GFM-RAG: Graph Foundation Model for Retrieval Augmented Generation

Linhao Luo[1]*, Zicheng Zhao[2]*, Gholamreza Haffari[1], Chen Gong[3], Dinh Phung[1], Shirui Pan[4]†
[1]Monash University, [2]Nanjing University of Science and Technology,
[3]Shanghai Jiao Tong University, [4]Griffith University,
{Linhao.Luo,gholamreza.haffari,dinh.phung}@monash.edu,
zicheng.zhao@njust.edu.cn, chen.gong@sjtu.edu.cn, s.pan@griffith.edu.au
Project page: https://rmanluo.github.io/gfm-rag

Abstract

Retrieval-augmented generation (RAG) has proven effective in integrating knowledge into large language models (LLMs). However, conventional RAGs struggle to capture complex relationships between pieces of knowledge, limiting their performance in intricate reasoning that requires integrating knowledge from multiple sources. Recently, graph-enhanced retrieval augmented generation (GraphRAG) builds graph structure to explicitly model these relationships, enabling more effective and efficient retrievers. Nevertheless, its performance is still hindered by the noise and incompleteness within the graph structure. To address this, we introduce GFM-RAG, a novel graph foundation model (GFM) for retrieval augmented generation. GFM-RAG is powered by an innovative graph neural network that reasons over graph structure to capture complex query-knowledge relationships. The GFM with 8M parameters undergoes a two-stage training process on large-scale datasets, comprising 60 knowledge graphs with over 14M triples and 700k documents. This results in impressive performance and generalizability for GFM-RAG, making it the first graph foundation model applicable to unseen datasets for retrieval without any domain-specific fine-tuning required. Extensive experiments on three multi-hop QA datasets and seven domain-specific RAG datasets demonstrate that GFM-RAG achieves state-of-the-art performance while maintaining efficiency and alignment with neural scaling laws, highlighting its potential for further improvement.

1 Introduction

Recent advancements in large language models (LLMs) [47, 42, 70] have greatly propelled the evolution of natural language processing, positioning them as foundational models for artificial general intelligence (AGI). Despite the remarkable reasoning ability [48], LLMs are still limited in accessing real-time information and lack of domain-specific knowledge, which is outside the pre-training corpus. To address these limitations, retrieval-augmented generation (RAG) [12] has become a popular paradigm in adding new knowledge to the static LLMs by retrieving relevant documents into the context of LLM generation.

Existing RAG methods typically retrieve documents independently, making it difficult to capture complex relationships between pieces of knowledge [30, 5, 43]. This limitation hampers the performance of LLMs in integrating knowledge across document boundaries, particularly in multi-hop reasoning tasks [72, 63] and real-world applications like legal judgment [28] and medical diagnoses [25], which

*Equal Contribution.
†Corresponding author.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

require reasoning over multiple sources. Although recent methods have expanded the retrieval process into multiple steps and incorporate LLM reasoning, they still encounter high computational costs due to iterative retrieval and reasoning with LLMs [64, 59, 26].

Recently, graph-enhanced retrieval augmented generation (GraphRAG) [51, 17] has emerged as a novel solution that builds a graph structure to explicitly model the intricate relationships between knowledge. This enables the development of a graph-enhanced retriever to identify relevant information using graphs. The structural nature of graphs allows GraphRAG to capture global context and dependencies among documents, significantly improving reasoning across multiple sources [9]. Methods like HippoRAG [16] enhance retrieval by employing a personalized PageRank algorithm to locate relevant knowledge with graphs. However, these algorithms rely solely on the graph structure, which is often noisy or incomplete, limiting their overall performance. Alternative methods [41, 18] incorporate graph neural networks (GNNs) into the retrieval process. These methods have shown impressive performance due to GNNs' powerful multi-hop reasoning capabilities on graphs [73]. Nevertheless, they still face limitations in generalizability since they require training from scratch on new datasets.

Nowadays, the search for a foundation GNN model that can transfer and generalize across different datasets has been an active research topic. Ideally, a foundation GNN or graph foundation model (GFM) can benefit from large-scale training and generalize across diverse graphs [40, 37]. Efforts have been made to identify transferable graph tokens (e.g., motifs, sub-trees, and relation graphs) [11, 66, 68] that can be shared among different graphs for GFM design. However, these methods primarily focus on graph-related tasks (e.g., node classification and link prediction), leaving the design of a GFM to enhance LLMs' reasoning ability unexplored.

To bridge the gap, in this paper, we propose an effective, efficient, and general graph foundation model for retrieval augmented generation (`GFM-RAG`), thereby enhancing LLMs' reasoning ability. As shown in Figure 1, we create a *knowledge graph index* (KG-index) from documents in each dataset. The KG-index consists of interconnected factual triples pointing to the original documents, which serves as a structural knowledge index across multiple sources, enhancing the integration of diverse knowledge for complex reasoning tasks [16]. Then, we present the *graph foundation model retriever* (GFM retriever), driven by a query-dependent GNN that captures complex query-knowledge relationships in a unified, transferable space of semantics and graph structure. Through multi-layer message passing, the GFM retriever enables efficient multi-hop retrieval in a single step, surpassing previous multi-step methods. The GFM retriever, with 8M parameters, undergoes a two-stage training: *self-supervised KG completion pre-training* and *supervised document retrieval fine-tuning* on large-scale datasets, including 60 knowledge graphs with over 14M triples and 700k documents. This large-scale training ensures the generalizability of GFM retriever to be applied to unseen datasets without further training.

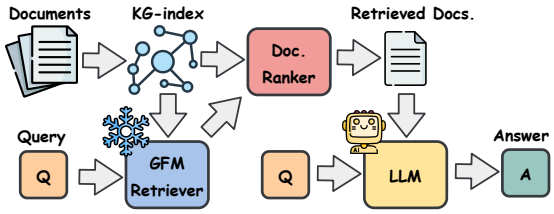

Figure 1: The overview framework of `GFM-RAG`.

In experiments, `GFM-RAG` achieves state-of-the-art performance across three multi-hop QA datasets, demonstrating its effectiveness and efficiency in multi-hop reasoning. It also generalizes well across seven RAG datasets from diverse domains, such as biomedical, customer service, and general knowledge, without requiring additional training. Furthermore, `GFM-RAG` follows the neural scaling law [19], whose performance benefits from training data and model size scaling, emphasizing its potential as a foundational model for future improvements. The main contributions of this paper are as follows:

- We introduce a graph foundation model for retrieval augmented generation (`GFM-RAG`), powered by a novel query-dependent GNN to enable efficient multi-hop retrieval within a single step.
- We train a large-scale model with 8M parameters, marking the first graph foundation model (GFM) that can be applied directly to various unseen datasets for retrieval augmented generation.
- We evaluate `GFM-RAG` on three multi-hop QA datasets and seven domain-specific RAG datasets, achieving state-of-the-art performance across all, demonstrating its effectiveness, efficiency, generalizability, and potential as a foundational model for further enhancement.

2 Related Work

Retrieval-augmented generation (RAG) [12] provides an effective way to integrate external knowledge into large language models (LLMs) by retrieving relevant documents to facilitate LLM generation. Early works adopt the pre-trained dense embedding model to encode documents as separate vectors [30, 5, 34, 43], which are then retrieved by calculating the similarity to the query. Despite efficiency and generalizability, these methods struggle to capture complex document relationships. Subsequent studies have explored multi-step retrieval, where LLMs guide an iterative process to retrieve and reason over multiple documents [64, 24, 58]. However, this approach is computationally expensive.

Graph-enhanced retrieval augmented generation (GraphRAG) [51, 17] is a novel approach that builds graphs to explicitly model the complex relationships between knowledge, facilitating comprehensive retrieval and reasoning. Early research focuses on retrieving information from existing knowledge graphs (KGs), such as WikiData [65] and Freebase [3], by identifying relevant facts or reasoning paths [33, 38, 50]. Recent studies have integrated documents with KGs to improve knowledge coverage and retrieval [9, 35]. A graph structure is built from these documents to aid in identifying relevant content for LLM generation [8]. Based on graphs, LightRAG [15] incorporates graph structures into text indexing and retrieval, enabling efficient retrieval of entities and their relationships. HippoRAG [16] enhances multi-hop retrieval by using a personalized PageRank algorithm to locate relevant knowledge with graphs. However, the graph structure can be noisy and incomplete, leading to suboptimal performance. Efforts to incorporate GNNs into graph-enhanced RAG [41, 18] have shown impressive results due to the multi-hop graph reasoning capabilities of GNNs in handling incomplete graphs [73]. Nonetheless, these methods still limit in generalizability due to the lack of a graph foundational model.

Graph Foundation models (GFM) aims to be a large-scale model that can generalize to various datasets [40, 37]. The main challenge in designing GFMs is identifying graph tokens that capture invariance across diverse graph data. For instance, ULTRA [11] employs four fundamental relational interactions in knowledge graphs (KGs) to create a GFM with 0.2M parameters for link prediction. OpenGraph [68] develops a graph tokenizer that converts graphs into a unified node token representation, enabling transformer-like GFMs for tasks such as link prediction and node classification. GFT [66] introduces a transferable tree vocabulary to construct a GFM that demonstrates effectiveness across various tasks and domains in graph learning. Despite these successful efforts, most methods primarily focus on conventional graph-related tasks, and transformer-like GFMs [61, 60] struggle with large-scale graphs and capture logical association [52]. How to design a GNN-based GFM to enhance the reasoning of LLM remains an open question.

3 Approach

The proposed `GFM-RAG` essentially implements a GraphRAG paradigm by constructing graphs from documents and using a graph-enhanced retriever to retrieve relevant documents.

GFM-RAG Overview. Given a set of documents $\mathcal{D} = \{D_1, D_2, \ldots, D_{|\mathcal{D}|}\}$, we construct a knowledge graph $\mathcal{G} = \{(e, r, e') \in \mathcal{E} \times \mathcal{R} \times \mathcal{E}\}$, where $e, e' \in \mathcal{E}$ and $r \in \mathcal{R}$ denote the set of entities and relations extracted from $\mathcal{D}$, respectively. For a user query q, we aim to design a graph-enhanced retriever to obtain relevant documents from $\mathcal{D}$ by leveraging the knowledge graph $\mathcal{G}$. The whole `GFM-RAG` process can be formulated as:

$$\mathcal{G} = \text{KG-index}(\mathcal{D}), \tag{1}$$

$$\mathcal{D}^K = \text{GFM-Retriever}(q, \mathcal{D}, \mathcal{G}), \tag{2}$$

$$a = \text{LLM}(q, \mathcal{D}^K). \tag{3}$$

In the first step, KG-index$(\cdot)$ constructs a knowledge graph index $\mathcal{G}$ from the document corpus $\mathcal{D}$, followed by our proposed *graph foundation model retriever* (GFM-Retriever), which is pre-trained on large-scale datasets. It retrieves top-K documents based on any user query q and knowledge graph index $\mathcal{G}$. The retrieved documents $\mathcal{D}^K$, along with the query q, are then input into a large language model (LLM) to generate the final answer a. These three main components in `GFM-RAG` are illustrated in Figure 2 and will be detailed next.

