HyperGraphRAG: Retrieval-Augmented Generation via Hypergraph-Structured Knowledge Representation

Haoran Luo[1,2], **Haihong E**[1*], **Guanting Chen**[1], **Yandan Zheng**[2], **Xiaobao Wu**[2], **Yikai Guo**[3], **Qika Lin**[4], **Yu Feng**[5], **Zemin Kuang**[6], **Meina Song**[1], **Yifan Zhu**[1], **Luu Anh Tuan**[2]
[1]Beijing University of Posts and Telecommunications [2]Nanyang Technological University
[3]Beijing Institute of Computer Technology and Application [4]National University of Singapore
[5]China Mobile Research Institute [6]Beijing Anzhen Hospital, Capital Medical University
haoran.luo@ieee.org, ehaihong@bupt.edu.cn, anhtuan.luu@ntu.edu.sg

Abstract

Standard Retrieval-Augmented Generation (RAG) relies on chunk-based retrieval, whereas GraphRAG advances this approach by graph-based knowledge representation. However, existing graph-based RAG approaches are constrained by binary relations, as each edge in an ordinary graph connects only two entities, limiting their ability to represent the n-ary relations ($n \geq 2$) in real-world knowledge. In this work, we propose **HyperGraphRAG**, the first hypergraph-based RAG method that represents n-ary relational facts via hyperedges. HyperGraphRAG consists of a comprehensive pipeline, including knowledge hypergraph construction, retrieval, and generation. Experiments across medicine, agriculture, computer science, and law demonstrate that HyperGraphRAG outperforms both standard RAG and previous graph-based RAG methods in answer accuracy, retrieval efficiency, and generation quality. Our data and code are publicly available[1].

1 Introduction

Retrieval-Augmented Generation (RAG) [10, 6] has advanced knowledge-intensive tasks by integrating knowledge retrieval with large language models (LLMs) [17, 28], thereby enhancing factual awareness and generation accuracy. Standard RAG typically relies on chunk-based retrieval, segmenting documents into fixed-length text chunks retrieved via dense vector similarity, which overlooks the relationships between entities. Recently, GraphRAG [2] has emerged as a promising direction that structures knowledge as a graph to capture inter-entity relations, with the potential to improve retrieval efficiency and knowledge-driven generation [18].

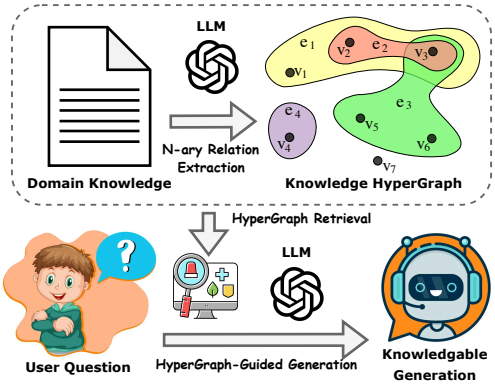


Figure 1: An illustration of HyperGraphRAG.

However, since each edge in an ordinary graph connects only two entities, existing graph-based RAG approaches [2, 7, 1, 8] are all restricted to **binary relations**, making them insufficient for modeling the **n-ary relations among more than two entities** that are widespread in real-world domain knowledge [25]. For example, in the medical domain, as illustrated in Figure 2, representing

[*] Corresponding author.
[1] https://github.com/LHRLAB/HyperGraphRAG

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

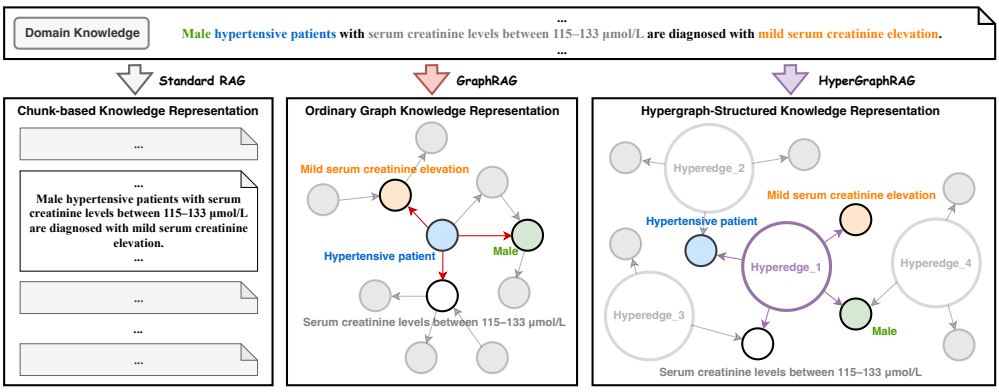


Figure 2: Comparison of knowledge representation: standard RAG uses chunks as units, GraphRAG captures binary relations with graphs, and HyperGraphRAG models n-ary relations with hyperedges.

the fact that "*Male hypertensive patients with serum creatinine levels between 115–133 μmol/L are diagnosed with mild serum creatinine elevation*" requires decomposing it into several binary relational triples, such as *Gender*:(*Hypertensive patient*, *Male*) and *Diagnosed_with*:(*Hypertensive patient*, *Mild serum creatinine elevation*), leading to representation sparsity during conversion process.

To address these limitations, we propose **HyperGraphRAG**, as illustrated in Figure 1, a novel graph-based RAG method built upon **hypergraph-structured knowledge representation**. In contrast to prior graph-based RAG methods constrained to binary relations, HyperGraphRAG leverages hyperedges to represent n-ary relational facts, where each hyperedge connects n entities ($n \geq 2$), e.g. *Hyperedge*:(*Hypertensive patient*, *Male*, *Serum creatinine levels between 115–133 μmol/L*, *Mild serum creatinine elevation*), and each hyperedge is expressed through natural language descriptions. This design ensures knowledge completeness, structural expressiveness, and inferential capability, thereby providing more comprehensive support for knowledge-intensive applications.

Our proposed HyperGraphRAG is built upon three key steps. First, we propose a **knowledge hypergraph construction method**, leveraging LLM-based n-ary relation extraction to extract and structure multi-entity relationships. The resulting hypergraph is stored in a bipartite graph database, with separate vector databases for entities and hyperedges to facilitate efficient retrieval. Second, we develop a **hypergraph retrieval strategy** that employs vector similarity search to retrieve relevant entities and hyperedges, ensuring that the knowledge retrieved is both precise and contextually relevant. Lastly, we introduce a **hypergraph-guided generation mechanism**, which combines retrieved n-ary facts with traditional chunk-based RAG passages, thereby improving response quality.

To validate the effectiveness, we conduct experiments in multiple knowledge-intensive domains [7], including medicine, agriculture, computer science, and law. Results demonstrate that HyperGraphRAG outperforms standard RAG and previous graph-based RAG methods in **answer accuracy**, **retrieval efficiency**, and **generation quality**, showcasing its strong potential for real-world applications.

2 Related Work

Graph-based RAG. GraphRAG [2] is the first graph-based RAG method that improves LLM generation via graph-based retrieval. Based on GraphRAG, several methods [26, 22, 11, 4, 23] focus on building graph-based RAG for different applications. LightRAG [7] enhances efficiency via graph indexing and updates. PathRAG [1] and HippoRAG2 [8] refine retrieval with path pruning and Personalized PageRank. However, all rely on binary relations, limiting knowledge expressiveness. In this work, we propose HyperGraphRAG, the first graph-based RAG method via hypergraph-structured knowledge representation. We compare several existing methods with HyperGraphRAG in Table 1.

Hypergraph Representation. Hypergraph-structured knowledge representation aims to overcome ordinary graph's limitations in modeling n-ary relations [15]. Early methods [25, 27, 12, 21] employ various embedding techniques to represent n-ary relational entities. Later methods [5, 24, 14] utilize GNN or attention to enhance embedding. However, existing methods mainly focus on link prediction, while hypergraphs also show potential for enhancing knowledge representation in graph-based RAG.

Table 1: Comparison of knowledge construction and retrieval methods for NaiveGeneration, StandardRAG, partial GraphRAG baselines, and our proposed HyperGraphRAG, where $\mathcal{K}$ represents the overall constructed knowledge, and K_q^* represents retrieved knowledge when given a user question q.

Method	Knowledge Construction	Knowledge Retrieval
NaiveGeneration	$\mathcal{K} = \emptyset$.	$K_q^* = \emptyset$.
StandardRAG	$\mathcal{K} = \{c_i\}_{i=1}^N$, where c_i is a chunk.	$K_q^* = K_{\text{chunk}} = \text{Top}_k\{\, c \in \mathcal{K} \mid \text{sim}(h_q, h_c)\,\}$
GraphRAG [2]	$\mathcal{K} = S = \{s_g \mid g \in \text{Community}(G)\}$, where S is the community summary set.	$K_q^* = \text{Detect}\{s_g \in S \mid q\}$, where detected community summaries are retrieved.
LightRAG [7]	$\mathcal{K} = G = (V, E)$, where V & E are entity & relation sets.	$K_q^* = \mathcal{F}\{v \in V, e \in E \mid q\} \cup K_{\text{chunk}}$, where entities & relations are retrieved with chunks.
PathRAG [1]	$\mathcal{K} = G = (V, E)$, where G is the same as LightRAG's.	$K_q^* = \text{Prune}\{p \in P_q \mid q\}$, where relational paths are retrieved via pruning.
HippoRAG2 [8]	$\mathcal{K} = G = (V \cup M,\ E)$, where V & M are phrase & passage nodes.	$K_q^* = \text{PageRank}\{m \in M \mid q\}$, where passages are retrieved via Personalized PageRank.
HyperGraphRAG (ours)	$\mathcal{K} = G_H = (V, E_H)$, where G_H is structured as a hypergraph.	$K_q^* = \mathcal{F}_n\{v \in V \mid q\} \cup \mathcal{F}_n\{e \in E_H \mid q\} \cup K_{\text{chunk}}$, where n-ary relational facts are retrieved with chunks.

3 Preliminaries

Definition 1: RAG. Given a question q and domain knowledge K, standard RAG first selects relevant document fragments d from K based on q, and then generates an answer y based on q and d. The probability model is formulated as:

$$P(y|q) = \sum_{d \in K} P(y|q, d)P(d|q, K). \tag{1}$$

Definition 2: Graph-based RAG. Graph-based RAG optimizes retrieval by representing knowledge as a graph structure $G = (V, E)$, where V is the set of entities and E is the set of relationships between entities. G consists of facts represented as $F = (e, V_e) \in G$, where e is the relation and V_e is the entity set connected to e. Given a question q, the retrieval process is defined as:

$$P(y|q) = \sum_{F \in G} P(y|q, F)P(F|q, G). \tag{2}$$

Definition 3: Hypergraph. A hypergraph $G_H = (V, E_H)$ [29] is a generalized graph, where V is the entity set, E_H is the hyperedge set, and each hyperedge $e_H \in E_H$ connects 2 or more entities:

$$V_{e_H} = (v_1, v_2, ..., v_n), \quad n \geq 2. \tag{3}$$

Unlike ordinary graphs, where relationships are binary $V_e = (v_h, v_t)$, hypergraphs model n-ary relational facts $F_n = (e_H, V_{e_H}) \in G_H$.

4 Method: HyperGraphRAG

In this section, we introduce the proposed HyperGraphRAG, as shown in Figure 3, including knowledge hypergraph construction, hypergraph retrieval strategy, and hypergraph-guided generation.

4.1 Knowledge Hypergraph Construction

To represent and store knowledge, we propose a knowledge hypergraph construction method that includes n-ary relational extraction, bipartite hypergraph storage, and vector representation storage.

N-ary Relation Extraction. To construct the knowledge hypergraph G_H, our first step is to extract multiple n-ary relational facts F_n from natural language documents $d \in K$. Unlike traditional hyper-relations [21], events [13], or other n-ary relation models [15], in the era of LLMs, to preserve richer and more diverse n-ary relations among entities, we propose a new n-ary relation representation $F_n = (e_H, V_{e_H})$, utilizing **natural language descriptions**, instead of structured relations, to represent hyperedges e_H among multiple entities V_{e_H} as follows.

