Judge and Improve: Towards a Better Reasoning of Knowledge Graphs with Large Language Models

Mo Zhiqiang[1], **YangHua** [1*], **Jiahui Li**[1,2†], **Yuan Liu**[1]
Shawn Wong[1], **Jianmin Huang**[1]
[1]Ant Group [2]Zhejiang University
mozhiqiang.mzq@antgroup.com, kaoen.yh@antgroup.com, jiahuil@zju.edu.cn
zhongling.ly@antgroup.com , shawnwong.ws@antgroup.com, caiqiong@antgroup.com

Abstract

Graph Neural Networks (GNNs) have shown immense potential in improving the performance of large-scale models by effectively incorporating structured relational information. However, current approaches face two key challenges: (1) achieving robust semantic alignment between graph representations and large models, and (2) ensuring interpretability in the generated outputs. To address these challenges, we propose **ExGLM** (**Ex**plainable **G**raph **L**anguage **M**odel), a novel training framework designed to seamlessly integrate graph and language modalities while enhancing transparency. Our framework introduces two core components: (1) a **graph-language synergistic alignment module**, which aligns graph structures with language model to ensure semantic consistency across modalities; and (2) a **Judge-and-Improve paradigm**, which allows the language model to iteratively evaluate, refine, and prioritize responses with higher interpretability, thereby improving both performance and transparency. Extensive experiments conducted on three benchmark datasets—ogbn-arxiv, Cora, and PubMed—demonstrate that ExGLM not only surpasses existing methods in efficiency but also generates outputs that are significantly more interpretable, effectively addressing the primary limitations of current approaches.

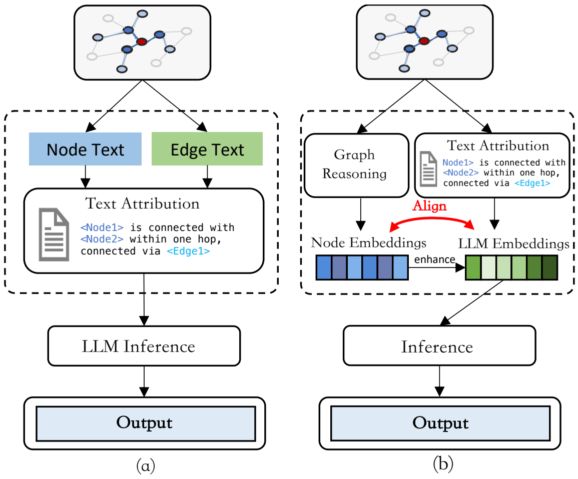

Figure 1: Overview of two mainstream methods (a) textualizing graph and inference via LLM and (b) aligning the semantic representation of LLMs and GNNs.

1 Introduction

Large Language Models (LLMs) have demonstrated remarkable success across various natural language processing tasks, including dialogue generation (Aboussalah and Ed-dib, 2025), machine translation (Zhu et al., 2024), question answering (Zhang et al., 2024), and text summarization (Zhang et al., 2025). These models exhibit an impressive capacity for understanding and generating human-like text. However, LLMs face inherent limitations in effectively modeling structured knowledge, such as graphs, which are essential for capturing complex relationships and dependencies in diverse real-world domains like social networks, biological systems, and knowledge graphs. To address these limitations, recent research has explored the integration of GNNs (Kipf and Welling, 2017; Hamilton et al., 2018; Veličković et al., 2018) with LLMs (OpenAI et al., 2024; Yang et al., 2024; DeepSeek-AI et al., 2025), leveraging GNNs' strengths in modeling structured information alongside LLMs' powerful language capabilities, creating opportunities for enhanced performance in graph-related tasks.

Current approaches (Yang et al., 2021; Zhao et al., 2023; Xue et al., 2024) to combining GNNs and LLMs can be broadly classified into two categories. The first category involves **textualizing graph structures** and feeding them into LLMs (Figure 1(a)). For example, some methods (Zhao et al., 2023; Wang et al., 2024; Chen et al., 2024; Wu et al., 2025) describe nodes and their relationships using natural language templates to generate textual representations of subgraphs.

*Authors contributed equally.
†Corresponding Author.

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 5303–5320
November 4-9, 2025

Other approaches (Ye et al., 2024; Tang et al., 2024) employ special tokens to represent nodes and edges, effectively converting graph structures into sequences compatible with LLM processing. However, these methods have notable pitfalls: the textualization process can result in the loss of structural information, and the sequential representations may fail to fully capture the intricate relationships within the graph. Additionally, these methods face scalability challenges due to the token length constraints of LLMs, making them unsuitable for handling large graphs with extensive neighborhood information.

The second category of approaches (Chai et al., 2023; Tang et al., 2024; Liu et al., 2024) focuses on **aligning the representation spaces** of GNNs and LLMs in the semantic domain (Figure 1(b)). For instance, certain methods (Xia et al., 2024; Huang et al., 2023, 2024; Guo et al., 2025) project GNN-generated node embeddings into the embedding space of LLMs to achieve semantic consistency. Other techniques, such as those employing attention mechanisms (Ying et al., 2021; Kuang et al., 2022), integrate graph structure information directly into the language model's representations. While these approaches improve the integration of graph and language modalities, challenges remain. The alignment process may not be optimal, leading to performance bottlenecks in tasks requiring a precise understanding of graph structures and language semantics. Moreover, such methods often suffer from a lack of interpretability, making it difficult to elucidate how the model leverages graph information to make decisions or derive outputs.

To address the limitations of existing approaches, we propose **ExGLM (Explainable Graph Language Model)**, a novel framework designed to effectively and interpretably integrate graph structures with LLMs. Our framework introduces a **graph-language synergistic alignment module** to achieve semantic consistency between graph structures and LLM outputs, while also maintaining interpretability. Specifically, we assign a textual attribute to each node in the graph, describing its adjacent relationships, with different nodes represented by special tokens. We then perform reasoning using the LLM and enhance its representation by incorporating the graph representation into the hidden state. To further improve interpretability, we propose a **Judge-and-Improve paradigm** where the LLM evaluates and selects responses with better interpretability. These optimized responses are subsequently used to refine the GNN-LLM model.

Our main contribution can be summarized as follows:

- We propose a novel graph-language synergistic alignment module that effectively bridges the gap between graph-structured data and LLM outputs, ensuring robust semantic consistency across modalities.
- We propose a Judge-and-Improve paradigm, enabling the model to iteratively evaluate and refine its responses for enhanced interpretability and generation quality, thereby improving both performance and transparency.
- We conduct comprehensive experiments on multiple datasets, demonstrating the superior performance and effectiveness of our approach compared to existing methods.

2 Related Work

2.1 Graph-Large Language Models

LLMs (OpenAI et al., 2024; Yang et al., 2024; DeepSeek-AI et al., 2025) achieve state-of-the-art performance on various natural language tasks, however, it lacks explicit mechanisms to effectively model structured information, such as graphs. To address this limitation, recent studies (Shu et al., 2024; Tang et al., 2024) have explored ways to integrate the benefits of GNNs into LLM-based frameworks. For instance, Zhang et al. (2020) adapts the self-attention mechanism of BERT (Devlin et al., 2019) to capture the relational structure of nodes and edges within a graph. However, its performance is highly dependent on the presence and quality of node features, which may limit its applicability when such features are sparse or noisy. InstructGLM (Ye et al., 2024) leverages the natural language modeling capabilities of LLMs to describe multi-scale geometric structures within graphs, thereby improving representation and analysis of graph data. Nonetheless, it suffers from token-length limitations, making it challenging to process large graphs with extensive neighbor information. Jin et al. (2024) propose a framework named Graph-COT that enhances LLMs by encouraging them to perform iterative reasoning over graph structures. However, fine-tuning LLMs within this framework remains challenging, and potential misalignment between the graph structure and the text attribution can lead to inaccuracies. Another recent work, PromptGFM (Zhu et al., 2025), explicitly prompts LLMs to mimic the workflow of GNNs within the text space, achieving naturally alignment between graph representations and textual modeling. While this approach improves graph-text integration, it struggles to differentiate between graphs with similar semantic structures. In this work, we propose a novel graph-language synergistic alignment module that aligns GNNs and LLMs at both the text attribution and semantic representation levels. This alignment enables seamless and effective incorporation of the strengths of GNNs and LLMs.

