VizRAG: Enhancing Retrieval-Augmented Generation with Hypergraph Visualization

Anonymous ACL submission

Abstract

Hypergraph-based Retrieval-Augmented Generation (RAG) systems surpass traditional graph-based approaches by organizing complex n-ary atomic facts among entities, rather than relying solely on binary relationships. Despite the advancements in multimodal large language models (MLLMs) with enhanced visual capabilities, current hypergraph-based RAG frameworks predominantly restrict knowledge retrieval and reconstruction to a unimodal, text-centric paradigm. This limitation prevents them from fully leveraging the powerful visual perception capabilities of modern MLLMs. To address this gap, we systematically explore the integration of hypergraph awareness in RAG systems through visual cues. By incorporating visual representations of hypergraphs into the RAG pipeline, we introduce VizRAG, the first RAG system to support visual hypergraph structure awareness. Experimental results demonstrate that VizRAG significantly outperforms strong baselines, validating the promising potential of hypergraph visualization as a novel approach for RAG systems.

1 Introduction

Retrieval-Augmented Generation (RAG) has become a key framework for improving LLM performance in knowledge-intensive tasks like question answering, document analysis, and intelligent assistance (Lewis et al., 2020; Gao et al., 2023; Li et al., 2024). By leveraging external knowledge sources, RAG reduces hallucinations in LLMs, enhancing both the reliability and quality of generated outputs. Moreover, RAG enables seamless integration of private or domain-specific knowledge bases, allowing LLMs to adapt effectively to specialized contexts.

However, conventional RAG systems predominantly utilize chunk-level retrieval based on vector similarity (Asai et al., 2023; Yang et al., 2024). While this approach achieves reasonable effectiveness, it often overlooks inter-chunk relationships and hierarchical semantic structures. As a result, retrieved content may lack coherence and exhibit weak connectivity, limiting the model's ability to structurally interpret knowledge, a critical factor for producing context-rich and coherent responses.

To address these shortcomings, recent studies have explored graph-based structures within RAG frameworks. For instance, GraphRAG (Edge et al., 2024) and LightRAG (Guo et al., 2024) utilize knowledge graphs to enhance entity-level indexing and retrieval by explicitly modeling semantic relationships. Hyper-RAG (Feng et al., 2025a) and Cog-RAG (Hu et al., 2026) further extend this paradigm by using hypergraphs to capture higher-order relationships among multiple entities, offering greater expressiveness than traditional graphs. However, the approaches remain predominantly text-centric, restricting knowledge retrieval and reasoning to unimodal paradigms and failing to leverage advanced visual perception capabilities of modern multimodal large language models (MLLMs).

In contrast, humans often rely on visual intuition to perceive and reason about complex structures, such as loops, clusters, or sets, within graphical representations. This cognitive advantage suggests that integrating visualized hypergraphs into the RAG process could complement text-based retrieval with multimodal benefits, such as intuitive reasoning shortcuts and enriched structural understanding. Inspired by this insight, we systematically analyze the **advantages (Sec. 2.1)** and **feasibility (Sec. 2.2)** of using hypergraph visualization to enhance RAGs, to reveal the untapped potential of hypergraph visualization as a **novel direction**.

Nevertheless, incorporating hypergraph visualization into RAG systems poses unique challenges (Sec. 2.3), including **visual congestion** and **rendering bias**. Addressing these challenges is essential to fully realize the potential of visual hypergraphs in RAG systems. To this end, we propose **HyperViz** (Sec. 3), a toolkit that provides scientifically

grounded solutions to these challenges, making hypergraph visualization an effective and accessible plugin for RAG systems.

Building on **HyperViz**, we introduce **VizRAG** (Sec. 4), a novel RAG framework that integrates hypergraph visualization to enhance the retrieval and reasoning capabilities of MLLMs. By combining visual hypergraph awareness deeply into the retrieval strategies, **VizRAG** distinguishes itself from the prior works and significantly improves the coherence and quality of generated responses over strong baselines in extensive experiments.

Our contributions are summarized as follows:

- We introduce **hypergraph visualization** as a novel and promising direction for enhancing RAG systems, systematically analyzing its potential, feasibility, and challenges.
- We develop **HyperViz**, a plug-and-play, well-encapsulated toolkit that addresses key challenges in hypergraph visualization.
- We propose **VizRAG**, a visualization-driven RAG framework that incorporates hypergraph visual awareness into the entire RAG pipeline, enabling multimodal retrieval and reasoning.
- Extensive experiments show that **VizRAG** significantly outperforms strong RAG baselines on multiple knowledge-intensive tasks, highlighting the promising enhancement of hypergraph visualization for RAG.

2 Hypergraph Visualization in RAG

In this section, we clarify the situation of hypergraph visualization in RAG, including its 1) advantages, 2) feasibility, and 3) specific challenges.

2.1 Advantages

We first highlight why hypergraph visualization is a meaningful extension for hypergraph-based RAG.

1) Structural Discrimination. Text-only retrieval often over-relies on lexical overlap and shallow semantics. In hypergraph-based RAG, many failure cases arise from *structurally mismatched but lexically similar* evidence: candidate passages may share keywords with the query but encode different high-order relations among entities. Hypergraph visualization provides explicit spatial and grouping cues (e.g., overlap patterns, bridge entities, and cross-group links), helping MLLMs separate true relational matches from superficial textual matches. This enables finer-grained structural disambiguation before final answer generation.

Note 2.1. *Hypergraph visualization enables RAG systems to achieve finer-grained structural discrimination in hypergraphs.*

2) Multi-hop Reasoning Composition. As the reasoning hop increases, relevant evidence becomes more scattered across entities, hyperedges, and chunks, making path composition fragile in text-only settings. Hypergraph visualization offers an externalized structural workspace: it exposes candidate evidence chains as visible substructures, making it easier for MLLMs to identify plausible multi-hop paths and reject shortcuts that break relational constraints. This is particularly important for deep reasoning, where correctness depends on composing multiple structurally consistent steps rather than selecting a single highly similar passage.

Note 2.2. *Visual structural cues bring increasingly prominent benefits for complex deep multi-hop reasoning tasks.*

To empirically validate these advantages, we present experimental results in Sec. 5.3.

2.2 Feasibility

Apart from the benefits, we argue that hypergraph visualization is feasible in current RAG systems:

(1) Usable MLLM Hypergraph Perception: Recent off-the-shelf MLLMs already demonstrate strong visual parsing ability for graph-like structures, including patterns like region overlap and bridge-like connectors (as evidenced in Sec. 5.3). These capabilities are tightly aligned with the key signals required to understand the hypergraph knowledge structure in RAGs, such as identifying overlapping entities, analyzing multi-hop connections, and interpreting relational constraints encoded in hyperedges. Because current MLLMs can effectively parse hypergraph visualizations without fine-tuning, integrating visualization into RAG systems can be seamless.

(2) Lightweight System Cost: Although visualization introduces extra rendering and transfer overhead, this overhead is lightweight relative to end-to-end RAG latency, especially when retrieval and generation already dominate the runtime. A detailed efficiency comparison is provided in Sec. 5.3, where the extra cost of visualization is very small compared to retrieval time. Therefore, hypergraph visualization does not introduce a significant com-

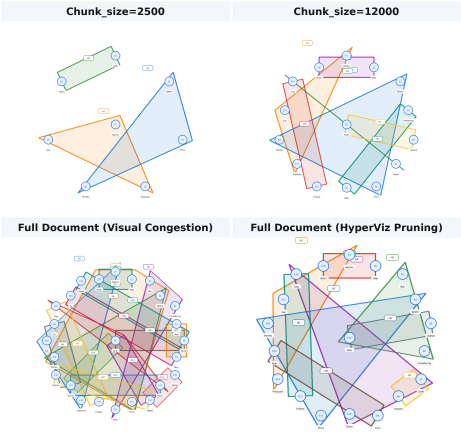

Figure 1: As the chunk size increases to even encompass the full document, more nodes and hyperedges are included in the visualization, leading to visual congestion that reduces structure clarity.

putational burden, making it suitable for deployment in existing RAG frameworks.

2.3 Specific Challenges

Though beneficial and feasible for integrating hypergraph visualization into RAG, it still faces two non-trivial challenges that hinder its effectiveness.

Visual Congestion. Knowledge-intensive corpora often contain a high density of entities and multi-entity hyperedges, leading to visual clutter in hypergraph visualization. As shown in Fig. 1, the bigger chunk size results in more node aggregation, edge crossing, and structural overlap. Although these features reveal critical structural cues, excessive clutter can obscure reasoning paths and degrade performance. Dense node clusters may cause localized visual overload, whereas sparse low-hop structures risk leaving large blank spaces. Balancing information density and clarity requires employing reasonable sampling strategies and adaptive element size adjustments. To address these challenges, effective structure filtering mechanisms and spatial layout optimization techniques are essential for preserving key information and ensuring readability across varying graph densities.

