RAGE: RAG Enhanced LLM Explainer for Heterogeneous Graphs

Raghvi Baloni
raghvib@uw.edu
University of Washington Tacoma
Washington, USA

Yihui Chong
cyihui@dso.org.sg
DSO National Laboratories
Singapore

Ankur Teredesai
ankurt@uw.edu
University of Washington/CueZen Inc.
Seattle, Washington, USA

Abstract

Generating accurate and interpretable explanations for predictions on heterogeneous graphs remains a significant challenge due to their multi-typed structures and complex relational dependencies. While Large Language Models (LLMs) have demonstrated strong performance in natural language tasks, their ability to provide grounded explanations for heterogeneous graphs is still underexplored. In this work, we introduce RAGE (Retrieval-Augmented Graph Explainer), a novel framework that enhances explanation quality by integrating Retrieval-Augmented Generation (RAG) with structured graph retrieval. RAGE retrieves subgraphs directly relevant to a given query, ensuring that explanations remain closely aligned with the dataset's inherent structure.

We evaluate RAGE on two heterogeneous graph datasets, DBLP and Goodreads, across multiple LLMs. Through comprehensive experiments, we demonstrate that RAGE achieves comparable or superior predictive performance to metapath-based approach, while improving scalability. Furthermore, our qualitative evaluation highlights that RAGE produces more coherent and contextually accurate explanations, reducing the hallucination risks associated with indirect explanation approaches.

By offering a directly interpretable alternative to metapath-based explanation, RAGE provides a compelling framework for enhancing LLM-based explanation over heterogeneous graphs.

CCS Concepts

• **Heterogeneous Graphs**; • **Explainable AI**; • **Retrieval Augmented Generation(RAG)**; • **Large Language Model**; • **Text-Attributed Graphs**;

Keywords

Heterogeneous Graphs, Explainable AI, Retrieval Augmented Generation(RAG), Large Language Model

ACM Reference Format:

Raghvi Baloni, Yihui Chong, and Ankur Teredesai. 2025. RAGE: RAG Enhanced LLM Explainer for Heterogeneous Graphs. In . ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

KDD'25 MLoG-GenAI Workshop, Toronto, ON, Canada

ACM ISBN 978-x-xxxx-xxxx-x/YYYY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Introduction

Producing accurate explanations for predictions on heterogeneous graph datasets remains a significant challenge due to their complex, multi-typed structure. While many methods have been developed to achieve high-performing prediction tasks on such datasets, generating reliable and interpretable explanations for these predictions is still an area of active exploration. For example, Li et al. [13] demonstrated the potential of large language models (LLMs) in few-shot node classification tasks, particularly for incomplete graphs, while Bi et al. [1] explored the scalability of link prediction using LLMs, highlighting their ability to learn and generalize from heterogeneous graph data effectively. Although these studies underline the promise of LLMs in graph-based tasks, they focus mainly on prediction accuracy and scalability, leaving the generation of interpretable explanations as an underexplored avenue.

Large Language Models (LLMs), known for their ability to accurately parse, interpret, and answer complex questions based on text by understanding context, semantics, and linguistic patterns, have shown impressive performance on textual data[22][19]. They excel in providing detailed explanations for their decisions, making them promising tools for enhancing interpretability in complex tasks. Despite this, the application of LLMs to heterogeneous graphs—composed of multiple types of nodes and edges—remains challenging due to the non-textual and diverse nature of graph data.

Heterogeneous graphs are characterized by diverse node relationships and edge types, posing significant challenges for LLMs to process effectively. Recent studies have proposed graph-to-text transformation methods to address this incompatibility. For instance, Jin et al. [9] utilized neural encoders to preserve structural and semantic information during transformation, while Chai et al. [3] demonstrated the efficacy of incorporating graph embeddings into LLM prompts to improve reasoning over graph-based data. These advancements underscore the potential of reformatting graph data for seamless integration with LLMs, paving the way for improved reasoning and explanation generation.

Building on this foundation, our work leverages the Retrieval-Augmented Generation (RAG) framework to enrich LLM prompts with relevant subgraph data. By embedding contextually relevant subgraphs, we aim to enable LLMs to generate more accurate predictions and interpretable explanations while minimizing hallucinated or irrelevant outputs. This approach addresses the dual challenge of improving both prediction accuracy and explainability, particularly in the context of heterogeneous graph datasets.

Our contributions can be summarized as follows:

- **RAG Pipeline for Heterogeneous Graphs:** We introduce a novel RAG pipeline tailored for heterogeneous graphs, retrieving the subgraph most relevant to the query, ensuring meaningful context for LLM predictions.
- **LLM as an Explainer:** By integrating subgraphs into RAG-enhanced prompts, we propose using LLMs as explainers for heterogeneous graph databases. This ensures factually correct and interpretable explanations for the predictions generated by LLMs.
- **Comprehensive Evaluation:** Through rigorous experiments on public datasets, including DBLP and Goodreads, across various LLMs, we demonstrate the robustness and performance of our model.