2.2 Self-Judge-and-Improve Paradigm

The self-Judge-and-Improve paradigm highlights the capacity of LLMs to autonomously evaluate and enhance their own performance and capabilities, thereby reducing dependence on external supervision. This approach enables models to internally refine their understanding and outputs. For instance, Self-Insturct (Wang et al., 2022) embodies this paradigm through a two-step process to improve instruction-following abilities. First, the model generates sample outputs and evaluates them using its internal mechanisms, filtering out suboptimal results. These filtered samples are then leveraged to fine-tune the model. Similarly, Self-Refine (Madaan et al., 2023) demonstrates how LLMs can provide feedback on their own generations and use this feedback to optimize their outputs iteratively. Expanding on this concept, Yuan et al. (2025) introduced self-rewarding language models, wherein LLMs assign self-generated rewards to their outputs. Preference pairs selected based on these rewards are then utilized to optimize the models using DPO (Rafailov et al., 2023). While these approaches effectively minimize external intervention, the quality of self-judgment is inherently constrained by the performance of the LLM. To address this limitation, we propose the Judge-and-Improve paradigm, which incorporates a superior language model to evaluate the generated outputs. By introducing an external judgment mechanism, our approach enhances the reliability and accuracy of evaluations, enabling more effective refinement of the model's outputs.

3 Method

The training framework of our method, illustrated in Figure 2, is composed of two key modules: (1) Graph-language synergistic alignment module and (2) Judge-and-Improve paradigm. The graph-language synergistic alignment module ensures effective integration between the GNNs and the LLMs by aligning textual attributes and semantic representations, thereby maintaining consistency across modalities. The Judge-and-Improve paradigm operates in two stages: first, it generates and selects accurate and explainable results through prompting, creating a supervised fine-tuning (SFT) and preference dataset; second, it uses these two datasets to optimize the model, progressively enhancing both performance and interpretability.

3.1 Problem Setup

Graph structure. Generally, a graph can be formally defined as $G = (V, E, X)$, where $V = \{v_1, v_2, \ldots, v_n\}$ represents the set of nodes, $E \subseteq V \times V$ represents the set of edges, encoding pairwise relationships between nodes, and $X \in \mathbb{R}^{n\times d}$ is the node feature matrix. Each $x_i \in \mathbb{R}^d$ corresponds to the feature vector of node v_i, where d represents the dimensionality of the node features.

Node classification with LLM. Consider a node classification problem over a graph $G = (V, E, X)$, where the goal is to assign one of k discrete class labels to each node. Let $Y = \{1, 2, \ldots, k\}$ denote the set of class labels. The training data consists of labeled examples (x_i, y_i), where $x_i \in \mathbb{R}^d$ represents the graph feature vector of node $v_i \in V$, and $y_i \in Y$ is the corresponding class label. The objective is to learn a classifier $f : X \xrightarrow{G} Y$, such that $f(x_i)$ accurately predicts the class label y_i for each node. In this work, we first derive textual attributions T_v of each node v, capturing its structural and feature information in a textual format. We then leverage both the graph structure and an LLM to perform reasoning. Consequently, the classification objective is refined to learning a classifier $f : T \xrightarrow{G-LLM} Y$ where T represents the textual descriptions derived from the graph's structural and feature information. This approach integrates the representational strengths of both GNNs and LLMs, enabling a more semantically rich and powerful node classification framework.

Classification with interpretability. In real-world applications where interpretability is paramount, it is essential for classification models to not only make accurate decisions but also provide clear explanations for those decisions. Therefore, our ultimate goal is to train a classifier that not only performs classification tasks but also generates an-

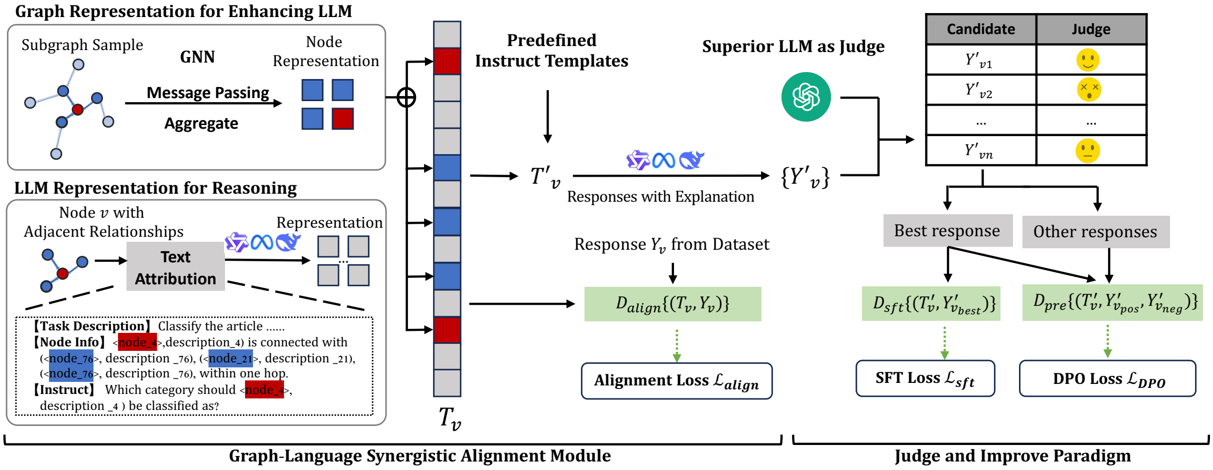

Figure 2: The training framework of ExGLM.

【Task introduction】

Classify the article according to its topic into one of the following categories: 【Label Set】. Node represents academic paper with a specific topic, link represents a citation between the two papers. Pay attention to the multi-hop link relationship between the nodes.

【Node Info】

Example1: (<$node_v$>, $description_v$) is connected with 【k-hop neighbor nodes】(<$node_{n_1}$>,$description_{n_1}$), ..., (<$node_{n_i}$>,$description_{n_i}$) within k hops.

Example2: (<$node_v$>, $description_v$) is connected with 【k-hop neighbor nodes】(<$node_{n_1}$>,$description_{n_1}$), ..., (<$node_{n_i}$>,$description_{n_i}$) within k hops through 【nodes under the path，if k>1】, respectively.

【Instruct】

Example1: Which category should (<$node_v$>, $description_v$) be classified as?

Example2: Should (<$node_v$>, $title_v$) be classified as 【random label】?

Figure 3: Example of text attribution.

alytical content to elucidate its decision-making process. This can be represented as a function $f : X \xrightarrow{G} Y, Analysis$. where $Analysis$ provides the explanatory content.

3.2 Graph-Language Synergistic Alignment Module

To effectively leverage both structural information from graphs and textual attributes from LLMs, we integrate GNNs and LLMs to obtain node representations. To bridge the gap between these two modalities, we propose a graph-language synergistic alignment module. This module consists of two core components:(1) Textual attribution of adjacency relationships, which captures the textual representation of graph structures. (2) Incorporating graph semantic information into textual attribution, which enriches textual descriptions with graph-based semantics. We detail these components below.