Rendering Bias. Another major challenge in hypergraph visualization is the uncontrollable rendering bias, where a single visualization emphasizes specific hypergraph properties, leading to partial or skewed structural understanding. Specifically, existing hypergraph layouts can be broadly categorized into two paradigms: i) *Set-based layouts*, which represent high-order relationships (i.e., hyperedges) as set elements, such as through region containment, to highlight entity grouping relations. ii) *Induced layouts*, which focus on direct belonging-ship between entities and their belonging groups by representing hyperedges as special nodes. In this approach, the nodes within each hyperedge are explicitly linked to the corresponding special node via pairwise edges. While both approaches provide valuable perspectives, they inherently emphasize different aspects of the hypergraph structure, which can lead to biased or incomplete interpretations depending on the chosen visualization method. To quantify this, we evaluate the Gemini-3 Flash model's structural perception on *overlap-core identification* and *bridge hyperedge detection*, where tasks are illustrated in Fig. 2(b), with 2,000 hypergraphs derived from the UltraDomain Legal doc/chunks (Qian et al., 2025). As shown in Fig. 2(c), significant discrepancies emerge between set-based and Induced visualizations. However, combining both layouts effectively mitigates unilateral perception deviations, with the strongest complementary effect observed when pairing set-based and induced views. This highlights the need for a unified paradigm in RAG systems to harness layout-induced biases for improved performance.

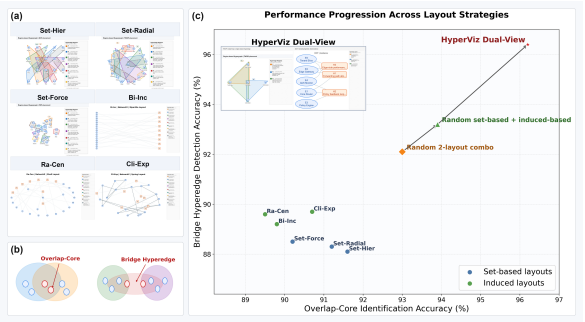

Figure 2: Rendering bias illustration. (a) Layout variants for the same hypergraph; (b) Structure perception tasks; (c) Performance gap caused by rendering bias.

3 HyperViz Toolkit

This section introduces **HyperViz**, a toolkit for hypergraph visualization in RAG systems. Taking the document hypergraph and text descriptions of entities/hyperedges, HyperViz generates visualization views that tackle visual congestion and rendering bias. By integrating structural and semantic considerations, it ensures document-based hypergraph visualization aligns with RAG requirements. To keep focus, we highlights its solutions to the aforementioned challenges in this section, with additional features are leaved in App. B.1.

3.1 Solution to Visual Congestion

To address visual congestion, HyperViz employs a multi-step optimization strategy:

(1) Weighted Space Allocation: Given a hypergraph $G = (V, E)$, where V represents entity nodes and E represents hyperedges, we assign distinct space weight coefficients w_v for nodes and w_e for hyperedges[1]. The total visual occupation cost is expressed as $C = \sum_{v \in V} A_v \cdot w_v + \sum_{e \in E} L_e \cdot w_e$, where A_v and L_e denote the actual visual areas and lengths occupied by entity nodes and hyperedges, respectively. To ensure clarity, a maximum allowable cost C_{max} is defined as $C_{\text{max}} = \gamma A$, where A is the area of the predefined canvas and γ controls the congestion threshold. Together, these formulations quantify the visual occupation by area and define an upper bound for maintaining clarity.

(2) Pruning Low-Importance Elements: When $C > C_{\text{max}}$, we rank all entities in G by an importance score Importance(e), defined as Importance$(e) = \alpha \cdot$ Centrality$(e) + \beta \cdot$ Relevance(e), where Centrality(e) is the degree centrality of the entity e in G, and Relevance(e) reflects a query-specific semantic alignment by the BM25 score between the text descriptions of e and Query Q, and α, β denote weighting coefficients. Based on this ranking, pruning is applied to the low-importance nodes, and they are eliminated from their associated hyperedges until $C \leq C_{\text{max}}$. This ensures that the resulting pruned hypergraph retains critical high-order relationships while significantly reducing visual clutter, improving interpretability and usability.

(3) Global Uniform Scaling: The filtered hypergraph is then scaled to fit the canvas dimensions, maximizing canvas utilization and eliminating marginal blank spaces to avoid space wasting.

In summary, the multi-step pipeline effectively mitigates visual congestion. As an illustration shown in Fig. 1, the pruned visualization generated by HyperViz *preserves clarity* while *retaining key structural and semantic information* in the document knowledge.

3.2 Solution to Rendering Bias

To fit diverse scenarios and increase the usability of Hyperviz, we categorize the addressing of rendering bias into two scenarios:

(1) User-driven Customization: To support tasks requiring specific rendering biases, HyperViz employs a layout-centric design encompassing both set-based and Induced paradigms, each offering three layout variants. The set-based layouts include i) Set-Hier, ii) Set-Radial, and iii) Set-Force, while the Induced layouts consist of i) Bi-Inc, ii) Ra-Cen, and iii) Cli-Exp. They are illustrated in Fig. 2(a), with detailed descriptions provided in App. B.3. Users can flexibly select a single layout or combine multiple layouts as needed.

(2) Automated Bias Mitigation: To address uncontrolled rendering biases, HyperViz incorporates a **probe-driven dual-view** mechanism as the default approach. Specifically, a small batch of k queries ($k = 15$ in our experiments, random from training sets) is used to evaluate all nine combinations of set-based and induced-graph layouts, inspired by their complementary strengths. The layout combination that best enhances hypergraph perception on *overlap-core identification* and *bridge hyperedges detection* (illustrated in Fig. 2(b))[2] is selected, and the two selected layouts are rendered side-by-side in a dual-view format within a single image, as demonstrated in Fig. 2(c) left-top. Results in Fig. 2(c) show that the dynamic probe-driven dual-view significantly improves hypergraph perception on both tasks compared to either single layouts or their random combinations. This finally benefits the RAG quality (Sec. 5.3).

3.3 Plug-and-Play Integration

HyperViz serves as a middleware that bridges knowledge hypergraphs and visualization workflows tailored to RAG scenarios. It seamlessly integrates with existing RAG pipelines in a plug-and-play manner, offering well-encapsulated interfaces to ensure ease of use for users.

4 VizRAG Framework

In this section, we introduce the VizRAG framework (illustrated in Fig. 3), where visualization plays a dual role in both indexing and retrieving.

4.1 Vision-awared Bottom-up Indexing

Given a corpus $\mathcal{D}$ (e.g., books, reports, or manuals), we segment it into a set of overlapping chunks using a fixed-length sliding window to preserve semantic continuity:

$$\mathcal{D} = \{D_1, D_2, \ldots, D_N\},$$

where D_i represents the i-th chunk, serving as the fundamental unit for subsequent processing.

[1] $w_v, w_e \in [1, 10]$ are grid-searched on large knowledge hypergraphs without data leakage under Gemini-3 Flash supervision via tasks in Fig.2. More hyperparameter details are in App. B.2

[2] Both tasks are chosen by their similarity to essential RAG requirements, such as identifying key entities within groups and uncovering critical relationships among entity clusters.

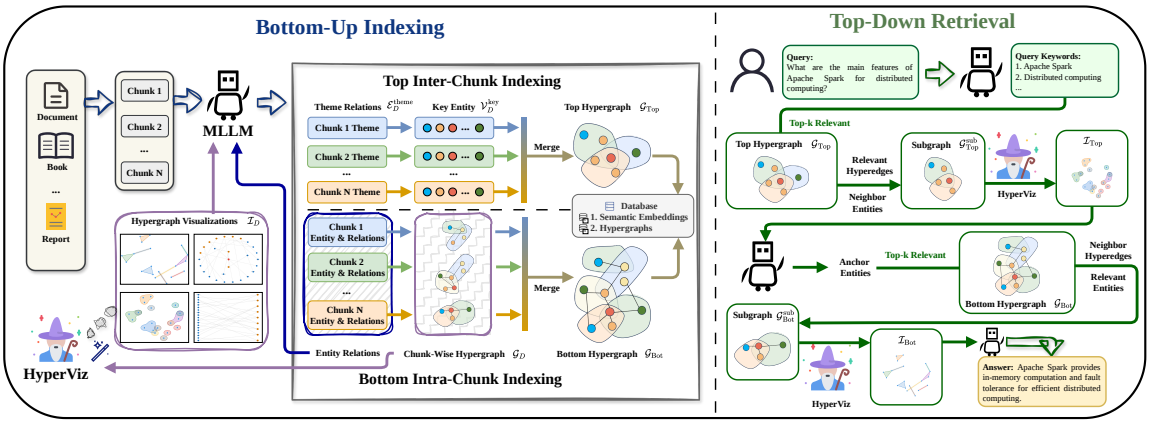

Figure 3: The overall framework of VizRAG.

We employ a two-layer hierarchical hypergraph indexing approach: (i) bottom-layer intra-chunk indexing, capturing fine-grained intra-chunk semantics. (ii) top-layer inter-chunk indexing, capturing global relationships across chunks.