2 Related Works

LLMs are proven to face challenges in reasoning, factual precision, and explainability in various domains[2][5][4][15][16]. Lei et al. utilized LLMs as surrogate models to explain black-box recommender systems [11]. They employed alignment techniques—behavior, intention, and hybrid alignment—to generate natural language explanations for recommendations based on user profiles. Similarly Fang et al. introduced a method to augment text-attributed graphs (TAGs) through prompt engineering and LLM-based textual attribute perturbation, integrating the augmented features into generative models and GNNs for improved performance [7]. While both approaches focus on leveraging LLMs, our framework diverges by targeting heterogeneous graph datasets and employing LLMs as both predictors and explainers.

The integration of large language models (LLMs) and retrieval-augmented generation (RAG) techniques has garnered significant attention[6][23]. Li et al. addressed the challenge of hallucinations in LLMs by employing RAG pipelines to retrieve relevant context for domain-specific and time-sensitive queries [12]. They used curated datasets in formats such as HTML and PDF, which were divided into small chunks for effective retrieval. Mavromatis et al. combined GNNs with LLMs to retrieve and reason over dense subgraphs in knowledge graphs (KGs) [14]. Their extracted reasoning paths included both answer-containing and distractor paths, which were verbalized and used as input for LLMs.

Our framework, RAGE, extends beyond the methodologies discussed by focusing on heterogeneous graphs, which pose unique challenges in decision-making and explainability. By leveraging SentenceBERT embeddings along with structural graph information, we ensure robust representation of graph data. Rather than fine-tuning LLMs on user profiles or modifying TAG attributes, we dynamically retrieve and integrate subgraphs as contextual prompts using RAG, enabling LLMs to provide factually correct predictions and explanations without altering the original graph structure.
Unlike the approaches by Li et al. and GNN-RAG, which include direct answers within retrieved reasoning paths, RAGE ensures predictions are based solely on contextual understanding. This avoids the risk of embedding biases or errors from pre-determined answers, creating a more nuanced and generalizable framework for reasoning over complex datasets.

3 Background

Heterogeneous graphs represent a versatile yet complex data structure found in various real-world domains, from citation networks to healthcare and social media. This section provides a detailed overview of key concepts and methodologies that form the foundation of our proposed framework. We begin by introducing heterogeneous graphs and their unique challenges, followed by a discussion on textually attributed graphs (TAGs), which integrate structural and textual information. Next, we explore Retrieval-Augmented Generation (RAG), an approach used to retrieve and process relevant subgraphs to enhance prediction and explanation tasks. Finally, we highlight the capabilities of Large Language Models (LLMs) in generating interpretable explanations for predictions derived from heterogeneous graphs, bridging the gap between structured graph data and natural language.

Heterogeneous graphs are defined as directed graphs $G = (V, E, T_v, T_e)$ where V represents the set of nodes, E denotes the set of edges, and T_v and T_e are the sets of node and edge types, respectively. Each node $v \in V$ and edge $e \in E$ is associated with type mapping functions$\tau_v(v) : V \rightarrow T_v$ and $\tau_e(e) : E \rightarrow T_e$. Such graphs can be represented using a set of adjacency matrices$\{A_t\}_{t=1}^{|T_e|}$, or a three-dimensional tensor$A \in \mathbb{R}^{|V|\times|V|\times|T_e|}$, where $A_t \in \mathbb{R}^{|V|\times|V|}$ captures the adjacency relationships for the t-th edge type. Specifically, $A_t[i, j]$ encodes the weight of an edge of type t from node j to node i. In the special case where $|T_v| = 1$ and $|T_e| = 1$, the graph reduces to a homogeneous graph.

Heterogeneous Textually Attributed Graphs (TAGs) extend this framework by incorporating textual features at the node and edge levels. Formally, let $X \in \mathbb{R}^{|V|\times d}$ represent the textual features of nodes, where d is the dimension of the embedding space, and $F \in \mathbb{R}^{|E|\times f}$ denote edge-level textual attributes. Jointly modeling structural relationships through A and textual features through X and F is a non-trivial task, as it requires balancing the heterogeneous graph's relational complexity with the semantic richness of the text. Predicting node labels in such networks involves jointly modeling the graph's structural and textual attributes, a non-trivial challenge due to the diverse data modalities involved [9]. While accurate prediction on TAGs remains difficult, the greater challenge lies in generating explanations for these predictions. Capturing the interplay of textual and structural information in TAGs is crucial for achieving interpretability, especially when dealing with complex heterogeneous datasets.

RAG enhances large language model outputs by retrieving relevant context—such as subgraphs—from a database. This context enables LLMs to provide factually accurate predictions and explanations. In this framework, cosine similarity between query embeddings and graph embeddings is employed to identify relevant subgraphs. Rather than directly utilizing all retrieved information, a ranking mechanism prioritizes top-k nodes and edges. These are then used to construct subgraphs that optimize the retrieval process. The retrieved subgraph, converted into textual form, is subsequently fed into the LLM as part of the prompt, ensuring coherence and