Textual attribution of adjacency relationships. We derive the textual attribution of each node through a two-step process: (1) Subgraph sampling for node information. In the context of large graphs, subgraph sampling is crucial to mitigate computational complexity and enable scalable processing. In this work, we adopt a k-hop sampling strategy extract localized subgraphs centered around each node. Specifically, for a central node v, we sample its neighbors within k hops, and represent it as $\mathcal{N}_v^{(n)}$. v and $\mathcal{N}_v^{(n)}$ are then further utilized to derive the textual attribution of adjacency relationships.

(2) Node description via text. For each central node v, we construct multiple text descriptions. Each description is represented as a tuple T_v: (`task introduction`, `<node info>`, `instruction`). Here, `task introduction` provides a brief overview of the task, `<node info>` contains textual descriptions of the central node's neighbors sampled from i-hop neighborhoods (where $0 < i \leq k$, selected randomly), and `instruction` specifies a task-related question tailored to the node and its neighborhood information. Specifically, in the `<node info>` part, each individual graph node is represented by a special token $node_i$ with its brief text descriptions $description_i$. Detailed examples are provided in Figure 3. This approach enables the attribution of each graph node to be naturally expressed in textual form, bridging the structural information of graphs with the representational capabilities of LLMs.

Incorporating graph semantic information into textual attribution. Since the aforementioned special tokens for each node cannot be effectively represented by LLMs alone, we integrate them with representations derived from GNNs. To obtain the representations of the GNNs, we adopt GraphSAGE, which primarily consists of three steps: neighbor sampling, message aggregation, and node updating. For each node v, we sample its m-hop neighbors and denote them as $\mathcal{N}^{(m)}(v)$. In this paper, we set $m = 2$ in all scenarios. For the aggregation representation is calculated via mean-pooling of neighborhood features:

$$h_{\text{agg}}^{(l)} = \frac{1}{|\mathcal{N}^{(m)}(v)| + 1} \left(h_v^{(l-1)} + \sum_{u \in \mathcal{N}^{(m)}(v)} h_u^{(l-1)} \right), \tag{1}$$

in which $h_v^{(l-1)}$ denotes the representation of node v at layer $(l - 1)$ Then each node is updated via Nonlinear projection with learnable parameters which is denoted as:

$$h_v^{(l)} = \sigma \left(W^{(l)} \cdot h_{\text{agg}}^{(l)} \right), \tag{2}$$

$W^{(l)}$ is the layer-specific weight matrix, $sigma(\cdot)$ denotes the ReLU activation function, $d^{(l)}$ is the dimensionality at layer l.

After obtaining $h_v^{(l)}$, we directly add it to the LLM's hidden states corresponding to the special token v which is shown in Figure 2 left.

To achieve better alignment between the LLM and GNN in the semantic space, we perform joint training. First, we construct a dataset First $\mathcal{D}_{align} = \{(T_v, Y_v)\}, v \in V$, where $Y_v = [c_1, c_2, \ldots, c_n]$ denotes the label sequence associated with node v. The alignment is achieved using the Negative Log-Likelihood (NLL) loss function:

$$\mathcal{L}_{\text{align}} = -\sum_{t=1}^{n} \log P\left(c_t \mid c_{<t}, T_v; \theta_{\text{LLM}}, \theta_{\text{GNN}}\right), \tag{3}$$

where θ_{LLM} denotes the parameters of the LLM, and θ_{GNN} denotes the parameters of the GNN encoder.

3.3 Judge and Improve

Building upon the dual-projection constrained mechanism, we achieve a deep collaboration between Graph Neural Networks (GNNs) and Large Language Models (LLMs). Beyond mere decision-making, providing reasonable and trustworthy analyses significantly enhances the interpretability of these decisions, which is crucial for various real-world applications. To ensure the interpretability of model decisions, we require the LLMs to not only generate accurate answers but also provide comprehensive explanations for their decisions.

However, we have observed that the explanations generated by the LLMs are often suboptimal, indicating a need for further training. Considering the challenge of obtaining training data with annotated explanations, we adopt a Judge-and-Improve paradigm (Yuan et al., 2025) to enhance the interpretability of the LLMs. Specifically, our approach involves the following steps:

(1) Generating multiple responses: For a given input, the LLM generates multiple responses, each accompanied by an explanation.

(2) Judging quality: Superior LLM acts as a judge to evaluate these responses, selecting the one that is not only accurate but also provides a reasonable explanation.

(3) Optimizing through annotated data: The generated responses and explanations are then used to optimized the LLM, thereby improving the quality of its explanations.

Responses generation. As illustrated in Figure 3, we construct the text attributes of node v using a tuple T_v `(task introduction, <node info>, instruction)`. To assemble a high-quality and diverse dataset, we first replace the instruction with several predefined instruction templates that convey the same intent, denoted as T'_v. Subsequently, we generate a response Y'_v or each T'_v a set $\{(T'_v, Y'_v)\}$, for all $v \in V$.

Judging quality. We require the superior LLM such as GPT-4 to evaluate the generated responses based on three criteria: correctness of the response, adherence to instructions, and reasonableness of the explanation. If none of the samples meet all three criteria, we repeat the response generation procedure. Ultimately, for each T'_v, we obtain a set of candidates $(Y'_{v_{\text{best}}}, Y'_{v_1}, \ldots, Y'_{v_{n-1}})$, where n denotes the number of generated responses.

Optimizing through annotated data. Building upon the generated responses, we construct a supervised fine-tuning (SFT) dataset: $\mathcal{D}_{sft} = \{(T'_v, Y'_{v_{\text{best}}}\}, v \in V$, which aims to teach the LLM to learn the pattern of the best response. The loss is computed as follows:

$$\mathcal{L}_{\text{sft}} = -\sum_{t=1}^{n} \log P\left(c'_t \mid c'_{<t}, T'_v; \theta_{\text{LLM}}, \theta_{\text{GNN}}\right), \tag{4}$$

where $Y'_{v_{\text{best}}} = [c'_{\text{best}_1}, \ldots, c'_{\text{best}_n}]$ and best_i denotes the i-th token of the best response.

This procedure is trained alongside the alignment process, and the overall loss becomes:

$$\mathcal{L}_{\text{total}} = \lambda_1 \cdot \mathcal{L}_{\text{align}} + \lambda_2 \cdot \mathcal{L}_{\text{sft}} \tag{5}$$

where λ_1 and λ_2 denotes the hyperparameters.

Moreover, to enable the LLM to distinguish between good and bad responses, we construct a preference dataset: $\mathcal{D}_{\text{pre}} = \{(T'_v, Y'_{v_{\text{pos}}}, Y'_{v_{\text{neg}}})\}, v \in V$ where $Y'_{v_{\text{pos}}}$ denotes the best response (positive example), and $Y'_{v_{\text{neg}}}$ denotes any other response (negative example) corresponding to the same input T'_v. This dataset pairs each best response with its corresponding non-optimal responses for every node $v \in V$.

We then utilize the preference dataset to optimize the LLM using DPO loss:

$$\mathcal{L}_{\text{DPO}} = -\mathbb{E}_{(T'_v, Y'_{v_{\text{pos}}}, Y'_{v_{\text{neg}}}) \sim \mathcal{D}_{\text{pre}}} \Bigg[\log \sigma \Bigg(\beta \log \frac{\pi_\theta(Y'_{v_{\text{pos}}} | T'_v)}{\pi_{\text{ref}}(Y'_{v_{\text{pos}}} | T'_v)} - \beta \log \frac{\pi_\theta((Y'_{v_{\text{neg}}} | T'_v)}{\pi_{\text{ref}}((Y'_{v_{\text{neg}}} | T'_v)} \Bigg) \Bigg], \tag{6}$$

where π_{ref} denotes the reference model, which we adopt as the model before DPO training, and β is a hyperparameter.

