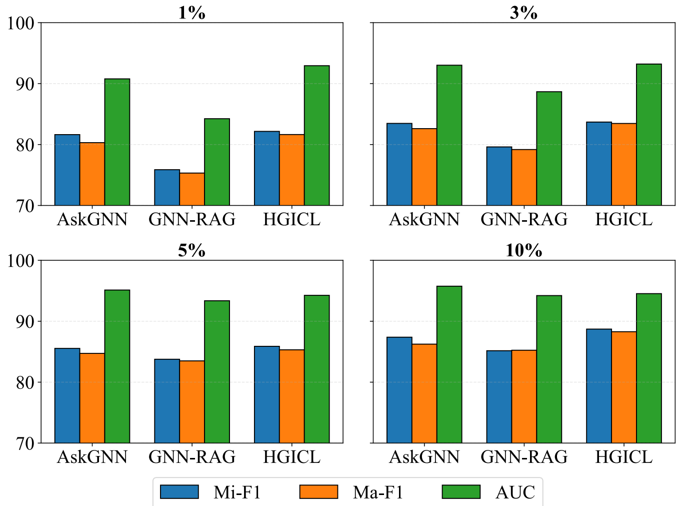


Fig. 3: Comparison of ICL Example Search Strategies.

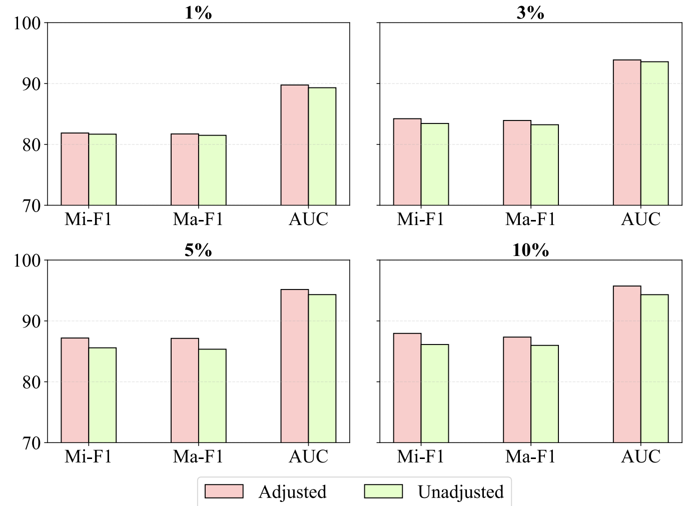


Fig. 4: ICL Example Weight Adjustment.

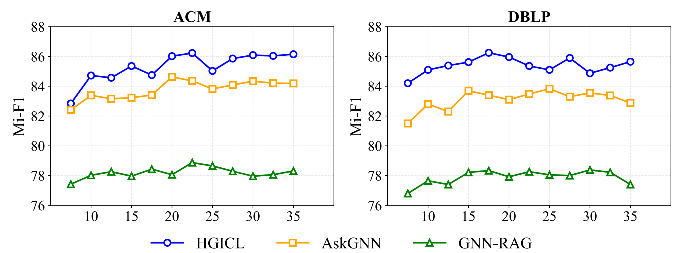


Fig. 5: Performance under different numbers of ICL examples. The left represents the ACM dataset, and the right represents the DBLP dataset.

related tasks, thereby acquiring stronger task adaptability and generalization capabilities, which are effectively utilized and enhanced within our framework.

## *D. Ablation study*

The proposed HGICL framework incorporates two core innovative modules: an ICL examples search method and a ICL examples weight adjustment mechanism. The following section provides a performance analysis of these two modules.

*1) Analysis of ICL Examples Search Methods:* To benchmark the ICL example selection capability of HGICL, we conduct a comparative study on the ACM dataset under the 1% training set setting. Using HGICL (with HAN+LLM) as our method, we compare its performance against two baseline selection strategies, GNN-RAG and AskGNN, to analyze how the search strategy affects final model performance.

As shown in Fig.3, the HGICL framework maintains a steady advantage over baseline methods across training set ratios from 1% to 10%. The key lies in its meta-path-guided search strategy, which jointly leverages the structural and semantic information of the heterogeneous graph. By retrieving examples that are both semantically relevant and topologically proximate, the strategy adheres to the graph's inherent topology. In heterogeneous datasets like ACM, this enables an effective synergy between structural and semantic information, driving HGICL's performance gains.

*2) Analysis of ICL Example Weight Adjustment:* Different ICL example nodes contribute unequally to the target node's representation learning, making it crucial to assign appropriate importance weights to them. To evaluate the effect of the ICL weight adjustment mechanism, we conduct a comparative study on the ACM dataset using the HGICL framework, which integrates HAN with DeepSeek. The weight adjustment module is assessed by comparing model performance with and without its inclusion. To isolate the mechanism's contribution, we replace the original meta-path-based search with a $K$-Hop search strategy, thereby reducing interference from the search process and emphasizing weight adjustment.