(a) **Hyperedge:** Given an input text d, it is parsed into several independent knowledge fragments, each treated as a hyperedge: $E_H^d = \{e_1, e_2, ..., e_k\}$. Each hyperedge $e_i = (e_i^{\text{text}}, e_i^{\text{score}})$ consists of two parts: a natural language description e_i^{text}, and a confidence score $e_i^{\text{score}} \in (0, 10]$ indicating the association degree between e_i and d.

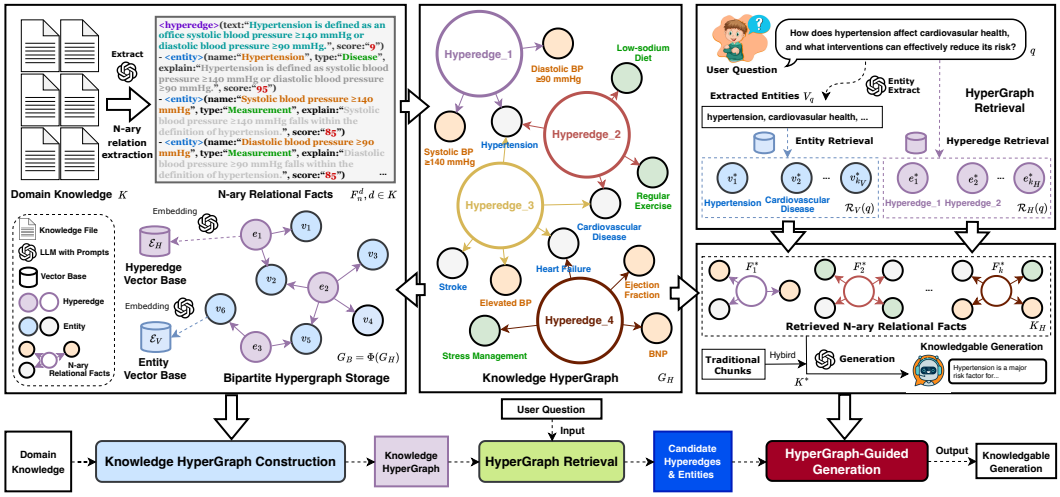


Figure 3: An overview of HyperGraphRAG, which constructs a knowledge hypergraph from domain knowledge, retrieves n-ary facts based on user questions, and generates knowledgeable responses.

(b) **Entity:** For each hyperedge e_i, entity recognition is performed to extract all contained entities: $V_{e_i} = \{v_1, v_2, ..., v_n\}$, where V_{e_i} is the entity set associated with e_i. Each entity $v_j = (v_j^{\text{name}}, v_j^{\text{type}}, v_j^{\text{explain}}, v_j^{\text{score}})$ consists of four parts: entity name $v_j^{\text{name}} \subseteq e_i^{\text{text}}$, type v_j^{type}, explanation v_j^{explain}, and confidence score $v_j^{\text{score}} \in (0, 100]$ indicating the extraction certainty.

Following this hypergraph-structured knowledge representation, we design an n-ary relation extraction prompt p_{ext}, detailed in Appendix A.1, to enable the LLM π to perform end-to-end knowledge fragment segmentation and entity recognition, thereby forming the n-ary relational fact set F_n^d:

$$F_n^d = \{f_1, f_2, ..., f_k\} \sim \pi(F_n | p_{\text{ext}}, d), \tag{4}$$

where each extracted n-ary relational fact $f_i = (e_i, V_{e_i})$ contains information about the corresponding hyperedge e_i and its associated entity set V_{e_i}. We convert all documents $d \in K$ into hyperedges and entities using n-ary relation extraction, forming a complete knowledge hypergraph G_H.

Proposition 1. *Hypergraph-structured knowledge representation is more comprehensive than binary.*

Proof. We provide experimental results in Section 5.4 and proofs in Appendix B.1. □

Bipartite Hypergraph Storage. After n-ary relation extraction, we store the constructed knowledge hypergraph G_H in a graph database to support an efficient query. We adopt an ordinary graph database represented as a bipartite graph structure $G_B = (V_B, E_B) = \Phi(G_H)$, to store the knowledge hypergraph $G_H = (V, E_H)$, where Φ is a transformation function defined as:

$$\Phi: \; V_B = V \cup E_H, \;\; E_B = \{(e_H, v) \mid e_H \in E_H, v \in V_{e_H}\}, \tag{5}$$

where V_B is the set of nodes in G_B, formed by merging the entity set V and the hyperedge set E_H from G_H. The edge set E_B captures the connections between each hyperedge $e_H \in E_H$ and its associated entities $v \in V_{e_H}$.

Based on G_B, we can efficiently query all entities associated with a hyperedge e_H or query all hyperedges linked to a specific entity v, thereby benefiting the optimized query efficiency of an ordinary graph database, as well as preserving the complete hypergraph-structured knowledge representation.

Moreover, G_B allows incremental updates through dynamically expansion: $G_B \leftarrow G_B \cup \Phi(G'_H)$, where G'_H represents newly added hypergraph information. The transformation of hyperedges and entities into the bipartite graph storage format enables seamless updates to the graph database.

Proposition 2. *A bipartite graph can losslessly preserve and query a knowledge hypergraph.*

Proof. We provide proofs in Appendix B.2. □

Vector Representation Storage. To support efficient semantic retrieval, we embed hyperedges $e_H \in E_H$ and entities $v \in V$ using the same embedding model f, ensuring that the vector representation of hyperedges and entities is in the same vector space as questions. Let Ψ be the vector function, then the vector representation storage for the knowledge hypergraph G_H is defined as: $\Psi(G_H) = (\mathcal{E}_H, \mathcal{E}_V)$, where $\mathcal{E}_H$ is the vector base of hyperedges and $\mathcal{E}_V$ is the vector base of entities:

$$\Psi: \ \mathcal{E}_H = \{\mathbf{h}_{e_H} \mid e_H \in E_H\}, \ \ \mathcal{E}_V = \{\mathbf{h}_v \mid v \in V\}, \tag{6}$$

where each hyperedge e_H and entity v in G_H is embedded into their vector representations: $\mathbf{h}_{e_H} = f(e_H)$, and $\mathbf{h}_v = f(v)$, respectively.

4.2 Hypergraph Retrieval Strategy

After constructing and storing the hypergraph G_H, we design an efficient retrieval strategy to match user questions with relevant hyperedges and entities.

Entity Retrieval. First, we extract key entities from the question q to facilitate subsequent matching. We design an entity extraction prompt $p_{\text{q\_ext}}$, detailed in Appendix A.2, along with the LLM π to extract the entity set V_q:

$$V_q \sim \pi(V|p_{\text{q\_ext}}, q). \tag{7}$$

After extracting entities, we retrieve the most relevant entities from the entity set V of the knowledge hypergraph G_H. We define the entity retrieval function $\mathcal{R}_V$, which retrieves the most relevant entities from $\mathcal{E}_V$ using cosine similarity:

$$\mathcal{R}_V(q) = \underset{v \in V}{\operatorname{argmax}}^{k_V} \left(\text{sim}(\mathbf{h}_{V_q}, \mathbf{h}_v) \odot v^{\text{score}}\right)_{>\tau_V}, \tag{8}$$

where $\mathbf{h}_{V_q} = f(V_q)$ is the concatenated text vector representation of the extracted entity set V_q, $\mathbf{h}_v \in \mathcal{E}_V$ is the vector representation of entity v, $\text{sim}(\cdot,\cdot)$ denotes the similarity function, $\odot$ represents element-wise multiplication between similarity and entity relevance score v^{score} determining the final ranking score, τ_V is the threshold for the entity retrieval score, and k_V is the limit on the number of retrieved entities.

Hyperedge Retrieval. Moreover, to expand the retrieval scope and capture complete n-ary relations within the hyperedge set E_H of the knowledge hypergraph G_H, we define the hyperedge retrieval function $\mathcal{R}_H$, which retrieves a set of hyperedges related to q:

$$\mathcal{R}_H(q) = \underset{e_H \in E_B}{\operatorname{argmax}}^{k_H} \left(\text{sim}(\mathbf{h}_q, \mathbf{h}_{e_H}) \odot e_H^{\text{score}}\right)_{>\tau_H}, \tag{9}$$

where $\mathbf{h}_q = f(q)$ is the text vector representation of q, $\mathbf{h}_{e_H} \in \mathcal{E}_H$ is the vector representation of the hyperedge e_H, $\odot$ represents element-wise multiplication between similarity and hyperedge relevance score e_H^{score} determining the final ranking score, τ_H is the threshold for the hyperedge retrieval score, and k_H limits the number of retrieved hyperedges.

4.3 Hypergraph-Guided Generation

To fully utilize the structured knowledge in the hypergraph, we propose a Hypergraph-Guided Generation mechanism, which consists of hypergraph knowledge fusion and generation augmentation.

Hypergraph Knowledge Fusion. The primary goal of hypergraph knowledge fusion is to expand and reorganize the retrieved n-ary relational knowledge to form a comprehensive knowledge input. Since q may only match partial entities or hyperedges, we further expand the retrieval scope. To obtain a complete set of n-ary relational facts, we design a bidirectional expansion strategy, that includes expanding hyperedges from retrieved entities and expanding entities from retrieved hyperedges.

First, given the entity set retrieved from q, denoted as $\mathcal{R}_V(q) = \{v_1, v_2, ..., v_{k_V}\}$, we retrieve all hyperedges in the knowledge hypergraph G_H that connect these entities:

$$\mathcal{F}_V^* = \bigcup_{v_i \in \mathcal{R}_V(q)} \{(e_H, V_{e_H}) \mid v_i \in V_{e_H}, e_H \in E_H\}. \tag{10}$$

Next, we expand the set of entities connected to the retrieved hyperedges $\mathcal{R}_H(q) = \{e_1, e_2, ..., e_{k_H}\}$:

$$\mathcal{F}_H^* = \bigcup_{e_i \in \mathcal{R}_H(q)} \{(e_i, V_{e_i}) \mid V_{e_i} \subseteq V\} \tag{11}$$

Finally, we merge the expanded hyperedge set $\mathcal{F}_V^*$ with the expanded entity set $\mathcal{F}_H^*$ to form a complete retrieved n-ary relational fact set $K_H = \mathcal{F}_V^* \cup \mathcal{F}_H^*$. This set contains all necessary n-ary relational knowledge for reasoning and generation, ensuring a comprehensive input for the LLM.

Generation Augmentation. Following hypergraph knowledge fusion, we augment the generation strategy to improve the accuracy and readability of the responses. We adopt a hybrid RAG fusion mechanism, combining hypergraph knowledge K_H with retrieved chunk-based text fragments K_{chunk} to form the final knowledge input. We define the final knowledge input $K^* = K_H \cup K_{\text{chunk}}$, where K_{chunk} consists of chunk-based text fragments retrieved using traditional RAG.

Finally, we use a retrieval-augmented generation prompt p_{gen}, detailed in Appendix A.3, that combines hypergraph knowledge K^* and the user question q as input to LLM π to generate final response y^*:

$$y^* \sim \pi(y|p_{\text{gen}}, K^*, q). \tag{12}$$

Proposition 3. *Retrieving knowledge on a knowledge hypergraph improves retrieval efficiency compared to methods based on ordinary binary graphs, leading to gains in generation quality.*

Proof. We provide experimental results in Sections 5.5 and 5.6 and proofs in Appendix B.3. □

5 Experiments

This section presents the experimental setup, main results, and analysis. We answer the following research questions (RQs): **RQ1:** Does HyperGraphRAG outperform other methods? **RQ2:** Does the main component of HyperGraphRAG work? **RQ3:** How effective is the knowledge hypergraph constructed by HyperGraphRAG across various domains? **RQ4:** Could the hypergraph retrieval strategy improve retrieval efficiency? **RQ5:** How effective is the generation quality of HyperGraphRAG? **RQ6:** How are the time and cost of HyperGraphRAG in construction and generation phases?