4 Experiments

4.1 Experimental Setup

Datasets. We employ three graph datasets of varying scales: Cora (Yang et al., 2016), PubMed (Namata et al., 2012), and ogbn-arxiv (Hu et al., 2020). For consistency, we adopt the dataset partitioning strategy introduced by (Ye et al., 2024). The key statistics of these datasets are summarized in Table 1.

Metrics. Following (Namata et al., 2012), we use accuracy as the primary metric to evaluate node classification performance. To assess the interpretability of the generated outputs, we utilize GPT-4 (Brown et al., 2020) as an automated evaluator. Additionally, to ensure a more robust and reliable assessment of interpretability, we complement this with a questionnaire-based survey, which provides valuable human-centered insights (Sperrle et al., 2021).

Table 1: Statistics of the graph datasets used in our experiment.

Dataset	#Nodes	#Edges
Cora	2,708	5,429
Pubmed	19,717	44,338
ogbn-arxiv	169,343	1,166,243

Baselines. We compare the proposed method against three categories of existing approaches: (1) GNN-based models, including GCN (Kipf and Welling, 2017),GraphSAGE(Hamilton et al., 2018),GAT(Veličković et al., 2018),TransGAT, (Louis et al., 2020) etc.; (2) Transformer-based models, such as Graphormer(Ying et al., 2021),GT(Dwivedi and Bresson, 2021) and CoarFormer (Kuang et al., 2022); and (3) LLM-based models, such as InstructGLM (Ye et al., 2024).

Implementations. In our implementation, we adopt LLaMA-7B (Touvron et al., 2023) and LLaMA3.1-8B-Instruct (Touvron et al., 2023) as the LLM backbones, and employ a two-layer GraphSAGE network to learn graph representations. The output dimension of the final GraphSAGE layer is aligned with the hidden size of the LLM backbones to enable seamless integration. When LLaMA3.1-8B-Instruct is used as the backbone for InstructGLM, only minimal input adjustments are made to meet its format requirements—for example, embedding dialogue templates and mapping node IDs to token IDs. During the data generation and annotation stage, we employ Qwen2.5-7B-Instruct (Bai et al., 2023) to produce decision-analysis content for node classification tasks. To control output randomness, we vary the temperature parameter across multiple values, thereby generating outputs of different qualities. For each node, we generate five outputs to ensure diversity. Additionally, we fix the node's output label and prompt Qwen2.5-7B-Instruct to produce responses conditioned on the correct label, from which we also sample five outputs. To improve annotation quality, we further utilize the more capable Qwen2.5-72B-Instruct model (Bai et al., 2023) as a "super annotator", automatically refining and validating the generated analyses. Model training is conducted on 8 A100 GPUs, with all experiments run for 1–3 epochs.

4.2 Performance Comparison

Tables 2 compares the performance of various models on the Cora and PubMed datasets, showcasing the effectiveness of different approaches.

Accuracy on Cora dataset. Among the GNN-based methods, ACM-GCN+ achieves the best accuracy on the Cora dataset (89.75%). Transformers-based methods, on the other hand, generally exhibit relatively lower performance. Notably, the hybrid InstructGLM approach, which combines GNN and LLM techniques, is the most comparable to our method, achieving competitive performance with an accuracy of 87.08% on Cora. In contrast, our proposed method achieves 88.8% accuracy, surpassing all existing Transformers-based and GNN-LLM-based approaches.

Accuracy on PubMed dataset. On the PubMed dataset, InstructGLM sets a strong baseline with the best performance among prior methods. Our method outperforms all baselines, achieving a new state-of-the-art accuracy of 94.6%. These results highlight the superiority of our training framework.

Table 2: Accuracy on Cora and PubMed datasets.

Method	Type	Cora (%)	PubMed (%)
MixHop	GNN	75.65	90.04
GAT	GNN	76.70	83.28
Geom-GCN	GNN	85.27	90.05
SGC-v2	GNN	85.48	85.36
GraphSAGE	GNN	86.58	86.85
GCN	GNN	87.78	88.90
BernNet	GNN	88.52	88.48
FAGCN	GNN	88.85	89.98
GCNII	GNN	88.93	89.80
RevGAT	GNN	89.11	88.50
Snowball-V3	GNN	89.59	91.44
ACM-GCN+	GNN	**89.75**	90.96
Graphormer	Transformers	80.41	88.24
GT	Transformers	86.42	88.75
CoarFormer	Transformers	88.69	89.75
InstructGLM	GNN-LLM	87.08	93.84
ExGLM	GNN-LLM	88.8	**94.6**

Accuracy on Ogbn-Arxiv dataset. Table 3 summarizes the performance of various models on the Ogbn-Arxiv dataset. Among traditional GNN-based approaches, DRGAT achieves the highest accuracy at 76.11%, outperforming simpler architectures such as GraphSAGE (74.35%) and GAT (74.15%), which exhibit moderate performance. Notably, methods that integrate large language models (LLMs) with GNN frameworks surpass all conventional GNN models, demonstrating the potential of combining structured graph data with the rich semantic understanding of LLMs. For instance, InstructGLM achieves an accuracy of 76.42%, further highlighting the effectiveness of this hybrid approach. Our proposed method achieves the highest overall accuracy at 77.4%, setting a new state-of-the-art performance on this task. This result underscores the advantages of our framework in effectively leveraging both graph structures and textual information to improve predictive performance.

Table 3: Accuracy on Ogbn-Arxiv dataset.

Method	Type	Accuracy (%)
GAT	GNN	74.15
GraphSAGE	GNN	74.35
GCN	GNN	73.29
AGDN	GNN	76.02
RvGAT	GNN	75.90
DRGAT	GNN	76.11
InstructGLM	GNN-LLM	76.42
ExGLM	GNN-LLM	**77.4**

Table 4: Performance comparison with different LLMs.

Method	Cora (%)	PubMed (%)
InstructGLM (LLaMA)	87.08	93.84
Ours (LLaMA)	**88.8**	**94.6**
InstructGLM (LLaMA3)	88.01	94.17
ExGLM (LLaMA3)	**89.30**	**94.42**

Accuracy with Other LLMs. Table 4 compares the performance of our proposed method against InstructGLM on the Cora and PubMed datasets, utilizing two different LLM backbones: LLaMA and LLaMA3. Two key observations can be drawn from the results: (1) Our method consistently outperforms the baseline InstructGLM across both datasets, regardless of the underlying LLM backbone. This demonstrates the robustness and effectiveness of our approach. (2) The use of a more advanced backbone does not always guarantee a significant performance improvement. While both methods perform slightly better with LLaMA3 compared to LLaMA, the relative gain is marginal. Notably, when applying LLaMA3, the performance on the PubMed dataset drops slightly from 94.6%

to 94.42%. This indicates that the integration mechanism and model design play a more critical role than simply using a stronger LLM.

GPT-4 evaluation. To evaluate the different methods more comprehensively, we use GPT-4 as a proxy for human judgment. Specifically, we task GPT-4 with performing pairwise evaluations to select the better response based on three key criteria: correctness of the response, adherence to instructions, and reasonableness of the explanation. The evaluation results, presented in Table 5, demonstrate that our method outperforms InstructGLM on both datasets. The low performance of InstructGLM may be attributed to its overfitting on the dataset, which can lead to less fluent or less adaptable language generation. Additionally, the integration of DPO enhances overall performance on both datasets.

Table 5: GPT-4 evaluation results with LLaMA3 as base model.