The experimental results are shown in Fig.4. When the training set ratio is 1%, the number of ICL examples available is limited, and the contribution disparity between nodes is not significant. Consequently, adjusting the weights has a relatively minor impact on overall model performance. As the proportion of the training set gradually increases, the number of ICL examples grows, and the differences in their representational contributions become more pronounced. The effect of the weight adjustment mechanism on the performance of the target node classification is progressively strengthened. This indicates that, with sufficient training data, distinguishing the importance of different ICL examples enables more effective aggregation of useful information, thereby improving the overall performance of the model.

## *E. Parameter experiment*

*1) Number of ICL examples:* To investigate the impact of the number of ICL examples on the performance of the HGICL framework, we systematically evaluate the model with varying numbers of ICL examples, as shown in the accompanying Fig.5. The experimental results show that the performance of various frameworks differs significantly under

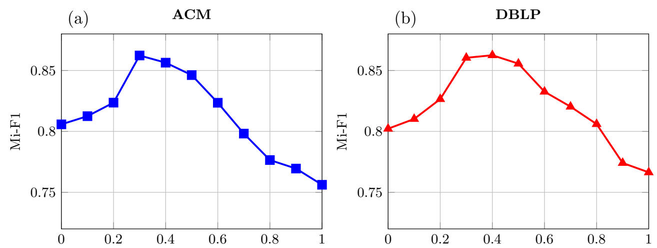


Fig. 6: Performance variation of hyperparameter $\lambda$.

different numbers of ICL examples; furthermore, for the same framework, the trend of performance change as the number of examples increases varies across different datasets. The study indicates that the optimal number of ICL examples is jointly influenced by the specific model framework and the characteristics of the dataset.

*2) Analysis of the hyperparameter $\lambda$:* To investigate the role of the weighting factor $\lambda$ in $L_{\text{total}}$, this section conducts a systematic hyperparameter analysis of $\lambda$ within the HGICL framework, which integrates HAN with DeepSeek. We use the ACM and DBLP datasets, across $\lambda$ values in $[0.0, 1.0]$, as shown in Fig.6. The results indicate that the model achieves optimal performance when $\lambda \approx 0.3$. This finding suggests that, under this configuration, an optimal balance is reached between the feedback-based loss term and the structure-based loss term, thereby maximizing performance.

## CONCLUSION

We propose HGICL, a novel framework that enhances node classification on heterogeneous graphs by deeply coupling HGNNs with LLMs. The framework introduces a meta-path-guided example search mechanism to construct high-quality ICL prompts, and leverages LLMs' semantic reasoning to generate feedback for optimizing HGNN parameters. HGICL effectively leverages rich semantic information in graphs and overcomes the limitations of existing $K$-hop retrieval and graph-text alignment methods. Extensive experiments demonstrate its superior accuracy and robustness over strong baselines. This work presents a new paradigm for collaborative LLM-HGNN optimization, laying the groundwork for potential extensions to link prediction, recommendation systems, and dynamic graph modeling.

## REFERENCES


[1] J. Li, Q. Dai, X. Duan, L. Chang, and S. Zhang, "MAGNN:A multi-view augmented graph neural network model for micro-video vlogger recommendation," in *37th IEEE International Conference on Tools with Artificial Intelligence, ICTAI 2025, Athens, Greece, November 3-5, 2025*, 2025, pp. 614–621.

[2] M. Guan, X. Cai, J. Shang, F. Hao, D. Liu, X. Jiao, and W. Ni, "HMSG: heterogeneous graph neural network based on metapath subgraph learning," *Knowl. Based Syst.*, vol. 279, p. 110930, 2023.

[3] C. Mavromatis and G. Karypis, "GNN-RAG: graph neural retrieval for efficient large language model reasoning on knowledge graphs," in *Findings of the Association for Computational Linguistics, ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, 2025, pp. 16 682–16 699.

[4] H. Lee, C. Yi, B. D. S. Aritra, and M. M. Islam, "Scalability matters: Overcoming challenges in instructglm with similarity-degree-based sampling," in *International Joint Conference on Neural Networks, IJCNN 2025, Rome, Italy, June 30 - July 5, 2025*, 2025, pp. 1–8.

[5] X. Li, D. Lian, Z. Lu, J. Bai, Z. Chen, and X. Wang, "Graphadapter: Tuning vision-language models with dual knowledge graph," in *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.

[6] S. Geng, W. Zhao, and A. M. Rush, "Great memory, shallow reasoning: Limits of knn-lms," in *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2025 - Volume 2: Short Papers, Albuquerque, New Mexico, April 29 - May 4, 2025*, 2025, pp. 471–482.