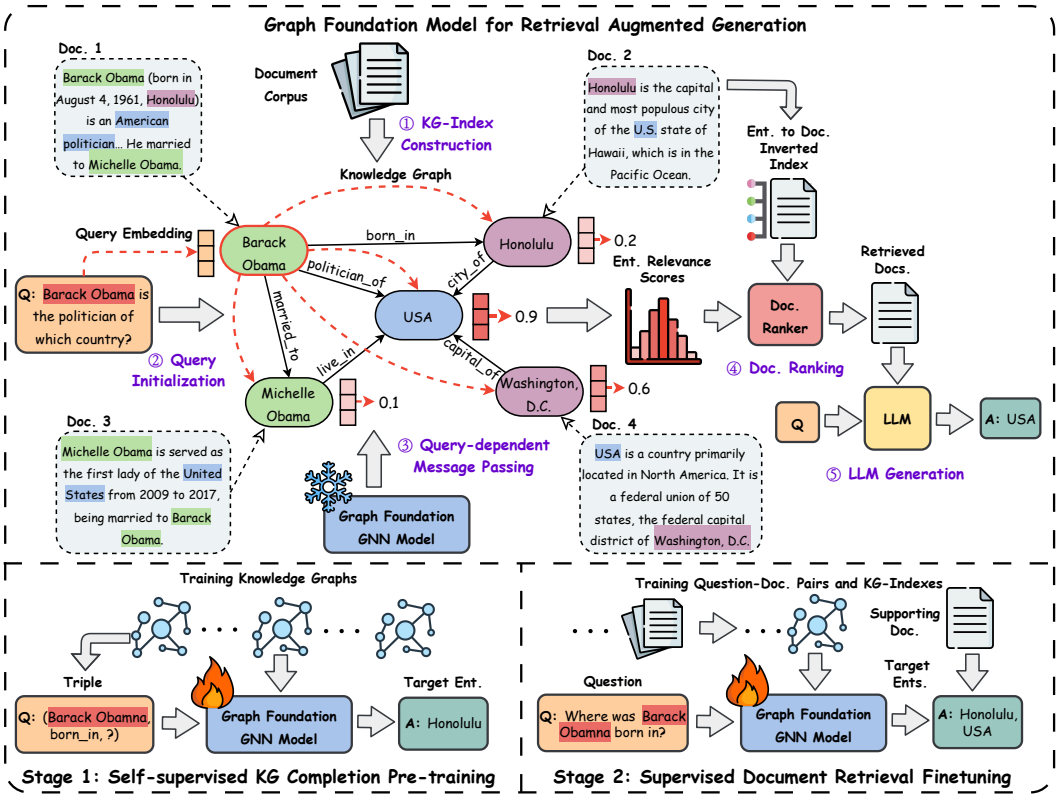

Figure 2: The detailed framework of GFM-RAG and training processes of graph foundation model. The GFM-RAG consists of three main components: A. *KG-index construction*, which constructs a knowledge graph index from document corpus (①); B. *graph foundation model retriever* (GFM retriever), which is pre-trained on large-scale datasets and could retrieve documents based on any user query and KG-index (②③); and C. *documents ranking and answer generation*, which ranks retrieved documents and generates final answer (④⑤).

3.1 KG-index Construction

Conventional embedding-based index methods encode documents as separate vectors [30, 5, 43], which are limited in modeling the relationships between them. Knowledge graphs (KGs), on the other hand, explicitly capturing the relationships between millions of facts, can provide a structural index of knowledge across multiple documents [9, 16]. The structural nature of the KG-index aligns well with the human hippocampal memory indexing theory [62], where the KG-index functions like an artificial hippocampus to store associations between knowledge memories, enhancing the integration of diverse knowledge for complex reasoning tasks [16].

To construct the KG-index, given a set of documents $\mathcal{D}$, we first extract entities $\mathcal{E}$ and relations $\mathcal{R}$ to form triples $\mathcal{T}$ from documents. Then, the entity to document inverted index $M \in \{0,1\}^{|\mathcal{E}|\times|\mathcal{D}|}$ is constructed to record the entities mentioned in each document. Such a process can be achieved by existing open information extraction (OpenIE) tools [1, 77, 49]. To better capture the connection between knowledge, we further conduct the entity resolution [13, 74] to add additional edges $\mathcal{T}^+$ between entities with similar semantics, e.g., (`USA`, `equivalent`, `United States of America`). Therefore, the final KG-index $\mathcal{G}$ is constructed as $\mathcal{G} = \{(e, r, e') \in \mathcal{T} \cup \mathcal{T}^+\}$. In implementation, we leverage an LLM [47] as the OpenIE tool (prompts are shown in Table 22) and a pre-trained dense embedding model [55] for entity resolution. Details can be found in Appendix D.1.

3.2 Graph Foundation Model (GFM) Retriever

The GFM retriever is designed to retrieve relevant documents based on any user query and the constructed KG-index. While the KG-index offers a structured representation of knowledge, it still

suffers from incompleteness and noise, resulting in suboptimal retrieval performance when solely relying on its structure [16]. Recently, graph neural networks (GNNs) [67] have shown impressive multi-hop reasoning ability by capturing the complex relationships between knowledge for retrieval or question answering [41, 18]. However, existing GNNs are limited in generalizability, as they are usually trained on specific graphs [40, 37], which limits their application to unseen corpora and KGs. Therefore, there is still a need for a graph foundation model that can be directly applied to unseen datasets and KGs without additional training.

To address these issues, we propose the first graph foundation model-powered retriever (GFM retriever), which harnesses the graph reasoning ability of GNNs to capture the complex relationships between queries, documents, and knowledge graphs in a unified and transferable space. The GFM retriever employs a query-dependent GNN to identify relevant entities in graphs that will aid in locating pertinent documents. After pre-training on large-scale datasets, the GFM retriever can be directly applied to new corpora and KGs without further training.

3.2.1 Query-dependent GNN

Conventional GNNs [14] follow the message passing paradigm, which iteratively aggregates information from neighbors to update entity representations. Such a paradigm is not suitable for the GFM retriever as it is graph-specific and neglects the relevance of queries. Recent query-dependent GNNs [78, 11] have shown promising results in capturing query-specific information and generalizability to unseen graphs, which is essential for the GFM retriever and can be formulated as:

$$H_q^L = \text{GNN}_q(q, \mathcal{G}, H^0), \tag{4}$$

where $H^0 \in \mathbb{R}^{|\mathcal{E}| \times d}$ denotes initial entity features, and H_q^L denotes the updated entity representations conditioned on query q after L layers of query-dependent message passing.

The query-dependent GNN is theoretically proven to exhibit multi-hop logical reasoning ability [21, 73, 52] (detailed in Appendix A), which is selected as the backbone of our GFM retriever. It allows the GFM retriever to dynamically adjust the message passing process based on user queries and find the most relevant information on the graph with multi-hop reasoning. The path interpretation for this multi-hop reasoning process is shown in Section 4.8.

Query Initialization. Given a query q, we first encode it into a query embedding with a sentence embedding model:

$$\boldsymbol{q} = \text{SentenceEmb}(q),\ \boldsymbol{q} \in \mathbb{R}^d, \tag{5}$$

where d denotes the dimension of the query embedding. Then, for all the entities mentioned in the query $e_q \in \mathcal{E}_q \subseteq \mathcal{E}$, we initialize their entity features as $\boldsymbol{q}$ while others as zero vectors:

$$H^0 = \begin{cases} \boldsymbol{q}, & e \in \mathcal{E}_q, \\ \mathbf{0}, & \text{otherwise}. \end{cases} \tag{6}$$

Query-dependent Message Passing. The query-dependent message passing will propagate the information from the question entities to other entities in the KG to capture their relevance to the query. The message passing process can be formulated as:

Triple-level:

$$h_r^0 = \text{SentenceEmb}(r),\ h_r^0 \in \mathbb{R}^d, \tag{7}$$

$$m_e^{l+1} = \text{Msg}(h_e^l, g^{l+1}(h_r^l), h_{e'}^l), (e, r, e') \in \mathcal{G}, \tag{8}$$

Entity-level:

$$h_e^{l+1} = \text{Update}(h_e^l, \text{Agg}(\{m_{e'}^{l+1} | e' \in \mathcal{N}_r(e), r \in \mathcal{R}\})), \tag{9}$$

where h_e^l, h_r^l denote the entity and relation embeddings at layer l, respectively. The relation embeddings h_r^0 are also initialized using the same sentence embedding model as the query, reflecting their semantics (e.g., "`born_in`"), and updated by a layer-specific function $g^{l+1}(\cdot)$, implemented as a 2-layer MLP. The $\text{Msg}(\cdot)$ is operated on all triples in the KG to generate messages, which is implemented with a non-parametric DistMult [71] following the architecture of NBFNet [78]. For each entity, we aggregate the messages from its neighbors $\mathcal{N}_r(e)$ with relation r using sum and update the entity representation with a single linear layer.

After L layers message passing, a final MLP layer together with a sigmoid function maps the entity embeddings to their relevance scores to the query:

$$P_q = \sigma(\text{MLP}(H_q^L)),\ P_q \in \mathbb{R}^{|\mathcal{E}|\times 1}. \tag{10}$$

Generalizability. Since the query, entity, and relation embeddings are initialized using the same sentence embedding model with identical dimensions, the query-dependent GNN can be directly applied to different queries and KGs. This allows it to learn complex relationships between queries and entities by taking into account both the semantics and structure of the KG through training on large-scale datasets.

3.2.2 Training Process

Training Objective. The training objective of the GFM retriever is to maximize the likelihood of the relevant entities to the query, which can be optimized by minimizing the binary cross-entropy (BCE) loss:

$$\mathcal{L}_{\text{BCE}} = -\frac{1}{|\mathcal{A}_q|}\sum_{e\in\mathcal{A}_q}\log P_q(e) - \frac{1}{|\mathcal{E}^-|}\sum_{|\mathcal{E}^-|}\log(1-P_q(e)), \tag{11}$$

where $\mathcal{A}_q$ denotes the set of target relevant entities to the query q, and $\mathcal{E}^- \subseteq \mathcal{E}\setminus\mathcal{A}_q$ denotes the set of negative entities sampled from the KG. However, due to the sparsity of the target entities, the BCE loss may suffer from the gradient vanishing problem [36]. To address this issue, we further introduce the ranking loss [2] to maximize the margin between the positive and negative entities:

$$\mathcal{L}_{\text{RANK}} = -\frac{1}{|\mathcal{A}_q|}\sum_{e\in\mathcal{A}_q}\frac{P_q(e)}{\sum_{e'\in\mathcal{E}^-}P_q(e')}. \tag{12}$$

The final training objective is the weighted combination of the BCE loss and ranking loss:

$$\mathcal{L} = \alpha\mathcal{L}_{\text{BCE}} + (1-\alpha)\mathcal{L}_{\text{RANK}}. \tag{13}$$

Self-supervised KG Completion Pre-training. To enhance the graph reasoning capability of the GFM retriever, we first pre-train it on a large-scale knowledge graph (KG) completion task. We sample a set of triples from the KG index and mask either the head or tail entity to create synthetic queries in the form $q = (e, r, ?)$ or $(?, r, e')$, with the masked entity serving as the target entity $\mathcal{A}_q = \{e\}$ or $\{e'\}$. The GFM retriever is then trained to predict the masked entity using both the query and the KG, as outlined in equation 13.

Supervised Document Retrieval Fine-tuning. After self-supervised pre-training, we supervised fine-tune the GFM retriever on a labeled document retrieval task. In this task, queries q are natural language questions, and target entities $\mathcal{A}_q$ are extracted from labeled supporting documents $\mathcal{D}_q$. The GFM retriever is trained to retrieve relevant entities from the KG index using the same training objective as in equation 13.

