

# MegaRAG: Multimodal Knowledge Graph-Based Retrieval Augmented Generation

Chi-Hsiang Hsiao<sup>1,\*</sup> Yi-Cheng Wang<sup>1,\*</sup> Tzung-Sheng Lin<sup>4</sup>  
Yi-Ren Yeh<sup>2</sup> Chu-Song Chen<sup>1,3</sup>

<sup>1</sup>Department of Computer Science and Information Engineering, National Taiwan University

<sup>2</sup>Department of Mathematics, National Kaohsiung Normal University

<sup>3</sup>FinTech Center, National Taiwan University

<sup>4</sup>E.SUN Financial Holding Co., Ltd.

<sup>1</sup>{r12922048, d13922033, chusong}@csie.ntu.edu.tw

<sup>4</sup>francis-17710@esunbank.com <sup>2</sup>yryeh@nknu.edu.tw

## Abstract

Retrieval-augmented generation (RAG) enables large language models (LLMs) to dynamically access external information, which is powerful for answering questions over previously unseen documents. Nonetheless, they struggle with high-level conceptual understanding and holistic comprehension due to limited context windows, which constrain their ability to perform deep reasoning over long-form, domain-specific content such as full-length books. To solve this problem, knowledge graphs (KGs) have been leveraged to provide entity-centric structure and hierarchical summaries, offering more structured support for reasoning. However, existing KG-based RAG solutions remain restricted to text-only inputs and fail to leverage the complementary insights provided by other modalities such as vision. On the other hand, reasoning from visual documents requires textual, visual, and spatial cues into structured, hierarchical concepts. To address this issue, we introduce a multimodal knowledge graph-based RAG that enables cross-modal reasoning for better content understanding. Our method incorporates visual cues into the construction of knowledge graphs, the retrieval phase, and the answer generation process. Experimental results across both global and fine-grained question answering tasks show that our approach consistently outperforms existing approaches on both textual and multimodal benchmarks. Our code is available on <https://github.com/AI-Application-and-Integration-Lab/MegaRAG>.

## 1 Introduction

Humans naturally integrate multiple modalities such as textual, visual, and layout to fluidly transition between abstract and detailed reasoning. However, multimodal large language models (MLLMs) (Bai et al., 2025; Grattafiori et al., 2024; Hurst et al., 2024; Team et al., 2023), despite recent progress,

remain limited by constrained context windows, restricting their ability to process long-form, domain-specific content. Interpreting a history textbook involves both global and localized observations, which remains challenging for MLLMs.

On the other hand, RAG can enhance LLMs by providing on-demand access to external knowledge. Early text-based RAG relied on sparse or dense retrieval but struggled with deep, multi-hop reasoning in multimodal documents. Recently, Graph-based RAG introduces structured abstraction via entity-relation graphs. With models like GraphRAG (Edge et al., 2024) and LightRAG (Guo et al., 2025), long-range knowledge retrieval of improved scalability are enhanced through KG-assisted retrieval pipelines. However, these methods excel in text-based multi-hop reasoning but remain constrained in handling complex, multimodal content. Current graph-based RAG methods face some key limitations. First, existing approaches remain unimodal, overlooking visual cues like diagrams, charts or maps, yielding disjointed representations that hinder multimodal reasoning. Additionally, due to context window constraints, most approaches segment documents into independent chunks, extracting entities separately rather than sequentially. This leads to fragmented KGs that miss cross-chunk relationships and key entities.

To our knowledge, while recent studies have explored manually constructed multimodal knowledge graphs (KGs) for RAG-based question answering (Lee et al., 2024), automatically building such KGs for RAG-assisted reasoning remains underexplored. To address this gap, we introduce MegaRAG, a multimodal, graph-based RAG method that enhances cross-modal reasoning.

To better handle the association of different modalities in visual documents, more relations beyond text-to-texts need to be extracted, such as text-to-figures and figure-to-figure relations. Although the parallel-reading-then-combining strategy can

\*Equal contribution.

refine entities and relations as in GraphRAG (Edge et al., 2024) and LightRAG (Guo et al., 2025), such refinement still relies on a single chunk while overlooking global document information. To address this limitation, we design a page-based, two-round approach for KG construction. Our solution initiates a KG by simply extracting entity-relation pairs in parallel for every page of a document using existing MLLMs, and the page-based relations are joined to form an initial graph. As the initial KG may not capture the inter-relationship between texts and visual elements sufficiently well, we conduct refinement processes in subsequent stage(s), where the initial KG(s) serve as global guidance to capture subtle relationships often lost in naïve, isolated extraction. In particular, to maintain scalability while incorporating long-range dependencies, we avoid injecting the entire initial KG into the MLLM inputs. Instead, we retrieve only a sub-graph of the entire KG for each page, yielding a lightweight yet context-aware input. This strategy enables progressive improvement of the graph’s structural coherence and cross-modal grounding.