[7] D. Wang, Y. Zuo, F. Li, and J. Wu, "Llms as zero-shot graph learners: Alignment of GNN representations with LLM token embeddings," in *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024.

[8] J. Tang, Y. Yang, W. Wei, L. Shi, L. Xia, D. Yin, and C. Huang, "Higpt: Heterogeneous graph language model," in *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2024, Barcelona, Spain, August 25-29, 2024*, 2024, pp. 2842–2853.

[9] R. Wang, M. Yang, and Y. Shen, "Bridging molecular graphs and large language models," in *AAAI-25, Sponsored by the Association for the Advancement of Artificial Intelligence, February 25 - March 4, 2025, Philadelphia, PA, USA*, 2025, pp. 21 234–21 242.

[10] J. Tang, Y. Yang, W. Wei, L. Shi, L. Su, S. Cheng, D. Yin, and C. Huang, "Graphgpt: Graph instruction tuning for large language models," in *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2024, Washington DC, USA, July 14-18, 2024*, 2024, pp. 491–500.

[11] Z. Hu, Y. Li, Z. Chen, J. Wang, H. Liu, K. Lee, and K. Ding, "Let's ask GNN: Empowering large language model for graph in-context learning," in *Findings of the Association for Computational Linguistics: EMNLP 2024*, Nov. 2024, pp. 1396–1409.

[12] J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun, "Graph neural networks: A review of methods and applications," *AI Open*, pp. 57–81, 2020.

[13] X. Wang, H. Ji, C. Shi, B. Wang, Y. Ye, P. Cui, and P. S. Yu, "Heterogeneous graph attention network," in *The World Wide Web Conference, WWW 2019*, 2019, pp. 2022–2032.

[14] H. Luo, H. E, Y. Yang, Y. Guo, M. Sun, T. Yao, Z. Tang, K. Wan, M. Song, and W. Lin, "HAHE: hierarchical attention for hyper-relational knowledge graphs in global and local level," in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, 2023, pp. 8095–8107.

[15] H. Hong, H. Guo, Y. Lin, X. Yang, Z. Li, and J. Ye, "An attention-based graph neural network for heterogeneous structural learning," in *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, 2020, pp. 4132–4139.

[16] Z. Hu, Y. Dong, K. Wang, and Y. Sun, "Heterogeneous graph transformer," in *WWW '20: The Web Conference 2020*, 2020, pp. 2704–2710.

[17] C. Zhang, D. Song, C. Huang, A. Swami, and N. V. Chawla, "Heterogeneous graph neural network," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019*, 2019, pp. 793–803.

[18] J. Liu, Z. Huang, C. Wang, X. Huang, C. Zhai, and E. Chen, "What makes in-context learning effective for mathematical reasoning," in *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*. OpenReview.net, 2025.

[19] Z. Jiang, F. F. Xu, L. Gao, Z. Sun, Q. Liu, J. Dwivedi-Yu, Y. Yang, J. Callan, and G. Neubig, "Active retrieval augmented generation," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, 2023, pp. 7969–7992.

[20] T. Nguyen and E. Wong, “In-context example selection with influences,” *CoRR*, vol. abs/2302.11042, 2023.

[21] A. Asai, Z. Wu, Y. Wang, A. Sil, and H. Hajishirzi, “Self-rag: Learning to retrieve, generate, and critique through self-reflection,” in *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*, 2024.

[22] Z. Wu, Y. Wang, J. Ye, Z. Wu, J. Feng, J. Xu, and Y. Qiao, “Openicl: An open-source framework for in-context learning,” in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics: System Demonstrations, ACL 2023, Toronto, Canada, July 10-12, 2023*, 2023, pp. 489–498.

[23] J. Zhao, X. Wang, C. Shi, Z. Liu, and Y. Ye, “Network schema preserving heterogeneous information network embedding,” in *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*. ijcai.org, 2020, pp. 1366–1372.

[24] X. Fu, J. Zhang, Z. Meng, and I. King, “MAGNN: metapath aggregated graph neural network for heterogeneous graph embedding,” in *WWW '20: The Web Conference 2020, Taipei, Taiwan, April 20-24, 2020*, 2020, pp. 2331–2341.

[25] DeepSeek-AI, “Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning,” *CoRR*, vol. abs/2501.12948, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2501.12948

[26] T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” in *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

[27] M. S. Schlichtkrull, T. N. Kipf, P. Bloem, R. van den Berg, I. Titov, and M. Welling, “Modeling relational data with graph convolutional networks,” in *The Semantic Web - 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3-7, 2018, Proceedings*. Springer, 2018, pp. 593–607.