3.3 Documents Ranking and Answer Generation

Given the entity relevance scores $P_q \in \mathbb{R}^{|\mathcal{E}|\times 1}$ predicted by the GFM retriever, we first retrieve the top-T entities $\mathcal{E}_q^T$ with the highest relevance scores as:

$$\mathcal{E}_q^T = \arg\text{top-}T(P_q),\ \mathcal{E}_q^T = \{e_1, \ldots, e_T\}. \tag{14}$$

These retrieved entities are then used by the document ranker to obtain the final documents. To diminish the influence of popular entities, we weight the entities by the inverse of their frequency as entities mentioned in the document inverted index $M \in \{0,1\}^{|\mathcal{E}|\times|\mathcal{D}|}$ and calculate the final document relevance scores by summing the weights of entity mentioned in documents:

$$F_e = \begin{cases} \frac{1}{\sum_{d\in\mathcal{D}} M[e,d]}, & e \in \mathcal{E}_q^T, \\ 0, & \text{otherwise}, \end{cases} \tag{15}$$

$$P_d = M^\top F_e,\ P_d \in \mathbb{R}^{|\mathcal{D}|\times 1}. \tag{16}$$

The top-K documents are retrieved based on the document relevance scores P_d and fed into the context of LLMs, with a retrieval augmented generation manner, to generate the final answer:

$$\mathcal{D}^K = \arg\text{top-}K(P_d),\ \mathcal{D}^K = \{D_1, \ldots, D_K\}, \tag{17}$$

$$a = \text{LLM}(q, \mathcal{D}^K). \tag{18}$$

4 Experiment

In experiments, we aim to address the following research questions: (1) How does GFM-RAG perform in multi-hop retrieval and QA tasks? (Sections 4.2 and 4.3); (2) What are the efficiency and effectiveness of GFM-RAG in multi-hop retrieval? (Section 4.4); (3) How well does GFM-RAG generalize to unseen datasets as a foundation model? (Section 4.6); (4) How does the performance of GFM-RAG scale with training as a foundation model? (Section 4.7); (5) How to interpret GFM-RAG in multi-hop reasoning? (Section 4.8).

4.1 Experimental Setup

Datasets. We first evaluate the effectiveness of GFM-RAG on three widely-used multi-hop QA datasets, including HotpotQA [72], MuSiQue [63], and 2WikiMultiHopQA (2Wiki) [20]. We also evaluate the performance of GFM-RAG on seven RAG datasets from three domains, including biomedical [25], custom support [54, 44, 39, 4], and general knowledge [45, 27], to demonstrate the generalizability of GFM-RAG as the foundation model. The detailed statistics of the test datasets are shown in the Appendix B.

Baselines. We compare against several widely used retrieval methods under three categories: (1) *single-step naive methods*: BM25 [53], Contriever [22], GTR [46], ColBERTv2 [55], RAPTOR [56], Proposition [6]; (2) *graph-enhanced methods*: GraphRAG (MS) [9], LightRAG [15], HippoRAG [16], SubgraphRAG [32], G-retriever [18]; (3) *multi-step methods*: Adaptive-RAG [23], FLARE [24], and IRCoT [64] framework that can be integrated with arbitrary retrieval methods to conduct multi-step retrieval and reasoning. The detailed introduction of the baselines are shown in the Appendix C.

Metrics. For retrieval performance, we use recall@2 (R@2) and recall@5 (R@5) as evaluation metrics. For final QA performance, we use the EM score and F1 score following previous works [16].

Implementation Details. The GFM retriever is implemented with 6 query-dependent message passing layers with the hidden dimension set to 512. The pre-trained all-mpnet-v2 [57] is adopted as the sentence embedding model and is frozen during training. The total parameters of the GFM retriever are 8M, which is trained on 8 NVIDIA A100s (80G) with batch size 4, learning rate 5e-4, and loss weight $\alpha = 0.3$. The training data contains 60 KGs with over 14M triples constructed from 700k documents extracted from the training set. The statistics of training data are shown in Table 5, and the implementations are detailed in Appendix D.

4.2 Retrieval Performance

We first evaluate the retrieval performance of GFM-RAG against the baselines on three multi-hop QA datasets. As shown in Table 1, GFM-RAG achieves the best performance on all datasets, outperforming the SOTA IRCoT + HippoRAG by 16.8%, 8.3%, 19.8% in R@2 on HotpotQA, MuSiQue, and 2Wiki, respectively. The results demonstrate the effectiveness of GFM-RAG in multi-hop retrieval. From the result, we can observe that the naive single-step retrievers (e.g., BM25, RAPTOR) are outperformed by graph-enhanced HippoRAG, which highlights the significance of graph structure in multi-hop retrieval. Although GraphRAG (MS) and LightRAG use the graph structure, it struggles with multi-hop QA tasks as its retriever is designed for summarization and lacks multi-hop reasoning capability. With the help of LLMs, the multi-step retrieval pipeline IRCoT improves the performance of all single-step methods through iterative reasoning and retrieval. However, GFM-RAG still outperforms the multi-step methods by a large margin even with a single-step retrieval. This indicates that the GFM-RAG can effectively conduct the multi-hop reasoning in a single step (detailed in Section 4.8 and Appendix E.8), which is more efficient and effective than the multi-step retrieval pipeline (detailed in Section 4.4).

4.3 Question Answering Performance

We then evaluate the QA performance of GFM-RAG, as it is directly influenced by retrieval quality. We adopt the GPT-4o-mini [47] as LLM and use the top-5 retrieved documents for generating answers. From the results shown in Table 2, the single-step GFM-RAG has already achieved state-of-the-art

Table 1: Retrieval performance comparison.

Category	Method	HotpotQA		MuSiQue		2Wiki	
		R@2	R@5	R@2	R@5	R@2	R@5
Single-step	BM25	55.4	72.2	32.3	41.2	51.8	61.9
	Contriever	57.2	75.5	34.8	46.6	46.6	57.5
	GTR	59.4	73.3	37.4	49.1	60.2	67.9
	ColBERTv2	64.7	79.3	37.9	49.2	59.2	68.2
	RAPTOR	58.1	71.2	35.7	45.3	46.3	53.8
	Proposition	58.7	71.1	37.6	49.3	56.4	63.1
	GraphRAG (MS)	58.3	76.6	35.4	49.3	61.6	77.3
	LightRAG	38.8	54.7	24.8	34.7	45.1	59.1
	HippoRAG (Contriever)	59.0	76.2	41.0	52.1	71.5	89.5
	HippoRAG (ColBERTv2)	60.5	77.7	40.9	51.9	70.7	89.1
	SubgraphRAG	61.5	73.0	42.1	49.3	70.7	85.5
	G-retriever	53.3	65.5	38.8	45.1	60.8	67.8
Multi-step	Adaptive-RAG	61.0	76.4	35.1	44.7	44.7	61.4
	FLARE	73.1	81.3	44.3	55.1	67.1	73.1
	IRCoT + BM25	65.6	79.0	34.2	44.7	61.2	75.6
	IRCoT + Contriever	65.9	81.6	39.1	52.2	51.6	63.8
	IRCoT + ColBERTv2	67.9	82.0	41.7	53.7	64.1	74.4
	IRCoT + HippoRAG (Contriever)	65.8	82.3	43.9	56.6	75.3	93.4
	IRCoT + HippoRAG (ColBERTv2)	67.0	83.0	45.3	57.6	75.8	93.9
Single-step	GFM-RAG	**78.3**	**87.1**	**49.1**	**58.2**	**90.8**	**95.6**

Table 2: Question answering performance comparison.

Category	Retriever	HotpotQA		MuSiQue		2Wiki	
		EM	F1	EM	F1	EM	F1
Single-step	None	30.4	42.8	12.5	24.1	31.0	39.0
	ColBERTv2	43.4	57.7	15.5	26.4	33.4	43.3
	GraphRAG (MS)	35.3	54.6	13.4	29.5	28.3	46.9
	LightRAG	36.8	48.3	18.1	27.5	45.1	49.5
	HippoRAG (ColBERTv2)	41.8	55.0	19.2	29.8	46.6	59.5
Multi-step	Adaptive-RAG	45.5	59.6	13.8	25.6	48.9	62.8
	FLARE	48.7	60.6	16.2	28.4	46.7	65.4
	IRCoT (ColBERTv2)	45.5	58.4	19.1	30.5	35.4	45.1
	IRCoT + HippoRAG (ColBERTv2)	45.7	59.2	21.9	33.3	47.7	62.7
Single-step	GFM-RAG	51.6	66.9	30.2	40.4	69.8	77.7
Multi-step	IRCoT + GFM-RAG	**56.0**	**71.8**	**36.6**	**49.2**	**72.5**	**80.8**

performance against all other baselines. Meanwhile, we also integrate GFM-RAG with IRCoT to conduct multi-step retrieval and reasoning, which further improves the performance by 8.5%, 21.2%, 3.9% in EM on three datasets, respectively. The results demonstrate the effectiveness and great compatibility of GFM-RAG with an arbitrary multi-step framework in multi-hop reasoning tasks.

4.4 Efficiency Analysis

GFM-RAG achieves great efficiency in performing multi-step reasoning in a single step. As shown in Table 3, while the naive single-step methods get the best efficiency whose performance is not satisfying. Admittedly, the multi-step framework IRCoT could improve the performance, but it suffers from high computational costs due to the iterative retrieval and reasoning with LLMs. In contrast, GFM-RAG conducts multi-hop reasoning within a single-step GNN reasoning, which is more effective than single-step methods and more efficient than multi-step ones.

Table 3: Retrieval efficiency and performance comparison.

Method	HotpotQA		MuSiQue		2Wiki	
	Time (s)	R@5	Time (s)	R@5	Time (s)	R@5
ColBERTv2	**0.035**	79.3	**0.030**	49.2	**0.029**	68.2
HippoRAG	0.255	77.7	0.251	51.9	0.158	89.1
LightRAG	0.861	54.7	1.109	34.7	0.911	59.1
GraphRAG (MS)	2.759	76.6	3.037	49.3	1.204	77.3
IRCoT + ColBERTv2	1.146	82.0	1.152	53.7	2.095	74.4
IRCoT + HippoRAG	3.162	83.0	3.104	57.6	3.441	93.9
GFM-RAG	0.107	**87.1**	0.124	**58.2**	0.060	**95.6**

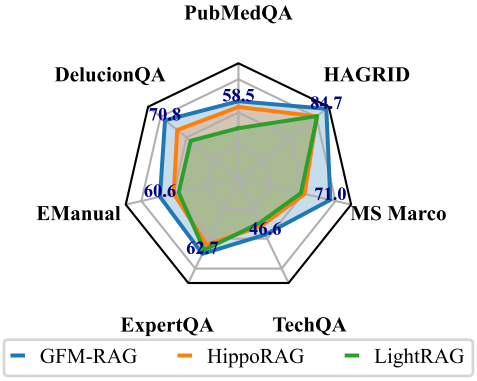

Figure 3: Model generalizability comparison.

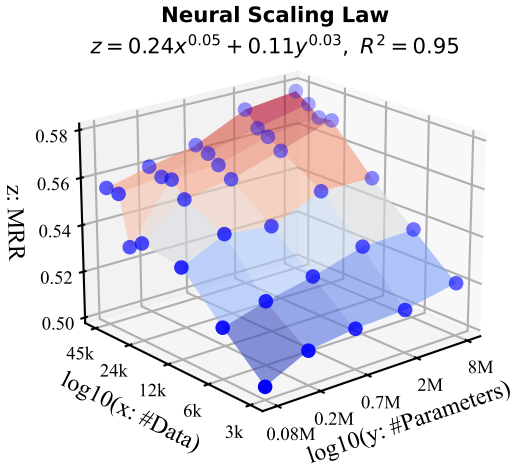

Figure 4: Neural scaling law of GFM-RAG.