ExGLM *vs.*	**Dataset**	**Win (%)**	**Lose (%)**
InstructGLM	Cora	81.61	0
	PubMed	92.45	0
ExGLM (w/o DPO)	Cora	8.46	6.80
	PubMed	5.81	4.42

Link Prediction. To evaluate the generality of our approach, we further extended it to the link prediction task. For this purpose, we modified the Cora and CiteSeer datasets by removing the nodes originally included in the node classification test set, thereby constructing new graphs for model training. The edges connected to these removed nodes were then used as prediction targets. In all experiments, the ratio of positive to negative samples was fixed at 1:1, and training was conducted for three epochs. As shown in Table 6, our model achieved link prediction accuracies of 95.40% on Cora and 76.14% on CiteSeer, indicating the strong generalization capability of our method.

Table 6: Accuracy for Link Prediction Tasks.

Method	**TYPE**	**Cora**	**CiteSeer**
InstructGLM	GNN-LLM	92.49	66.88
OURS	GNN-LLM	95.43	76.14

4.3 Interpretability

DPO influence for accuracy. In the Judge-and-Improve paradigm, DPO is utilized to prioritize generations that exhibit better interpretability. However, it remains essential to evaluate how this prioritization affects reasoning accuracy. The results presented in Table 7 demonstrate that enhancing interpretability does not compromise accuracy and may even lead to improvements in reasoning performance.

Table 7: Ablation study of DPO with LLaMA as base model.

Method	**Cora (%)**	**PubMed (%)**
ExGLM w/o dpo	88.92	94.85
ExGLM	90.03	94.75

A showcase. We present a showcase in Figure 4 to illustrate the interpretability of our method in comparison with GNN-based approaches. While GNN-based methods provide explanations for their reasoning through attention weights, these weights may not accurately capture the underlying inference process and can be challenging for humans to interpret. In contrast, our method generates natural language explanations directly, thereby enhancing comprehensibility and interpretability. More cases can be found in the appendix.

Human evaluation. We aim to evaluate whether the use of DPO in the Judge-and-Improve paradigm enhances interpretability. However, assessing interpretability is challenging due to the lack of a standardized metric. To address this, we conducted a human evaluation. Specifically, we designed a questionnaire involving 20 human participants, each answering 20 questions. Participants were asked to select the response they deemed more interpretable based on three key criteria: coherency, logical consistency, and factuality. The results of this evaluation, presented in Table 8, demonstrate the effectiveness of our approach. The baseline InstructGLM suffers from overfitting on the training dataset, which harms its language generation capabilities and limits its ability to provide meaningful explanations.

Table 8: Human evaluation results with LLaMA3 as base model.

ExGLM *vs.*	**Dataset**	**Win (%)**	**Lose (%)**
InstructGLM	Cora	100.00	0
	PubMed	100.00	0
ExGLM (w/o DPO)	Cora	23.25	13.75
	PubMed	9.75	9.50

Discussion on Plausibility versus Faithfulness

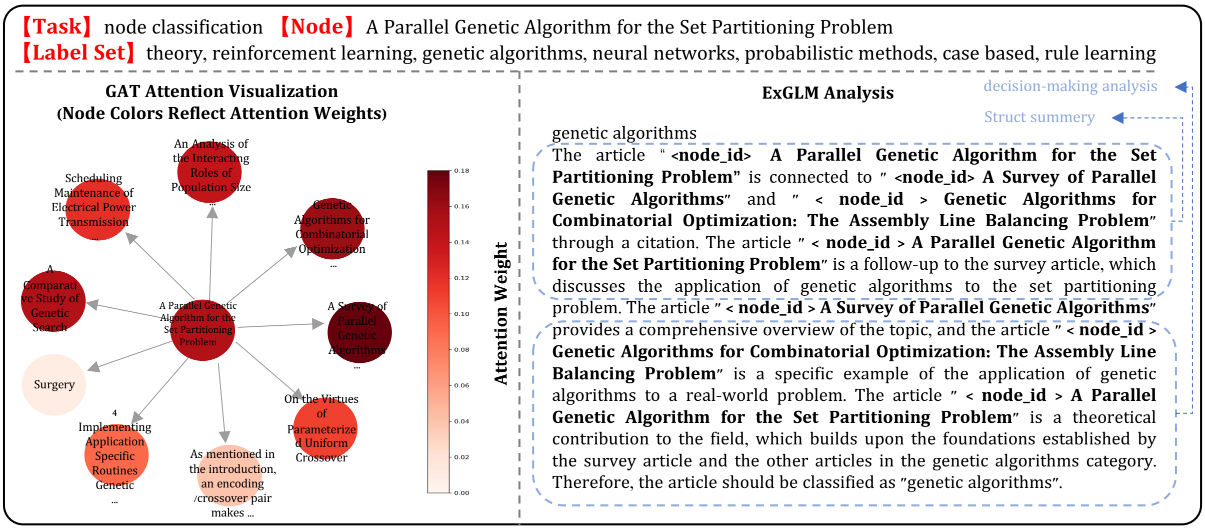

Figure 4: A show case of explanation provided by (left) GNN-based method and (right) ExGLM.

within Interpretability. In terms of interpretability, this paper focuses on the trade-off between **plausibility** and **faithfulness.** The study is situated in a recommendation scenario involving node classification, where both prediction results and explanations are presented directly to end-users. As illustrated by the examples in the Appendix, providing intuitive, coherent, and logically consistent explanations helps improve user acceptance and engagement with the recommendations. Therefore, the current experiment emphasizes plausibility to enhance comprehensibility and persuasiveness at the application level. Regarding faithfulness, although complex scenarios such as medical diagnostics require strict tracing of the model's decision path, the decision logic in our citation network node classification task is relatively straightforward: the model relies on the node's own features and the structure of its neighbors for inference. This process aligns with the explanatory basis we focus on in terms of factuality. Thus, the requirement for faithfulness is relatively limited in the current task. We intend to conduct further research on faithfulness mechanisms in more complex scenarios in subsequent work.

5 Conclusion

This work investigates how to better leverage LLMs for reasoning with structured data. Concretely, we aim to address two main limitations identified in recent studies: cross-modality alignment and interpretability. We propose a novel training framework named **ExGLM**, within which a graph-language synergistic alignment module is introduced to ensure semantic consistency across modalities. Additionally, we introduce a Judge-and-Improve paradigm that adopts a superior language model to evaluate and select generated responses with better interpretability. The selected data is subsequently utilized to optimize the reasoning model. Experiments across various scenarios demonstrate the effectiveness of our approach, showcasing its potential to advance reasoning with structured data.

6 Limitations

While our work achieves promising results, there are several limitations that warrant attention. First, the effectiveness of the Judge-and-Improve module depends heavily on the performance of the superior language model used for evaluation. If the evaluating model introduces biases or provides inaccurate assessments, the refinement process may be suboptimal, potentially constraining the overall improvement of the target model's outputs. Second, the current framework does not implement the judgment-and-improvement process iteratively. Iterative refinement, which involves multiple rounds of evaluation and optimization, could further enhance the quality and robustness of the model's outputs. However, this remains an unexplored avenue and is left for future work.

Acknowledgements

This work was supported by the merchant strategy recommendation business of Sesame, and Ant Group Postdoctoral Programme. All opinions in this paper only represents the personal views of the authors.

References

Amine Mohamed Aboussalah and Abdessalam Ed-dib. 2025. Are gnns doomed by the topology of their input graph? *Preprint*, arXiv:2502.17739.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, and 1 others. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Ziwei Chai, Tianjie Zhang, Liang Wu, Kaiqiao Han, Xiaohai Hu, Xuanwen Huang, and Yang Yang. 2023. Graphllm: Boosting graph reasoning ability of large language model. *Preprint*, arXiv:2310.05845.

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

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, and 181 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186.

Vijay Prakash Dwivedi and Xavier Bresson. 2021. A generalization of transformer networks to graphs. *Preprint*, arXiv:2012.09699.