We validate MegaRAG across global (book-level) and local (page/slide-level) QA benchmarks, spanning both text-only and multimodal datasets. Experimental results demonstrate that MegaRAG consistently outperforms strong baselines, particularly in scenarios requiring deep cross-modal integration and structured abstraction. Our contributions are summarized as follows.

- We introduce MegaRAG, an easy-to-use system that automatically constructs Multimodal KGs for visual document question answering with MLLMs.
- We develop a novel refinement process that enhances cross-modal grounding while addressing limitations in independent KG construction.
- We demonstrate that MegaRAG outperforms strong baselines on both global and local QA tasks, including GraphRAG and LightRAG.

## 2 Related Work

We briefly review several major directions of RAG: including retrieving information directly from raw data sources such as documents and images, and integrating structured knowledge through KGs.

**RAG with Raw Data Source.** Early RAG methods (Gua et al., 2020; Lewis et al., 2020) retrieve text chunks from corpora to support answer generation, primarily relying on retrieval strategies either sparse or dense. Sparse methods exemplified by

TF-IDF (Salton et al., 1975) and BM25 (Robertson and Zaragoza, 2009) depend on lexical heuristics to match queries with relevant text segments. They offer computational efficiency but lack deeper semantic comprehension. Dense techniques (Karpukhin et al., 2020; Khattab and Zaharia, 2020; Santhanam et al., 2022) project queries and documents into a shared embedding space, significantly improving retrieval performance of lexical variations. Subsequent works have enhanced this pipeline using LLM recently: HyDE (Gao et al., 2023) generates a hypothetical answer to enrich the retrieval query, Self-RAG (Asai et al., 2024) introduces reflection tokens to enable adaptive retrieval and self-critique within a single LLM, while RQ-RAG (Chan et al., 2024) decomposes the query into sub-queries to improve context coverage. Despite their strong performance on text-based RAG tasks, these methods often struggle with multimodal documents involving complex texts, layouts and visual elements.

**Multimodal RAG (MMRAG).** To tackle the limitations, more recent studies have focused on multimodal retrieval methods that better retain the structural information of documents. DSE (Ma et al., 2024) treats document screenshots as unified inputs and directly encodes their visual layout, text, and images into a single vector embedding. ColPaLi (Faysse et al., 2025) continues this direction by encoding document images into multi-vector embeddings, effectively capturing fine-grained visual cues. Its variant, ColQwen, replaces the PaLI-Gemma (Beyer et al., 2024) with Qwen2-VL (Wang et al., 2024b) and achieves improved retrieval performance. Moving beyond retrieval, VisRAG (Yu et al., 2025) integrates MLLMs into the full RAG pipeline. Instead of extracting text, it embeds document images directly for retrieval and incorporates them into the generation stage, allowing the model to jointly reason over visual and textual content.

The above methods excel in text-to-image retrieval but fail to solve tasks involving a mixture of single-modality (e.g., text-to-text), cross-modality (e.g., text-to-image), and fused-modality (text+image-to-text+image) retrieval. GME (Zhang et al., 2025) tackles this by introducing a unified embedding model that encodes diverse modality combinations and enables flexible retrieval within a shared representation space.

While these approaches significantly enhance document understanding, they neglect the long-range corpus-level structure, which is essential for