5.1 Experimental Setup

Datasets. To evaluate the performance of HyperGraphRAG across multiple domains, we select four knowledge contexts from UltraDomain [19], as used in LightRAG [7]: **Agriculture**, Computer Science (**CS**), **Legal**, and a mixed domain (**Mix**). In addition, we include the latest international hypertension guidelines [16] as the foundational knowledge for the **Medicine** domain. For each of the five domains, we sample knowledge fragments one, two, and three hops away to construct questions with ground-truth answers verified by human annotators. We then categorize the questions into **Binary Source** and **N-ary Source**, based on whether the sampled knowledge of the question contains facts among n entities ($n > 2$). More details can be found in Appendix D.

Baselines. We compare HyperGraphRAG against six publicly available baseline methods: **Naive-Generation** [17], which directly generates responses using LLM; **StandardRAG** [6], a traditional chunk-based RAG approach; **GraphRAG** [2], **LightRAG** [7], **PathRAG** [1], and **HippoRAG2** [8], which are four selected available graph-based RAG methods described in Table 1. To ensure fairness, we use the same generation prompt, which can be found in Appendix E.

Evaluation Metrics. We evaluate the answer accuracy, retrieval efficiency, and generation quality of HyperGraphRAG and its baselines using 3 key metrics: **F1**, Retrieval Similarity (**R-S**), and Generation Evaluation (**G-E**). F1 measures word-level similarity between the generated answer and the ground-truth answer, following FlashRAG [9]. R-S assesses the semantic similarity between the retrieved knowledge and the ground-truth knowledge used to construct the question, in line with RAGAS [3]. G-E, inspired by HelloBench [20], is a metric that uses LLM-as-a-judge to evaluate generation quality in 7 dimensions and reports the average score. Details are provided in Appendix E.

Implementation Details. We use OpenAI's `GPT-4o-mini` for extraction and generation, and `text-embedding-3-small` for vector. During retrieval, we set the following parameters: entity retrieval $k_V = 60$, $\tau_V = 50$; hyperedge retrieval $k_H = 60$, $\tau_H = 5$; and chunk retrieval $k_C = 5$, $\tau_C = 0.5$. All experiments were conducted on a server with an 80-core CPU and 512GB RAM.

5.2 Main Results (RQ1)

To evaluate the effectiveness of HyperGraphRAG, we compare its performance with various baselines across multiple domains. The results are shown in Table 2.

Table 2: Performance comparison across different domains. **Bold** indicates the best performance.

Method	Medicine			Agriculture			CS			Legal			Mix		
	F1	R-S	G-E	F1	R-S	G-E	F1	R-S	G-E	F1	R-S	G-E	F1	R-S	G-E
Binary Source															
NaiveGeneration	12.63	0.00	44.70	11.71	0.00	45.76	18.93	0.00	48.79	22.91	0.00	50.00	18.58	0.00	46.14
StandardRAG	26.87	61.08	56.24	28.31	42.69	57.58	28.87	49.44	57.10	37.19	52.21	59.85	47.57	46.79	67.42
GraphRAG	17.13	54.56	48.19	20.67	40.90	52.41	23.75	37.65	53.17	31.09	34.26	54.62	23.62	25.01	48.12
LightRAG	12.16	52.38	44.15	17.70	41.24	50.32	22.59	41.86	51.62	33.63	45.54	56.42	29.98	34.22	54.50
PathRAG	14.74	52.30	45.36	21.97	42.21	53.13	25.28	41.49	53.28	32.32	43.60	55.45	40.87	33.36	60.75
HippoRAG2	21.12	57.50	51.08	12.60	16.85	44.56	16.94	21.05	47.29	20.10	34.13	46.77	21.10	18.34	45.83
HyperGraphRAG (ours)	**36.45**	**69.91**	**60.65**	**34.80**	**61.97**	**59.99**	**31.60**	**60.94**	**57.54**	**44.42**	**60.87**	**63.53**	**51.51**	**67.34**	**68.76**
N-ary Source															
NaiveGeneration	13.15	0.00	41.83	13.78	0.00	47.93	18.37	0.00	48.94	20.37	0.00	48.09	15.29	0.00	45.16
StandardRAG	28.93	64.06	55.08	26.55	48.93	56.62	28.99	47.35	56.69	37.50	51.16	60.09	38.83	47.73	61.82
GraphRAG	18.07	57.22	47.09	21.90	41.27	53.49	22.90	39.97	53.76	29.12	34.11	53.76	14.93	24.32	42.32
LightRAG	13.43	54.67	41.86	18.78	42.44	50.92	22.85	41.19	52.20	29.64	44.47	54.65	24.08	33.22	50.83
PathRAG	15.14	54.08	42.77	20.64	42.53	51.83	28.18	42.29	54.97	30.27	44.47	55.26	33.27	34.11	57.47
HippoRAG2	21.56	61.54	48.06	12.66	20.32	45.14	17.75	26.92	48.44	16.95	34.72	45.09	21.95	18.49	46.87
HyperGraphRAG (ours)	**34.26**	**70.48**	**58.06**	**32.98**	**62.58**	**59.59**	**31.00**	**59.25**	**58.35**	**43.20**	**60.07**	**63.70**	**45.91**	**69.09**	**65.04**
Overall															
NaiveGeneration	12.89	0.00	43.27	12.74	0.00	46.85	18.65	0.00	48.87	21.64	0.00	49.05	16.93	0.00	45.65
StandardRAG	27.90	62.57	55.66	27.43	45.81	57.10	28.93	48.40	56.89	37.34	51.68	59.97	43.20	47.26	64.62
GraphRAG	17.60	55.89	47.64	21.28	41.08	52.95	23.33	38.81	53.47	30.11	34.18	54.19	19.27	24.67	45.22
LightRAG	12.79	53.52	43.00	18.24	41.84	50.62	22.72	41.53	51.91	31.64	45.00	55.53	27.03	33.72	52.67
PathRAG	14.94	53.19	44.06	21.30	42.37	52.48	26.73	41.89	54.13	31.29	44.03	55.36	37.07	33.73	59.11
HippoRAG2	21.34	59.52	49.57	12.63	18.58	44.85	17.34	23.99	47.87	18.53	34.42	45.93	21.53	18.42	46.35
HyperGraphRAG (ours)	**35.35**	**70.19**	**59.35**	**33.89**	**62.27**	**59.79**	**31.30**	**60.09**	**57.94**	**43.81**	**60.47**	**63.61**	**48.71**	**68.21**	**66.90**

Overall Comparison Across Methods. HyperGraphRAG consistently outperforms all baselines across F1, R-S, and G-E metrics. Compared to StandardRAG, it achieves gains of +7.45 (F1), +7.62 (R-S), and +3.69 (G-E). Interestingly, existing graph-based RAG baselines often underperform StandardRAG, as their reliance on binary relational graphs causes knowledge fragmentation, sparsified retrieval, and incomplete context reconstruction during generation.

Comparison Across Source Types. HyperGraphRAG maintains strong gains under both Binary and N-ary settings. For Binary Source, it improves F1, R-S, and G-E by +8.6, +8.8, and +4.4; for N-ary Source, the improvements are +5.3, +6.4, and +2.9, confirming its robustness.

Comparison Across Domains. Performance gains are consistent across domains, with the largest improvements in Medicine and Legal (over +7 F1), and stable advantages in Agriculture and CS. HyperGraphRAG adapts well to both highly structured and more general knowledge tasks.

5.3 Ablation Study (RQ2)

As shown in Figure 4, we conduct an ablation study in the Medicine domain by removing entity retrieval (w/o ER), hyperedge retrieval (w/o HR), and their combination (w/o ER & HR). We also remove chunk retrieval fusion (w/o CR), and all modules (w/o ER & HR & CR):

Impact of Entity Retrieval (ER). ER is critical for precise retrieval by anchoring key concepts. Without ER, F1 falls from 35.4 to 29.8, underscoring its importance in selecting relevant entities for accurate generation.

Impact of Hyperedge Retrieval (HR). HR captures n-ary, multi-entity facts necessary for complex reasoning. Removing HR drops F1 from 35.4 to 26.4, highlighting its unique role beyond mere entity retrieval.

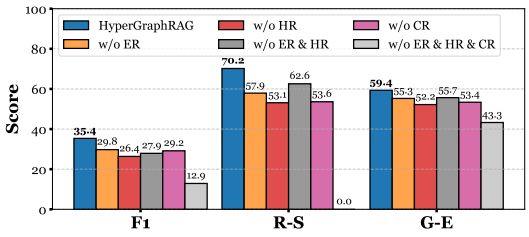
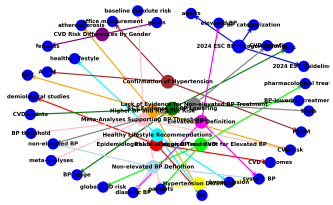


Figure 4: Results of the ablation study.

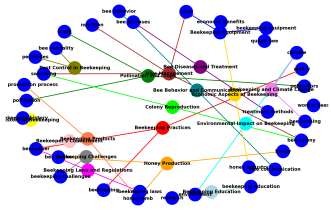
Impact of Chunk Retrieval Fusion (CR). CR enhances retrieval by integrating unstructured text with hypergraph data. Excluding CR reduces F1 from 35.4 to 29.2, demonstrating that the fusion leads to more complete and fluent generation.

5.4 Analysis of Hypergraph-structured Knowledge Representation (RQ3)

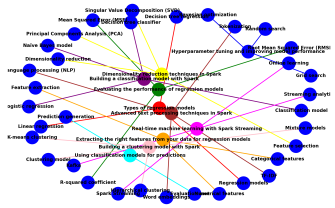
As shown in Figure 5, we assess HyperGraphRAG's knowledge representation across 5 domains:



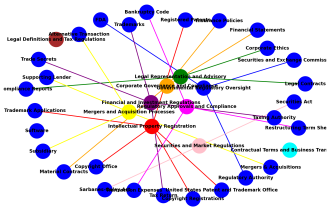
(a) Medicine HyperGraph



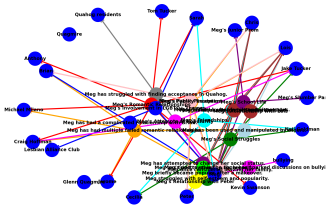
(b) Agriculture HyperGraph



(c) CS HyperGraph



(d) Legal HyperGraph



(e) Mix HyperGraph

	Med.	Agric.	CS	Legal	Mix
#Knowl. Token	179k	382k	795k	940k	122k
GraphRAG					
#Entity	329	699	1449	1711	225
#Community	256	523	930	517	59
LightRAG					
#Entity	3,725	5,032	8,967	5,354	2,229
#Relation	1,304	3,105	5,632	6,002	940
HyperGraphRAG					
#Entity	7,675	16,805	19,913	11,098	6,201
#Hyperedge	4,818	16,102	26,902	18,285	4,356

(f) Statistics of Construction

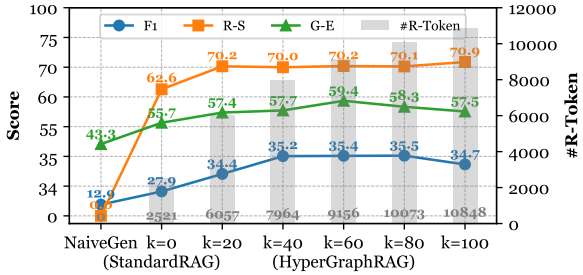
Figure 5: (a-e) Visualizations of knowledge hypergraphs constructed in 5 domains. (f) Statistical comparison highlights HyperGraphRAG's richer expressiveness over GraphRAG and LightRAG.