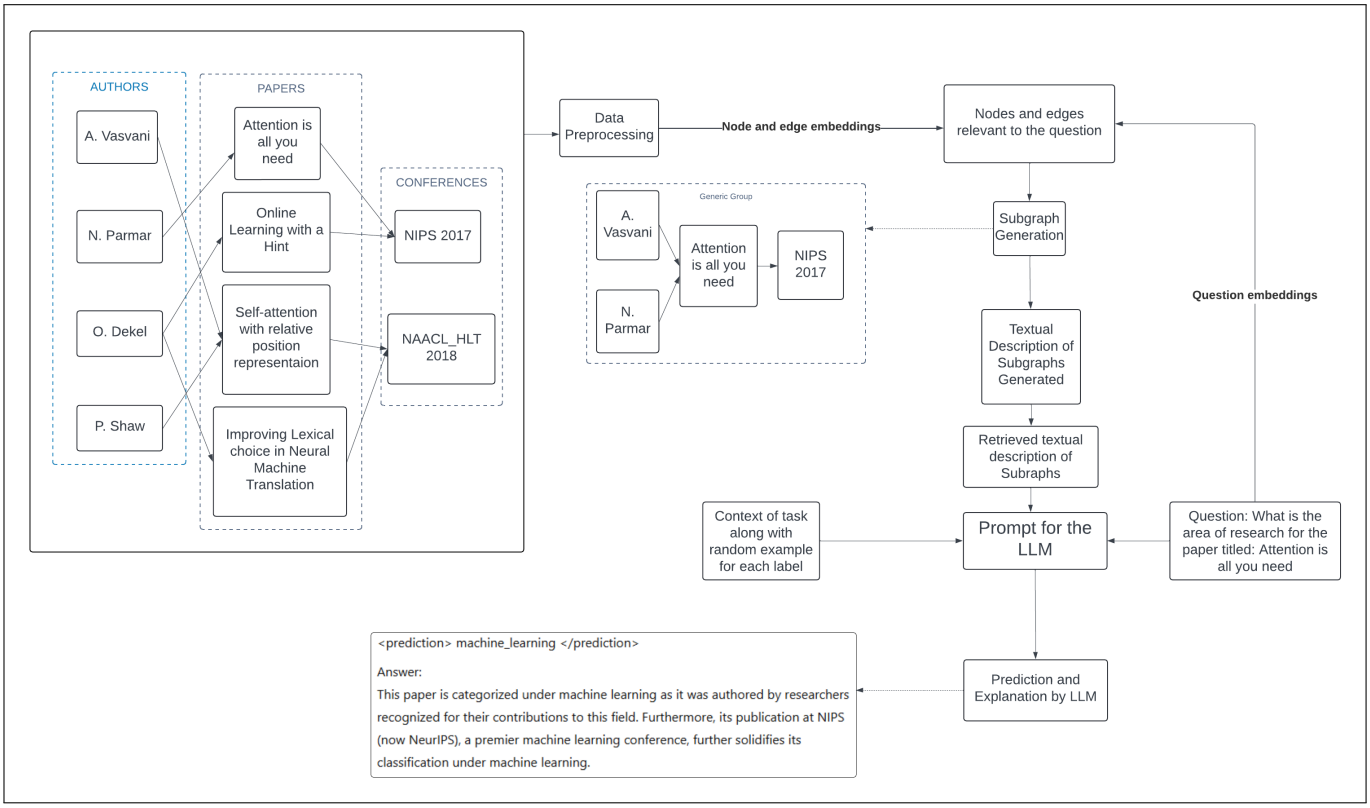

Figure 1: RAGE Framework: Retrieval-Augmented Graph Explainer approach to enhance explanations for Heterogeneous Graphs. It leverages a heterogeneous graph database to retrieve subgraphs relevant to the question. The prompt, consisting of the textual description of the retrieved subgraph, the question, and additional context, guides the LLM's response. This approach minimizes hallucinations while enhancing explanation quality by utilizing the rich structure of heterogeneous graphs.

context relevancy. This step significantly mitigates the issue of hallucination and enhances the quality of explanations.

Large Language Models (LLMs), such as GPT-based models, have shown exceptional capabilities in natural language processing tasks, including question answering[14], summarization, and text generation. They have also demonstrated utility in heterogeneous graph prediction tasks[9].

In our framework, LLMs are leveraged not just for predictions but also for generating explanations. Since LLMs are inherently limited in understanding graph structures, a transformation process is employed where heterogeneous graphs are "textualized" into a sequence format suitable for language models. By integrating the retrieved subgraphs as part of the prompt, the LLM can generate accurate and interpretable explanations for its predictions.

This hybrid approach of graph retrieval and language-based explanation opens new avenues for improving interpretability in heterogeneous graph learning while addressing key challenges associated with complexity and multi-modal data representation.

4 Methodology

In this section, we outline a systematic approach aimed at improving the quality of explanations for predictions made on heterogeneous graph datasets. Due to the intricate relationships and varied edge types within these graphs, generating grounded and interpretable explanations remains a significant challenge. Our methodology leverages the Retrieval-Augmented Generation (RAG) framework to provide rich, graph-contextualized prompts to large language models (LLMs), thereby addressing this challenge.

The RAG framework enables the extraction and embedding of relevant subgraph data into the LLM's input. This enriched context allows the LLM to generate predictions and explanations that are more specific, accurate, and aligned with the underlying graph data. By bridging the gap between heterogeneous graph structures and LLM capabilities, our approach aims to not only improve the interpretability of LLM output, but also to enhance the overall prediction accuracy.