4.5 Ablation Study

We conduct ablation studies to investigate the effectiveness of different components in GFM-RAG, including: different sentence embedding models (Appendix E.1), pre-training strategies (Appendix E.2), loss weighting strategies (Appendix E.3), ranking methods (Appendix E.4), training datasets (Appendix E.5), and the construction of KG-index (Appendix E.9). The results show that GFM-RAG is not sensitive to different sentence embedding models, and the pre-training strategy, as well as the loss weighting strategy, are both crucial for the performance of GFM-RAG.

4.6 Model Generalizability

To demonstrate the generalizability of GFM-RAG as a foundation model, we test the performance (R@5) of GFM-RAG on seven RAG datasets without any domain-specific fine-tuning. Specifically, we first build the KG-index from the documents in each dataset. Then, given the query, we use the pre-trained GFM retriever to retrieve the top-K documents with the help of the corresponding KG-index. As shown in Figure 3, GFM-RAG achieves the best performance on all datasets, outperforming the SOTA HippoRAG by 18.9% on average. The results demonstrate the generalizability of GFM-RAG as the foundation model which can be directly applied to various unseen datasets without any domain-specific fine-tuning. Additionally, results in Appendix E.6 demonstrate GFM-RAG's strong transferability for further performance improvement when fine-tuned on domain-specific datasets.

4.7 Model Neural Scaling Law

We further investigate the neural scaling law of GFM-RAG, which quantifies how model performance grows with the scale of training data and model parameter size. It has been validated in the recent

Table 4: Path interpretations of GFM for multi-hop reasoning, where r^{-1} denotes the inverse of original relation.

Question	What *football club* was owned by the singer of "*Grow Some Funk of Your Own*"?
Answer	Watford Football Club
Sup. Doc.	["Grow Some Funk of Your Own", "Elton John"]
Paths	1.095: (grow some funk of your own, is a song by, elton john) → (elton john, equivalent, sir elton hercules john) → (sir elton hercules john, named a stand after^{-1}, **watford football club**) 0.915: (grow some funk of your own, is a song by, elton john) → (elton john, equivalent, sir elton hercules john) → (sir elton hercules john, owned, **watford football club**)
Question	When was the judge born who made notable contributions to the trial of the man who tortured, raped, and murdered eight student nurses from *South Chicago Community Hospital* on the night of *July 13-14, 1966*?
Answer	June 4, 1931
Sup. Doc.	["Louis B. Garippo", "Richard Speck"]
Paths	0.797: (south chicago community hospital, committed crimes at^{-1}, richard speck) → (richard speck, equivalent, trial of richard speck) → (trial of richard speck, made contributions during^{-1}, **louis b garippo**) 0.412: (south chicago community hospital, were from^{-1}, eight student nurses) → (eight student nurses, were from, south chicago community hospital) → (south chicago community hospital, committed crimes at$^{-}1$, **richard speck**)

foundation models [29, 7]. As shown in Figure 4, the performance of GFM-RAG (MRR: z) scales well with the training data (x) and the model size (y), which can be fitted by the power-law scaling law $z \propto 0.24x^{0.05} + 0.11y^{0.03}$. The results demonstrate the scalability of GFM-RAG as the foundation model and potential for further improvement. The detailed analysis of the neural scaling law is shown in Appendix E.7.

4.8 Path Interpretations

GFM-RAG exhibits the multi-hop reasoning ability powered by the multi-layer GFM. We provide path interpretations of GFM-RAG for multi-hop reasoning in Table 4. Inspired by NBFNet [78], the paths' importance to the final prediction can be quantified by the partial derivative of the prediction score with respect to the triples at each layer (hop), defined as:

$$s_1, s_2, \ldots, s_L = \arg\text{top-}k \frac{\partial p_e(q)}{\partial s_l}. \tag{19}$$

The top-k path interpretations can be obtained by the top-k longest paths with beam search. We illustrate the path interpretations in Table 4. In the first example, GFM-RAG successfully deduces that the singer of the song has a football club named after him and that he owned it. In the second example, GFM-RAG identifies two paths related to the murder case and the judge presiding over the trial. These interpretations show that GFM-RAG exhibits the ability of multi-hop reasoning within single-step retrieval. We also illustrate the distribution the multi-hop prediction in Appendix E.8.

5 Conclusion

In this paper, we introduce the first graph foundation model for retrieval augmented generation. By leveraging the knowledge graph index, GFM-RAG explicitly models the complex relationships between knowledge and documents, facilitating a more effective and efficient retrieval process. Powered by a query-dependent GNN pre-trained on large-scale datasets, GFM-RAG can effectively perform multi-hop reasoning over the graph structure to find relevant knowledge in a single step. Extensive experiments across three benchmark datasets and seven domain-specific datasets demonstrate that GFM-RAG significantly outperforms state-of-the-art methods in effectiveness, efficiency, and generalizability. Its alignment with scaling laws also suggests the potential for scaling to even larger datasets. In the future, we plan to conduct larger-scale training and further explore GFM-RAG's capabilities in other challenging scenarios such as knowledge graph completion and question answering.

Acknowledgments

S Pan was partly funded by Australian Research Council (ARC) under grants FT210100097 and DP240101547 and the CSIRO – National Science Foundation (US) AI Research Collaboration Program. C Gong is supported by NSF of China (Nos: 62336003, 12371510).

References

[1] Gabor Angeli, Melvin Jose Johnson Premkumar, and Christopher D Manning. Leveraging linguistic structure for open domain information extraction. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 344–354, 2015.

[2] Aijun Bai, Rolf Jagerman, Zhen Qin, Le Yan, Pratyush Kar, Bing-Rong Lin, Xuanhui Wang, Michael Bendersky, and Marc Najork. Regression compatible listwise objectives for calibrated ranking with binary relevance. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 4502–4508, 2023.

[3] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250, 2008.

[4] Vittorio Castelli, Rishav Chakravarti, Saswati Dana, Anthony Ferritto, Radu Florian, Martin Franz, Dinesh Garg, Dinesh Khandelwal, Scott McCarley, Michael McCawley, Mohamed Nasr, Lin Pan, Cezar Pendus, John Pitrelli, Saurabh Pujar, Salim Roukos, Andrzej Sakrajda, Avi Sil, Rosario Uceda-Sosa, Todd Ward, and Rong Zhang. The TechQA dataset. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1269–1278, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.117. URL `https://aclanthology.org/2020.acl-main.117/`.

[5] Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation, 2023.

[6] Tong Chen, Hongwei Wang, Sihao Chen, Wenhao Yu, Kaixin Ma, Xinran Zhao, Hongming Zhang, and Dong Yu. Dense X retrieval: What retrieval granularity should we use? In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 15159–15177, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.845. URL `https://aclanthology.org/2024.emnlp-main.845/`.

[7] Mostafa Dehghani, Josip Djolonga, Basil Mustafa, Piotr Padlewski, Jonathan Heek, Justin Gilmer, Andreas Peter Steiner, Mathilde Caron, Robert Geirhos, Ibrahim Alabdulmohsin, et al. Scaling vision transformers to 22 billion parameters. In *International Conference on Machine Learning*, pages 7480–7512. PMLR, 2023.

[8] Jialin Dong, Bahare Fatemi, Bryan Perozzi, Lin F Yang, and Anton Tsitsulin. Don't forget to connect! improving rag with graph-based reranking. *arXiv preprint arXiv:2405.18414*, 2024.

[9] Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, and Jonathan Larson. From local to global: A graph rag approach to query-focused summarization. *arXiv preprint arXiv:2404.16130*, 2024.

[10] Robert Friel, Masha Belyi, and Atindriyo Sanyal. Ragbench: Explainable benchmark for retrieval-augmented generation systems. *arXiv preprint arXiv:2407.11005*, 2024.

[11] Mikhail Galkin, Xinyu Yuan, Hesham Mostafa, Jian Tang, and Zhaocheng Zhu. Towards foundation models for knowledge graph reasoning. In *The Twelfth International Conference on Learning Representations*, 2024.

[12] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, and Haofen Wang. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*, 2023.

[13] Dan Gillick, Sayali Kulkarni, Larry Lansing, Alessandro Presta, Jason Baldridge, Eugene Ie, and Diego Garcia-Olano. Learning dense representations for entity retrieval. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 528–537, 2019.

[14] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *International conference on machine learning*, pages 1263–1272. PMLR, 2017.

[15] Zirui Guo, Lianghao Xia, Yanhua Yu, Tu Ao, and Chao Huang. Lightrag: Simple and fast retrieval-augmented generation. 2024.

[16] Bernal Jiménez Gutiérrez, Yiheng Shu, Yu Gu, Michihiro Yasunaga, and Yu Su. Hipporag: Neurobiologically inspired long-term memory for large language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL `https://openreview.net/forum?id=hkujvAPVsg`.

[17] Haoyu Han, Yu Wang, Harry Shomer, Kai Guo, Jiayuan Ding, Yongjia Lei, Mahantesh Halappanavar, Ryan A Rossi, Subhabrata Mukherjee, Xianfeng Tang, et al. Retrieval-augmented generation with graphs (graphrag). *arXiv preprint arXiv:2501.00309*, 2024.

[18] Xiaoxin He, Yijun Tian, Yifei Sun, Nitesh Chawla, Thomas Laurent, Yann LeCun, Xavier Bresson, and Bryan Hooi. G-retriever: Retrieval-augmented generation for textual graph understanding and question answering. *Advances in Neural Information Processing Systems*, 37:132876–132907, 2024.

[19] Joel Hestness, Sharan Narang, Newsha Ardalani, Gregory Diamos, Heewoo Jun, Hassan Kianinejad, Md Mostofa Ali Patwary, Yang Yang, and Yanqi Zhou. Deep learning scaling is predictable, empirically. *arXiv preprint arXiv:1712.00409*, 2017.

[20] Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6609–6625, 2020.

[21] Xingyue Huang, Miguel Romero, Ismail Ceylan, and Pablo Barceló. A theory of link prediction via relational weisfeiler-leman on knowledge graphs. *Advances in Neural Information Processing Systems*, 36:19714–19748, 2023.

[22] Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. Unsupervised dense information retrieval with contrastive learning. *Transactions on Machine Learning Research*, 2022.

[23] Soyeong Jeong, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, and Jong C Park. Adaptive-rag: Learning to adapt retrieval-augmented large language models through question complexity. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 7029–7043, 2024.

[24] Zhengbao Jiang, Frank F Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. Active retrieval augmented generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7969–7992, 2023.

[25] Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William Cohen, and Xinghua Lu. PubMedQA: A dataset for biomedical research question answering. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2567–2577, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1259. URL `https://aclanthology.org/D19-1259/`.

[26] Ashutosh Joshi, Sheikh Muhammad Sarwar, Samarth Varshney, Sreyashi Nag, Shrivats Agrawal, and Juhi Naik. Reaper: Reasoning based retrieval planning for complex rag systems. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, pages 4621–4628, 2024.

[27] Ehsan Kamalloo, Aref Jafari, Xinyu Zhang, Nandan Thakur, and Jimmy Lin. Hagrid: A human-llm collaborative dataset for generative information-seeking with attribution. *arXiv preprint arXiv:2307.16883*, 2023.

[28] Xiaoxi Kang, Lizhen Qu, Lay-Ki Soon, Zhuang Li, and Adnan Trakic. Bridging law and data: Augmenting reasoning via a semi-structured dataset with irac methodology. *arXiv preprint arXiv:2406.13217*, 2024.