Visualization of Knowledge Structures. As shown in Figure 5(a)-5(e), unlike previous graph-based RAG methods, which only model binary relations, HyperGraphRAG connects multiple entities via hyperedges, forming a more interconnected and expressive network.

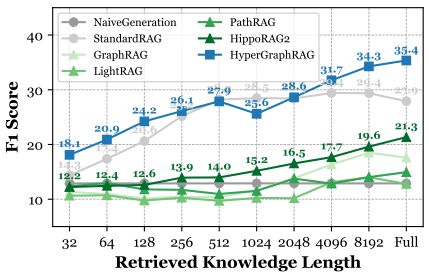
Statistical Analysis. As shown in Figure 5(f), HyperGraphRAG surpasses GraphRAG and LightRAG in all domains. For instance, in CS, it constructs 26,902 hyperedges, whereas GraphRAG has 930 communities and LightRAG 5,632 relations, showing a stronger capacity for capturing knowledge.

5.5 Analysis of Hypergraph Retrieval Efficiency (RQ4)

As shown in Figure 6, to evaluate retrieval efficiency, we conduct two experiments: (a) examining how HyperGraphRAG's retrieval efficiency and token length scales with different top-k values and (b) comparing its F1 scores with other methods under varying retrieval length limits:



(a) Impact of Top-k on Retreval Efficency & Token Length.



(b) F1 Comparison under Limited Lengths.

Figure 6: Experimental results in the Medicine domain analyzing hypergraph retrieval efficiency.

Impact of Retrieved Hyperedge Quantity. As shown in Figure 6(a), increasing the top-k hyperedges improves F1, R-S, and G-E, along with the rise in token count. Performance saturates around k = 60, indicating that HyperGraphRAG achieves strong retrieval quality with limited input.

Performance under Constrained Retrieval Length. As illustrated in Figure 6(b), HyperGraphRAG outperforms all binary graph-based RAG methods even under retrieval length limits, demonstrating the efficiency of n-ary representations and highlighting the semantic loss inherent in binary structures.

5.6 Analysis of Hypergraph-Guided Generation Quality (RQ5)

As shown in Figure 7, we evaluate the quality of the generation in seven dimensions:

Best Overall Generation Quality. HyperGraphRAG achieves the highest Overall score (61.5), significantly outperforming all baseline methods, indicating the comprehensive advantage in hypergraph-guided generation.

Lead on Key Dimensions. HyperGraphRAG achieves notable improvements in Correctness (64.8), Relevance (66.0), and Factuality (64.2), outperforming both standard RAG and binary graph-based methods. These gains indicate its strong capacity to produce accurate, context-aware, and knowledge-grounded responses.

Balanced Performance. Although the Diversity score (47.0) is relatively lower than other dimensions, HyperGraphRAG still exceeds all baselines, indicating that it maintains a balanced dimension-wise performance, effectively combining content richness with structural consistency for stable and high-quality generation.

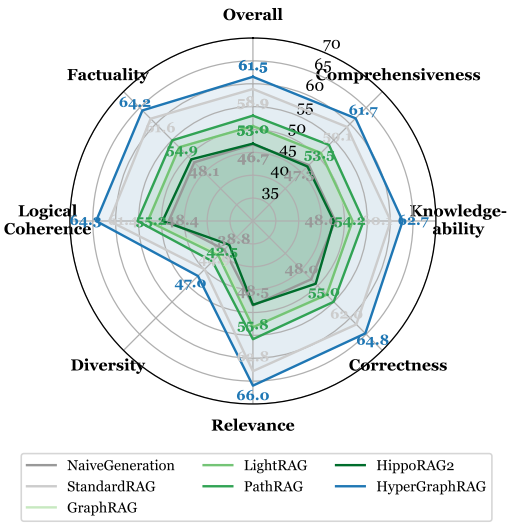


Figure 7: Generation Equality Evaluations.

5.7 Analysis of Time and Cost in Construction and Generation Phases (RQ6)

As shown in Table 3, to evaluate the efficiency and cost of HyperGraphRAG, we compare different methods in terms of knowledge construction and generation. We assess time consumption per 1k tokens (TP1kT), cost per 1k tokens (CP1kT), time per query (TPQ), and cost per 1k query (CP1kQ).

Time & Cost in Construction Phase. HyperGraphRAG demonstrates efficient knowledge construction with a time cost of 3.084 seconds per 1k tokens (TP1kT) and a monetary cost of $0.0063 per 1k tokens (CP1kT). This places it between the faster HippoRAG2 (2.758s, $0.0056) and slower GraphRAG (9.272s, $0.0058). While its cost is slightly higher than GraphRAG, HyperGraphRAG achieves a better balance between speed, expressiveness, and structure, offering a more compact yet richer representation of n-ary relational knowledge.

Table 3: Time & Cost Comparisons.

Method	Construction		Generation	
	TP1kT	CP1kT	TPQ	CP1kQ
NaiveGeneration	0 s	0 $	0.131 s	0.059 $
StandardRAG	0 s	0 $	0.147 s	1.016 $
GraphRAG	9.272 s	0.0058 $	0.221 s	1.836 $
LightRAG	5.168 s	0.0081 $	0.359 s	3.359 $
PathRAG	5.168 s	0.0081 $	0.436 s	3.496 $
HippoRAG2	2.758 s	0.0056 $	0.240 s	3.438 $
HyperGraphRAG	3.084 s	0.0063 $	0.256 s	3.184 $

Time & Cost in Generation Phase. During the generation phase, HyperGraphRAG requires 0.256 seconds per query (TPQ) and incurs a cost of $3.184 per 1k queries (CP1kQ). This is moderately higher than StandardRAG (0.147s, $1.016) but significantly lower than PathRAG (0.436s, $3.496) and LightRAG (0.359s, $3.359). Compared to GraphRAG (0.221s, $1.836), HyperGraphRAG slightly increases time and cost but compensates with better retrieval quality and generation outcomes. The results suggest that HyperGraphRAG achieves a favorable trade-off between generation efficiency and output quality, suitable for real-world knowledge-intensive applications.

6 Conclusion

In this work, we present HyperGraphRAG, a retrieval-augmented generation framework that models knowledge as hypergraphs to capture n-ary relational structures. By introducing novel methods for knowledge hypergraph construction, retrieval, and generation, HyperGraphRAG addresses limitations of binary graph-based RAG methods. Experimental results across diverse domains demonstrate consistent improvements in answer accuracy, retrieval relevance, and generation quality, confirming the effectiveness and generalizability of hypergraph-guided retrieval and generation.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (Grant No. 62473271, Grant No. 62176026, and Grant No. 62406036) and the Engineering Research Center of Information Networks, Ministry of Education, China.

References

[1] Boyu Chen, Zirui Guo, Zidan Yang, Yuluo Chen, Junze Chen, Zhenghao Liu, Chuan Shi, and Cheng Yang. Pathrag: Pruning graph-based retrieval augmented generation with relational paths, 2025.

[2] 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, 2024.

[3] Shahul Es, Jithin James, Luis Espinosa Anke, and Steven Schockaert. RAGAs: Automated evaluation of retrieval augmented generation. In Nikolaos Aletras and Orphee De Clercq, editors, *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 150–158, St. Julians, Malta, March 2024. Association for Computational Linguistics.

[4] Tianyu Fan, Jingyuan Wang, Xubin Ren, and Chao Huang. Minirag: Towards extremely simple retrieval-augmented generation, 2025.

[5] Mikhail Galkin, Priyansh Trivedi, Gaurav Maheshwari, Ricardo Usbeck, and Jens Lehmann. Message passing for hyper-relational knowledge graphs. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7346–7359, Online, November 2020. Association for Computational Linguistics.

[6] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. Retrieval-augmented generation for large language models: A survey, 2024.

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

[8] Bernal Jiménez Gutiérrez, Yiheng Shu, Weijian Qi, Sizhe Zhou, and Yu Su. From rag to memory: Non-parametric continual learning for large language models, 2025.

[9] Jiajie Jin, Yutao Zhu, Xinyu Yang, Chenghao Zhang, and Zhicheng Dou. Flashrag: A modular toolkit for efficient retrieval-augmented generation research. *CoRR*, abs/2405.13576, 2024.

[10] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc., 2020.

[11] Lei Liang, Mengshu Sun, Zhengke Gui, Zhongshu Zhu, Zhouyu Jiang, Ling Zhong, Yuan Qu, Peilong Zhao, Zhongpu Bo, Jin Yang, Huaidong Xiong, Lin Yuan, Jun Xu, Zaoyang Wang, Zhiqiang Zhang, Wen Zhang, Huajun Chen, Wenguang Chen, and Jun Zhou. Kag: Boosting llms in professional domains via knowledge augmented generation, 2024.

[12] Yu Liu, Quanming Yao, and Yong Li. Generalizing tensor decomposition for n-ary relational knowledge bases. In *Proceedings of The Web Conference 2020*, WWW '20, page 1104–1114, New York, NY, USA, 2020. Association for Computing Machinery.

[13] Yaojie Lu, Hongyu Lin, Jin Xu, Xianpei Han, Jialong Tang, Annan Li, Le Sun, Meng Liao, and Shaoyi Chen. Text2Event: Controllable sequence-to-structure generation for end-to-end event extraction. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2795–2806, Online, August 2021. Association for Computational Linguistics.

[14] Haoran Luo, Haihong E, Yuhao Yang, Yikai Guo, Mingzhi Sun, Tianyu Yao, Zichen Tang, Kaiyang Wan, Meina Song, and Wei Lin. HAHE: Hierarchical attention for hyper-relational knowledge graphs in global and local level. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8095–8107, Toronto, Canada, July 2023. Association for Computational Linguistics.

[15] Haoran Luo, Haihong E, Yuhao Yang, Tianyu Yao, Yikai Guo, Zichen Tang, Wentai Zhang, Kaiyang Wan, Shiyao Peng, Meina Song, Wei Lin, Yifan Zhu, and Luu Anh Tuan. Text2nkg: Fine-grained n-ary relation extraction for n-ary relational knowledge graph construction, 2024.

[16] John William McEvoy, Cian P McCarthy, Rosa Maria Bruno, Sofie Brouwers, Michelle D Canavan, Claudio Ceconi, Ruxandra Maria Christodorescu, Stella S Daskalopoulou, Charles J Ferro, Eva Gerdts, et al. 2024 esc guidelines for the management of elevated blood pressure and hypertension. *Giornale italiano di cardiologia (2006)*, 25(11):1e–107e, 2024.

[17] OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, et al. Gpt-4 technical report, 2024.

[18] Shirui Pan, Linhao Luo, Yufei Wang, Chen Chen, Jiapu Wang, and Xindong Wu. Unifying large language models and knowledge graphs: A roadmap. *IEEE Transactions on Knowledge and Data Engineering*, 36(7):3580–3599, 2024.

[19] Hongjin Qian, Peitian Zhang, Zheng Liu, Kelong Mao, and Zhicheng Dou. Memorag: Moving towards next-gen rag via memory-inspired knowledge discovery, 2024.

[20] Haoran Que, Feiyu Duan, Liqun He, Yutao Mou, Wangchunshu Zhou, Jiaheng Liu, Wenge Rong, Zekun Moore Wang, Jian Yang, Ge Zhang, Junran Peng, Zhaoxiang Zhang, Songyang Zhang, and Kai Chen. Hellobench: Evaluating long text generation capabilities of large language models, 2024.

[21] Paolo Rosso, Dingqi Yang, and Philippe Cudré-Mauroux. Beyond triplets: Hyper-relational knowledge graph embedding for link prediction. In *Proceedings of The Web Conference 2020*, WWW '20, page 1885–1896, New York, NY, USA, 2020. Association for Computing Machinery.