As illustrated in **Figure 1** the pipeline begins with the preprocessing of heterogeneous graph data to extract key relationships and subgraph structures. These subgraphs are then integrated into LLM

prompts, forming a context-rich input for prediction and explanation generation. The following subsections detail each step, including graph data preprocessing, RAG-based prompt engineering, prediction generation, and explanation evaluation.

4.1 Dataset Preprocessing and Graph Embedding Construction

The preprocessing step involves extracting key metadata from the raw dataset and ensuring data consistency through cleaning and organization. This stage is crucial for refining the graph structure, eliminating noise, and preparing a manageable and representative subset of data for analysis. The processed data serves as the foundation for constructing subgraphs used in downstream task.

The cleaned and sampled data is used to generate graph embeddings. In this process, the extracted metadata form nodes, while directed edges represent relationships between these nodes, constructing a heterogeneous graph structure. This graph serves as the input for our proposed framework. We use SBERT (Sentence-BERT) to generate embeddings of the nodes and edges, capturing both semantic and structural information[17]. This is particularly beneficial for heterogeneous graphs where textual attributes (such as paper abstracts in DBLP or book descriptions in Goodreads) serve as primary sources of information. These embeddings are stored in a vector database to facilitate efficient retrieval during downstream tasks.

4.2 Retrieval and Sub-graph Creation

To enhance the relevance of information provided to the Large Language Model (LLM), we implement a focused subgraph creation process inspired by the methodology described by Xiaoxin He et al[8]. The subgraphs are created using the the Prize-Collecting Steiner Tree (PCST) algorithm[21].

We generate these subgraphs to refine the contextual information passed to the LLM, ensuring the model focuses on query-relevant data while minimizing noise. By using the PCST algorithm, we extract the key nodes and their connecting edges, forming coherent subgraphs that represent only the most essential relationships within the broader graph. These extracted subgraphs are then transformed into textual descriptions through a structured serialization process. Each description includes the node attributes, edge relationships, and contextual metadata, which are then stored in a database for efficient retrieval during the LLM's prompt generation.

PCST-Based Subgraph:

In this approach as shown in **Figure 2**, each retrieved node in the graph is assigned a "prize" based on its cosine similarity with the query embedding, indicating its relevance to the query. Similarly, retrieved edges are assigned prizes reflecting their relevance based on similarity to the query embedding's attributes. To balance relevance with conciseness, each edge is also assigned a cost, which encourages the inclusion of high-prize nodes and edges while minimizing the overall cost of the resulting subgraph.

(1) **Node and Edge Selection:** To identify the most relevant nodes and edges in response to a query x_q, we utilize a k-nearest neighbors (k-NN) retrieval approach. The query is encoded into an embedding $z_q = LM(x_q) \in \mathbb{R}^d$, ensuring consistent handling of textual data across queries and graph components. The retrieval process involves computing the cosine similarity between z_q and the embeddings of nodes (z_n) and edges (z_e) in the graph. The top-k relevant nodes (V_k) and edges (E_k) are identified as:

$$V_k = \arg\text{top-k}_{n \in V} \cos(z_q, z_n),$$
$$E_k = \arg\text{top-k}_{e \in E} \cos(z_q, z_e).$$

Here, $\cos(\cdot, \cdot)$ denotes the cosine similarity function, and the arg top-k operator retrieves the top-k elements based on the similarity scores.

(2) **Constructing the Subgraph:** Using the selected V_k and E_k, we construct a Prize-Collecting Steiner Tree (PCST) to create a cohesive and cost-effective subgraph. Higher prize values are assigned to nodes and edges that exhibit greater relevance to the query, with the top-k nodes/edges receiving descending prize values from k to 1. The node prize function is defined as:

$$\text{prize}(n) = \begin{cases} k - i, & \text{if } n \in V_k \text{ and } n \text{ is the top } i \text{ node,} \\ 0, & \text{otherwise.} \end{cases}$$

Edge prizes are computed similarly. The subgraph $S^* = (V^*, E^*)$ is then optimized to maximize the total prize of its nodes and edges, minus the cost associated with its size:

$$S^* = \underset{\substack{S \subseteq G, \\ S \text{ is connected}}}{\arg\max} \sum_{n \in V_S} \text{prize}(n) + \sum_{e \in E_S} \text{prize}(e) - \text{cost}(S),$$

where the cost of a subgraph is defined as:$\text{cost}(S) = |E_S| \cdot C_e$.

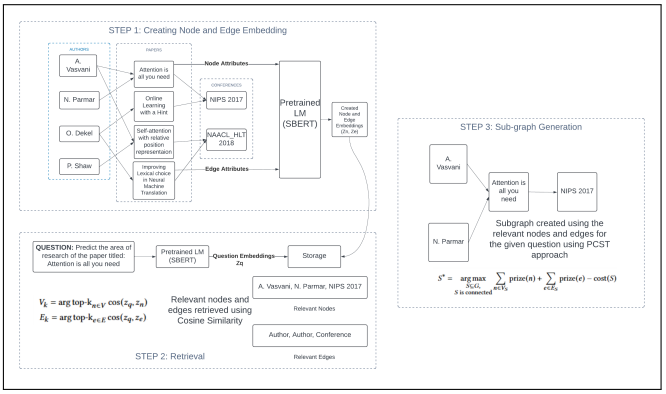

Figure 2: Three-step approach for creating a Prize-Collecting Steiner Tree (PCST)-based subgraph 1. Node and Edge Embeddings: Nodes and edges are embedded using a pretrained language model (SBERT). 2. Retrieval: A query is encoded into an embedding using SBERT. Relevant nodes and edges are retrieved based on cosine similarity with the query embedding. 3. Subgraph Generation: The top-k nodes and edges are used to construct a subgraph using the PCST approach. The final subgraph maximizes relevance to the query while maintaining structural coherence and minimizing redundancy.