[29] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

[30] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, 2020.

[31] Chankyu Lee, Rajarshi Roy, Mengyao Xu, Jonathan Raiman, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. Nv-embed: Improved techniques for training llms as generalist embedding models. *arXiv preprint arXiv:2405.17428*, 2024.

[32] Mufei Li, Siqi Miao, and Pan Li. Simple is effective: The roles of graphs and large language models in knowledge-graph-based retrieval-augmented generation. In *The Thirteenth International Conference on Learning Representations*, 2025.

[33] Shiyang Li, Yifan Gao, Haoming Jiang, Qingyu Yin, Zheng Li, Xifeng Yan, Chao Zhang, and Bing Yin. Graph reasoning for question answering with triplet retrieval. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 3366–3375, 2023.

[34] Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. Towards general text embeddings with multi-stage contrastive learning. *arXiv preprint arXiv:2308.03281*, 2023.

[35] Lei Liang, Mengshu Sun, Zhengke Gui, Zhongshu Zhu, Zhouyu Jiang, Ling Zhong, Yuan Qu, Peilong Zhao, Zhongpu Bo, Jin Yang, et al. Kag: Boosting llms in professional domains via knowledge augmented generation. *arXiv preprint arXiv:2409.13731*, 2024.

[36] Zhutian Lin, Junwei Pan, Shangyu Zhang, Ximei Wang, Xi Xiao, Shudong Huang, Lei Xiao, and Jie Jiang. Understanding the ranking loss for recommendation with sparse user feedback. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 5409–5418, 2024.

[37] Jiawei Liu, Cheng Yang, Zhiyuan Lu, Junze Chen, Yibo Li, Mengmei Zhang, Ting Bai, Yuan Fang, Lichao Sun, Philip S Yu, et al. Graph foundation models: Concepts, opportunities and challenges. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

[38] LINHAO LUO, Yuan-Fang Li, Reza Haf, and Shirui Pan. Reasoning on graphs: Faithful and interpretable large language model reasoning. In *The Twelfth International Conference on Learning Representations*, 2024.

[39] Chaitanya Malaviya, Subin Lee, Sihao Chen, Elizabeth Sieber, Mark Yatskar, and Dan Roth. Expertqa: Expert-curated questions and attributed answers. *arXiv preprint arXiv:2309.07852*, 2023.

[40] Haitao Mao, Zhikai Chen, Wenzhuo Tang, Jianan Zhao, Yao Ma, Tong Zhao, Neil Shah, Mikhail Galkin, and Jiliang Tang. Position: Graph foundation models are already here. In *Forty-first International Conference on Machine Learning*, 2024.

[41] Costas Mavromatis and George Karypis. Gnn-rag: Graph neural retrieval for large language model reasoning. *arXiv preprint arXiv:2405.20139*, 2024.

[42] Meta. Build the future of ai with meta llama 3, 2024. URL https://llama.meta.com/llama3/.

[43] Gabriel de Souza P Moreira, Radek Osmulski, Mengyao Xu, Ronay Ak, Benedikt Schifferer, and Even Oldridge. Nv-retriever: Improving text embedding models with effective hard-negative mining. *arXiv preprint arXiv:2407.15831*, 2024.

[44] Abhilash Nandy, Soumya Sharma, Shubham Maddhashiya, Kapil Sachdeva, Pawan Goyal, and NIloy Ganguly. Question answering over electronic devices: A new benchmark dataset and a multi-task learning based QA framework. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4600–4609, Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-emnlp.392. URL https://aclanthology.org/2021.findings-emnlp.392/.

[45] Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. Ms marco: A human generated machine reading comprehension dataset. November 2016.

[46] Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hernandez Abrego, Ji Ma, Vincent Zhao, Yi Luan, Keith Hall, Ming-Wei Chang, et al. Large dual encoders are generalizable retrievers. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9844–9855, 2022.

[47] OpenAI. Hello gpt-4o, 2024. URL https://openai.com/index/hello-gpt-4o/.

[48] OpenAI. Learning to reason with llms, 2024. URL https://openai.com/index/learning-to-reason-with-llms/.

[49] Liu Pai, Wenyang Gao, Wenjie Dong, Lin Ai, Ziwei Gong, Songfang Huang, Li Zongsheng, Ehsan Hoque, Julia Hirschberg, and Yue Zhang. A survey on open information extraction from rule-based model to large language model. *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 9586–9608, 2024.

[50] Pranoy Panda, Ankush Agarwal, Chaitanya Devaguptapu, Manohar Kaul, and Prathosh Ap. Holmes: Hyper-relational knowledge graphs for multi-hop question answering using llms. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13263–13282, 2024.

[51] Boci Peng, Yun Zhu, Yongchao Liu, Xiaohe Bo, Haizhou Shi, Chuntao Hong, Yan Zhang, and Siliang Tang. Graph retrieval-augmented generation: A survey. *arXiv preprint arXiv:2408.08921*, 2024.

[52] Haiquan Qiu, Yongqi Zhang, Yong Li, et al. Understanding expressivity of gnn in rule learning. In *The Twelfth International Conference on Learning Representations*, 2024.

[53] Stephen E Robertson and Steve Walker. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *SIGIR'94: Proceedings of the Seventeenth Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval, organised by Dublin City University*, pages 232–241. Springer, 1994.

[54] Mobashir Sadat, Zhengyu Zhou, Lukas Lange, Jun Araki, Arsalan Gundroo, Bingqing Wang, Rakesh Menon, Md Parvez, and Zhe Feng. DelucionQA: Detecting hallucinations in domain-specific question answering. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 822–835, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.59. URL https://aclanthology.org/2023.findings-emnlp.59/.

[55] Keshav Santhanam, Omar Khattab, Jon Saad-Falcon, Christopher Potts, and Matei Zaharia. Colbertv2: Effective and efficient retrieval via lightweight late interaction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3715–3734, 2022.

[56] Parth Sarthi, Salman Abdullah, Aditi Tuli, Shubh Khanna, Anna Goldie, and Christopher D Manning. Raptor: Recursive abstractive processing for tree-organized retrieval. In *The Twelfth International Conference on Learning Representations*, 2024.

[57] SBERT. Sentence-transformers all-mpnet-base-v2, 2021. URL `https://huggingface.co/sentence-transformers/all-mpnet-base-v2`.

[58] Weihang Su, Yichen Tang, Qingyao Ai, Zhijing Wu, and Yiqun Liu. DRAGIN: Dynamic retrieval augmented generation based on the real-time information needs of large language models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12991–13013, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.702. URL `https://aclanthology.org/2024.acl-long.702/`.

[59] Jiashuo Sun, Chengjin Xu, Lumingyuan Tang, Saizhuo Wang, Chen Lin, Yeyun Gong, Lionel Ni, Heung-Yeung Shum, and Jian Guo. Think-on-graph: Deep and responsible reasoning of large language model on knowledge graph. In *The Twelfth International Conference on Learning Representations*, 2024.

[60] Jiabin Tang, Yuhao Yang, Wei Wei, Lei Shi, Lixin Su, Suqi Cheng, Dawei Yin, and Chao Huang. Graphgpt: Graph instruction tuning for large language models. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 491–500, 2024.

[61] Jiabin Tang, Yuhao Yang, Wei Wei, Lei Shi, Long Xia, Dawei Yin, and Chao Huang. Higpt: Heterogeneous graph language model. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 2842–2853, 2024.

[62] Timothy J Teyler and Pascal DiScenna. The hippocampal memory indexing theory. *Behavioral neuroscience*, 100(2):147, 1986.

[63] Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Musique: Multihop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554, 2022.

[64] Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10014–10037, 2023.

[65] Denny Vrandečić and Markus Krötzsch. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85, 2014.

[66] Zehong Wang, Zheyuan Zhang, Nitesh V Chawla, Chuxu Zhang, and Yanfang Ye. GFT: Graph foundation model with transferable tree vocabulary. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL `https://openreview.net/forum?id=0MXzbAv8xy`.

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

[68] Lianghao Xia, Ben Kao, and Chao Huang. Opengraph: Towards open graph foundation models. *arXiv preprint arXiv:2403.01121*, 2024.

[69] Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. C-pack: Packaged resources to advance general chinese embedding, 2023.

[70] 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, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zhihao Fan. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2024.

[71] Bishan Yang, Scott Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. Embedding entities and relations for learning and inference in knowledge bases. In *Proceedings of the International Conference on Learning Representations (ICLR) 2015*, 2015.

[72] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, 2018.

[73] Michihiro Yasunaga, Hongyu Ren, Antoine Bosselut, Percy Liang, and Jure Leskovec. Qa-gnn: Reasoning with language models and knowledge graphs for question answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 535–546, 2021.

[74] Alexandros Zeakis, George Papadakis, Dimitrios Skoutas, and Manolis Koubarakis. Pre-trained embeddings for entity resolution: an experimental analysis. *Proceedings of the VLDB Endowment*, 16(9):2225–2238, 2023.

[75] Xinyu Zhang, Nandan Thakur, Odunayo Ogundepo, Ehsan Kamalloo, David Alfonso-Hermelo, Xiaoguang Li, Qun Liu, Mehdi Rezagholizadeh, and Jimmy Lin. Making a miracl: Multilingual information retrieval across a continuum of languages. *arXiv preprint arXiv:2210.09984*, 2022.

[76] Lingfeng Zhong, Jia Wu, Qian Li, Hao Peng, and Xindong Wu. A comprehensive survey on automatic knowledge graph construction. *ACM Computing Surveys*, 56(4):1–62, 2023.

[77] Shaowen Zhou, Bowen Yu, Aixin Sun, Cheng Long, Jingyang Li, and Jian Sun. A survey on neural open information extraction: Current status and future directions. In Lud De Raedt, editor, *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pages 5694–5701. International Joint Conferences on Artificial Intelligence Organization, 7 2022. doi: 10.24963/ijcai.2022/793. URL `https://doi.org/10.24963/ijcai.2022/793`. Survey Track.

[78] Zhaocheng Zhu, Zuobai Zhang, Louis-Pascal Xhonneux, and Jian Tang. Neural bellman-ford networks: A general graph neural network framework for link prediction. *Advances in Neural Information Processing Systems*, 34:29476–29490, 2021.

Appendix

Table of Contents

A Query-dependent GNNs for Multi-hop Reasoning and Retrieval

We provide a detailed explanation about why query-dependent GNNs can be used for multi-hop reasoning and retrieval. Recent studies [21, 52] have theoretically proven that query-dependent GNNs are effective for capturing the multi-hop logical associations on KGs to answer queries, such as:

$$\exists y : \texttt{politician\_of}(\text{Barack Obama}, y) \leftarrow \texttt{work\_in}(\text{Barack Obama}, z_1) \wedge \texttt{city\_of}(z_1, y), \quad (20)$$

where the right part denotes the logical associations can be executed to answer the query on the left, i.e., "`politician_of`(Barack Obama,y)".

This query is semantic equivalent to the nature language question: "`Barack Obama is the politician of which country?`". By treating the input question as a "soft query" (query in nature language), we apply the query-dependent GNN (GFM) to bridge the gap between nature language and logical query. The GFM tries to understand the semantic of the questions and learn to conduct complex logical reasoning (e.g., multi-hop reasoning) on KGs for retrieval [73]. The learned logical associations for reasoning are shown in Section 4.8.

Table 5: Statistics of the query-doc pairs and KGs used for training.