[22] Kartik Sharma, Peeyush Kumar, and Yunqing Li. Og-rag: Ontology-grounded retrieval-augmented generation for large language models, 2024.

[23] Jinyu Wang, Jingjing Fu, Rui Wang, Lei Song, and Jiang Bian. Pike-rag: specialized knowledge and rationale augmented generation, 2025.

[24] Quan Wang, Haifeng Wang, Yajuan Lyu, and Yong Zhu. Link prediction on n-ary relational facts: A graph-based approach. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 396–407, Online, August 2021. Association for Computational Linguistics.

[25] Jianfeng Wen, Jianxin Li, Yongyi Mao, Shini Chen, and Richong Zhang. On the representation and embedding of knowledge bases beyond binary relations. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, IJCAI'16, page 1300–1307. AAAI Press, 2016.

[26] Junde Wu, Jiayuan Zhu, Yunli Qi, Jingkun Chen, Min Xu, Filippo Menolascina, and Vicente Grau. Medical graph rag: Towards safe medical large language model via graph retrieval-augmented generation, 2024.

[27] Richong Zhang, Junpeng Li, Jiajie Mei, and Yongyi Mao. Scalable instance reconstruction in knowledge bases via relatedness affiliated embedding. In *Proceedings of the 2018 World Wide Web Conference*, WWW '18, page 1185–1194, Republic and Canton of Geneva, CHE, 2018. International World Wide Web Conferences Steering Committee.

[28] Wayne Xin Zhao, Kun Zhou, Junyi Li, et al. A survey of large language models, 2024.

[29] Dengyong Zhou, Jiayuan Huang, and Bernhard Schölkopf. Learning with hypergraphs: Clustering, classification, and embedding. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems*, volume 19. MIT Press, 2006.

Appendix

A Prompts Used in HyperGraphRAG

A.1 N-ary Relation Extraction Prompt

As shown in Figure 8, this prompt is designed for extracting structured n-ary relational facts from raw text. It guides LLM to segment the input into coherent knowledge fragments, assign a completeness score to each, and identify entities with their names, types, descriptions, and importance scores.

```
-Goal-
Given a text document that is potentially relevant to this activity and a list of entity types, identify all entities of those types from the text and all relationships among the identified entities.
Use {language} as output language.

-Steps-
1. Divide the text into several complete knowledge segments.  For each knowledge segment, extract the following information:
-- knowledge_segment: A sentence that describes the context of the knowledge segment.
-- completeness_score: A score from 0 to 10 indicating the completeness of the knowledge segment.
Format each knowledge segment as ("hyper-relation"{tuple_delimiter}<knowledge_segment>{tuple_delimiter}<completeness_score>)

2. Identify all entities in each knowledge segment. For each identified entity, extract the following information:
- entity_name: Name of the entity, use same language as input text. If English, capitalized the name.
- entity_type: Type of the entity.
- entity_description: Comprehensive description of the entity's attributes and activities.
- key_score: A score from 0 to 100 indicating the importance of the entity in the text.
Format each entity as ("entity"{tuple_delimiter}<entity_name>{tuple_delimiter}<entity_type>{tuple_delimiter}<entity_description>{tuple_delimiter}<key_score>)

3. Return output in {language} as a single list of all the entities and relationships identified in steps 1 and 2. Use **{record_delimiter}** as the list delimiter.

4. When finished, output {completion_delimiter}

######################
-Examples-
######################
{examples}

#############################
-Real Data-
######################
Text: {input_text}
######################
Output:
```

Figure 8: Prompt for n-ary relation extraction p_{ext} in Equation 4.

A.2 Entity Extraction Prompt

As shown in Figure 9, this prompt is used to extract key entities from a user query. LLM is instructed to return all identified entities in JSON format, ensuring the output is concise, human-readable, and aligned with the language of the input query. This facilitates entity-level retrieval in the hypergraph.

```
---Role---

You are a helpful assistant tasked with identifying entities in the user's query.

---Goal---

Given the query, list all entities.

---Instructions---

- Output the keywords in JSON format.

######################
-Examples-
######################
{examples}

#############################
-Real Data-
######################
Query: {query}
######################
The `Output` should be human text, not unicode characters. Keep the same language as `Query`.
Output:
```

Figure 9: Prompt for entity extraction $p_{\text{q\_ext}}$ in Equation 7.

A.3 Retrieval-Augmented Generation Prompt

To ensure a fair comparison across RAG baselines, we adopt a unified Chain-of-Thought (CoT)-based generation prompt p_{gen} in Equation 12 for all methods. We present this prompt together with the designed evaluation approach in Appendix E.

B Proof

B.1 Proof of Proposition 1

Proposition 1. *Hypergraph-structured knowledge representation is more comprehensive than binary.*

Proof. Given a universe of entities V, an n-ary fact with $n \geq 3$ is denoted as $F = \{v_1, \ldots, v_n\} \subseteq V$. For hypergraph representation, we represent it with a single hyperedge:

$$e_H = F, \quad G_H = (V, E_H), \quad e_H \in E_H, \tag{13}$$

so the representation function $\phi_H : F \mapsto e_H$ is naturally injective. For binary graph representation, we connect every pair of entities that co-occur in a fact. For any collection of facts $\mathcal{S} \subseteq \mathcal{P}(V)$, define the representation function:

$$\phi_B(\mathcal{S}) = (V_B, E_B), \quad V_B = \bigcup_{F \in \mathcal{S}} F, \quad E_B = \{(u, w) \mid u \neq w,\ \exists F \in \mathcal{S} :\ \{u, w\} \subseteq F\}, \tag{14}$$

where E_B consists of the binary edges activated by $\mathcal{S}$ within the complete graph $K_{|V|}$. Each E_B is a subset of some clique.

Let the random variable X range over all possible fact sets $\mathcal{S}$, with Shannon entropy:

$$H(X) = -\sum_{\mathcal{S}} p(\mathcal{S}) \log_2 p(\mathcal{S}), \tag{15}$$

measuring the total information to be represented. For hypergraph representation, since ϕ_H is injective and each fact can be uniquely recovered,

$$H(X \mid \phi_H(X)) = 0. \tag{16}$$

For binary representation, consider any three distinct entities $a, b, c \in V$, and define

$$\mathcal{S}_1 = \{\{a, b, c\}\}, \quad \mathcal{S}_2 = \big\{\{a, b\},\ \{a, c\},\ \{b, c\}\big\}. \tag{17}$$

Clearly, $\mathcal{S}_1 \neq \mathcal{S}_2$, but

$$\phi_B(\mathcal{S}_1) = \phi_B(\mathcal{S}_2) = \big(\{a, b, c\},\ \{(a, b), (a, c), (b, c)\}\big) = g, \tag{18}$$

since both activate the same set of binary edges. Thus,

$$\begin{aligned} &\big|\phi_B^{-1}(\phi_B(\mathcal{S}_1))\big| \geq 2, \quad \Rightarrow \quad 0 < \frac{p(\mathcal{S}_i)}{p(g)} < 1, \\ \Rightarrow \quad &H(X \mid \phi_B(X) = g) = -\sum_{\mathcal{S}_i \in \phi_B^{-1}(g)} \frac{p(\mathcal{S}_i)}{p(g)} \log_2 \frac{p(\mathcal{S}_i)}{p(g)} > 0, \end{aligned} \tag{19}$$

then, we can get

$$H(X \mid \phi_B(X)) = \sum_y p(y) H(X \mid \phi_B(X) = y) \geq p(g) H(X \mid \phi_B(X) = g) > 0, \tag{20}$$

where information is inevitably lost in binary representation.

More generally, as long as there exists at least one n-ary fact ($n \geq 3$) in the knowledge base, we can always construct a pair of distinct fact sets that activate the same binary edges through a merge-split transformation. Hence,

$$H(X \mid \phi_B(X)) > 0, \quad I(X; \phi_B(X)) = H(X) - H(X \mid \phi_B(X)) < H(X), \tag{21}$$

which proves that binary representation is lossy. In contrast, hypergraph representation satisfies $H(X \mid \phi_H(X)) = 0$, so the mutual information reaches its upper bound $H(X)$ and all information is preserved. In the special case where no n-ary facts with $n \geq 3$ exist, i.e., all facts are binary, then

$$\big|\phi_B^{-1}(g_B)\big| = 1, \quad H(X \mid \phi_B(X)) = 0, \tag{22}$$

so binary representation becomes injective and equivalent to hypergraph, with no information loss.

In conclusion, as long as the knowledge base contains at least one fact of arity three or higher, hypergraph-structured representation preserves more information with lossless representation, whereas binary representation inevitably loses information. Therefore, hypergraph representation is more comprehensive than binary in the information-theoretic sense. □

B.2 Proof of Proposition 2

Proposition 2. *A bipartite graph can losslessly preserve and query a knowledge hypergraph.*

Proof. Let the knowledge hypergraph be denoted as $G_H = (V, E_H), E_H \subseteq \{\, e_H \subseteq V \mid |e_H| \geq 2\}$. Each hyperedge is abstracted as a new node, and combined with the set of entity nodes to form a new vertex set $V_B = V \cup E_H$, with edges defined as $E_B = \{\, (e_H, v) \mid e_H \in E_H,\ v \in e_H\}$, resulting in the incidence bipartite graph $\Phi(G_H) = G_B = (V_B, E_B)$.

Ordering the vertices such that entities come first and hyperedges second, G_H can be represented by the binary incidence matrix

$$M \in \{0,1\}^{|V|\times|E_H|}, \quad M_{v,e_H} = 1 \iff v \in e_H, \tag{23}$$

and the adjacency matrix of G_B becomes

$$A_{G_B} = \begin{pmatrix} 0 & M \\ M^\top & 0 \end{pmatrix} \tag{24}$$

where M uniquely determines A_{G_B}, and conversely, M can be recovered from the top-right block of A_{G_B}. Therefore, there exists an inverse mapping:

$$\Phi^{-1} : G_B \to G_H, \quad \Phi^{-1}(V_B, E_B) = (V, \{\, N_{G_B}(e_H) \mid e_H \in E_H\}), \tag{25}$$

where

$$N_{G_B}(e_H) = \{v \in V \mid (e_H, v) \in E_B\}. \tag{26}$$

Clearly,

$$\Phi^{-1} \circ \Phi = \mathrm{id}_{G_H}, \quad \Phi \circ \Phi^{-1} = \mathrm{id}_{G_B}, \tag{27}$$

which means that Φ is a bijection and the encoding is lossless.

The query equivalence can also be derived directly via matrix operations and path counting: the set of hyperedges containing an entity v corresponds to the support of the v-th row of M, and in the bipartite graph this is equivalent to the neighborhood $N_{G_B}(v)$, given by the right block of $\mathbf{e}_v^\top A_{G_B} = (0,\ \mathbf{e}_v^\top M)$. Likewise, the entity set of a hyperedge e_H is the support of the e_H-th column of M, which matches the left block of $\mathbf{f}_{e_H}^\top A_{G_B}$. To determine whether two entities u, v co-occur in some hyperedge, it suffices to check whether

$$(MM^\top)_{uv} = (A_{G_B}^2)_{uv} \neq 0, \tag{28}$$

since $(A_{G_B}^2)_{uv}$ counts all 2-step paths from u through a hyperedge node to v. For a given subset of entities $S \subseteq V$, hyperedges that contain all of them can be found by summing the corresponding rows $\sum_{v\in S} M_{v,*}$ and selecting columns where the sum equals $|S|$; in the bipartite graph, this corresponds to the intersection

$$\bigcap_{v\in S} N_{G_B}(v). \tag{29}$$

All operations run in time $O(|E_B|)$, which matches the complexity of equivalent queries over G_H.