Here, C_e represents the per-edge cost, which can be adjusted to control the size of the resulting subgraph. This formulation ensures that the subgraph maximizes relevance to the query while maintaining structural coherence and minimizing redundancy.

The final subgraph, which contains the top 5 most relevant nodes and associated edges, effectively captures the relationships and contextual information pertinent to the query. This subgraph is then converted into a textual description, serving as a grounded input for the LLM, which leverages it for more accurate prediction and explanation.

4.3 Loading the Model and Prompt Generation

We begin the evaluation process by loading the test dataset, consisting of 2,000 samples, and the pre-trained Large Language Model (LLM). The key to effective predictions and explanations lies in constructing a well-engineered prompt, leveraging the **Retrieval-Augmented Generation (RAG)** approach to enhance both the accuracy and relevance of the LLM's output.

The prompt generation process is carefully designed to provide the model with sufficient contextual information, ensuring precise predictions while minimizing hallucinations. For each sample (in this case, a research paper), the prompt is constructed as follows:

(1) **Question Framing:** Each prompt starts by clearly posing the question. For instance, if the task is to predict the research field of a paper, the question will be: "Predict the research area for the paper titled: 'X'."
(2) **Contextual Guidance (One-shot Learning:)** To guide the LLM in producing structured responses, we use a one-shot example for each label. This step provides the model with a template for generating answers. For example, the model is shown how to answer for a paper in Machine Learning and similarly for Computer Networking and Theoretical Computer Science. This serves as an anchor, helping the LLM maintain consistency in its output format.
(3) **Including the Retrieved Subgraph:** The most critical part of the prompt is embedding the retrieved subgraph, which is constructed using the Prize-Collecting Steiner Tree (PCST) approach. This subgraph, consisting of the top 5 most relevant nodes and their associated edges, reflects the key relationships between different concepts or papers. By embedding this structured information, the LLM is given context grounded in actual data, reducing the risk of generating fabricated or irrelevant explanations.

The LLM then processes this complete prompt and returns:

- A **predicted node**, which represents the research field or label for the paper.
- An **explanation** detailing why this node was chosen, based on the relationships within the subgraph (e.g., co-citations, shared authorship, or similar research topics). This grounded explanation ensures transparency in the model's decision-making process.

By integrating both the question and relevant subgraph into the prompt, we aim to improve the model's interpretability and reduce reliance on generic or uninformed outputs.

5 Experimentation

Dataset Description: We evaluate our model on two public heterogeneous graph datasets: **DBLP**[10] and **Goodreads**[20]. These datasets contain multiple node and edge types, with textual attributes available for specific node categories. To ensure consistency in our analysis, we restrict the target labels to three per dataset. In DBLP, papers are categorized into *machine learning, computer networking, and theoretical computer science*. The graph consists of three node types—**papers (P), authors (A), and conferences (C)**—and four types of edges capturing author-paper and conference-paper relationships. Paper abstracts serve as the primary textual feature.

Similarly, the Goodreads dataset is a book-oriented graph where books are labeled as *fiction, non-fiction, and romance*. This dataset comprises five node types—**books (B), authors (A), publishers (P), formats (F), and language codes (L)**—with multiple relationships connecting books to other entities, such as authors, publishers, and formats. The primary textual component in this dataset is the book description, which serves as the key feature for book nodes.

Baseline: To evaluate the effectiveness of our proposed framework, we compare it against Metapath of Thought (MoT)[18], a structured reasoning approach designed for node classification and explanation tasks in heterogeneous graphs. Instead of directly retrieving subgraphs, MoT generates metapaths—sequences of connected node types that capture meaningful relational patterns within the graph. These metapaths serve as structured reasoning chains, guiding the large language model (LLM) in both prediction and explanation generation. By explicitly incorporating metapath-based reasoning, MoT ensures that predictions are grounded in structured graph-derived insights.

To evaluate our framework's effectiveness, we conduct experiments using two widely adopted commercial LLMs: **Claude** and **GPT-4**. These models were selected due to their widespread use in research and industry, and demonstrated success in handling complex natural language tasks. The inclusion of multiple LLMs allows us to validate the robustness of our approach across different architectures, ensuring that our findings are not model-specific but instead highlight the general effectiveness of retrieval-augmented explainability in heterogeneous graph explanation.

5.1 Task 1: Node Prediction

The node prediction task evaluates the model's ability to correctly classify target nodes within a heterogeneous graph. Given the complex structure of these graphs, which include multiple node and edge types, achieving high predictive accuracy requires effective contextualization of both structural and textual information.