4.1.1 Bottom Intra-Chunk Indexing

For each chunk D_i, we extract entities (e.g., persons, events, organizations) and their descriptions using LLMs. These entities form the vertex set $\mathcal{V}_{D_i}$ of the fine-grained entity hypergraph. Semantic relationships among the entities are used to construct hyperedges $\mathcal{E}_{D_i}$, capturing high-order associations such as co-occurrence in events or causal links:

$$\begin{cases} \mathcal{V}_{D_i} = \text{LLM}(\mathcal{P}_{\text{ext_entity}}(D_i)), \\ \mathcal{E}_{D_i} = \text{LLM}(\mathcal{P}_{\text{ext_relation}}(D_i, \mathcal{V}_{D_i})), \end{cases}$$

where $\mathcal{P}_{\text{ext_entity}}$ and $\mathcal{P}_{\text{ext_relation}}$ are prompts designed for entity and hyperedge extraction (detailed in Appendix G).

Once the hypergraph $\mathcal{G}_{D_i} = \{\mathcal{V}_{D_i}, \mathcal{E}_{D_i}\}$ is constructed, we generate its hypergraph visualization using HyperViz:

$$\mathcal{I}_{D_i} = \text{HyperViz}(\mathcal{G}_{D_i}).$$

4.1.2 Top Inter-Chunk Indexing

Using predefined prompts $\mathcal{P}_{\text{ext_theme}}$ and $\mathcal{P}_{\text{ext_key}}$ (in App. G), MLLMs are guided to perform semantic parsing on each chunk D_i, extracting its theme $\mathcal{E}_{D_i}^{\text{theme}}$ and a subset of key entities $\mathcal{V}_{D_i}^{\text{key}}$ relevant to the theme. Importantly, the intra-chunk knowledge structure visualized in $\mathcal{I}_{D_i}$ is integrated into the MLLM to enhance semantic understanding:

$$\begin{cases} \mathcal{E}_{D_i}^{\text{theme}} = \text{MLLM}(\mathcal{P}_{\text{ext_theme}}(D_i, \mathcal{I}_{D_i})), \\ \mathcal{V}_{D_i}^{\text{key}} = \text{MLLM}(\mathcal{P}_{\text{ext_key}}(\mathcal{E}_{D_i}^{\text{theme}}, D_i, \mathcal{I}_{D_i}, \mathcal{V}_{D_i})). \end{cases}$$

4.1.3 Hierarchical Hypergraph Merging

Merging all extracted entities, their relations, and chunk themes from all chunks, we get a fine-grained bottom hypergraph $\mathcal{G}_{\text{Bot}}$ and a theme-aware top hypergraph $\mathcal{G}_{\text{top}}$, which are connected via shared vertices (i.e., theme-related key entities $\{V_{\text{key}}\}$). Both hypergraphs are stored in a hypergraph database.

$$\begin{cases} \mathcal{G}_{\text{Bot}} = \{\bigcup_{D_i \in \mathcal{D}} \mathcal{V}_{D_i}, \bigcup_{D_i \in \mathcal{D}} \mathcal{E}_{D_i}\}, \\ \mathcal{G}_{\text{Top}} = \{\bigcup_{D_i \in \mathcal{D}} \mathcal{V}_{D_i}^{\text{key}}, \bigcup_{D_i \in \mathcal{D}} \mathcal{E}_{D_i}^{\text{theme}}\}. \end{cases}$$

4.2 Vision-enhanced Top-down Retrieval

VizRAG employs a visualization-guided top-down retrieval strategy, leveraging rendered hypergraph images as structural bridges from the global theme layer $\mathcal{G}_{\text{Top}}$ to the fine-grained entity layer $\mathcal{G}_{\text{Bot}}$. Given a query q, the keywords are first extracted:

$$\mathcal{X}_{\text{query}} = \text{LLM}(\mathcal{P}_{\text{keyword}}(q)),$$

where $\mathcal{X}_* = \{x_1, x_2, \ldots\}$ and $\mathcal{P}_{\text{keyword}}$ is in App. G.

4.2.1 Top Retrieval and Locating

These keywords $\mathcal{X}_{\text{query}}$ are used to retrieve the top-k relevant hyperedges $\mathcal{E}_{rel}$ from $\mathcal{G}_{\text{Top}}$ based on vector similarity $\mathcal{R}(\cdot,\cdot)$, where $\mathcal{E}^{\text{Top}}$ are edges in $\mathcal{G}_{\text{Top}}$:

$$\mathcal{E}_{\text{rel}} = \bigcup_{x_i \in \mathcal{X}_{\text{query}}} \mathcal{R}(x_i, \mathcal{E}^{\text{Top}}).$$

We then construct a h-hop induced subgraph $\mathcal{G}_{\text{Top}}^{\text{sub}}$ centered around the retrieved hyperedges $\mathcal{E}_{\text{rel}}$ and visualize it using HyperViz:

$$\mathcal{I}_{\text{Top}} = \text{HyperViz}(\text{Subgraph}(\mathcal{G}_{\text{Top}}, \mathcal{E}_{\text{rel}}, \text{hops} = h)).$$

VizRAG then exploits the MLLM's visual structural perception to locate key entities from $\mathcal{I}_{\text{Top}}$,

union with the direct neighbors of hyperedges $\mathcal{E}_{\text{rel}}$:

$$\hat{\mathcal{V}} = \text{MLLM}(\mathcal{P}_{\text{locate}}(q, \mathcal{I}_{\text{Top}}, \mathcal{C}_{\text{e_rel}})) \cup \mathcal{V}_{dif},$$
$$\mathcal{V}_{dif} = \bigcup_{e_i \in \mathcal{E}_{rel}} \mathcal{N}(e_i, \mathcal{G}_{\text{Top}}),$$

where $\mathcal{N}$ denotes retrieving neighbors from the hypergraph, $\mathcal{C}_{\text{e_rel}}$ is the relevant context of e_rel $\in \mathcal{E}_{\text{rel}}$, $\mathcal{P}_{\text{locate}}$ prompts the MLLM to locate query-relevant entities in the rendered subgraph structure.

4.2.2 Bottom Retrieval and Answer Generation

With $\hat{\mathcal{V}}$ as entity anchors, we retrieve the top-k matching vertices from $\mathcal{G}_{\text{Bot}}$ and expand context via their neighbor hyperedges, where $\mathcal{V}_{\text{Bot}} = \{\mathcal{V}_{D_i}\}$:

$$\mathcal{V}_{\text{rel}} = \bigcup_{v_i \in \hat{\mathcal{V}}} \mathcal{R}(v_i, \mathcal{V}_{\text{Bot}}),$$
$$\mathcal{E}_{\text{dif}} = \bigcup_{v_i \in \mathcal{V}_{\text{rel}}} \mathcal{N}(v_i, \mathcal{G}_{\text{Bot}}).$$

The retrieved entity subgraph is rendered as a visual cue $\mathcal{I}_{\text{Bot}}$ to expose fine-grained evidence paths:

$$\mathcal{I}_{\text{Bot}} = \text{HyperViz}(\text{Subgraph}(\mathcal{G}_{\text{Bot}}, \mathcal{V}_{\text{rel}}, \text{hops} = h)).$$

Finally, $\mathcal{V}_{\text{rel}}$, $\mathcal{E}_{\text{dif}}$, their corresponding contexts $\mathcal{C}_{\text{v_rel}}, \mathcal{C}_{\text{e_dif}}$, and the visualization $\mathcal{I}_{\text{Bot}}$ are jointly fed to the MLLM to generate the final answer:

$$\mathcal{A} = \text{MLLM}(q, \mathcal{I}_{\text{Bot}}, \mathcal{V}_{\text{rel}}, \mathcal{E}_{\text{dif}}, \mathcal{C}_{\text{v_rel}}, \mathcal{C}_{\text{e_dif}}).$$

5 Experiments

In this section, we perform extensive experiments to evaluate the proposed methods. The experimental setup is aligned with prior works (Hu et al., 2026; Feng et al., 2025b).

5.1 Experimental Setup

To systematically evaluate our method across diverse application scenarios, we adopt five datasets from two benchmarks: Mix, CS, and Agriculture from the UltraDomain benchmark (Qian et al., 2025), and Neurology and Pathology from the MIRAGE benchmark (Xiong et al., 2024). UltraDomain covers typical RAG applications across different domains, while MIRAGE focuses on medical question answering and domain-specific knowledge coverage. Data statistics are in App. C.

Based on domain consistency and semantic correlation within the texts, we categorize the datasets into three types to enable a comprehensive analysis of the model's adaptability: ***Cross-domain Sparse*** (Mix): Fragmented passages from unrelated domains with weak semantic coherence. ***Intra-domain Sparse*** (CS, Agriculture): Domain-specific documents with weak inter-passage context. ***Intra-domain Dense*** (Neurology, Pathology): Highly structured medical textbooks with strong semantic continuity from MIRAGE. Additionally, we follow the data processing and query procedure of LightRAG, utilizing Gemini-3 Flash to generate complex, document-related queries. Besides, to enhance the robustness and reduce stochasticity, we evaluate 10 times for each result. All experiments are conducted on 4 A100-80G GPUs.

Baselines. We compared our approach with the state-of-the-art and popular Graph-enhanced RAG methods: GraphRAG (Edge et al., 2024), LightRAG (Guo et al., 2024), HiRAG (Huang et al., 2025), and hypergraph RAGs including HyperRAG (Feng et al., 2025a) and Cog-RAG (Hu et al., 2026). The baseline details are in App. D.