In conclusion, the bijection Φ guarantees full structural reversibility, while adjacency and path-based reasoning preserve the semantics of all queries involving entity–hyperedge membership. Therefore, a bipartite graph can losslessly preserve and query a knowledge hypergraph. □

B.3 Proof of Proposition 3

Proposition 3. *Retrieving knowledge on a knowledge hypergraph improves retrieval efficiency compared to methods based on ordinary binary graphs, leading to gains in generation quality.*

Proof. Let the ground-truth knowledge set required for a query q be modeled as a discrete random variable $X \subseteq \mathcal{P}(V)$, with probability measure μ defined over the measurable space $(\mathcal{P}(V), \mathcal{B})$. For any n-ary fact $F = \{v_1, \ldots, v_n\}$ with $n \geq 3$, we define two encoders:

$$\varphi_H \colon F \longmapsto e_H = F, \quad \varphi_B \colon F \longmapsto \{(v_i, v_j) \mid 1 \leq i < j \leq n\}. \tag{30}$$

Let the encoded knowledge sets be random variables $Y_H = \varphi_H(X)$ and $Y_B = \varphi_B(X)$. Since φ_H is injective, the conditional entropy is zero:

$$H\left(X \mid Y_H\right) = 0, \quad \text{and hence} \quad I(X; Y_H) = H(X). \tag{31}$$

However, when $\mu(\{|F| \geq 3\}) > 0$, the encoder φ_B becomes non-injective. There exist $x_1 \neq x_2$ such that $Y_B(x_1) = Y_B(x_2)$, leading to:

$$H\left(X \mid Y_B\right) = \mathbb{E}_{Y_B}\left[-\sum_{x \in \varphi_B^{-1}(Y_B)} \mu(x \mid Y_B) \log_2 \mu(x \mid Y_B)\right] > 0, \tag{32}$$

$$I(X; Y_B) = H(X) - H(X \mid Y_B) < H(X). \tag{33}$$

To study encoding efficiency, consider encoding $Y_\star$ ($\star \in \{H, B\}$) using an optimal prefix code. Let the expected code length be $\mathcal{L}_\star = \mathbb{E}[\ell(Y_\star)]$. According to Shannon's source coding theorem:

$$\mathcal{L}_\star \in [H(Y_\star), H(Y_\star) + 1). \tag{34}$$

Define the information efficiency density (information per bit) as:

$$\eta_\star = \frac{I(X; Y_\star)}{\mathcal{L}_\star}. \tag{35}$$

This metric quantifies the amount of effective information transmitted per bit. Since $I(X; Y_H) = H(X)$ while $I(X; Y_B) < H(X)$, and $H(Y_B) \geq H(Y_H)$ (as the pairwise representation introduces a larger outcome space), we have:

$$\eta_H - \eta_B = \frac{H(X)}{\mathcal{L}_H} - \frac{H(X) - \delta}{\mathcal{L}_B}, \quad \delta > 0, \quad \mathcal{L}_B - \mathcal{L}_H \geq 0, \tag{36}$$

which is strictly positive when $\delta > 0$. This shows that the hypergraph representation transmits more effective information per bit. Let the maximum retrievable context budget for a language model be L, and define the coverage function:

$$\mathcal{C}_\star(L) = \Pr\big(I(X; Y_\star) \leq L\big) = \mu\left(\{x \mid \eta_\star \cdot \ell(Y_\star(x)) \leq L\}\right), \tag{37}$$

$\mathcal{C}_\star(L)$ is a non-decreasing function of L and is differentiable almost everywhere. Given $\eta_H > \eta_B$, the chain rule yields:

$$\frac{d}{dL}\mathcal{C}_H(L) = \int_{\ell(Y_H) = L/\eta_H} \frac{\partial \mu}{\partial \ell} \cdot \frac{d\ell}{dL}\, d\sigma \geq \int_{\ell(Y_B) = L/\eta_B} \frac{\partial \mu}{\partial \ell} \cdot \frac{d\ell}{dL}\, d\sigma = \frac{d}{dL}\mathcal{C}_B(L), \tag{38}$$

which implies $\mathcal{C}_H(L) \geq \mathcal{C}_B(L)$ with strict inequality on intervals where $\mu(\{|F| \geq 3\}) > 0$. Let generation quality E (e.g., G-E score) be a differentiable function $E = g(I(X; Y_\star), \mathcal{N}_\star)$, where $\mathcal{N}_\star$ denotes the noise introduced by irrelevant or redundant edges, and satisfies:

$$\frac{\partial g}{\partial I} > 0, \quad \frac{\partial g}{\partial \mathcal{N}} < 0. \tag{39}$$

Here, noise $\mathcal{N}_\star$ is defined as the set of edges retrieved under budget L that are irrelevant to the ground-truth $X^\dagger$. Under the same bit budget, higher η_H implies fewer edges per bit, and thus:

$$\mathbb{E}[\mathcal{N}_H] \leq \mathbb{E}[\mathcal{N}_B]. \tag{40}$$

Treating L as an independent variable, we apply the chain rule:

$$\frac{d}{dL}\big[E_H(L) - E_B(L)\big] = \frac{\partial g}{\partial I}(\theta_L)\left[\frac{d}{dL}I(X; Y_H) - \frac{d}{dL}I(X; Y_B)\right] + \frac{\partial g}{\partial \mathcal{N}}(\theta_L)\left[\frac{d}{dL}\mathcal{N}_H - \frac{d}{dL}\mathcal{N}_B\right], \tag{41}$$

where θ_L is an intermediate state between the two systems. From Equation 38 and Equation 40, we know: (1) The first term is strictly positive if high-arity facts exist; (2) The second term is always non-positive, as higher information density leads to lower redundancy. Therefore, the total derivative is strictly positive. Integrating over $[0, L]$, we obtain:

$$E_H(L) - E_B(L) = \int_0^L \frac{d}{d\beta}\left[E_H(\beta) - E_B(\beta)\right] d\beta > 0, \quad \text{unless } \mu(\{|F| \geq 3\}) = 0. \tag{42}$$

Equation 42 formally proves that if there exists at least one fact with arity $n \geq 3$ in the knowledge base, then under any fixed retrieval budget L, the generation quality under hypergraph encoding strictly exceeds that of the binary encoding. In the degenerate case where all facts are binary, both encodings reduce to the same mapping, and the conclusion naturally becomes an equality. □

C HyperGraphRAG Algorithm Detail

Hypergraph Construction. To provide a clear overview of our system pipeline, we present the detailed procedures of HyperGraphRAG in the form of pseudocode. As shown in Algorithm 1, we first construct a knowledge hypergraph from raw documents via LLM-based extraction of n-ary relational facts. Each extracted fact forms a hyperedge connecting multiple entities, and the resulting hypergraph is stored in a bipartite structure for efficient indexing and retrieval. We further compute dense embeddings for all entities and hyperedges to support semantic retrieval.

Algorithm 1 Hypergraph Construction

Require: Document collection $\mathcal{D}$
Ensure: Knowledge hypergraph $\mathcal{G}_H = (V, E_H)$
Initialize entity set $V \leftarrow \emptyset$, hyperedge set $E_H \leftarrow \emptyset$
for each document $d \in \mathcal{D}$ **do**
  Extract n-ary facts: $\mathcal{F}_d = \{(e_i, V_{e_i})\}_{i=1}^{k} \sim \pi(d)$
  $V \leftarrow V \cup \bigcup_{i=1}^{k} V_{e_i}$
  $E_H \leftarrow E_H \cup \{e_i\}_{i=1}^{k}$
end for
Store (V, E_H) as bipartite graph $\mathcal{G}_B = \Phi(\mathcal{G}_H)$
Compute embeddings: $E_V = \{f(v) \mid v \in V\}$, $E_{E_H} = \{f(e) \mid e \in E_H\}$
return $\mathcal{G}_H = (V, E_H)$

Complexity Analysis. Given a corpus of D documents, assume each document contains at most r relational facts, and each fact involves up to n entities. The LLM-based extraction step has complexity $\mathcal{O}(D)$ under the assumption of constant-time per document prompt. Constructing the hypergraph involves inserting up to $\mathcal{O}(D \cdot r)$ hyperedges and $\mathcal{O}(D \cdot r \cdot n)$ entities (with deduplication), resulting in a total construction time of $\mathcal{O}(D \cdot r \cdot n)$. Embedding all nodes and hyperedges requires $\mathcal{O}(|V| + |E_H|)$ calls to the encoder, typically parallelizable.

Hypergraph Retrieval and Generation. Once the hypergraph is constructed, the generation process begins with a query input, as detailed in Algorithm 2. We first extract relevant entities from the query and perform top-k similarity search to retrieve both entity and hyperedge candidates. We then perform bidirectional neighborhood expansion over the hypergraph to assemble a knowledge set, which may optionally be combined with chunk-level retrieval. Finally, we format the retrieved knowledge into a prompt and generate an answer using a large language model. This modular pipeline ensures efficient, expressive, and accurate generation grounded in structured knowledge.

Algorithm 2 Hypergraph Retrieval and Generation

Require: Query q, knowledge hypergraph $\mathcal{G}_H = (V, E_H)$
Ensure: Final answer y^*
Extract query entities: $V_q \sim \pi(q)$
Retrieve top-k entities: $V_r \leftarrow \textsc{TopKSim}(V_q, E_V)$
Retrieve top-k hyperedges: $E_r \leftarrow \textsc{TopKSim}(q, E_{E_H})$
Expand neighbors: $F_V^* = \bigcup_{v \in V_r} \text{Nbr}(v), \quad F_E^* = \bigcup_{e \in E_r} \text{Nbr}(e)$
Assemble retrieved knowledge: $K_H = F_V^* \cup F_E^*$
Retrieve additional chunks (optional): $K_{\text{chunk}} = \textsc{RetrieveChunks}(q)$
Combine all knowledge: $K^* = K_H \cup K_{\text{chunk}}$
Generate answer: $y^* \sim \pi(q, K^*)$
return y^*

Complexity Analysis. Given a query q, entity and hyperedge retrieval involves computing top-k similarity against all entity and hyperedge embeddings. With $|V|$ entities and $|E_H|$ hyperedges, this results in $\mathcal{O}(|V| + |E_H|)$ embedding comparisons. The neighborhood expansion step is bounded by the degree of retrieved nodes, i.e., $\mathcal{O}(k \cdot d)$ where d is average node degree. Finally, generation is treated as a black-box LLM inference, typically $\mathcal{O}(L)$ where L is the prompt length.

In summary, HyperGraphRAG achieves efficient inference with precomputed indices, and its overall retrieval-generation time is dominated by vector similarity lookup and prompt generation, both of which scale linearly with hypergraph size and are highly parallelizable in practice.

D Dataset Construction

D.1 Knowledge Domains

The dataset used for HyperGraphRAG evaluation covers five domains, with data sourced as follows:

Medicine: Derived from the latest international hypertension guidelines [16], covering medical diagnosis, treatment plans, and clinical indicators. **Agriculture:** Extracted from the UltraDomain dataset [19], including knowledge on agricultural production, crop management, and pest control. **Computer Science (CS):** Sourced from the UltraDomain dataset, encompassing computer architecture, algorithms, and machine learning. **Legal:** Based on the UltraDomain dataset, covering legal provisions, judicial precedents, and regulatory interpretations. **Mix:** A combination of multiple domains to assess the model's generalization ability across interdisciplinary tasks.

D.2 Question Sampling Strategies

To construct a fair and comprehensive evaluation benchmark, we design a uniform sampling strategy for both binary and n-ary sources. Specifically, for each domain, we sample a total of **512 questions**, consisting of:

Binary Source (256 samples): 128 facts are selected via 1-hop traversal, 64 facts via 2-hop traversal, 64 facts via 3-hop traversal. These facts are composed of binary relations (i.e., pairwise entity connections) and are used to build the binary knowledge source.

N-ary Source (256 samples): 128 facts are sampled via 1-hop traversal, 64 facts via 2-hop traversal, 64 facts via 3-hop traversal. These facts involve multi-entity ($n \geq 3$) relational structures and are used to construct the n-ary knowledge source.