To standardize evaluation, we adopt Micro F1 and Macro F1 scores as key performance metrics. The Micro F1 score captures overall accuracy by considering all predictions equally, while the Macro F1 score accounts for per-label performance, ensuring that the model

DATASET	APPROACH	Micro F1	Macro F1
DBLP	Best RAGE performance with Claude	0.8575	0.8584
	Best MoT performance with Claude	**0.90259**	**0.90533**
	Best RAGE performance with GPT-4	0.8915	0.8916
	Best MoT performance with GPT-4	0.81481	0.81086
GOODREADS	Best RAGE performance with Claude	0.805	0.825
	Best MoT performance with Claude	0.904	0.909
	Best RAGE performance with GPT-4	**0.906**	**0.910**
	Best MoT performance with GPT-4	0.876	0.866

Table 1: Performance Comparison of RAGE and MoT across datasets and models

is not biased toward more frequent classes. These metrics help assess both the general effectiveness and class-wise robustness of the model.

Additionally, the confusion matrix provides insights into label-wise performance, identifying common misclassifications and highlighting areas where the model struggles. By analyzing misclassification patterns, we assess the robustness of the model across different categories of research papers, particularly in distinguishing between closely related fields.

5.2 Task 2: Qualitative Analysis of Explanation

In addition to evaluating node prediction performance, we assess the quality of explanations provided by RAGE and Metapath of Thought (MoT) to determine how well each approach justifies its predictions. Since generating factually accurate, clear, and contextually grounded explanations is crucial for interpretability in heterogeneous graph classification, we employ a structured ranking and rating system to compare the responses from both models.

For a given research paper, both RAGE and MoT generate an explanation along with a predicted research area. These responses are then ranked based on clarity and factual correctness, with the more informative and precise explanation receiving a higher rank. Additionally, each response is rated on a 1-3 scale across five key dimensions:

- Clarity: Measures how well-structured and comprehensible the explanation is.
- Relevance: Assesses how directly the response addresses the research area of the given paper.
- Depth: Evaluates whether the explanation provides meaningful insight into the model's reasoning.
- Accuracy: Determines whether the response is factually correct and free of hallucinations.
- Consistency: Ensures the explanation is logically coherent and aligns with known domain knowledge.

A score of 3 denotes a clear, well-structured, and highly relevant explanation, while a score of 1 indicates a vague, off-topic, or factually incorrect response. By aggregating these scores across multiple test samples, we quantitatively compare the explanation quality of RAGE and MoT to identify strengths and weaknesses in their respective approaches.

6 Results

We evaluate the performance of RAGE and MoT across two heterogeneous graph datasets (DBLP and Goodreads) using two different LLMs (Claude and GPT-4). Table 1 presents the Micro F1 and Macro F1 scores for both approaches.The results indicate that both approaches achieve comparable node classification performance across datasets and models suggesting that retrieving subgraphs instead of relying on metapath-based reasoning can be equally or more effective in heterogeneous graph learning tasks.

For the DBLP dataset, MoT achieves the highest performance with Claude (Micro F1: 0.9025, Macro F1: 0.9053), while RAGE attains its best performance with GPT-4 (Micro F1: 0.8915, Macro F1: 0.8916). The difference in scores is minimal, highlighting that both frameworks offer strong predictive capabilities when provided with appropriate contextual information.

In contrast, for the Goodreads dataset, RAGE demonstrates a clear advantage over MoT with GPT-4, surpassing all MoT scores with a 0.906 Micro F1 and 0.910 Macro F1. This reinforces that retrieving direct subgraphs can be an effective alternative to metapath-based approaches, particularly for datasets rich in textual attributes like Goodreads.

Table 2 presents a comparative analysis of RAGE and the metapath-based approach under varying prompt configurations, evaluating the impact of 1, 3, and 5 random context examples per label. The results reveal that while both methods maintain competitive performance, RAGE exhibits greater robustness across different configurations.

In addition to quantitative evaluation, we assess the quality of explanations generated by RAGE and MoT using a structured ranking and rating system. Table 3 presents a comparative analysis of explanation quality, evaluating responses based on clarity, factual accuracy, and relevance.

The results indicate a significant advantage of RAGE in generating high-quality explanations. Specifically, in the DBLP dataset, RAGE explanations received a higher percentage of top ratings (90.10% with a score of 3). Similarly, for the Goodreads dataset, RAGE achieved 95.45% of top-rated explanations, far surpassing MoT's 72.71%. Furthermore, RAGE consistently ranked higher in terms of explanation quality, with 70.72% of its responses earning

DATASET	APPROACH	Context + 1 example		Context + 3 example		Context + 5 example	
		Micro f1	Macro f1	Micro f1	Macro f1	Micro f1	Macro f1
DBLP	RAGE performance with Claude	0.8575	0.8584	0.841	0.845	0.853	0.849
	MoT performance with Claude	-	-	-	-	0.876	0.882
	RAGE performance with GPT-4	0.8835	0.8834	0.8725	0.8671	**0.8915**	**0.8916**
	MoT performance with GPT-4	0.801	0.815	0.791	0.78927	0.8181	0.81086
GOODREADS	RAGE performance with Claude	0.805	0.825	0.7938	0.7812	0.827	0.831
	MoT performance with Claude	-	-	-	-	0.877	0.886
	RAGE performance with GPT-4	0.8545	0.8844	0.8693	0.8751	**0.906**	**0.910**
	MoT performance with GPT-4	0.827	0.8192	0.8519	0.8563	0.8764	0.8661

Table 2: Parameter Variation across approaches

Dataset	Approach	Rank 1	% of score 3 for quality of explanation	% of score 2 for quality of explanation
DBLP	**RAGE**	**70.72**	**90.10**	9.90
	MOT	29.28	84.60	14.40
GOODREADS	**RAGE**	**83.76**	**95.45**	4.55
	MOT	16.24	72.71	27.29

Table 3: Comparison of quality of Explanations generated for the two approaches

Rank 1 in DBLP and 83.76% in Goodreads, demonstrating its ability to provide clearer and more factually accurate justifications.