Implementation Details. To ensure fairness and consistency for both the baselines and proposed methods, we validate on five different LLMs for information extraction and question answering, including Gemini-3 Flash (Pichai et al., 2025), GPT-5 Mini (Singh et al., 2025), GLM-5V Turbo (Hong et al., 2026), Claude Haiku 4.5 (Anthropic, 2025), and Qwen3VL-8B (Bai et al., 2025). The result evaluation is default on Gemini-3 Flash, as well as the text-embedding-3-small embedding model for vector encoding and retrieval tasks.

Evaluation Metrics. Following the recent works, we adopt two evaluation strategies: Selection-based (Guo et al., 2024; Huang et al., 2025) and Score-based (Wang et al., 2024; Feng et al., 2025a), providing both relative and absolute perspectives on model performance. ***The Selection-based*** evaluation uses LLMs to report win rates of answer quality between two methods. ***The Score-based*** evaluation employs LLMs to score responses of different methods. Both strategies assess models from six dimensions: Comprehensiveness, Empowerment, Relevance, Consistency, Clarity, and Logical. We report both per-dimension and overall average scores. More Evaluation Details are in App. E.

5.2 Main Results

Our primary results are presented in Tab. 1 and Fig. 4, and more results are provided in App. F. VizRAG consistently outperforms all baselines across multiple dimensions. Additionally, we have several key insights:

1) **VizRAG consistently wins against all base-**

	Mix		CS		Agriculture		Neurology		Pathology	
	GraphRAG	**VizRAG**	GraphRAG	**VizRAG**	GraphRAG	**VizRAG**	GraphRAG	**VizRAG**	GraphRAG	**VizRAG**
Comp.	28.7%	**71.3%**	24.5%	**75.5%**	20.8%	**79.2%**	22.7%	**77.3%**	20.5%	**79.5%**
Empo.	24.6%	**75.4%**	23.2%	**76.8%**	14.8%	**85.2%**	15.6%	**84.4%**	11.7%	**88.3%**
Rele.	33.9%	**66.1%**	27.5%	**72.5%**	23.1%	**76.9%**	25.2%	**74.8%**	19.6%	**80.4%**
Cons.	28.9%	**71.1%**	23.7%	**76.3%**	17.9%	**82.1%**	19.5%	**80.5%**	18.6%	**81.4%**
Clar.	34.7%	**65.3%**	24.8%	**75.2%**	26.5%	**73.5%**	24.2%	**75.8%**	18.5%	**81.5%**
Logi.	27.8%	**72.2%**	25.1%	**74.9%**	15.9%	**84.1%**	21.6%	**78.4%**	17.8%	**82.2%**
Overall	29.8%	**70.2%**	24.8%	**75.2%**	19.8%	**80.2%**	21.5%	**78.5%**	17.8%	**82.2%**
	LightRAG	**VizRAG**	LightRAG	**VizRAG**	LightRAG	**VizRAG**	LightRAG	**VizRAG**	LightRAG	**VizRAG**
Comp.	30.5%	**69.5%**	22.8%	**77.2%**	16.6%	**83.4%**	20.9%	**79.1%**	22.9%	**77.1%**
Empo.	22.2%	**77.8%**	19.5%	**80.5%**	12.6%	**87.4%**	15.7%	**84.3%**	17.6%	**82.4%**
Rele.	28.9%	**71.1%**	19.8%	**80.2%**	17.7%	**82.3%**	20.7%	**79.3%**	24.8%	**75.2%**
Cons.	26.8%	**73.2%**	21.5%	**78.5%**	13.7%	**86.3%**	17.8%	**82.2%**	19.7%	**80.3%**
Clar.	30.6%	**69.4%**	16.7%	**83.3%**	14.8%	**85.2%**	18.5%	**81.5%**	18.8%	**81.2%**
Logi.	27.7%	**72.3%**	21.6%	**78.4%**	15.6%	**84.4%**	18.8%	**81.2%**	18.9%	**81.1%**
Overall	27.8%	**72.2%**	20.3%	**79.7%**	15.2%	**84.8%**	18.7%	**81.3%**	20.4%	**79.6%**
	HiRAG	**VizRAG**	HiRAG	**VizRAG**	HiRAG	**VizRAG**	HiRAG	**VizRAG**	HiRAG	**VizRAG**
Comp.	36.6%	**63.4%**	32.8%	**67.2%**	33.7%	**66.3%**	27.5%	**72.5%**	32.9%	**67.1%**
Empo.	31.5%	**68.5%**	28.7%	**71.3%**	28.5%	**71.5%**	23.6%	**76.4%**	29.6%	**70.4%**
Rele.	37.7%	**62.3%**	39.8%	**60.2%**	36.7%	**63.3%**	27.6%	**72.4%**	33.8%	**66.2%**
Cons.	31.6%	**68.4%**	32.6%	**67.4%**	29.6%	**70.4%**	24.7%	**75.3%**	29.9%	**70.1%**
Clar.	38.6%	**61.4%**	42.7%	**57.3%**	36.8%	**63.2%**	23.5%	**76.5%**	32.7%	**67.3%**
Logi.	32.6%	**67.4%**	32.7%	**67.3%**	30.8%	**69.2%**	23.8%	**76.2%**	28.6%	**71.4%**
Overall	34.8%	**65.2%**	34.9%	**65.1%**	32.7%	**67.3%**	25.1%	**74.9%**	31.2%	**68.8%**
	Hyper-RAG	**VizRAG**	Hyper-RAG	**VizRAG**	Hyper-RAG	**VizRAG**	Hyper-RAG	**VizRAG**	Hyper-RAG	**VizRAG**
Comp.	37.7%	**62.3%**	38.6%	**61.4%**	40.8%	**59.2%**	31.6%	**68.4%**	33.8%	**66.2%**
Empo.	34.9%	**65.1%**	35.7%	**64.3%**	31.7%	**68.3%**	28.6%	**71.4%**	28.8%	**71.2%**
Rele.	47.2%	**52.8%**	40.7%	**59.3%**	38.7%	**61.3%**	39.2%	**60.8%**	28.7%	**71.3%**
Cons.	34.6%	**65.4%**	36.2%	**63.8%**	36.8%	**63.2%**	29.8%	**70.2%**	26.6%	**73.4%**
Clar.	46.6%	**53.4%**	39.7%	**60.3%**	33.6%	**66.4%**	32.7%	**67.3%**	24.7%	**75.3%**
Logi.	35.6%	**64.4%**	37.6%	**62.4%**	34.7%	**65.3%**	26.8%	**73.2%**	28.7%	**71.3%**
Overall	39.4%	**60.6%**	38.1%	**61.9%**	36.0%	**64.0%**	31.4%	**68.6%**	28.5%	**71.5%**
	Cog-RAG	**VizRAG**	Cog-RAG	**VizRAG**	Cog-RAG	**VizRAG**	Cog-RAG	**VizRAG**	Cog-RAG	**VizRAG**
Comp.	40.7%	**59.3%**	42.6%	**57.4%**	46.6%	**53.4%**	37.7%	**62.3%**	39.7%	**60.3%**
Empo.	39.5%	**60.5%**	40.5%	**59.5%**	37.8%	**62.2%**	34.6%	**65.4%**	34.7%	**65.3%**
Rele.	44.7%	**55.3%**	44.6%	**55.4%**	42.5%	**57.5%**	43.7%	**56.3%**	34.7%	**65.3%**
Cons.	40.6%	**59.4%**	41.6%	**58.4%**	41.7%	**58.3%**	35.8%	**64.2%**	33.5%	**66.5%**
Clar.	47.6%	**52.4%**	45.7%	**54.3%**	39.8%	**60.2%**	38.7%	**61.3%**	29.8%	**70.2%**
Logi.	41.6%	**58.4%**	43.7%	**56.3%**	40.7%	**59.3%**	32.6%	**67.4%**	34.7%	**65.3%**
Overall	42.4%	**57.6%**	43.1%	**56.9%**	41.5%	**58.5%**	37.2%	**62.8%**	34.5%	**65.5%**

Table 1: Average win rates of six evaluation metrics across five datasets. The comparison is made between baselines and VizRAG. Among them, we refer to the six metrics as Comp. (Comprehensiveness), Empo. (Empowerment), Rele. (Relevance), Cons. (Consistency), Clar. (Clarity), and Logi. (Logical).

lines on selection-based evaluation. As shown in Tab. 1, the overall win rates are 70.2%–82.2% vs. GraphRAG, 72.2%–84.8% vs. LightRAG, and 65.1%–74.9% vs. HiRAG across five datasets. VizRAG also outperforms strong hypergraph baselines, with 60.6%–71.5% vs. Hyper-RAG and 56.9%–65.5% vs. Cog-RAG. This confirms the effectiveness of VizRAG on pairwise comparison.

2) **VizRAG brings robust and comprehensive improvement on response quality.** Figs. 4(a) and 4(b) show that VizRAG achieves higher scores across *all datasets* and *all evaluation dimensions*, and Fig. 4(c) shows the same trend across *all tested LLM backbones*. These results verify that adding visual hypergraph cues delivers persistent and robust gains for hypergraph-based RAG systems.