For each sampled fact, we prompt GPT to generate a corresponding question and its golden answer. All generated question-answer pairs are manually verified to ensure factual accuracy, relevance, and diversity. This process is repeated independently for every domain to ensure consistent scale and structure across evaluation sets. All datasets undergo manual review to ensure the accuracy of annotated answers and the fairness of model evaluation.

E Evaluation Details

Unified Generation Prompt. To ensure a fair comparison across all baselines, we adopt a unified generation prompt for all methods, as shown in Figure 10. Specifically, we insert the knowledge retrieved by each method into a fixed prompt template that guides the model to first perform reasoning within a `<think>` block and then provide the final answer within an `<answer>` block, preserving benefits of zero-shot CoT reasoning while maintaining consistency across different retrieval strategies.

```
---Role---

You are a helpful assistant responding to questions based on given knowledge.

---Knowledge---

{d['knowledge']}

---Goal---

Answer the given question.
You must first conduct reasoning inside <think>...</think>.
When you have the final answer, you can output the answer inside <answer>...</answer>.

Output format for answer:
<think>
...
</think>
<answer>
...
</answer>

---Question---

{d['question']}
```

Figure 10: The unified prompt for generation p_{gen} in Equation 12.

We evaluate model performance using three complementary metrics that assess different aspects of retrieval-augmented generation: factual alignment, retrieval quality, and generation fluency.

(i) F1 Score. Following FlashRAG [9], we compute the word-level F1 score between each generated answer and its ground-truth reference, and then average over all questions. This metric captures reflects factual alignment with the expected answer.

$$\text{F1} = \frac{1}{N}\sum_{i=1}^{N}\frac{2 \cdot P_i \cdot R_i}{P_i + R_i}, \quad P_i = \frac{|\text{Pred}_i \cap \text{GT}_i|}{|\text{Pred}_i|}, \quad R_i = \frac{|\text{Pred}_i \cap \text{GT}_i|}{|\text{GT}_i|} \tag{43}$$

where Pred_i and GT_i denote the set of words in the predicted and ground-truth answers for the i-th question, and N is the total number of evaluated questions.

(ii) Retrieval Similarity (R-S). Inspired by RAGAS [3], R-S quantifies the semantic similarity between the retrieved knowledge and the ground-truth knowledge used to construct the question. For each question, we concatenate all retrieved knowledge into a single string k_{retr} and all golden knowledge into k_{gold}, then compute the cosine similarity between their embeddings. The final R-S score is the average similarity across the dataset:

$$\text{R-S} = \frac{1}{N}\sum_{i=1}^{N}\cos\left(f(k_{\text{retr}}^{(i)}),\ f(k_{\text{gold}}^{(i)})\right) \tag{44}$$

where $f(\cdot)$ is the embedding function (e.g., SimCSE), and N is the total number of questions.

(iii) Generation Evaluation (G-E). Adapted from HelloBench [20], G-E uses GPT-4o-mini as an LLM judge to evaluate generation quality along seven dimensions: *Correctness*, *Relevance*, *Factuality*, *Comprehensiveness*, *Knowledgeability*, *Logical Coherence*, and *Diversity*. For each question, we compute the average of the seven dimension scores, then combine it with the question's F1 score by taking their mean. The final G-E score is obtained by averaging this combined score:

$$\text{G-E} = \frac{1}{N}\sum_{i=1}^{N}\text{mean}\left(\frac{1}{7}\sum_{d=1}^{7}s_{i,d}, ; F1_i\right) \tag{45}$$

where $s_{i,d}$ denotes the score for dimension d on question i, $F1_i$ is the word-level F1 score for the i-th question, and N is the total number of evaluated questions. This formulation encourages alignment between LLM-judged quality and factual correctness.

G-E Prompt. Figure 11 and Figure 12 show our generation evaluation prompts. Figure 11 presents the unified prompt used to score each dimension on a 0–10 scale, while Figure 12 provides the detailed scoring rubric for all seven dimensions, ensuring consistency and fairness across evaluations.

```
---Role---

You are a helpful assistant evaluating the **{title}** of a generated response.

---Question---

{question}

---Golden Answers---

{str(answers)}

---Evaluation Goal---

Evaluate **{goal}** using a **0–10 integer scale**.

{rubric}

Output format:
<score>
your_score_here (an integer from 0 to 10)
</score>
<explanation>
Explain why you gave this score.
</explanation>

---Generation to be Evaluated---

{generation}
```

Figure 11: Prompt for G-E.

```
{
        "comprehensiveness": (
            "comprehensiveness",
            "whether the thinking considers all important aspects and is thorough",
            """Scoring Guide (0–10):
– 10: Extremely thorough, covering all relevant angles and considerations with depth.
– 8–9: Covers most key aspects clearly and thoughtfully; only minor omissions.
– 6–7: Covers some important aspects, but lacks depth or overlooks notable areas.
– 4–5: Touches on a few relevant points, but overall lacks substance or completeness.
– 1–3: Sparse or shallow treatment of the topic; misses most key aspects.
– 0: No comprehensiveness at all; completely superficial or irrelevant."""
        ),
        "knowledgeability": (
            "knowledgeability",
            "whether the thinking is rich in insightful, domain-relevant knowledge",
            """Scoring Guide (0–10):
– 10: Demonstrates exceptional depth and insight with strong domain-specific knowledge.
– 8–9: Shows clear domain knowledge with good insight; mostly accurate and relevant.
– 6–7: Displays some understanding, but lacks depth or has notable gaps.
– 4–5: Limited knowledge shown; understanding is basic or somewhat flawed.
– 1–3: Poor grasp of relevant knowledge; superficial or mostly incorrect.
– 0: No evidence of meaningful knowledge."""
        ),
        "correctness": (
            "correctness",
            "whether the reasoning and answer are logically and factually correct",
            """Scoring Guide (0–10):
– 10: Fully accurate and logically sound; no flaws in reasoning or facts.
– 8–9: Mostly correct with minor inaccuracies or small logical gaps.
– 6–7: Partially correct; some key flaws or inconsistencies present.
– 4–5: Noticeable incorrect reasoning or factual errors throughout.
– 1–3: Largely incorrect, misleading, or illogical.
– 0: Entirely wrong or nonsensical."""
        ),
        "relevance": (
            "relevance",
            "whether the reasoning and answer are highly relevant and helpful to the question",
            """Scoring Guide (0–10):
– 10: Fully focused on the question; highly relevant and helpful.
– 8–9: Mostly on point; minor digressions but overall useful.
– 6–7: Generally relevant, but includes distractions or less helpful parts.
– 4–5: Limited relevance; much of the response is off-topic or unhelpful.
– 1–3: Barely related to the question or largely unhelpful.
– 0: Entirely irrelevant."""
        ),
        "diversity": (
            "diversity",
            "whether the reasoning is thought-provoking, offering varied or novel perspectives",
            """Scoring Guide (0–10):
– 10: Exceptionally rich and original; demonstrates multiple fresh and thought-provoking ideas.
– 8–9: Contains a few novel angles or interesting perspectives.
– 6–7: Some variety, but generally safe or conventional.
– 4–5: Mostly standard thinking; minimal diversity.
– 1–3: Very predictable or monotonous.
– 0: No diversity or originality at all."""
        ),
        "logical_coherence": (
            "logical coherence",
            "whether the reasoning is internally consistent, clear, and well-structured",
            """Scoring Guide (0–10):
– 10: Highly logical, clear, and easy to follow throughout.
– 8–9: Well-structured with minor lapses in flow or clarity.
– 6–7: Some structure and logic, but a few confusing or weakly connected parts.
– 4–5: Often disorganized or unclear; logic is hard to follow.
– 1–3: Poorly structured and incoherent.
– 0: Entirely illogical or unreadable."""
        ),
        "factuality": (
            "factuality",
            "whether the reasoning and answer are based on accurate and verifiable facts",
            """Scoring Guide (0–10):
– 10: All facts are accurate and verifiable.
– 8–9: Mostly accurate; only minor factual issues.
– 6–7: Contains some factual inaccuracies or unverified claims.
– 4–5: Several significant factual errors.
– 1–3: Mostly false or misleading.
– 0: Completely fabricated or factually wrong throughout."""
        )
    }
```

Figure 12: Seven Evaluation Dimensions for Generation Quality.

F Baseline Details

We compare HyperGraphRAG against six representative baselines that cover retrieval-free, chunk-based, and binary graph-based RAG paradigms:

NaiveGeneration is a retrieval-free baseline where the LLM directly answers questions without any external knowledge input. This serves as a lower bound for retrieval-augmented generation.

StandardRAG follows the original RAG design, retrieving top-k text chunks from a flat corpus using dense vector similarity and feeding them into the generator.

GraphRAG [2] constructs a binary relational graph and retrieves community-level summaries linked to query-relevant entities. It uses entity overlap to detect relevant subgraphs.

LightRAG [7] enhances retrieval efficiency by using graph indexing and lightweight entity-relation matching over the binary graph, and then combines results with chunk-level retrieval.

PathRAG [1] improves graph-based retrieval by selecting paths through the graph that are semantically relevant to the query, using path pruning strategies to reduce redundancy.

HippoRAG2 [8] introduces a high-precision multi-hop retrieval mechanism over binary graphs, using Personalized PageRank to select passage-level nodes connected to the query.

To ensure fairness, all baselines use the same generation prompt (Figure 10) and are evaluated under identical conditions, with retrieved knowledge constrained to equivalent token budgets. Each method's construction and retrieval mechanism is summarized in Table 1.

G Hyperparameter Settings

For all methods, we adopt a unified set of hyperparameters for all models across both the main evaluation in Table 2 and the time/cost experiments in Table 3 to ensure fair and consistent comparison. For chunk-based methods (e.g., StandardRAG), we retrieve the top-5 chunks using dense similarity. For graph-based methods, including GraphRAG, LightRAG, PathRAG, and HippoRAG2, we retrieve the top-60 relevant elements according to their respective retrieval strategies. HyperGraphRAG performs dual top-60 retrieval over entities and hyperedges, followed by neighborhood expansion. All methods are run using 16 parallel cores and the same generation model (GPT-4o-mini) with temperature 1.0 and a maximum generation length of 32k tokens. Table 4 summarizes the detailed hyperparameter configurations used throughout our experiments.

Table 4: Hyperparameter settings for all methods.

Method	Retrieval Type	Top-k Units	Parallel Cores	Generation Model
NaiveGeneration	None	–	16	GPT-4o-mini
StandardRAG	Chunk	5 chunks	16	GPT-4o-mini
GraphRAG	Entity $\rightarrow$ Community	60	16	GPT-4o-mini
LightRAG	Entity + Relation	60	16	GPT-4o-mini
PathRAG	Graph Path	60	16	GPT-4o-mini
HippoRAG2	PageRank over Graph	60	16	GPT-4o-mini
HyperGraphRAG (ours)	Entity + Hyperedge	60	16	GPT-4o-mini

H Case Study

To better understand how different methods perform in complex, knowledge-intensive scenarios, we present a case study on the question: *"What type of renal denervation has been shown to demonstrate BP-lowering efficacy over 24 hours in several randomized, sham-controlled trials, particularly for resistant hypertension?"* As shown in Table 5, this question requires synthesizing clinical evidence and distinguishing between treatment modalities (e.g., ultrasound vs. catheter-based techniques).

HyperGraphRAG provides the most comprehensive and accurate response, clearly identifying ultrasound renal denervation as the answer, and justifying it with domain-relevant reasoning and structured knowledge. It achieves the highest scores across all dimensions, including **Correctness (10.00)**, **Factuality (9.50)**, and **Knowledgeability (10.00)**, resulting in an overall G-E score of **92.86**.

Table 5: Case study comparing RAG methods on a complex medical question. The reported metrics reflect the actual scores for this specific example. HyperGraphRAG generates the most accurate and comprehensive answer, achieving the highest scores across all evaluation metrics, while other baselines exhibit factual omissions or ambiguity.