Zirui Guo, Lianghao Xia, Yanhua Yu, Yuling Wang, Kangkang Lu, Zhiyong Huang, and Chao Huang. 2025. Graphedit: Large language models for graph structure learning. *Preprint*, arXiv:2402.15183.

William L. Hamilton, Rex Ying, and Jure Leskovec. 2018. Inductive representation learning on large graphs. *Preprint*, arXiv:1706.02216.

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.

Xuanwen Huang, Kaiqiao Han, Dezheng Bao, Quanjin Tao, Zhisheng Zhang, Yang Yang, and Qi Zhu. 2023. Prompt-based node feature extractor for few-shot learning on text-attributed graphs. *Preprint*, arXiv:2309.02848.

Xuanwen Huang, Kaiqiao Han, Yang Yang, Dezheng Bao, Quanjin Tao, Ziwei Chai, and Qi Zhu. 2024. Can gnn be good adapter for llms? In *Proceedings of the ACM Web Conference 2024*, WWW '24, page 893–904, New York, NY, USA. Association for Computing Machinery.

Bowen Jin, Chulin Xie, Jiawei Zhang, Kashob Kumar Roy, Yu Zhang, Zheng Li, Ruirui Li, Xianfeng Tang, Suhang Wang, Yu Meng, and Jiawei Han. 2024. Graph chain-of-thought: Augmenting large language models by reasoning on graphs. *Preprint*, arXiv:2404.07103.

Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. *Preprint*, arXiv:1609.02907.

Weirui Kuang, Zhen WANG, Yaliang Li, Zhewei Wei, and Bolin Ding. 2022. Coarformer: Transformer for large graph via graph coarsening.

Zheyuan Liu, Xiaoxin He, Yijun Tian, and Nitesh V. Chawla. 2024. Can we soft prompt llms for graph learning tasks? In *Companion Proceedings of the ACM Web Conference 2024*, WWW '24, page 481–484, New York, NY, USA. Association for Computing Machinery.

Steph-Yves Louis, Yong Zhao, Alireza Nasiri, Xiran Wang, Yuqi Song, Fei Liu, and Jianjun Hu. 2020. Graph convolutional neural networks with global attention for improved materials property prediction. *Physical Chemistry Chemical Physics*, 22(32):18141–18148.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, and 1 others. 2023. Self-refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems*, 36:46534–46594.

Gerald Namata, Ben London, Andrey Kolobov, German Mart'ınez-Mu noz, and Kristian Kersting. 2012. Query-driven active surveying for collective classification. In *Proceedings of the 10th international conference on Knowledge discovery and data mining*, pages 446–461. Springer.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, and 262 others. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn.

2023. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36:53728–53741.

Dong Shu, Tianle Chen, Mingyu Jin, Chong Zhang, Mengnan Du, and Yongfeng Zhang. 2024. Knowledge graph large language model (kg-llm) for link prediction. *Preprint*, arXiv:2403.07311.

Fabian Sperrle, Mennatallah El-Assady, Grace Guo, Rita Borgo, D Horng Chau, Alex Endert, and Daniel Keim. 2021. A survey of human-centered evaluations in human-centered machine learning. In *Computer Graphics Forum*, volume 40, pages 543–568. Wiley Online Library.

Jiabin Tang, Yuhao Yang, Wei Wei, Lei Shi, Lixin Su, Suqi Cheng, Dawei Yin, and Chao Huang. 2024. 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 '24, page 491–500, New York, NY, USA. Association for Computing Machinery.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, and 1 others. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph attention networks. *Preprint*, arXiv:1710.10903.

Jianing Wang, Junda Wu, Yupeng Hou, Yao Liu, Ming Gao, and Julian McAuley. 2024. InstructGraph: Boosting large language models via graph-centric instruction tuning and preference alignment. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 13492–13510, Bangkok, Thailand. Association for Computational Linguistics.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2022. Self-instruct: Aligning language models with self-generated instructions. *arXiv preprint arXiv:2212.10560*.

Menghua Wu, Russell Littman, Jacob Levine, Lin Qiu, Tommaso Biancalani, David Richmond, and Jan-Christian Huetter. 2025. Contextualizing biological perturbation experiments through language. In *The Thirteenth International Conference on Learning Representations*.

Lianghao Xia, Ben Kao, and Chao Huang. 2024. OpenGraph: Towards open graph foundation models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 2365–2379, Miami, Florida, USA. Association for Computational Linguistics.

Rui Xue, Xipeng Shen, Ruozhou Yu, and Xiaorui Liu. 2024. Efficient end-to-end language model fine-tuning on graphs. *Preprint*, arXiv:2312.04737.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, and 43 others. 2024. Qwen2 technical report. *Preprint*, arXiv:2407.10671.

Junhan Yang, Zheng Liu, Shitao Xiao, Chaozhuo Li, Defu Lian, Sanjay Agrawal, Amit S, Guangzhong Sun, and Xing Xie. 2021. Graphformers: GNN-nested transformers for representation learning on textual graph. In *Advances in Neural Information Processing Systems*.

Zhilin Yang, William W. Cohen, and Ruslan Salakhutdinov. 2016. Revisiting semi-supervised learning with graph embeddings. *Preprint*, arXiv:1603.08861.

Ruosong Ye, Caiqi Zhang, Runhui Wang, Shuyuan Xu, and Yongfeng Zhang. 2024. Language is all a graph needs. *EACL*.

Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. 2021. Do transformers really perform badly for graph representation? *Advances in neural information processing systems*, 34:28877–28888.

Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Xian Li, Sainbayar Sukhbaatar, Jing Xu, and Jason Weston. 2025. Self-rewarding language models. *Preprint*, arXiv:2401.10020.

Ce Zhang, Taixi Lu, Md Mohaiminul Islam, Ziyang Wang, Shoubin Yu, Mohit Bansal, and Gedas Bertasius. 2024. A simple llm framework for long-range video question-answering. *Preprint*, arXiv:2312.17235.

Jiawei Zhang, Haopeng Zhang, Congying Xia, and Li Sun. 2020. Graph-bert: Only attention is needed for learning graph representations. *Preprint*, arXiv:2001.05140.

Yang Zhang, Hanlei Jin, Dan Meng, Jun Wang, and Jinghua Tan. 2025. A comprehensive survey on process-oriented automatic text summarization with exploration of llm-based methods. *Preprint*, arXiv:2403.02901.

Jianan Zhao, Meng Qu, Chaozhuo Li, Hao Yan, Qian Liu, Rui Li, Xing Xie, and Jian Tang. 2023. Learning on large-scale text-attributed graphs via variational inference. In *The Eleventh International Conference on Learning Representations*.

Wenhao Zhu, Hongyi Liu, Qingxiu Dong, Jingjing Xu, Shujian Huang, Lingpeng Kong, Jiajun Chen, and Lei Li. 2024. Multilingual machine translation with large language models: Empirical results and analysis. *Preprint*, arXiv:2304.04675.

Xi Zhu, Haochen Xue, Ziwei Zhao, Wujiang Xu, Jingyuan Huang, Minghao Guo, Qifan Wang, Kaixiong Zhou, and Yongfeng Zhang. 2025. Llm as gnn: Graph vocabulary learning for text-attributed graph foundation models. *arXiv preprint arXiv:2503.03313*.

A Implementation Details and Hyperparameters

A.1 Neighbor Node Sampling

We employed node sampling in two components of the graph–language synergistic alignment module: (1) textual attribution, and (2) GNN representation learning. For textual attribution, node-level attributions were derived by randomly sampling neighboring nodes within 1–3 hops. For GNN training, subgraph information was collected by sampling nodes within a fixed 2-hop neighborhood to enable efficient representation learning.