5.3 Ablation Study

Effects of Method Components. We report score-based ablations on three representative datasets in Tab. 2 to validate the necessity of key components in our proposed methods: 1) visualization in indexing, 2) visualization in retrieval, 3) HyperViz for addressing visual congestion, and 4) HyperViz for addressing rendering bias. As the results shown in Tab. 2, removing any component leads to obvious performance degradation on all datasets, confirming their empirical effectiveness.

Advantages of Structural Discrimination. To evaluate whether hypergraph visualization im-

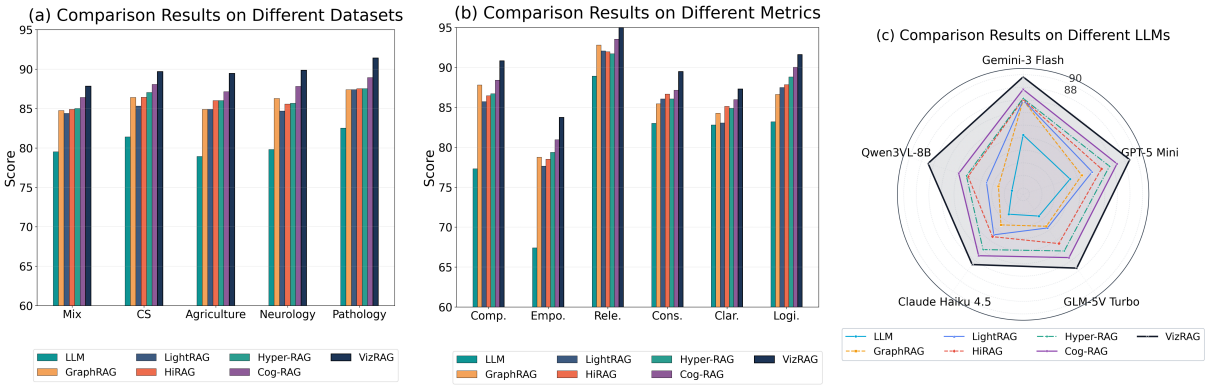

Figure 4: Test results by scoring. (a) is the comparison results on five datasets; (b) is the comparison results on six dimensions; (c) shows the evaluation results on different LLMs.

Models	Mix	CS	Neurology
VizRAG	88.0	89.7	89.9
w./o. Indexing Visualization	87.3	88.5	89.4
w./o. Retrieval Visualization	86.8	87.3	88.2
w./o. HyperViz (Address Visual Congestion)	87.5	88.1	88.9
w./o. HyperViz (Probe-driven Dual-View)	87.9	88.4	89.2

Table 2: Ablation results by overall scores. All ablated variants underperform full VizRAG on all datasets.

proves structural disambiguation, we define a binary classification task: distinguishing between structurally aligned positive candidates and lexically similar but structurally mismatched negative candidates. Using 200 query groups from the UltraDomain dataset (Qian et al., 2025), each query is paired with one correct match (positive) and one distractor (negative). The goal is to classify each candidate as "positive" or "negative". As shown in Tab. 3, integrating hypergraph visualization into the Hypergraph RAG pipeline (Feng et al., 2025b) improves accuracy from 71.5% to 79.0%, highlighting its effectiveness in resolving structural ambiguities.

Method	Accuracy (%)
Hypergraph RAG w/o Visualization	71.5
Hypergraph RAG w/ Visualization	79.0

Table 3: Structural disambiguation results on lexically matched hard negative samples.

Enhancing Multi-hop Reasoning Composition. To assess whether hypergraph visualization improves multi-hop reasoning, we evaluate 900 questions (evenly split across 2-hop, 3-hop, and 4-hop) from MusiQue (Trivedi et al., 2022) and HotpotQA (Yang et al., 2018). Using Exact Match (EM) as the metric, we compare the Hyper-RAG pipeline (Feng et al., 2025b) with and without visual hypergraph input. Results in Tab. 4 show consistent improvements across all reasoning depths, with the most significant gains in 3-hop (+6.5%) and 4-hop (+10.6%) tasks, highlighting the utility of hypergraph visualization in complex reasoning.

MLLM Hypergraph Perception Capability. We also evaluate the hypergraph perception capabilities of representative frozen off-the-shelf MLLMs

Reasoning EM	w/o visualization	w/ visualization
2-hop Reasoning	72.3	74.0
3-hop Reasoning	60.5	67.0
4-hop Reasoning	48.7	59.3

Table 4: Visualization effects on multi-hop reasoning.

Frozen MLLM	Overlap-core Identification	Bridge Hyperedge Detection
Qwen3VL-8B	86.8%	83.2%
Gemini-3 Flash	90.5%	88.9%

Table 5: Accuracy (%) of off-the-shelf MLLMs on key structural detection tasks.

through two key tasks: *overlap-core identification* and *bridge-hyperedge detection*. Both tasks (illustrated in Fig. 2(b)) are conducted on a fixed 1,000-sample subset from the UltraDomain dataset. As shown in Tab. 5, all MLLMs demonstrate notable accuracy in identifying these structural cues from hypergraphs, which serves as a strong foundation for integrating hypergraph visualization into their reasoning processes within RAG pipelines.

Efficiency Analysis. We analyze the retrieval efficiency for the VizRAG and baselines. As shown in Fig. 5, VizRAG achieves the highest overall score with low retrieval time, striking a balance between efficiency and performance. The optimized visualization adds only 0.21 seconds per query, a lightweight overhead compared to the baseline, while yielding significant score gains (Tab. 2). This makes VizRAG ideal for scenarios demanding both accuracy and speed.

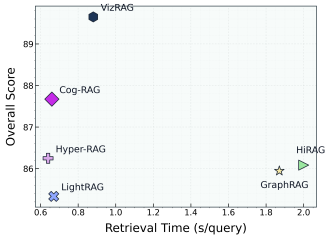

Figure 5: Trade-off between retrieval time and score.

6 Conclusion

We introduce hypergraph visualization as a novel enhancement for RAG systems, along with the plug-and-play HyperViz toolkit and VizRAG framework, enabling multimodal hypergraph reasoning and achieving significant gains in real-world knowledge-intensive tasks.

Limitations

While VizRAG demonstrates promising results, it faces several limitations. Its performance heavily depends on the chosen MLLM's ability to perceive visual structures, which may limit its effectiveness in scenarios involving complex visual reasoning tasks. Rendering combined with dual-view probing introduces additional system overhead, particularly in latency-sensitive applications. Furthermore, the reliance on LLM-as-a-judge evaluations may lead to evaluator bias. Future work will focus on developing adaptive visualization strategies that dynamically adjust to task-specific requirements and incorporating broader, human-centered evaluation methods to reduce bias and improve robustness.

References

AI Anthropic. 2025. System card: Claude opus 4 & claude sonnet 4. *Claude-4 Model Card*.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2023. Self-rag: Learning to retrieve, generate, and critique through self-reflection. In *The Twelfth International Conference on Learning Representations*.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, and 45 others. 2025. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*.

Alain Bretto. 2013. Hypergraph theory. *An introduction. Mathematical Engineering. Cham: Springer*, 1:209–216.

Haeyong Chung, Santhosh Nandhakumar, and Seungwon Yang. 2020. Gridset: Visualizing individual elements and attributes for analysis of set-typed data. *IEEE Transactions on Visualization and Computer Graphics*, 28(8):2983–2998.

Christopher Collins, Gerald Penn, and Sheelagh Carpendale. 2009. Bubble sets: Revealing set relations with isocontours over existing visualizations. *IEEE transactions on visualization and computer graphics*, 15(6):1009–1016.

Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, Dasha Metropolitansky, Robert Osazuwa Ness, and Jonathan Larson. 2024. From local to global: A graph rag approach to query-focused summarization. *Preprint*, arXiv:2404.16130.

Yifan Feng, Hao Hu, Xingliang Hou, Shiquan Liu, Shihui Ying, Shaoyi Du, Han Hu, and Yue Gao. 2025a. Hyper-rag: Combating llm hallucinations using hypergraph-driven retrieval-augmented generation. *Preprint*, arXiv:2504.08758.

Yifan Feng, Hao Hu, Xingliang Hou, Shiquan Liu, Shihui Ying, Shaoyi Du, Han Hu, and Yue Gao. 2025b. Hyper-rag: Combating llm hallucinations using hypergraph-driven retrieval-augmented generation. *arXiv preprint arXiv:2504.08758*.

Emden R Gansner and Stephen C North. 2000. An open graph visualization system and its applications to software engineering. *Software: practice and experience*, 30(11):1203–1233.

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

Zirui Guo, Lianghao Xia, Yanhua Yu, Tu Ao, and Chao Huang. 2024. Lightrag: Simple and fast retrieval-augmented generation. *Preprint*, arXiv:2410.05779.

Aric Hagberg, Pieter Swart, and Daniel S Chult. 2008. Exploring network structure, dynamics, and function using networkx. Technical report, Los Alamos National Lab.(LANL), Los Alamos, NM (United States).

Wenyi Hong, Xiaotao Gu, Ziyang Pan, Zhen Yang, Yuting Wang, Yue Wang, Yuanchang Yue, Yu Wang, Yanling Wang, Yan Wang, and 1 others. 2026. Glm-5v-turbo: Toward a native foundation model for multimodal agents. *arXiv preprint arXiv:2604.26752*.