Query		What type of renal denervation has been shown to demonstrate BP-lowering efficacy over 24 hours in several randomized, sham-controlled trials, particularly for resistant hypertension?									
Golden Answer		"ULTRASOUND RENAL DENERVATION"									
		F1	**R-S**	**G-E**	**Comprehen -siveness**	**Knowledge -ability**	**Correctness**	**Relevance**	**Diversity**	**Logical Coherence**	**Factuality**
NaiveGeneration	*Generation*	<think>\nRenal denervation is a procedure used to treat resistant hypertension by disrupting the nerves near the renal arteries, which are believed to contribute to the regulation of blood pressure. Several techniques can be utilized for renal denervation, including radiofrequency, ultrasound, and chemical methods. In the context of studies, the method that has been heavily studied and demonstrated BP-lowering efficacy over a 24-hour period is typically the radiofrequency renal denervation. This approach has been rigorously tested in various randomized, sham-controlled trials and has shown significant persistence of blood pressure reduction, particularly in patients with resistant hypertension. Therefore, the radiofrequency method stands out among the available techniques.\n</think>\n<answer>\n **Radiofrequency renal denervation has been shown to demonstrate BP-lowering efficacy over 24 hours in several randomized, sham-controlled trials, particularly for resistant hypertension.**\n</answer>									
	Evaluation Score	16.00	0.00	48.00	43.00	48.00	53.00	58.00	33.00	48.00	53.00
StandardRAG	*Generation*	<think>\nThe text mentions that several devices have been investigated for treating resistant hypertension, but it specifically highlights catheter-based renal denervation as having the most evidence for BP-lowering efficacy. It notes that both radiofrequency and ultrasound renal denervation have been demonstrated to lower blood pressure over 24 hours in a broad spectrum of hypertension cases, including resistant hypertension. Thus, both radiofrequency and ultrasound renal denervation seem to be effective based on the trials referenced.\n</think>\n<answer>\n **Catheter-based renal denervation, particularly radiofrequency and ultrasound renal denervation, has been shown to demonstrate BP-lowering efficacy over 24 hours in several randomized, sham-controlled trials for resistant hypertension**.\n</answer>									
	Evaluation Score	20.00	65.61	51.43	50.00	50.00	55.00	55.00	40.00	55.00	55.00
GraphRAG	*Generation*	<think>\nThe question refers to renal denervation methods that have been tested in randomized, sham-controlled trials and have demonstrated BP-lowering efficacy. Renal denervation is a procedure designed to treat resistant hypertension by disrupting the nerve signals to the kidneys, which can lead to long-term reductions in blood pressure. \n\nSeveral clinical trials, such as the SYMPLICITY HTN-3 and those involving catheter-based ultrasound techniques, have evaluated the effectiveness of renal denervation for treating resistant hypertension. More specifically, some trials have shown a sustained blood pressure-lowering effect over a 24-hour period, which is essential for patients suffering from resistant hypertension.\n\nAfter reviewing the context provided, catheter-based renal denervation appears to be the specific type that has shown significant efficacy over 24 hours in these trials, particularly targeting patients who do not respond well to traditional antihypertensive medications.\n</think>\n<answer>\n **Catheter-based renal denervation has been shown to demonstrate BP-lowering efficacy over 24 hours in several randomized, sham-controlled trials, particularly for resistant hypertension.**\n</answer>									
	Evaluation Score	16.00	57.53	48.00	43.00	48.00	48.00	58.00	38.00	48.00	53.00
LightRAG	*Generation*	<think>\nRenal denervation is a procedure aimed at reducing high blood pressure (BP) by disrupting renal nerve activity. The document discusses various types of renal denervation, particularly highlighting catheter-based methods. These methods use radiofrequency and ultrasound technologies to denervate renal arteries. The trials mentioned demonstrate BP-lowering efficacy over a 24-hour period for these approaches. Given the context, the specific types of renal denervation shown to effectively lower BP, especially in resistant hypertension, are radiofrequency and ultrasound-based techniques.\n</think>\n<answer>\n **Catheter-based renal denervation using radiofrequency and ultrasound has been shown to demonstrate BP-lowering efficacy over 24 hours in several randomized, sham-controlled trials, particularly for resistant hypertension.**\n</answer>									
	Evaluation Score	20.69	35.29	53.20	50.34	55.34	60.34	55.34	40.34	55.34	55.34
PathRAG	*Generation*	<think>\nThe question is about renal denervation specifically focusing on its type that shows efficacy in blood pressure (BP) lowering. Renal denervation is a procedure that involves modifying nerve signals in the kidney to help control hypertension. The trials mentioned specifically pertain to randomized, sham-controlled studies which evaluate the effectiveness of different techniques or devices used during the procedure. The answer is likely \"catheter-based renal denervation,\" as it is a common method used in such trials, and several studies have reported its effectiveness in managing resistant hypertension.\n</think>\n<answer>\n **Catheter-based renal denervation has been shown to demonstrate BP-lowering efficacy over 24 hours in several randomized, sham-controlled trials, particularly for resistant hypertension.**\n</answer>									
	Evaluation Score	16.00	60.44	46.57	48.00	43.00	48.00	53.00	33.00	48.00	53.00
HippoRAG2	*Generation*	<think>\nThe provided text mentions that catheter-based renal denervation has been studied significantly in treating resistant hypertension, with a focus on two technologies: radiofrequency and ultrasound renal denervation. The sentence indicates that there is available evidence from several randomized, sham-controlled trials which confirmed BP-lowering efficacy for these methods over a 24-hour period. Therefore, the answer should highlight that catheter-based renal denervation, particularly with radiofrequency and ultrasound techniques, demonstrated this efficacy.\n</think>\n<answer>\n **Catheter-based renal denervation, particularly using radiofrequency and ultrasound methods, has been shown to demonstrate BP-lowering efficacy over 24 hours in several randomized, sham-controlled trials, particularly for resistant hypertension.**\n</answer>									
	Evaluation Score	19.35	66.15	53.25	54.68	54.68	54.68	59.68	39.68	54.68	54.68
HyperGraphRAG (ours)	*Generation*	<think>\nThe question asks about a type of renal denervation that has demonstrated BP-lowering efficacy over 24 hours specifically in the context of several randomized, sham-controlled trials related to resistant hypertension. Based on the provided knowledge, both ultrasound renal denervation and radiofrequency renal denervation have been mentioned to show BP-lowering efficacy. The hyperedges also state that several sham-controlled trials have been published demonstrating this efficacy for these two methods. Therefore, it's essential to identify one type that is prominently recognized for this efficacy in published studies. Both types could potentially qualify, but the information about ultrasound renal denervation being mentioned in conjunction with randomized trials makes it a likely candidate. \nThus, ultrasound renal denervation is a key method that fits the criteria given in the question.\n</think>\n<answer>\n **Ultrasound renal denervation**\n</answer>									
	Evaluation Score	**100.00**	**70.87**	**92.86**	**90.00**	**95.00**	**100.00**	**100.00**	**75.00**	**95.00**	**95.00**

In contrast, baseline methods like **NaiveGeneration** and **StandardRAG** produce factually plausible but imprecise or overly generic answers (e.g., mentioning radiofrequency denervation instead), leading to lower scores, particularly in **Correctness** and **Factuality**. Graph-based baselines such as **GraphRAG**, **LightRAG**, and **PathRAG** improve coherence but still fall short in domain specificity. Even the best-performing baseline, **HippoRAG2**, fails to precisely isolate the correct answer, with reduced clarity and comprehensiveness compared to HyperGraphRAG. This case highlights the strength of HyperGraphRAG in integrating multi-entity clinical knowledge through hyperedges, enabling more precise, interpretable, and fact-grounded responses in real-world expert-level tasks.

I Limitations and Future Work

I.1 Multimodal HyperGraphRAG

While our current framework focuses on textual knowledge, real-world information often spans multiple modalities, including images, tables, and structured metadata. A promising direction is to extend HyperGraphRAG to the multimodal setting by constructing hypergraphs that integrate both textual and non-textual entities (e.g., medical images, diagrams, or structured EHR fields). This would allow the model to reason over complex multimodal relationships, such as "image + report + diagnosis" or "chart + claim + textual guideline," and enable broader deployment in domains like medicine, science, and law. Future work will explore how to encode, align, and retrieve multimodal hyperedges effectively, while maintaining the structural advantages of hypergraph representations.

I.2 HyperGraphRAG with Reinforcement Learning

Another important extension lies in incorporating reinforcement learning (RL) to guide both retrieval and generation. In our current setup, retrieval is driven by fixed similarity metrics, which may not fully capture downstream utility. By formulating hypergraph-based retrieval as a sequential decision-making process, we can apply RL to optimize entity and hyperedge selection policies based on long-term generation rewards—such as factuality, coherence, or user feedback. This would allow HyperGraphRAG to dynamically adapt retrieval strategies to different tasks and domains, leading to more efficient and effective use of structured knowledge.

I.3 Federated HyperGraphRAG for Privacy-Preserving Retrieval

Many real-world applications involve sensitive or distributed data that cannot be centralized due to privacy constraints. To address this, we propose to integrate HyperGraphRAG with federated learning techniques, allowing hypergraph construction, retrieval, and generation to occur across decentralized data silos. Each local client can construct its own partial hypergraph and share only anonymized or encrypted embeddings, preserving privacy while contributing to global retrieval. This federated HyperGraphRAG would be particularly beneficial in domains like healthcare or finance, where data sharing is restricted but collective knowledge is crucial for robust decision-making.

I.4 Toward a Foundation Model for HyperGraph-based Retrieval

As large language models continue to scale and generalize across domains, a natural extension is to explore the development of a foundation model for HyperGraphRAG. Rather than constructing and retrieving from hypergraphs on a per-task or per-domain basis, we envision a pretrained hypergraph reasoning model that jointly learns representations of entities, relations, and higher-order hyperedges across diverse corpora. This model would encode structural, semantic, and contextual signals in a unified way, and could be adapted to new domains via lightweight fine-tuning. Such a foundation model could also enable transfer learning across knowledge-intensive tasks, reducing the need for domain-specific engineering and improving the sample efficiency of retrieval and generation pipelines. Building this requires scalable hypergraph pretraining objectives, efficient storage formats, and robust generalization strategies, which we leave as future work.

I.5 Scaling to Harder Tasks and Broader Applications

Finally, we plan to evaluate HyperGraphRAG on more challenging tasks and diverse real-world applications. This includes settings that require deeper compositional reasoning, such as multi-hop question answering, legal argument generation, or complex scientific synthesis. Additionally, we aim to apply HyperGraphRAG to broader domains beyond the current benchmarks, including policy analysis, education, and open-domain dialogue. These tasks will test the framework's ability to generalize across domains, handle larger and more diverse knowledge bases, and maintain high-quality generation under increasingly demanding conditions.

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 abstract and introduction accurately reflect the paper's core contributions in hypergraph-based knowledge representation, retrieval, and generation.

   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 present limitations and future work in Appendix I.

   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: [Yes]

Justification: We present complete proofs in Appendix B.

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 provide code and data in an anonymous GitHub link.

   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 provide sufficient instructions to faithfully reproduce the main experimental results.

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 provide all implementation details.

   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: [Yes]

   Justification: We unify the generated prompts to ensure fairness.

   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: We provide compute resources in the implementation details.

   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 fully complies with the NeurIPS Code of Ethics in all respects.

   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: [Yes]

    Justification: We discuss both the potential positive and negative societal impacts of the work performed.

    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 poses no such risks.

    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: [NA]

    Justification: The paper does not use existing assets.

    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: The paper does not release new assets.

    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: [Yes]

    Justification: We pay the human annotators.

    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: [NA]

Justification: The core method development in this research does not involve LLMs.

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.