A.2 SFT and DPO

During the SFT phase, we perform a grid search over learning rates $\{1 \times 10^{-5}, 3 \times 10^{-5}, 8 \times 10^{-5}\}$ and batch sizes $\{8, 16, 32, 64\}$, training for 1–3 epochs. In the DPO (Rafailov et al., 2023) phase, we use a smaller learning rate in the range of 1×10^{-6} to 5×10^{-6} and fix the batch size to 8. For LLM training in both the SFT and DPO stages, we adopt LoRA-based fine-tuning.

Table 9: Full hyperparameter settings.

Stage	**Hyperparameter**	**Value**
Model Backbone	Number of GraphSAGE layers	2
	Final GraphSAGE hidden size (dim)	4096
Graph Sampling	Textual attribution hop sampling	1–3
	GNN node representation hop sampling	2
Generation	Temperature	0.1, 0.3, 0.5, 0.7, 1.0
	Outputs per node (unconditioned)	5
	Outputs per node (conditioned)	5
SFT Phase	Max length	1024
	Learning rate	1×10^{-5}, 3×10^{-5}, 8×10^{-5}
	Batch size	8, 16, 32, 64
	Epochs	1–3
	λ_1	0.99
	λ_2	0.01
DPO Phase	Max length	1024
	Learning rate	1×10^{-6} – 5×10^{-6}
	Batch size	8
	weight_decay	0.05
	beta	0.1
Lora	lora_alpha	16
	dropout	0.05
	lora_r	16

B Analysis of Multi-turn Judge-and-Improve Paradigm

We conducted an additional round of judge-and-improve on the outputs from the first iteration of the model. However, the results exhibited minimal improvement, and in some cases, certain anomalies became more pronounced. For example, with complex subgraphs, the model tended to produce excessively long interpretations, often overemphasizing the description of connections between nodes, as illustrated in Table 10. We posit that, unlike aligning LLMs with broader human preference patterns, the explanation style for node classification tasks is inherently more uniform—primarily focusing on node attributes and local subgraph structures. Consequently, a simpler feedback loop proves to be more intuitive and effective in this context.

Table 10: Examples of overemphasizing the description of connections between nodes

"Expected Mistake Bound Model for On-Line Reinforcement Learning" is connected to "On the Applicability of Neural Network and Machine Learning Methodologies to Natural Language Processing" through a citation. The paper "On the Applicability of Neural Network and Machine Learning Methodologies to Natural Language Processing" is connected to "Visualizing High-Dimensional Structure with the Incremental Grid Growing Neural Network" through a citation. The paper "Visualizing High-Dimensional Structure with the Incremental Grid Growing Neural Network" is connected to "Exploration in Machine Learning" through a citation. The paper "Exploration in Machine Learning" is connected to "Issues in Using Function Approximation for Reinforcement Learning" through a citation...

C Details of Human Evaluation

For each dataset, we sampled 400 instances as test cases and distributed them among 20 human participants. As there are no strict quantitative criteria for coherence or logical consistency, evaluators were primarily asked to assess: (1) whether the explanation was reasonable, (2) whether any semantic inconsistencies were present, and (3) whether the given instructions were strictly followed. For factuality, we focused on two aspects: first, whether the inferred conclusions were consistent with the ground-truth labels; and second, whether the descriptions of node information and subgraph structures accurately reflected the underlying data.

D Case Study

We provide illustrative examples of both high-quality and low-quality cases.

Table 11: Some accurately predicted showcases.

Question1 :
Classify the article according to its topic into one of the following categories:[theory, reinforcement learning, genetic algorithms, neural networks, probabilistic methods, case based, rule learning]. Node represents academic paper with a specific topic, link represents a citation between the two papers. Pay attention to the multi-hop link relationship between the nodes.

Input
(<Node_index>,An Experimental and Theoretical Comparison of Model Selection Methods on simple model selection problems, the) is connected with (<Node_index>,Algorithmic Stability and Sanity-Check Bounds for Leave-One-Out Cross-Validation), (<Node_index>,Toward Efficient Agnostic Learning), (<Node_index>,A Fast, Bottom-Up Decision Tree Pruning Algorithm with Near-Optimal Generalization), (<Node_index>,Towards Robust Model Selection using Estimation and Approximation Error Bounds), (<Node_index>,Rigorous Learning Curve Bounds from Statistical Mechanics), (<Node_index>,Characterizing the generalization performance of model selection strategies), (<Node_index>,Model Selection based on Minimum Description Length), (<Node_index>,A Bound on the Error of Cross Validation Using the Approximation and Estimation Rates, with), (<Node_index>,Preventing "Overfitting" of Cross-Validation Data), (<Node_index>,A New Metric-Based Approach to Model Selection) within one hop. Which category should (<Node_index>,An Experimental and Theoretical Comparison of Model Selection Methods on simple model selection problems, the) be classified as? Please answer classification result in the first line.
Then provide accurate and critical explanation.
Ground truth label:
theory
Response from ExGLM:
theory
The article "An Experimental and Theoretical Comparison of Model Selection Methods on simple model selection problems" is classified as "theory" because it compares and analyzes different model selection methods, which is a theoretical approach. The article discusses the experimental and theoretical aspects of model selection, which is a key aspect of theoretical research in machine learning and data science.

Question :
Classify the article according to its topic into one of the following categories:[theory, reinforcement learning, genetic algorithms, neural networks, probabilistic methods, case based, rule learning]. Node represents academic paper with a specific topic, link represents a citation between the two papers. Pay attention to the multi-hop link relationship between the nodes.

Input
(<Node_index>,Explanations of Empirically Derived Reactive Plans) is connected with (<Node_index>,Using a Genetic Algorithm to Learn Strategies for Collision Avoidance and Local Navigation), (<Node_index>,AN ENHANCER FOR REACTIVE PLANS), (<Node_index>,LEARNING CONCEPTS BY ASKING QUESTIONS), (<Node_index>,Improving Tactical Plans with Genetic Algorithms), (<Node_index>,Learning Sequential Decision Rules Using Simulation Models and Competition), (<Node_index>,Simulation-Assisted Learning by Competition: Effects of Noise Differences Between Training Model and Target Environment) within one hop. Which category should (<Node_index>,Explanations of Empirically Derived Reactive Plans) be classified as?
Please answer classification result in the first line.
Then provide accurate and critical explanation.
Ground truth label:
genetic algorithms
Response from ExGLM:
theory
The article "Explanations of Empirically Derived Reactive Plans" is connected to "Using a Genetic Algorithm to Learn Strategies for Collision Avoidance and Local Navigation" within one hop. This suggests that the article is related to genetic algorithms, which are a type of optimization technique that uses a population of solutions to find the best solution to a problem. The article likely discusses the application of genetic algorithms to the problem of planning reactive behaviors in robots.

Table 12: Some accurately predicted showcases.

Question1 :
Classify the article according to its topic into one of the following categories:[theory, reinforcement learning, genetic algorithms, neural networks, probabilistic methods, case based, rule learning]. Node represents academic paper with a specific topic, link represents a citation between the two papers. Pay attention to the multi-hop link relationship between the nodes.