Hao Hu, Yifan Feng, Ruoxue Li, Rundong Xue, Xingliang Hou, Zhiqiang Tian, Yue Gao, and Shaoyi Du. 2026. Cog-rag: Cognitive-inspired dual-hypergraph with theme alignment retrieval-augmented generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 31032–31040.

Haoyu Huang, Yongfeng Huang, Junjie Yang, Zhenyu Pan, Yongqiang Chen, Kaili Ma, Hongzhi Chen, and James Cheng. 2025. Retrieval-augmented generation with hierarchical knowledge. *Preprint*, arXiv:2503.10150.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474.

Alexander Lex, Nils Gehlenborg, Hendrik Strobelt, Romain Vuillemot, and Hanspeter Pfister. 2014. Upset: visualization of intersecting sets. *IEEE transactions on visualization and computer graphics*, 20(12):1983–1992.

Zhuoqun Li, Xuanang Chen, Haiyang Yu, Hongyu Lin, Yaojie Lu, Qiaoyu Tang, Fei Huang, Xianpei Han, Le Sun, and Yongbin Li. 2024. Structrag: Boosting knowledge intensive reasoning of llms via inference-time hybrid information structurization. *Preprint*, arXiv:2410.08815.

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

Sundar Pichai, Demis Hassabis, and Koray Kavukcuoglu. 2025. A new era of intelligence with gemini 3. *Mountain View, CA: Google*.

Hongjin Qian, Zheng Liu, Peitian Zhang, Kelong Mao, Defu Lian, Zhicheng Dou, and Tiejun Huang. 2025. Memorag: Boosting long context processing with global memory-enhanced retrieval augmentation. In *Proceedings of the ACM on Web Conference 2025*, pages 2366–2377.

Bhaskarjit Sarmah, Dhagash Mehta, Benika Hall, Rohan Rao, Sunil Patel, and Stefano Pasquali. 2024. Hybridrag: Integrating knowledge graphs and vector retrieval augmented generation for efficient information extraction. In *Proceedings of the 5th ACM International Conference on AI in Finance*, pages 608–616, New York, NY, USA.

Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, Aidan McLaughlin, Aiden Low, AJ Ostrow, Akhila Ananthram, and 1 others. 2025. Openai gpt-5 system card. *arXiv preprint arXiv:2601.03267*.

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

Minzheng Wang, Longze Chen, Cheng Fu, Shengyi Liao, Xinghua Zhang, Bingli Wu, Haiyang Yu, Nan Xu, Lei Zhang, Run Luo, and 1 others. 2024. Leave no document behind: Benchmarking long-context llms with extended multi-doc qa. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 5627–5646, Miami, Florida, USA.

Yuan Xia, Jingbo Zhou, Zhenhui Shi, Jun Chen, and Haifeng Huang. 2025. Improving retrieval augmented language model with self-reasoning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 39, pages 25534–25542.

Guangzhi Xiong, Qiao Jin, Zhiyong Lu, and Aidong Zhang. 2024. Benchmarking retrieval-augmented generation for medicine. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 6233–6251, Bangkok, Thailand.

Xiao Yang, Kai Sun, Hao Xin, Yushi Sun, Nikita Bhalla, Xiangsen Chen, Sajal Choudhary, Rongze Gui, Ziran Jiang, Ziyu Jiang, and 1 others. 2024. Crag-comprehensive rag benchmark. In *Advances in Neural Information Processing Systems*, volume 37, pages 10470–10490, Red Hook, NY, USA.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pages 2369–2380.

Lingxi Zhang, Yue Yu, Kuan Wang, and Chao Zhang. 2024. ARL2: Aligning retrievers with black-box large language models via self-guided adaptive relevance labeling. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, pages 3708–3719, Bangkok, Thailand.

A Related Works

RAG Systems. Existing RAG systems can be categorized into three types based on their knowledge structures: (i) Document-based RAG methods (Asai et al., 2023; Zhang et al., 2024; Xia et al., 2025; Yang et al., 2024) primarily rely on flat document structures for retrieval and generation. However, they struggle with modeling semantic relationships and contextual dependencies, limiting the accuracy and completeness of generated responses. (ii) GraphRAG methods (Sarmah et al., 2024; Peng et al., 2024) address the limitations by incorporating knowledge graphs, which enhance the organization and semantic expression of content retrieval by structurally representing entities and their relationships. Despite these advancements, GraphRAG often constructs sparse graph structures, failing to fully capture higher-order semantic associations and complex contextual relations. (iii) HypergraphRAG (Feng et al., 2025a; Hu et al., 2026) extends this approach by utilizing hypergraphs, where hyperedges connect multiple nodes to represent higher-order semantic connections and group semantics, significantly enhancing the expressiveness of knowledge structures and retrieval quality. However, these HypergraphRAG methods solely rely on text input, which fail to adapt modern MLLMs' super vision capability, motivating our exploration of hypergraph visualization to enhance them.

Hypergraph Visualization. Existing hypergraph visualization methods can be broadly grouped into two categories. (i) **Set-based visualization methods** treat each hyperedge as a set and focus on overlap and intersection patterns. Representative designs include contour-based approaches such as Bubble Sets (Collins et al., 2009), matrix-based intersection analysis such as UpSet (Lex et al., 2014), and element-attribute analysis for dense set systems such as GridSet (Chung et al., 2020). (ii) **Induced visualization methods** convert a hypergraph into an incidence (bipartite) graph by introducing virtual nodes for hyperedges (Bretto, 2013), and then apply mature graph visualization techniques (Gansner and North, 2000; Hagberg et al., 2008) to the induced structure. The first preserves set semantics directly, while the second offers better compatibility with scalable graph layouts and interaction pipelines.

B HyperViz Toolkit Details

This appendix supplements the HyperViz toolkit's extra characteristics not covered in the main texts.

B.1 Additional HyperViz Characteristics

Customizable Visual Styles. HyperViz offers extensive customization options during rendering, including adjustable parameters for node shapes, edge thickness, background colors, node outline styles, and graph margins. For example, node shapes can be set to options such as `ellipse`, `circle`, `box`, or `polygon`, with defaults like `ellipse` and `rounded box` commonly used to differentiate entities and hyperedges. Edge thickness can be customized within a range of 0.1 to 10 pixels, while background colors support RGB, HEX, or standard color names. Node outline styles include options such as `solid`, `dashed`, and `dotted`. The spacing between the graph and canvas edges can be adjusted within a range of 0 to 100 pixels. Additionally, in the set-based paradigm, hyperedge region colors can be selected from predefined palettes (e.g., `pastel` or `vivid`) or defined by the user. HyperViz also provides toggle options such as `show_legend`, `show_title`, `show_footer`, and `show_edge_labels`, allowing users to control the display of legends, titles, footers, and edge labels. These features collectively enable HyperViz to adapt to a wide range of visualization tasks and requirements.

Interactive Visualization. Beyond static PNG outputs, HyperViz preserves stable visual IDs (entity IDs as `E1, E2,...`, hyperedge IDs as `H1, H2,...`) and exports synchronized graph-text mappings, which support interactive front-end behaviors without changing backend retrieval logic. In practical deployment, we bind these IDs to metadata tables and enable interaction primitives, including zoom, pan, hover-tooltip, and click-to-expand neighborhood. Clicking an entity can trigger one-hop or multi-hop expansion through $\mathcal{N}(v, \mathcal{G})$, while clicking a hyperedge opens its member set, relation description, source chunk ID, and evidence text. This ID-consistent design guarantees that interactive exploration and MLLM prompting refer to the same structural objects.

Scalability Optimization. HyperViz uses a staged compression-render pipeline for dense subgraphs. First, it applies budgeted subgraph construction with explicit caps (e.g., `max_edges` and

max_entities) and preserves high-utility structures through query-aware expansion. Second, for panel rendering, it uses mode-specific simplification: induced layouts keeps explicit E-H participation edges, and set-based layouts emphasizes group semantics by contour aggregation. Third, for multi-view usage, render_hypergraph_collage arranges complementary views in a bounded panel grid to avoid single-canvas overplotting. Together with the main-text pruning strategy (importance-based filtering under a congestion budget), these controls preserve readability while retaining key high-order evidence.

More detailed characteristics and features of HyperViz will be compiled into a comprehensive manual, which will be made available on a dedicated website following the acceptance of the paper, along with the code.

B.2 Hyperparameter Optimization Details

We adopt a unified grid search strategy to optimize the full set of hyperparameters in HyperViz, including w_v, w_e, γ, α, and β. Due to the inherent mutual dependencies across these hyperparameters, all parameters are jointly optimized on real-world knowledge hypergraphs, rather than adopting separate stage-wise tuning. All spatial weight parameters w_v, w_e and structural regularization parameters γ, α, β are searched simultaneously within their respective ranges of $[1, 10]$ and $[0, 1]$.

The optimization is performed entirely on real knowledge hypergraphs constructed from the UltraDomain Legal dataset (Qian et al., 2025). To strictly prevent data leakage, this tuning subset is excluded from all evaluation test sets throughout our experiments. The adopted real legal hypergraphs naturally cover diverse density distributions, with node degrees ranging from 1 to 9 and hyperedge sizes spanning $[2, 11]$, fully encompassing sparse, moderate, and dense structural patterns in real document scenarios.