Dataset	#Q-doc Pair	#Document	#KG	#Entity	#Relation	#Triple
HotpotQA	20,000	204,822	20	1,930,362	967,218	6,393,342
MuSiQue	20,000	410,380	20	1,544,966	900,338	4,848,715
2Wiki	20,000	122,108	20	916,907	372,554	2,883,006
Total	**60,000**	**737,310**	**60**	**4,392,235**	**2,240,110**	**14,125,063**

B Datasets

B.1 Multi-hop QA Datasets

Three multi-hop QA datasets are used in our experiments: HotpotQA [72], MuSiQue [63], and 2WikiMultiHopQA (2Wiki) [20]. We provide a brief overview of these datasets below.

Table 6: Statistics of the datasets and constructed KG-indexes used for testing.

Dataset	Domain	#Test	#Document	#Entity	#Relation	#Triple
HotpotQA	Multi-hop	1,000	9,221	87,768	45,112	279,112
MuSiQue	Multi-hop	1,000	6,119	48,779	20,748	160,950
2Wiki	Multi-hop	1,000	11,656	100,853	55,944	319,618
PubMedQA	Biomedical	2,450	5,932	42,389	20,952	149,782
DelucionQA	Customer Support	184	235	2,669	2,298	6,183
TechQA	Customer Support	314	769	10,221	4,606	57,613
ExpertQA	Customer Support	203	808	11,079	6,810	16,541
EManual	Customer Support	132	102	695	586	1,329
MS Marco	General Knowledge	423	3,481	24,740	17,042	63,995
HAGRID	General Knowledge	1,318	1,975	23,484	18,653	48,969

- HotpotQA [72] is a multi-hop QA dataset that requires reasoning over multiple documents to answer questions. The dataset consists of 97k question-answer pairs, where each question is associated with up to 2 supporting and several distracting documents. The questions are designed to be answerable using multiple pieces of information from the supporting documents.
- MuSiQue [63] is a challenging multi-hop QA dataset with 25k 2-4 hop questions. It requires coherent multi-step reasoning to answer questions that span multiple documents.
- 2WikiMultiHopQA (2Wiki) [20] is a multi-hop QA dataset that requires reasoning over multiple Wikipedia articles to answer questions. The dataset consists of 192k questions, which are designed to be answerable using information from 2 or 4 articles.

In experiments, we adhere to the official data split to obtain the training samples and follow existing methods [64, 16] to use the same 1,000 samples from each validation set to avoid data leakage. We merge the candidate passages as the document corpus for KG-index construction. The statistics of the training and test data are presented in Table 5 and Table 6, respectively.

B.2 Domain-specific RAG Datasets

To test the generalizability of `GFM-RAG`, we evaluate it on seven domain-specific RAG datasets [10] including, (1) *biomedical*: PubMedQA [25]; (2) *customer support*: DelucionQA [54], TechQA [4], ExpertQA [39], EManual [44]; (3) *general knowledge*: MS Marco [45], HAGRID [27]. We provide a brief overview of these datasets below.

- PubMedQA [25] is a collection of PubMed research abstracts with corresponding questions paired with 4 abstract chunks.
- DelucionQA [54] is a domain-specific RAG dataset leveraging Jeep's 2023 Gladiator model manual as the source of knowledge, where each question is associated with 4 context documents and only 1 relevant passage.
- TechQA [4] is a collection of real-world user questions posted on IBMDeveloper and DeveloperWorks forums, along with 10 technical support documents relating to each question.
- ExpertQA [39] is a collection of curated questions from domain experts in various fields of science, arts, and law. The dataset also contains expert-curated passages relevant to each question.
- EManual [44] is a question-answering dataset comprising consumer electronic device manuals and realistic questions about them composed by human annotators, where each question is related with up to 3 context documents.
- MS Marco [45] is an open-domain question-answering dataset sourced from Bing search engine user query logs. Each question is associated with 10 context passages retrieved via Bing web search.
- HAGRID [27] is a multi-lingual information retrieval dataset with questions and passages from MIRACL [75].

In experiments, we use test sets constructed by RAGBench [10] and merge all the candidate passages as document corpus for KG-index construction. The statistics of the test dataset are detailed in Table 6.

C Baselines

In experiments, we compare with several widely used retrieval methods under three categories: (1) *single-step naive methods*: BM25 [53], Contriever [22], GTR [46], ColBERTv2 [55], RAPTOR [56], Proposition [6]; (2) *graph-enhanced methods*: GraphRAG (MS) [9], LightRAG [15], HippoRAG [16]; (3) *multi-step methods*: Adaptive-RAG [23], FLARE [24], and IRCoT [64]. The detailed introduction of the baselines is as follows.

Single-step Naive Methods are widely adopted in real-world applications due to their great efficiency and generalizability.

- BM25 [53] is a classic information retrieval method based on the probabilistic model that ranks a set of documents based on the query terms frequency appearing in each document.
- Contriever [22] trains a dense retriever with contrastive learning on a large-scale corpus to retrieve relevant documents for a given query.
- GTR [46] develops a scale-up T5-based dense retriever that could generalize across different datasets and domains.
- ColBERTv2 [55] is a state-of-the-art dense retriever that couples an aggressive residual compression mechanism with a denoised supervision strategy to simultaneously improve the retrieval quality.
- RAPTOR [56] is an LLM-augmented retriever that recursively embeds, clusters, and summarizes chunks of text, constructing a tree with differing levels of summarization to enable accurate retrieval.
- Proposition [6] enhances the performance of dense retrievers by leveraging LLMs to generate a natural language proposition that captures the essential information of the document.

Graph-enhanced Methods design a retriever that is built upon a graph structure to conduct effective retrieval and reasoning.

- GraphRAG (MS) [9] is a graph-enhanced retrieval method originally proposed by Microsoft. It builds a graph structure from the document corpus and use hierarchical community detection to cluster the documents into communities and generate a summary for each community. The summary together with the original documents are retrieved by the retriever for LLM generation.
- LightRAG [15] is an innovative graph-enhanced RAG method that incorporates graph structures into text indexing and retrieval, enabling efficient retrieval of entities and their relationships. It employs a dual-level retrieval system to gather both low-level and high-level knowledge for LLM generation.
- HippoRAG [16] is a state-of-the-art, training-free graph-enhanced retriever that uses the Personalized PageRank algorithm to assess entity relevance to a query and performs multi-hop retrieval on a document-based knowledge graph. It can be directly applied to various datasets.

Multi-step Methods are designed to conduct multi-hop reasoning by iteratively retrieving and reasoning over documents, which can be integrated with arbitrary retrieval methods.

- Adaptive-RAG [23] proposes an adaptive multi-step retrieval method that can dynamically select the most suitable retrieval strategy based on the complexity of the query.
- FLARE [24] introduces a multi-step retrieval method that actively decide when and how to retrieve documents. It also predicts the future content to the guide the retrieval in next steps.
- IRCoT [64] is a powerful multi-step retrieval pipeline that integrates the retrieval with the chain-of-thought (CoT) reasoning of LLMs. It guides the retrieval with CoT and in turn using retrieved documents to improve CoT. IRCoT can be compatible with arbitrary retrievers to conduct multi-step retrieval and reasoning.

Table 7: The detailed implementation and training settings of GFM-RAG.

	Setting	GFM-RAG
KG-index Construction	OpenIE	GPT-4o-mini
	Entity resolution	ColBERTv2
	τ	0.8
GFM Model	# Layer	6
	Hidden dim	512
	Message	DistMult
	Aggregation	Sum
	$g^l(\cdot)$	2-layer MLP
	Sentence embedding model	all-mpnet-v2
	Doc. ranker entities T	20
KGC Pre-training	α	1
	Optimizer	AdamW
	Learning rate	5e-4
	Batch size	4
	Training steps	30,000
	# Negative sample	128
Supervised Retrieval Fine-tuning	α	0.3
	Optimizer	AdamW
	Learning rate	5e-4
	Batch size	4
	Training epochs	5
	# Negative sample	$\mathcal{E} \setminus \mathcal{A}_q$

D Implementations and Training Details

D.1 Training Data Construction

We extract 60,000 samples from the training set of HotpotQA, MuSiQu, and 2Wiki to construct KG-indexes and conduct large-scale training. Specifically, we merge the candidate passages as the document corpus. In the KG-index construction, we use the GPT-4o-mini [47] with the OpenIE prompts described in HippoRAG [16] to extract the entities, relations, and triples from the document corpus. Then, we use the ColBERTv2 [55] to conduct the entity resolution by computing the similarity between entities as

$$s(e_i, e_j) = \text{Emb.}(e_i)^\top \text{Emb.}(e_j), \tag{21}$$

where a new triple $(e_i, \texttt{equivalent}, e_j)$ is generated if $s(e_i, e_j) > \tau$ and $e_i \neq e_j$. We set the threshold τ as 0.8 in our experiments. We divide the samples into groups of approximately 1k questions and 10k documents each to control the constructed KG-index size. In the end, we obtain 60 different KG-indexes and associated question-document pairs for model training.

D.2 Model Settings

In GFM-RAG, the GFM is implemented as a 6-layer query-dependent GNN with the hidden dimension of 512, DistMult message function, and sum aggregation. The relation update function $g^l(\cdot)$ is implemented as a 2-layer MLP. We use the all-mpnet-v2 as the sentence embedding model with a dimension of 768. The total training parameters of the GFM is 8M. In the retrieval stage, we select top $T = 20$ entities for the document ranker.

D.3 Training Settings

In KG completion pre-training, we randomly sample triples (e, r, t) from knowledge graphs and mask out either the head or the tail entity to create a synthetic query $q = (e, r, ?)$ and answer $a = \{e\}$ in a self-supervised manner. For example, given a triple (`Barack Obama`, `born_in`, `Honolulu`), we

Table 8: Comparison of different sentence embedding models used in GFM-RAG.

Sentence Embedding Model	HotpotQA		MuSique		2Wiki	
	R@2	R@5	R@2	R@5	R@2	R@5
sentence-transformers/all-mpnet-base-v2	**70.2**	**82.1**	46.0	55.1	**81.1**	85.6
BAAI/bge-large-en	68.1	81.1	45.9	**55.9**	80.7	**86.3**
Alibaba-NLP/gte-Qwen2-1.5B-instruct	69.9	81.5	46.0	55.0	79.8	86.2
Alibaba-NLP/gte-Qwen2-7B-instruct	68.5	81.5	45.5	55.1	80.8	85.6
nvidia/NV-Embed-v2	69.2	81.4	**46.3**	54.9	80.3	85.5

Table 9: Comparison of GFM-RAG with pre-trained and fine-tuned sentence embedding models.

Method	HotpotQA		MuSiQue		2Wiki	
	R@2	R@5	R@2	R@5	R@2	R@5
GFM-RAG	**78.3**	**87.1**	**49.1**	**58.2**	**90.8**	**95.6**
all-mpnet-v2 (pre-trained)	59.4	73.3	33.2	46.3	48.5	59.4
all-mpnet-v2 (finetuned)	67.0	82.3	41.7	55.0	65.1	76.7

can create a query as $(\texttt{Barack Obama}, \texttt{born\_in}, ?)$, which is encoded as a sentence embedding and fed into the GFM to predict the target entity Honolulu on graphs.

In supervised document retrieval fine-tuning, we obtain natural language questions and supporting documents from the multi-hop QA datasets. For each question, we identify the entities from its supporting documents as the targets. For instance, given the question *"Where was Barack Obama born in?"*, we can extract two entities such as [Honolulu, USA] from its supporting documents (e.g., Doc. 2 in Figure 2). The GFM is trained to maximize the likelihood of these two target entities.