While MoT and RAGE achieve comparable F1 scores, RAGE demonstrates enhancement in the quality of explanations, providing more precise, contextually relevant, and factually consistent justifications for predictions. These results reinforce that RAGE not only maintains strong predictive capabilities but also enhances interpretability by providing more precise and contextually relevant justifications for its predictions. RAGE offers several advantages that make it a more **scalable and efficient approach for heterogeneous graph reasoning**. One fundamental distinction between the two frameworks lies in how they retrieve contextual information. MoT relies on a multi-step pipeline, where metapaths are first generated using a GNN, followed by prompting an LLM with explanations of these metapaths before generating predictions. In contrast, RAGE directly retrieves relevant subgraphs from the heterogeneous graph, eliminating the need for an intermediate GNN-based metapath generation step. This direct retrieval ensures that the contextual information remains **closely aligned with the original dataset** without additional abstraction layers.

Additionally, RAGE provides better scalability and computational efficiency. The metapath generation process in MoT is inherently computationally intensive and often requires task-specific fine-tuning, limiting its adaptability to new datasets. RAGE, however, scales naturally across heterogeneous graphs by retrieving relevant subgraphs without requiring dataset-specific pre-processing or training on metapath structures. This flexibility makes it a more generalizable solution for various heterogeneous graph datasets.

Another key advantage of RAGE is its **stronger grounding in the original dataset**, as it retrieves actual subgraph structures rather than relying on metapaths. This ensures greater factual consistency in predictions and explanations. In contrast, since MoT depends on predefined metapath templates, it is more susceptible to biases introduced by GNN-generated paths, which may not always capture the most relevant contextual relationships in the original data. As a result, RAGE offers a more direct, interpretable, and scalable framework for improving explanation quality and reducing hallucinations in LLM-generated outputs for heterogeneous graphs.

7 Conclusion

In this work, we introduced RAGE, a retrieval-augmented framework designed to enhance both prediction accuracy and explanation quality for heterogeneous graphs. By retrieving and integrating relevant subgraphs directly into LLM prompts, RAGE eliminates the need for intermediate metapath generation, offering a more direct, interpretable, and computationally efficient alternative to metapath-based approaches. Our extensive evaluations on DBLP and Goodreads datasets, across multiple LLMs, demonstrate that RAGE achieves competitive node classification performance while significantly improving explanation clarity, relevance, and factual accuracy.

A key strength of RAGE lies in its scalability and adaptability. The framework utilizes a vector database for efficient retrieval. While this approach proves effective for textually attributed heterogeneous graphs, future work could explore the applicability of GraphDB-based retrieval, particularly for non-TAG heterogeneous graphs where structural relationships may be more dominant than textual attributes.

Beyond academic benchmarks, RAGE holds significant potential for real-world applications. Its ability to generate interpretable explanations makes it valuable for scientific literature analysis, recommendation systems, and biomedical knowledge graphs, where explainability is crucial for trust and transparency. Expanding RAGE to include more diverse and complex heterogeneous graph datasets, would further validate its generalizability and establish its broader applicability across domains.

While we evaluated the performance of RAGE across multiple LLMs, our primary objective was to assess its impact on explanation quality for heterogeneous graph tasks rather than to compare the LLMs themselves. Our core contribution lies in demonstrating the effectiveness of structured subgraph retrieval in enhancing explanation clarity, factual consistency, and interpretability. Future work can explore how different LLM architectures influence retrieval-augmented explanations, further refining the balance between contextual grounding and model adaptability.

This research lays the foundation for more interpretable, scalable, and efficient heterogeneous graph learning, bridging the gap between structured graph explanation and natural language understanding.

References

[1] Baolong Bi, Shenghua Liu, Yiwei Wang, Lingrui Mei, and Xueqi Cheng. 2024. LPNL: Scalable Link Prediction with Large Language Models. arXiv:2401.13227 [cs.CL] https://arxiv.org/abs/2401.13227

[2] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. arXiv:2005.14165 [cs.CL] https://arxiv.org/abs/2005.14165

[3] 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. arXiv:2310.05845 [cs.CL] https://arxiv.org/abs/2310.05845

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv:1810.04805 [cs.CL] https://arxiv.org/abs/1810.04805

[5] Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified Language Model Pre-training for Natural Language Understanding and Generation. arXiv:1905.03197 [cs.CL] https://arxiv.org/abs/1905.03197