All grid search trials are supervised by the structural perception capability of Gemini-3 Flash, utilizing the overlap-core identification and bridge hyperedge detection tasks illustrated in Fig. 2. The final parameter combination is selected to maximize structural perception accuracy, which we equate to visual readability for MLLM-centric visualization evaluation.

B.3 Description of Layouts in Hyperviz

HyperViz provides six layout variants under two complementary paradigms: set-based views for group semantics and Induced views for explicit incidence relations. In implementation, these layouts are achieved by customizing layout algorithms in Graphviz (Gansner and North, 2000) with paradigm-specific post-processing (e.g., contour generation for set regions and bipartite incidence expansion for induced views).

Set-based layouts represent high-order relationships (i.e., hyperedges) as set elements through region containment to highlight entity grouping relations. In Hyperviz, it is implemented in three diverse layout variants:

- **Set-Hier layout** is achieved by the Graphviz dot layout algorithm. It emphasizes top-down hierarchy and dependency chains, making parent-child style organization and progressive evidence flow easier to inspect.
- **Set-Radial layout** is achieved by the Graphviz twopi layout algorithm. It emphasizes center-periphery organization around high-degree entities, which is useful for identifying hubs, overlap cores, and radius-like influence patterns.
- **Set-Force layout** is achieved by the Graphviz fdp/sfdp layout algorithms. It emphasizes community structure and local cluster separation, helping reveal dense semantic groups and weak inter-group bridges.

Induced layouts emphasize direct element-group relationships by representing hyperedges as special nodes, with pairwise edges explicitly connecting these nodes to their corresponding hyperedge nodes.

- **Bi-Inc:** uses NetworkX bipartite_layout to separate entity nodes and virtual hyperedge nodes into two partitions, making membership links directly readable. Incidence edges are drawn with light dashed strokes, entity nodes are styled as blue-outlined circles, and virtual hyperedge nodes are styled as orange-outlined squares.
- **Ra-Cen:** first selects the center as the highest-degree entity in the induced graph, then computes shortest-path distance shells and renders them using NetworkX (Hagberg et al.,

2008) `shell_layout`. This design emphasizes center-periphery organization for neighborhood expansion and multi-hop anchor tracing; if no valid center is available, it falls back to `spring_layout`.

- **Cli-Exp:** keeps the induced incidence backbone and additionally expands each hyperedge $e = \{v_1, v_2, \ldots, v_k\}$ into pairwise entity links (i.e., $\binom{k}{2}$ clique edges). The panel is laid out by NetworkX `spring_layout` (seed=42, $k = 0.95$), where dashed incidence edges preserve E–H semantics and solid clique edges improve local connectivity perception.

To maximize the interpretability of visualized hypergraphs, explicit semantic annotations are embedded: **(1) Node-Level Annotation.** Intuitive text labels are attached to all entity nodes, presenting clear semantic information. **(2) Hyperedge-Level Annotation.** Internal inclusion relations of hyperedges are summarized and displayed via unified legend labels, clarifying hierarchical associations.

C Dataset Statistics

Statistics	Mix	CS	Agriculture	Neurology	Pathology
Total Documents	61	10	12	1	1
Total Chunks	560	1992	1813	1790	824
Total Tokens	615,355	2,190,803	1,993,515	1,968,716	905,760

Table 6: Statistical information of the reported datasets.

Table 6 reports the dataset statistics used in this paper.

D Baseline Details

We compare VizRAG with representative methods spanning graph and hypergraph RAG families.
GraphRAG (Edge et al., 2024): entity-relation graph construction with graph-guided retrieval and community-level summaries.
LightRAG (Guo et al., 2024): lightweight graph-enhanced retrieval with dual-level graph context integration.
HiRAG (Huang et al., 2025): hierarchical graph retrieval with multi-level structural traversal.
Hyper-RAG (Feng et al., 2025a): hypergraph retrieval modeling high-order relations through hyperedges.
Cog-RAG (Hu et al., 2026): cognitive-inspired dual-hypergraph retrieval with theme-aligned top-down evidence activation.

For fairness, we keep the chunking strategy, embedding model, and evaluation protocol aligned with Cog-RAG (Hu et al., 2026) and keep consistent across methods.

E Evaluation Protocol Details

We evaluated the proposed method and baseline across six key dimensions: Comprehensiveness, Empowerment, Relevance, Consistency, Clarity, and Logical Coherence.

E.1 Selection-based Evaluation

For qualitative comparison, we report win rates between methods. To ensure fairness, the order of answers in each method pair was alternated in the prompts, and the average result was calculated. The evaluation dimensions are described below:

- **Comprehensiveness (Comp.)**: Measures how thoroughly the answer addresses all aspects and details of the question.
- **Empowerment (Empo.)**: Assesses the extent to which the answer helps the reader understand the topic and make informed decisions.
- **Relevance (Rele.)**: Evaluates how accurately the response focuses on the key aspects of the question without including irrelevant information.
- **Consistency (Cons.)**: Examines the ability of the system to synthesize and integrate information from multiple sources into a coherent and logical response.
- **Clarity (Clar.)**: Gauges whether the response avoids unnecessary verbosity and redundancy while providing complete and easily understandable information.
- **Logical Coherence (Logi.)**: Checks whether the response maintains consistent and rational arguments without contradictions throughout.

E.2 Score-based Evaluation

For quantitative assessment, we employed LLMs to score the responses based on the following indicators:

- **Comprehensiveness (0-100)**: Evaluates whether the response comprehensively covers all critical aspects of the question and identifies any omissions.

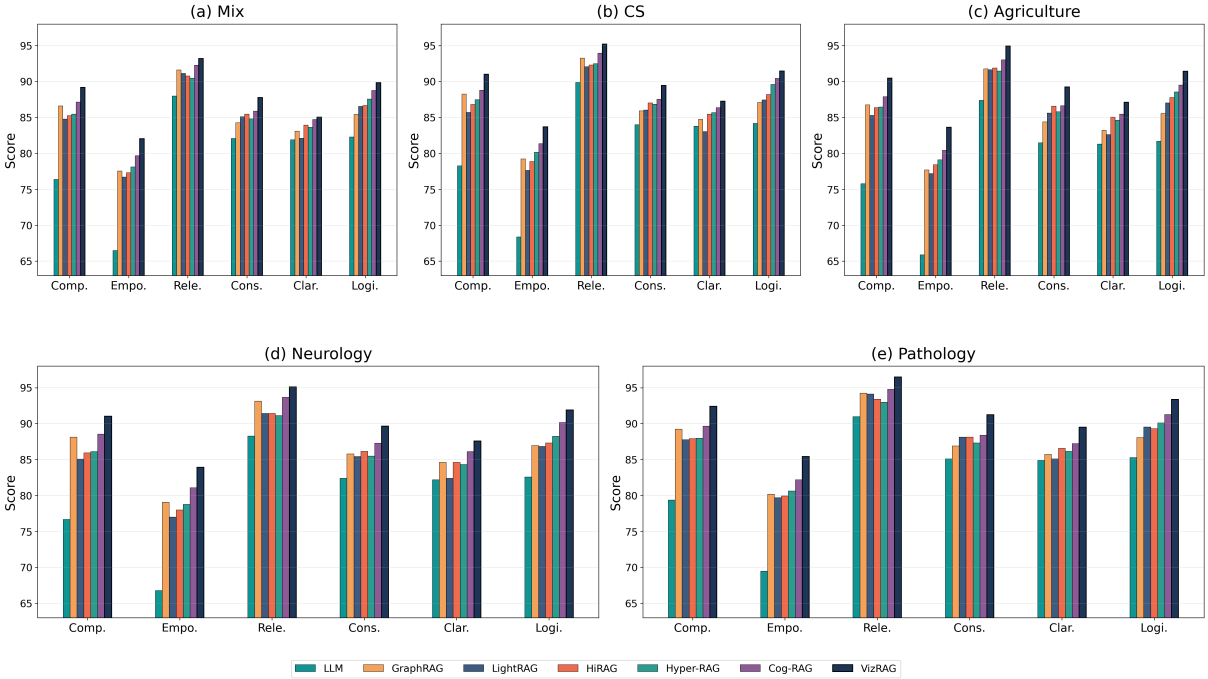

Figure 6: Score-based metric comparison of full five datasets.

- **Empowerment (0-100)**: Measures the credibility and persuasiveness of the response. High scores are given to answers that provide strong evidence or cite authoritative sources.

- **Relevance (0-100)**: Determines the degree to which the response stays focused on the question and avoids unnecessary digressions.

- **Consistency (0-100)**: Assesses whether the response is logically organized, flows smoothly, and the components are well-connected and mutually supportive.

- **Clarity (0-100)**: Examines whether the response is expressed clearly and concisely, using precise language and unambiguous definitions.

- **Logical Coherence (0-100)**: Evaluates whether the response maintains logical consistency and is free from contradictions.

E.3 Rating Levels

Each evaluation dimension is assessed using five discrete levels, with clear scoring criteria to ensure consistency and transparency. For example, the **Comprehensiveness** dimension is rated as follows:

- **Level 1 (0-20)**: The response is highly incomplete and omits significant portions or key aspects of the question.