In the self-supervised KG completion pre-training, the GFM is trained on the mixture of 60 constructed KG-indexes for 30,000 steps. Then, we conduct the supervised document retrieval fine-tuning on the labeled question-document pairs for 5 epochs. The weight α between losses is set to 0.3. We use AdamW optimizer, learning rate of 5e-4 with batch sizes of both training stages set to 4. Each batch contains only one KG-index and training samples associated to it, where we randomly sample from different KG-indexes during training. The model is trained on 8 NVIDIA A100s (80G) with 14 hours pre-training and 5 hours supervised fine-tuning. The detailed settings are summarized in Table 7.

E Additional Experiments

E.1 Effectiveness of Different Sentence Embeddings

In this section, we first study the effectiveness of different sentence embeddings in the GFM. We compare the all-mpnet-v2 [57], bge-large-en [69], gte-Qwen2-1.5B-instruct and gte-Qwen2-7B-instruct [34] as well as NV-Embed-v2 [31]. We download the official pre-trained model from the Huggingface[3]. The details of the models are shown in Table 8. From the results, we can observe that the performance variance between different sentence embeddings is relatively small, where the all-mpnet-v2 achieves the best performance with respect to 3 metrics. This indicates that GFM-RAG is not sensitive to the choice of sentence embedding models. In experiments, we use the all-mpnet-v2 as the default sentence embedding model due to its efficiency. However, it has relative smaller context-size (512) which limits the length of input text. We leave the exploration of larger context-size sentence embedding models (e.g., NV-Embed-v2 with 32k context) for future work.

Then, we expand our ablation study to compare GFM-RAG with variants without GNN and using solely the pre-trained all-mpnet-v2 embeddings and those fine-tuned on multi-hop QA data, respectively. The results are shown in Table 9. We can observe that GNN plays a crucial role in retrieval. The sentence embedding model all-mpnet-v2 is pre-trained on large-scale text data and could potentially see the QA data. However, it is not specifically trained for the multi-hop QA task, which leads to

[3] https://huggingface.co/

Table 10: Effectiveness of KGC pre-training and supervised retrieval fine-tuning in GFM-RAG.

Method	HotpotQA		MuSique		2Wiki	
	R@2	R@5	R@2	R@5	R@2	R@5
GFM-RAG	**78.3**	**87.1**	**49.1**	58.2	**89.1**	**92.8**
GFM-RAG w/o Retrieval Fine-tune	21.0	32.8	18.3	25.9	44.6	53.4
GFM-RAG w/o KGC Pre-train	77.8	86.5	48.3	**58.3**	88.3	92.5

Table 11: Knowledge graph completion result of different training strategies.

Method	MRR	Hits@1	Hits@3	Hits@10
GFM-RAG	0.193	0.138	0.221	0.293
GFM-RAG w/o Retrieval Fine-tune	**0.304**	**0.234**	**0.323**	**0.451**
GFM-RAG w/o KGC Pre-train	0.029	0.007	0.022	0.067

suboptimal performance in capturing the relationship between question and supporting documents. The fine-tuned all-mpnet-v2 achieves better performance than the pre-trained one by supervised fine-tuning on the multi-hop QA data, but still inferior to GFM-RAG. This indicates that the GNN can effectively capture the relationship between knowledge and conduct multi-hop reasoning, which is not achievable by simply using the sentence embedding model.

E.2 Effectiveness of Different Training Strategies

In this section, we first study the effectiveness of the two training tasks used in GFM-RAG. We compare the performance by only conducting the KG completion pre-training (GFM-RAG w/o Fine-tune) and supervised document retrieval fine-tuning (GFM-RAG w/o Pre-train). The results are shown in Table 10. The results show that removing the supervised document retrieval fine-tuning significantly decreases the performance of GFM-RAG. This highlights the importance of supervised fine-tuning, as it enables the model to understand users' queries and better capture the relevance between questions and knowledge for improved retrieval.

Although the pre-training has a relatively small impact on the final performance, its primary purpose is to learn the general graph reasoning ability, following previous studies like ULTRA [11]. This would enhance the generalization and robustness of the GFM, which could be beneficial to its performance on other tasks, such as knowledge graph completion. To further validate this, we conduct an ablation study to compare GFM-RAG with different training strategies on the knowledge graph completion task. We report the knowledge graph completion (KGC) performance on the KG-index from the test set of the HotpotQA dataset. The results are shown in Table 11.

From the knowledge graph completion results, we can observe that the GFM-RAG undergoes only the pre-training (GFM-RAG w/o Fine-tune) achieves the best performance, which indicates that the pre-training is effective in learning the general graph reasoning ability. The performance of GFM-RAG with only supervised fine-tuning (GFM-RAG w/o Pre-train) is significantly lower than that of GFM-RAG with pre-training. This indicates that the supervised fine-tuning is only learning the specific downstream task, which would limit the generalization ability of GFM-RAG as the foundation model. The GFM trained with both pre-training and supervised fine-tuning achieves the second-best performance on the knowledge graph completion task and the best performance on the multi-hop QA task. This indicates that both training strategies are essential for GFM-RAG to learn the general graph reasoning ability and benefit specific downstream tasks.

E.3 Effectiveness of Loss Weights

In this section, we examine the effectiveness of the weights assigned to the BCE loss and ranking loss in training GFM-RAG. We compare performance by varying the weight α between the two losses: $\mathcal{L} = \alpha\mathcal{L}_{\text{BCE}} + (1-\alpha)\mathcal{L}_{\text{RANK}}$, with results presented in Table 12. The findings indicate that using only either the BCE loss or ranking loss leads to suboptimal performance ($\alpha = 0$ or 1). The best

Table 12: Effectiveness (MRR) for the weight α of two losses.

α	HotpotQA	MuSique	2Wiki
0	0.5189	0.3252	0.4425
1	0.5096	0.3214	0.4282
0.7	0.5202	0.3249	0.4348
0.3	**0.5243**	**0.3260**	**0.4490**

performance occurs when α is set to 0.3, which aligns with previous studies [36] suggesting that a smaller weight for BCE loss is preferable when positive samples are rare in the training data.

E.4 Effectiveness of Ranking Methods

In this section, we investigate the effectiveness of different ranking methods based on inverted index used in `GFM-RAG`. We compare four ranking methods including (1) *IDF + Top-T Pred*: Our proposed method (eqs. (14) to (16)), which maps the top-T entities predicted by GFM to documents using inverse document frequency (IDF)-weighted scores. (2) *IDF + All Pred*: Uses all predicted entities from GFM and weights them by IDF (w/o eq. (14)). (3) *Top-T Pred*: Uses only the top-T predicted entities without applying IDF weighting (w/o eq. (15)). (4) *All Pred*: Use all entity predictions and directly map to document scores (w/o eqs. (14) and (15)). The results are shown in Table 13. The results show that the proposed *IDF + Top-k Pred* performs the best. This indicates that the inverted index is a crucial component of `GFM-RAG`, which serves as a bridge between structured reasoning over KGs and the unstructured documents required by LLMs, necessitating a careful design.

We acknowledge the potential alternatives, and as a promising future direction, we plan to explore end-to-end models that can jointly reason over structured and unstructured knowledge without relying on an explicit inverted index.

Table 13: Comparison of different ranking methods.

Ranking Method	HotpotQA		MuSiQue		2Wiki	
	R@2	R@5	R@2	R@5	R@2	R@5
IDF + Top-T Pred (`GFM-RAG`)	**78.3**	**87.1**	**49.1**	**58.2**	**90.8**	**95.6**
IDF + All Pred (w/o eq. (14))	68.1	71.4	35.8	41.2	86.0	87.5
Top-T Pred (w/o eq. (15))	71.6	78.6	46.3	52.5	74.7	78.1
All Pred (w/o eqs. (14) and (15))	77.6	82.9	41.1	46.9	88.6	90.4

E.5 Ablation Study of Training Datasets

We further conducted ablation studies where `GFM-RAG` is trained separately on each dataset, and we report performance across all three benchmarks. Results are shown Table 14. These results show that `GFM-RAG` not only performs well on the trained datasets, but also generalizes well to other datasets. More importantly, the model trained on multi-domain datasets performs competitively across all datasets, validating its ability to generalize effectively across domains and benefit from training on diverse KGs by learning generalizable reasoning ability across domains.

E.6 Model Transferability

In this section, we evaluate `GFM-RAG`'s transferability by conducting domain-specific fine-tuning on the training split of dataset on each domain. As shown in 15, `GFM-RAG` performs well in zero-shot generalization, with further improvements achieved through domain-specific fine-tuning. This highlights its transferability when adapted to domain-specific datasets.

Table 14: Ablation study of GFM-RAG trained on each dataset. Best results are highlighted in **bold**. The second best is underlined.

Test Dataset	HotpotQA		MuSiQue		2Wiki	
Training Dataset	R@2	R@5	R@2	R@5	R@2	R@5
HotpotQA	**79.3**	**87.8**	46.9	57.2	86.6	92.4
MusiQue	68.8	81.8	47.6	57.5	84.4	89.6
2Wiki	72.2	77.9	46.6	55.5	89.3	93.2
All	78.3	87.1	**49.1**	**58.2**	**90.8**	**95.6**

Table 15: Model performance (R@5) and transferability comparsion.

Model	DelucionQA	EManual	ExpertQA	TechQA	MS Marco	HAGRID
HippoRAG (zero-shot)	59.0	50.0	55.1	39.5	51.1	75.5
LightRAG (zero-shot)	46.1	46.2	59.4	36.8	48.3	75.9
GFM-RAG (zero-shot)	70.8	60.6	62.7	46.6	71.0	84.7
GFM-RAG (domain-specific fine-tuning)	**82.7**	**75.9**	**60.8**	**49.5**	**77.5**	**86.6**

E.7 Details of Model Neural Scaling

In this section, we provide more details on the neural scaling experiments. We evaluate the changes of the model performance with respect to different parameter sizes and training data sizes. In GFM-RAG, the model parameter sizes are primarily influenced by the hidden dimension of the GFM. Thus, we vary the dimension from 32 to 512 which results in the model parameter sizes ranging from 0.08M to 8M. The detailed settings are shown in Table 16. We test models with different sizes on different scales of training data ranging from 3k to 45k samples. We separately report the fitted trend line of performance changing with model parameter size and training data size in Figure 5. From the trend line, we can observe that the performance of GFM-RAG increases with the model parameter size and training data size. Meanwhile, with the larger model parameter size a larger training data size is required to achieve the best performance. This indicates that the performance of GFM-RAG can be further improved by scaling up the model size and training data simultaneously.

To further investigate architectural design, we varied the number of GNN layers from 1 to 8 while keeping the hidden dimension fixed (512), and evaluated model performance across all datasets. The results are shown in Table 17. We observe that performance generally improves with deeper GNN layers, which we attribute to both the increased model sizes and the ability to capture more complex multi-hop associations. This trend aligns with the neural scaling laws observed in foundation models, where larger parameter counts typically yield better generalization.

Interestingly, we find that performance peaks around 4 layers in some cases. As discussed in Appendix A and Section 4.8, GFM-RAG is designed to capture logical associations from KGs through multi-hop message passing. However, since the maximum number of reasoning hops required by our datasets is 4, additional layers beyond this offer limited benefit, likely due to the absence of higher-hop training signals. This finding supports our hypothesis that GFM-RAG effectively learns query-relevant multi-hop reasoning paths, and that deeper architectures may not improve performance without datasets requiring more complex reasoning. In summary, these results demonstrate the effectiveness and interpretability of the proposed GNN-based architecture, and confirm that both model capacity

Table 16: The hidden dimension with corresponding model size and training batch size for scaling law analysis.