Input
(<Node_index>,Discretization of continuous attributes using ReliefF) is connected with (<Node_index>,Estimating Attributes: Analysis and Extensions of RELIEF), (<Node_index>,Estimating Attributes: Analysis and Extensions of RELIEF), (<Node_index>,Naive Bayesian classifier within ILP-R), (<Node_index>,Induction of decision trees using RELIEFF), (<Node_index>,SFOIL: Stochastic Approach to Inductive Logic Programming), (<Node_index>,Machine learning in prognosis of the femoral neck fracture recovery examples, estimating attributes, explanation ability,), (<Node_index>,Overcoming the myopia of inductive learning algorithms with RELIEFF), (<Node_index>,Induction of decision trees and Bayesian classification applied to diagnosis of sport injuries), (<Node_index>,Context-sensitive attribute estimation in regression),......, (<Node_index>,A counter example to the stronger version of the binary tree hypothesis), (<Node_index>,Irrelevant Features and the Subset Selection Problem), (<Node_index>,Naive Bayesian classifier within ILP-R), (<Node_index>,Discretization of continuous attributes using ReliefF), (<Node_index>,Stochastic search in inductive concept learning), (<Node_index>,Prognosing the Survival Time of the Patients with the Anaplastic Thyroid Carcinoma with Machine Learning) within two hops. Which category should (<Node_index>,Discretization of continuous attributes using ReliefF) be classified as?
Please answer classification result in the first line.
Then provide accurate and critical explanation.
Ground truth label:
rule learning
Response from ExGLM:
rule learning
The article "Discretization of continuous attributes using ReliefF" is connected to "Estimating Attributes: Analysis and Extensions of RELIEF" and "Induction of decision trees using RELIEF" through a citation. The article discusses the use of ReliefF, a rule learning algorithm, to discretize continuous attributes in a dataset. The article is connected to other papers that discuss the application of rule learning algorithms in machine learning, such as "Induction of decision trees and Bayesian classification applied to diagnosis of sport injuries" and "Overcoming the myopia of inductive learning algorithms with RELIEF". The article is also connected to papers that discuss the use of rule learning algorithms in other domains, such as "Stochastic search in inductive concept learning" and "Prognosing the Survival Time of the Patients with the Anaplastic Thyroid Carcinoma with Machine Learning". The article is connected to other papers through citations, indicating that it is a key contribution to the field of rule learning.

Table 13: Bad case: The inferred conclusion is inconsistent with the label

Question1 :
Classify the article according to its topic into one of the following categories:[theory, reinforcement learning, genetic algorithms, neural networks, probabilistic methods, case based, rule learning]. Node represents academic paper with a specific topic, link represents a citation between the two papers. Pay attention to the multi-hop link relationship between the nodes.

Input
(<Node_index>,Learning in Boltzmann Trees) is connected with (<Node_index>,Decimatable Boltzmann Machines vs. Gibbs Sampling), (<Node_index>,Boltzmann Machine learning using mean field theory and linear response correction), (<Node_index>,Boltzmann Chains and Hidden Markov Models), (<Node_index>,Unsupervised learning of distributions on binary vectors using two layer networks), (<Node_index>,Exploiting Tractable Substructures in Intractable Networks) within one hop. Which category should (<Node_index>,Learning in Boltzmann Trees) be classified as?
Please answer classification result in the first line.
Then provide accurate and critical explanation.
Ground truth label:
neural networks
Response from ExGLM:
neural networks
The article "Learning in Boltzmann Trees" is connected to "Decimatable Boltzmann Machines vs. Gibbs Sampling" within one hop. This suggests that the article is related to probabilistic methods, specifically Boltzmann machines and Gibbs sampling. The article is likely to discuss the application of Boltzmann machines and Gibbs sampling in learning, which is a key aspect of probabilistic methods.

Table 14: Bad case: It is coherent but has unclear logic. It describes too much structural information and cannot draw a conclusion quickly

Question1 :
Classify the article according to its topic into one of the following categories:[theory, reinforcement learning, genetic algorithms, neural networks, probabilistic methods, case based, rule learning]. Node represents academic paper with a specific topic, link represents a citation between the two papers. Pay attention to the multi-hop link relationship between the nodes.

Input
(<Node_index>,Data Analyses Using Simulated Breeding and Inductive Learning Methods) is connected with (<Node_index>,Genetic Algorithm Programming Environments), (<Node_index>,Geometric Comparison of Classifications and Rule Sets*), ... (<Node_index>,Automatic Parameter Selection by Minimizing Estimated Error), (<Node_index>,Induction of One-Level Decision Trees), (<Node_index>,Using Real-Valued Genetic Algorithms to Evolve Rule Sets for Classification), (<Node_index>,Putting the Genetics back into Genetic Algorithms), (<Node_index>,Surgery), (<Node_index>,An Evolutionary Approach to Combinatorial Optimization Problems), (<Node_index>,A Stochastic Search Approach to Grammar Induction) within two hops. Which category should (<Node_index>,Data Analyses Using Simulated Breeding and Inductive Learning Methods) be classified as?
Please answer classification result in the first line.
Then provide accurate and critical explanation.
Ground truth label:
genetic algorithms
Response from ExGLM:
genetic algorithms
The paper "Data Analyses Using Simulated Breeding and Inductive Learning Methods" is connected to "Genetic Algorithm Programming Environments" and "Geometric Comparison of Classifications and Rule Sets" through a citation relationship. The paper discusses the use of genetic algorithms to analyze data and make predictions, which is a key aspect of genetic algorithm programming. The paper is also connected to "Actively Searching for an Effective Neural-Network Ensemble" through a citation relationship, which suggests that the paper may be related to the use of neural networks in genetic algorithm programming. The paper is also connected to "Using Decision Trees to Improve Case-Based Learning" through a citation relationship, which suggests that the paper may be related to the use of decision trees in genetic algorithm programming. The paper is also connected to "Discovering Complex Othello Strategies Through Evolutionary Neural Networks" through a citation relationship, which suggests that the paper may be related to the use of neural networks in genetic algorithm programming. The paper is also connected to "Every Niching Method has its Niche: Fitness Sharing and Implicit Sharing Compared" through a citation relationship, which suggests that the paper may be related to the use of niching methods in genetic algorithm programming. The paper is also connected to "Solving Combinatorial Problems Using Evolutionary Algorithms" through a citation relationship, which suggests that the paper may be related to the use of evolutionary algorithms in genetic algorithm programming. The paper is also connected to "Multivariate Decision Trees" through a citation relationship, which suggests that the paper may be related to the use of decision trees in genetic algorithm programming. The paper is also connected to "Analysis of Neurocontrollers Designed by Simulated Evolution" through a citation relationship, which suggests that the paper may be related to the use of genetic algorithms in the design of neural controllers. The paper is also connected to "Evolution in Time and Space: The Parallel Genetic Algorithm" through a citation relationship, which suggests that the paper may be related to the use of parallel genetic algorithms in genetic algorithm programming.

Table 15: Bad case: The node information is described incorrectly. During training, the model attempted to memorize the node behind the special token but introduced deviations. In this particular example, the target classification node corresponds to 'Bayesian Probability Theory: A General Method for Machine Learning' rather than 'Theory of Mind.'

Question1 :
Classify the article according to its topic into one of the following categories:[theory, reinforcement learning, genetic algorithms, neural networks, probabilistic methods, case based, rule learning]. Node represents academic paper with a specific topic, link represents a citation between the two papers. Pay attention to the multi-hop link relationship between the nodes.

Input
Node represents academic paper with a specific topic, link represents a citation between the two papers. Pay attention to the multi-hop link relationship between the nodes. <Node_index> is connected with <Node_index> within one hop. Which category should <Node_index> be classified as?
Please answer classification result in the first line.
Then provide accurate and critical explanation.
Ground truth label:
probabilistic methods
Response from ExGLM:
theory
The article "Theory of Mind" is connected to "Reinforcement Learning" within one hop. This suggests that the article is related to the field of reinforcement learning, which is a subfield of machine learning that deals with the design of algorithms that can learn from positive and negative reinforcement. The article's title and content suggest that it is discussing the theory of mind, which is a concept in cognitive psychology that deals with the understanding of other people's mental states and beliefs. The article's connection to reinforcement learning suggests that it is discussing the theory of mind from a machine learning perspective, which is a reasonable classification.