[6] Wenqi Fan, Yujuan Ding, Liangbo Ning, Shijie Wang, Hengyun Li, Dawei Yin, Tat-Seng Chua, and Qing Li. 2024. A Survey on RAG Meeting LLMs: Towards Retrieval-Augmented Large Language Models. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (Barcelona, Spain) *(KDD '24)*. Association for Computing Machinery, New York, NY, USA, 6491–6501. doi:10.1145/3637528.3671470

[7] Yi Fang, Dongzhe Fan, Daochen Zha, and Qiaoyu Tan. 2024. GAugLLM: Improving Graph Contrastive Learning for Text-Attributed Graphs with Large Language Models. arXiv:2406.11945 [cs.LG] https://arxiv.org/abs/2406.11945

[8] Xiaoxin He, Yijun Tian, Yifei Sun, Nitesh V. Chawla, Thomas Laurent, Yann LeCun, Xavier Bresson, and Bryan Hooi. 2024. G-Retriever: Retrieval-Augmented Generation for Textual Graph Understanding and Question Answering. arXiv:2402.07630 [cs.LG] https://arxiv.org/abs/2402.07630

[9] Bowen Jin, Gang Liu, Chi Han, Meng Jiang, Heng Ji, and Jiawei Han. 2024. Large Language Models on Graphs: A Comprehensive Survey. arXiv:2312.02783 [cs.CL] https://arxiv.org/abs/2312.02783

[10] Bowen Jin, Yu Zhang, Qi Zhu, and Jiawei Han. 2023. Heterformer: Transformer-based Deep Node Representation Learning on Heterogeneous Text-Rich Networks. arXiv:2205.10282 [cs.CL] https://arxiv.org/abs/2205.10282

[11] Yuxuan Lei, Jianxun Lian, Jing Yao, Xu Huang, Defu Lian, and Xing Xie. 2024. RecExplainer: Aligning Large Language Models for Explaining Recommendation Models. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '24)*. ACM, 1530–1541. doi:10.1145/3637528.3671802

[12] Jiarui Li, Ye Yuan, and Zehua Zhang. 2024. Enhancing LLM Factual Accuracy with RAG to Counter Hallucinations: A Case Study on Domain-Specific Queries in Private Knowledge-Bases. arXiv:2403.10446 [cs.CL] https://arxiv.org/abs/2403.10446

[13] Yun Li, Yi Yang, Jiaqi Zhu, Hui Chen, and Hongan Wang. 2024. LLM-Empowered Few-Shot Node Classification on Incomplete Graphs with Real Node Degrees. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management* (Boise, ID, USA) *(CIKM '24)*. Association for Computing Machinery, New York, NY, USA, 1306–1315. doi:10.1145/3627673.3679861

[14] Costas Mavromatis and George Karypis. 2024. GNN-RAG: Graph Neural Retrieval for Large Language Model Reasoning. arXiv:2405.20139 [cs.CL] https://arxiv.org/abs/2405.20139

[15] Alec Radford and Karthik Narasimhan. 2018. Improving Language Understanding by Generative Pre-Training. https://api.semanticscholar.org/CorpusID:49313245

[16] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language Models are Unsupervised Multitask Learners. https://api.semanticscholar.org/CorpusID:160025533

[17] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. arXiv:1908.10084 [cs.CL] https://arxiv.org/abs/1908.10084

[18] Harshvardhan Solanki, Jyoti Singh, Yihui Chong, and Ankur Teredesai. 2024. Metapath of thoughts: Verbalized metapaths in heterogeneous graph as contextual augmentation to LLM. (2024). https://www.amazon.science/publications/metapath-of-thoughts-verbalized-metapaths-in-heterogeneous-graph-as-contextual-augmentation-to-llm

[19] Yiming Tan, Dehai Min, Yu Li, Wenbo Li, Nan Hu, Yongrui Chen, and Guilin Qi. 2023. Evaluation of ChatGPT as a Question Answering System for Answering Complex Questions. doi:10.48550/arXiv.2303.07992

[20] Mengting Wan, Rishabh Misra, Ndapa Nakashole, and Julian McAuley. 2019. Fine-Grained Spoiler Detection from Large-Scale Review Corpora. arXiv:1905.13416 [cs.CL] https://arxiv.org/abs/1905.13416

[21] Seongjun Yun, Minbyul Jeong, Sungdong Yoo, Seunghun Lee, Sean Yi, Raehyun Kim, Jaewoo Kang, and Hyunwoo Kim. 2022. Graph Transformer Networks: Learning meta-path graphs to improve GNNs. *Neural Networks* 153 (06 2022). doi:10.1016/j.neunet.2022.05.026

[22] Tianyi Zhang, Faisal Ladhak, Esin Durmus, Percy Liang, Kathleen McKeown, and Tatsunori B. Hashimoto. 2023. Benchmarking Large Language Models for News Summarization. arXiv:2301.13848 [cs.CL] https://arxiv.org/abs/2301.13848

[23] Siyun Zhao, Yuqing Yang, Zilong Wang, Zhiyuan He, Luna K. Qiu, and Lili Qiu. 2024. Retrieval Augmented Generation (RAG) and Beyond: A Comprehensive Survey on How to Make your LLMs use External Data More Wisely. arXiv:2409.14924 [cs.CL] https://arxiv.org/abs/2409.14924