- **Level 2 (20-40)**: The response includes some relevant content but misses many important elements, making it insufficiently comprehensive.

- **Level 3 (40-60)**: The response addresses the main aspects of the question but still has notable omissions.

- **Level 4 (60-80)**: The response is mostly comprehensive, covering most aspects of the question with minimal omissions.

- **Level 5 (80-100)**: The response is exceptionally thorough, addressing all aspects of the question without any omissions, providing a complete understanding to the reader.

F Additional Results

F.1 Detailed Scores on Individual Datasets

Fig.6 shows the results of scores under different evaluation dimensions on the individual Mix, CS, Agriculture, Neurology, and Pathology datasets.

G Prompt Templates

G.1 Core Prompt Set Aligned with VizRAG

Extracting Entities

Formulation: $\mathcal{P}_{\text{ext_entity}}(D_i)$

Prompt: *You are given one chunk D_i. Extract all important entities for retrieval and reasoning. Return JSON only with the schema:*
{ "entities": [{ "entity_name": str, "entity_type": one of [organization, person, geo, event, role, concept, method, artifact, disease, treatment, mechanism], "entity_description": str, "aliases": [str], "source_spans": [str] }] }
Requirements:
1) Use the same language as the input chunk.
2) Merge duplicates and aliases.
3) Keep entities that are structurally useful for downstream hyperedge construction.

Extracting Hyperedges

Formulation: $\mathcal{P}_{\text{ext_relation}}(D_i, \mathcal{V}_{D_i})$

Prompt: *You are given a chunk D_i and the extracted entity set $\mathcal{V}_{D_i}$. Construct hyperedges that represent semantically coherent multi-entity relations (including pairwise entity relationships as a special case). Return JSON only with the schema:*
{ "hyperedges": [{ "entities_set": [entity_name], "relation_description": str, "relation_keywords": [str], "relation_strength": number in [0,100], "evidence_spans": [str] }] }
Requirements:
1) Each hyperedge must contain at least 2 (including 2) valid entities from $\mathcal{V}_{D_i}$.
2) Prefer compact but meaningful sets; avoid trivial or redundant hyperedges.
3) If two hyperedges are near-duplicates, keep the one with clearer evidence.

Extracting Theme with Visual Cue

Formulation: $\mathcal{P}_{\text{ext_theme}}(D_i, \mathcal{I}_{D_i})$

Prompt: *You are given chunk text D_i and its hypergraph image $\mathcal{I}_{D_i}$. This visual input $\mathcal{I}_{D_i}$ encodes the intra-chunk knowledge structure over entity objects and their grouping membership. Infer a concise theme that is consistent with both text and structure. Return JSON only:*
{ "theme_description": str, "support_entity_ids": ["E"], "support_hyperedge_ids": ["H*"] }*
Requirements:
1) Theme must reflect the central semantic focus of the chunk.
2) Prioritize evidence supported by both text and visual structure.

Extracting Theme-related Key Entities with Visual Cue

Formulation:
$\mathcal{P}_{\text{ext_key}}(\mathcal{E}_{D_i}^{theme}, D_i, \mathcal{I}_{D_i}, \mathcal{V}_{D_i})$

Prompt: *You are given: (a) theme description, (b) chunk text D_i, (c) hypergraph image $\mathcal{I}_{D_i}$ with stable IDs, and (d) candidate entities $\mathcal{V}_{D_i}$. This visual input $\mathcal{I}_{D_i}$ encodes the intra-chunk knowledge structure over entity objects and their grouping membership. Select theme-critical key entities for top-layer indexing. Return JSON only:*
{ "key_entities": [{ "entity_name": str, "entity_id": "E" (if available), "key_entity_type": str, "key_entity_description": str, "key_score": number in [0,100], "selection_reason": str }] }*
Requirements:
1) Prioritize entities that are both theme-relevant and structurally central.
2) Avoid selecting many near-duplicate entities.

Extracting Query Keywords for Top Retrieval

Formulation: $\mathcal{P}_{\text{keyword}}(q)$

Prompt: *Given user query q, extract retrieval keywords for top-layer hyperedge search. Return JSON only:*
{ "theme_keywords": [str], "entity_keywords": [str] }
Requirements:
1) Keep keywords short and retrieval-oriented.
2) Cover both abstract intent (theme-level) and concrete anchors (entity-level).
3) Remove redundant synonyms when possible.

Locating Query-relevant Entities from Top-view Visualization

Formulation: $\mathcal{P}_{\text{locate}}(q, \mathcal{I}_{Top}, \mathcal{C}_{e_rel})$

Prompt: *You are given query q, top-layer subgraph visualization $\mathcal{I}_{Top}$, and context summaries $\mathcal{C}_{e_rel}$ of retrieved theme hyperedges. The visual input $\mathcal{I}_{Top}$ encodes top-layer knowledge structure, including entities, theme hyperedges, and their structural connectivity. Identify entities in the visual structure that are most relevant for downstream bottom retrieval. Return JSON only:*
{ "selected_entities": [{ "entity_id": "E", "entity_name": str, "relevance_score": number in [0,100], "reason": str }] }*
Requirements:
1) Use both visual structure (bridges, overlaps, hubs) and textual context.
2) Prefer entities that can connect to answer-critical evidence paths.

Final Answer Generation with Bottom-view Evidence

Formulation:
$\mathcal{P}_{\text{answer}}(q, \mathcal{I}_{Bot}, \mathcal{V}_{rel}, \mathcal{E}_{dif}, \mathcal{C}_{v_rel}, \mathcal{C}_{e_dif})$

Prompt: *You are given query q, bottom-layer visualization $\mathcal{I}_{Bot}$, retrieved entities $\mathcal{V}_{rel}$, expanded hyperedges $\mathcal{E}_{dif}$, and their textual contexts $\mathcal{C}_{v_rel}, \mathcal{C}_{e_dif}$. The visual input $\mathcal{I}_{Bot}$ encodes bottom-layer evidence structure over entities, hyperedges, and their local reasoning paths. Generate the final answer grounded in the provided evidence. Return JSON only:*
{ "answer": str, "support_entity_ids": ["E"], "support_hyperedge_ids": ["H*"], "confidence": number in [0,100] }*
Requirements:
1) Prefer evidence-consistent reasoning over unsupported speculation.
2) If evidence is insufficient, explicitly state uncertainty.
3) Keep the answer concise but complete for the question.

G.2 Evaluation Metrics

Selection-based Evaluation

Formulation: $\mathcal{P}_{\text{eval_scoring}}(q, \mathcal{A}_1, \mathcal{A}_2)$
q denotes user query, $\mathcal{A}_1$ and $\mathcal{A}_2$ denotes the response from two approaches.

Prompt: *You will evaluate two answers to the same question based on six criteria: Comprehensiveness, Empowerment, Relevance, Consistency, Clarity, and Logical.*
—Goal—
You will evaluate two answers to the same question by using the relevant documents based on six criteria: Comprehensiveness, Empowerment, Relevance, Consistency, Clarity, and Logical.
-Comprehensiveness: How much detail does the answer provide to cover all aspects and details of the question?
-Empowerment: How well does the answer help the reader understand and make informed judgments about the topic?
...,
-Logical: How well does the system maintain consistent logical arguments without contradicting itself across the response?

For each criterion, choose the better answer (either Answer 1 or Answer 2) and explain why. Then, select an overall winner based on these six categories.
Here is the question: q
Here are the two answers:
Answer 1: $\mathcal{A}_1$*;*
Answer 2: $\mathcal{A}_2$
Evaluate both answers using the six criteria listed above and provide detailed explanations for each criterion.

Scoring-based Evaluation

Formulation: $\mathcal{P}_{\text{eval_scoring}}(q, \mathcal{A}, \mathcal{C})$
q denotes user query, $\mathcal{A}$ denotes LLM response, $\mathcal{C}$ denotes the original text chunk that generated the question.

Prompt: *You are an expert tasked with evaluating answers to the questions by using the relevant documents based on five criteria: Comprehensiveness, Diversity, Empowerment, Logical, and Readability.*

—Goal—
You will evaluate the answers to the question by using the relevant documents based on six criteria: Comprehensiveness, Empowerment, Relevance, Consistency, Clarity, and Logical.

-Comprehensiveness-
Measure whether the answer comprehensively covers all key aspects of the question and whether there are omissions.
Level | score range | description
Level 1 | 0-20 | The answer is extremely one-sided, leaving out key parts or important aspects of the question.
Level 2 | 20-40 | The answer has some content, but it misses many important aspects of the question and is not comprehensive enough.
Level 3 | 40-60 | The answer is more comprehensive, covering the main aspects of the question, but there are still some omissions.
Level 4 | 60-80 | The answer is comprehensive, covering most aspects of the question, with few omissions.
Level 5 | 80-100 | The answer is extremely comprehensive, covering all aspects of the question with no omissions, enabling the reader to gain a complete understanding.
...,

For each indicator, please give the problem a corresponding Level based on the description of the indicator, and then give a score according to the score range of the level.
Here is the question: q
Here are the relevant document: $\mathcal{C}$
Here are the answer: $\mathcal{A}$
Evaluate all the answers using the six criteria listed above. For each criterion, provide a summary description, give a Level based on the description of the indicator, and then give a score based on the score range of the level.