Hidden Dim.	Parameter Size	Batch size (A100, 80G)
32	78,977	40
64	215,297	20
128	659,969	20
256	2,237,441	8
512	8,144,897	4

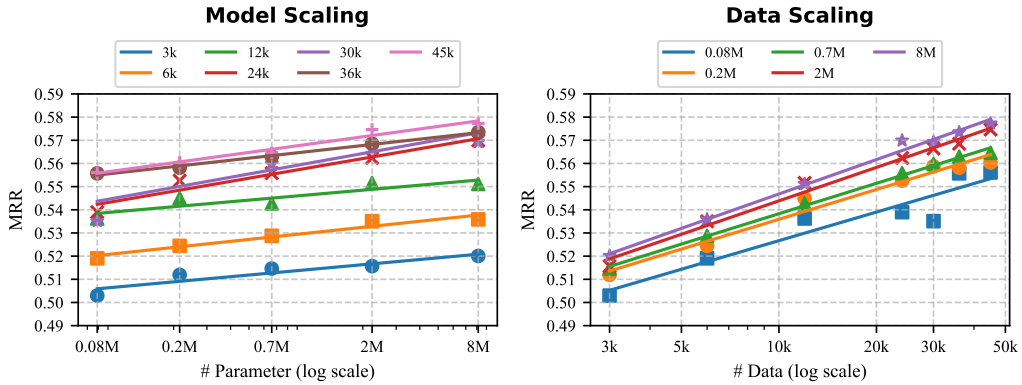

Figure 5: The illustration of the model and data scaling law of GFM-RAG.

Table 17: The different number of layers with corresponding model size and performance for scaling law analysis.

Hidden Dim. = 512	Averge		HotpotQA		MuSiQue		2Wiki	
L-Layer	R@2	R@5	R@2	R@5	R@2	R@5	R@2	R@5
1-layer (3M)	53.9	66.7	59.3	74.2	40.7	50.2	61.8	75.7
2-layer (4M)	69.9	78.6	73.6	85.4	47.6	57.0	88.6	93.3
4-layer (6M)	72.2	**80.1**	78.4	87.8	49.3	**60.1**	88.8	92.5
6-layer (8M)	71.9	79.6	78.0	87.0	48.4	58.7	89.3	93.1
8-layer (10M)	**73.0**	79.9	**79.7**	**87.8**	**49.7**	59.1	**89.5**	**92.8**

and logical expressibility contribute to GFM-RAG 's strong performance. We recognize the potential of other architectural designs and aim to explore them in the future, inspiring the community to do the same.

E.8 Visualization of the Distribution of Multi-hop Prediction

In this section, we visualize the distribution of the number of hops in the multi-hop reasoning process of GFM-RAG. We calculate the number of hops in the ground-truth reasoning path required for each question in the test set of HotpotQA, MuSiQue, and 2Wiki. Then, we compare the distribution of the number of hops in the reasoning path of the ground-truth and the predicted reasoning path by GFM-RAG as well as HippoRAG. The results are shown in Figure 6. We can observe that the distribution of GFM-RAG is closely aligned to the ground-truth, which indicates that GFM-RAG can effectively conduct the multi-hop reasoning within a single step. Meanwhile, the distribution of HippoRAG is relatively different from the ground-truth, especially in 2Wiki dataset. This indicates that HippoRAG may not be able to effectively capture the complex relationship to conduct multi-hop reasoning on graphs.

Table 18: The cost of the KG-index construction.

LLM	Price per 10k docs.	Total Price
GPT-4o-mini	$2.93	$216

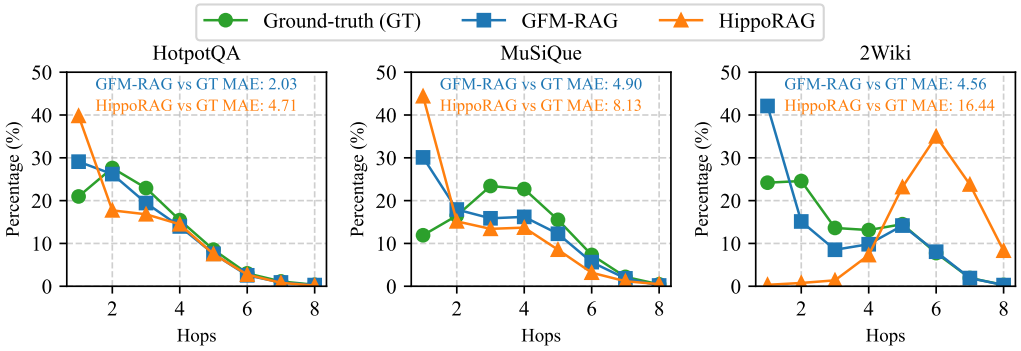

Figure 6: Statistics of the prediction hops of GFM-RAG and HippoRAG against the ground-truth.

Table 19: Token cost comparison for index construction

Method	# Tokens per 10k documents
LightRAG	5.5M
GraphRAG	7.6M
GFM-RAG	**4.8M**

E.9 Cost and Impact of LLMs on KG-index Construction

In this section, we first analyze the cost of the KG-index construction. In experiments, we utilize GPT-4o-mini[4] for OpenIE extraction and construct the KG-index for 737,310 documents. The cost is shown in Tables 18 and 19. Specifically, we find that constructing the KG-index requires approximately 4.8M tokens per 10k documents, which costs around $2.6 using GPT-4o-mini. LightRAG and GraphRAG cost 5.7M tokens and 7.6M tokens, respectively. Compared to other methods, GFM-RAG is more cost-effective as it does not require generating community-level summaries. In addition, we compare the graph index construction time of GFM-RAG in Table 20. Results show that GFM-RAG benefits from the quick index process during retrieval, as it does not construct a traditional vector database to store documents and entities.

Admittedly, using an LLM for KG index construction incurs computational costs. However, KG construction has been extensively studied, and numerous alternative methods exist that do not rely on LLMs [76]. Our implementation offers an easy interface for integration with any KG construction tools. We would explore the use of other KG construction methods in future work.

We further analyze the impact of LLMs used for KG-index construction on the performance of GFM-RAG. We conduct experiments using different LLMs for KG-index construction, including GPT-4o-mini and GPT-3.5-turbo[5]. Then, we reevaluate the performance of GFM-RAG and HippoRAG with the constructed KG-index. The results are shown in Table 21. From the results, the performance of both methods on the KG extracted by GPT-4o-mini is higher than the ones by GPT-3.5-turbo. This supports the opinion that GPT-4o-mini generally outperforms GPT-3.5-turbo in constructing high quality KG-index, which is crucial for the graph-enhanced retrieval. However, the performance

[4] https://platform.openai.com/docs/models/o4-mini
[5] https://platform.openai.com/docs/models/gpt-3-5-turbo

Table 20: Graph Indexing time comparison.

Method	Indexing time (s)
LightRAG	1430.32
GraphRAG (MS)	1796.43
GFM-RAG	**93.55**

Table 21: Comparison of the model performance under the KG-index constructed by different LLMs.

Method	HotpotQA		MuSiQue		2Wiki	
	R@2	R@5	R@2	R@5	R@2	R@5
GFM-RAG (gpt-4o-mini)	78.3	87.1	49.1	58.2	90.8	95.6
HippoRAG (gpt-4o-mini)	62.2	79.3	41.7	53.6	72.1	89.5
GFM-RAG (gpt-3.5-trubo)	75.6	84.7	46.1	55.8	85.2	90.4
HippoRAG (gpt-3.5-trubo)	60.5	77.7	40.9	51.9	70.7	89.1

of GFM-RAG is significantly higher than HippoRAG under both KG-indexes. This indicates that GFM-RAG is more robust to the quality of the KG-index, demonstrating the effectiveness of the GFM in graph reasoning and retrieval.

F Prompts

In experiments, we follow the prompts used in HippoRAG [16] to extract the triples from the document corpus, which is shown in Table 22.

G Limitations

The limitations of GFM-RAG are as follows: (1) The construction of KG-index can be costly and time-consuming, especially when using LLMs for OpenIE extraction. We would explore the use of efficient KG construction methods in future work and optimize the construction process. (2) The model size of the GFM-RAG is relatively small (8M) compared to other foundation models like large language models with billions of parameters. Although it is not faired to directly compare the GNN-based model with transformer-based LLMs, we would explore the scaling of GFM-RAG in future work to improve its performance and generalizability. (3) GFM-RAG is only evaluated on multi-hop QA tasks and KG completion tasks. We would explore the capabilities of GFM-RAG in other tasks such as knowledge graph question answering and knowledge graph reasoning in future work to validate its effectiveness as a foundation model.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

OpenIE Prompt

```
## Instruction:
Your task is to construct an RDF (Resource Description Framework
) graph from the given passages and named entity lists. Respond
with a JSON list of triples, with each triple representing a
relationship in the RDF graph. Pay attention to the following
requirements:
- Each triple should contain at least one, but preferably two,
of the named entities in the list for each passage.
- Clearly resolve pronouns to their specific names to maintain
clarity.

Convert the paragraph into a JSON dict, it has a named entity
list and a triple list.

## One-Shot Demonstration:
Paragraph:
‘‘‘
Radio City
Radio City is India’s first private FM radio station and was
started on 3 July 2001. It plays Hindi, English and regional
songs. Radio City recently forayed into New Media in May 2008
with the launch of a music portal - PlanetRadiocity.com that
offers music related news, videos, songs, and other music-
related features.
‘‘‘
{
    "named_entities":
    ["Radio City", "India", "3 July 2001", "Hindi", "English", "
    May 2008", "PlanetRadiocity.com"]
}
{
    "triples": [
            ["Radio City", "located in", "India"],
            ["Radio City", "is", "private FM radio station"],
            ["Radio City", "started on", "3 July 2001"],
            ["Radio City", "plays songs in", "Hindi"],
            ["Radio City", "plays songs in", "English"]
            ["Radio City", "forayed into", "New Media"],
            ["Radio City", "launched", "PlanetRadiocity.com"],
            ["PlanetRadiocity.com", "launched in", "May 2008"],
            ["PlanetRadiocity.com", "is", "music portal"],
            ["PlanetRadiocity.com", "offers", "news"],
            ["PlanetRadiocity.com", "offers", "videos"],
            ["PlanetRadiocity.com", "offers", "songs"]
    ]
}

## Input
Convert the paragraph into a JSON dict, it has a named entity
list and a triple list. Paragraph:
‘‘‘
INPUT PASSAGE
‘‘‘
```

Table 22: The prompt for OpenIE extraction.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims in the abstract and introduction accurately reflect the paper's contributions and scope.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We have discussed the limitations of the work in Appendix G.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: The paper does not include theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We have detailed data construction process, model settings, and training process in Appendix D to ensure the reproducibility of our results.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We have uploaded the code to an anonymous link in the paper.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We have detailed experiment settings in Appendix D.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [NA]

   Justification: The experiments are conducted with a fixed random seed and no error bars are reported.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The resources used for the experiments are detailed in Appendix D.3.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conducted in the paper complies with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: The proposed method focuses on the technical aspects of the problem and do not include societal impacts.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper utilizes existing datasets and pretrained models that are already released which have safeguards in place.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We have properly cited all code, data, and models used in our research and complied with the licensing agreements and terms of use set by the original authors.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: There are no new assets introduced in the paper.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: The usage of the LLM is described and discussed in Appendices D and D.1.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.