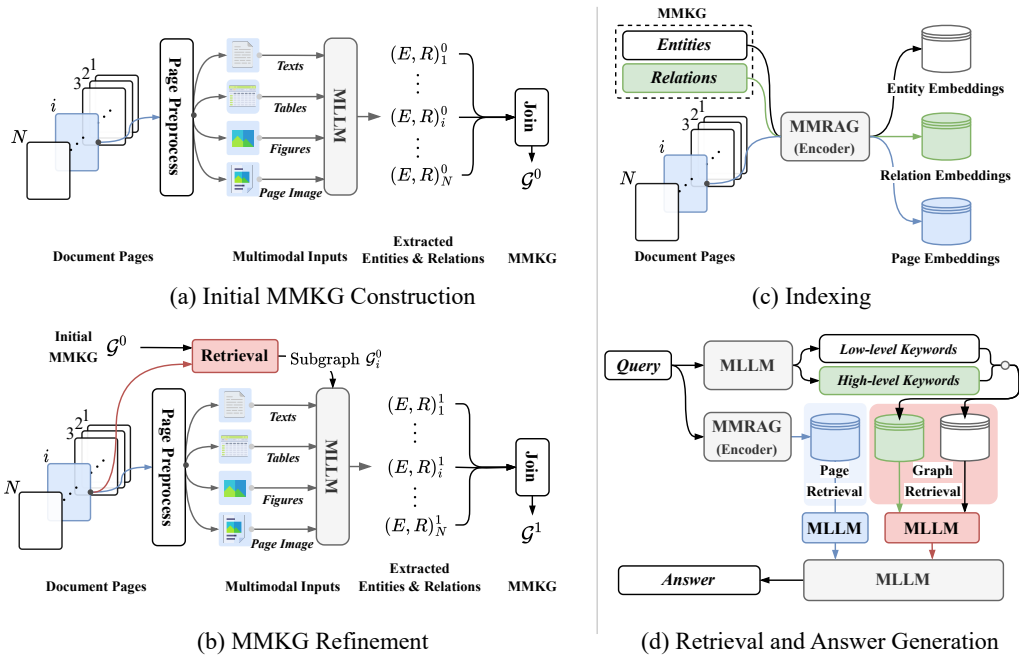


Figure 1: Overview of our MegaRAG for MMKG construction and MMKG-augmented generation. (a) Initial MMKG construction: an MLLM extracts page-level entities/relations  $(E, R)_i^0$  in parallel and merges them into  $\mathcal{G}^0$ . (b) Refinement: each page retrieves a subgraph  $\mathcal{G}_i^0$  to refine the graph, producing  $\mathcal{G}^1$ . (c) Indexing:  $\mathcal{G}^1$  and page images are encoded into dense entity, relation, and page embeddings. (d) Retrieval & generation: low-/high-level keywords retrieve relevant subgraphs/pages, followed by two-stage answer generation.

handling complex, multi-hop QA (Tanaka et al., 2023; Yang et al., 2018).

**RAG with Knowledge Graph.** Knowledge-augmented generation (Procko and Ochoa, 2024) leverages KGs to provide structured, factual context for LLMs. Within this line of research, SubgraphRAG (Li et al., 2025) enhances efficiency through lightweight scoring mechanisms for subgraph retrieval, while G-Retriever (He et al., 2024) frames subgraph selection as a Steiner Tree optimization problem to support large-scale textual graphs. Gao et al. (Gao et al., 2022) employ a learning-to-rank approach to improve retrieval from KGs. While these methods advance graph-based retrieval, they depend on manually constructed KGs, which are costly to build and require substantial domain expertise. Moreover, static KGs are inherently limited in addressing queries that require corpus-level reasoning beyond fixed KGs.

To address this limitation, GraphRAG (Edge et al., 2024) proposes building KGs directly from raw text using LLMs, followed by a hierarchical community detection algorithm (Traag et al., 2019) to cluster semantically related nodes. During inference, it prompts the LLM to generate intermediate answers for each community summary, scores them by confidence, and aggregates the top responses

into a final answer. Although this enables corpus-level reasoning, it incurs high computational cost due to repeated LLM queries over many community summaries. To improve efficiency, LightRAG (Guo et al., 2025) introduces a two-stage retrieval process: it first extracts local and global keywords from the query, then retrieves relevant nodes and their surrounding subgraphs using dense retrieval. This design reduces the need for repeated LLM inference and significantly improves scalability. which introduces a hybrid RAG framework that alternates between naive and graph-based retrieval. TOG-2 (Ma et al., 2025) introduces a hybrid RAG method that alternates between dense retrieval and graph reasoning. However, these approaches rely on manually curated KGs, which are costly to construct and limited in coverage.

However, these KG-augmented RAGs rely solely on textual KGs, limiting their ability to handle multimodal content such as images. To overcome this limitation, multimodal knowledge graphs (MMKGs) (Liu et al., 2019; Zhang et al., 2023) enrich KGs by associating entities with aligned visual (e.g., images), numeric (e.g., dates, measurements), and textual descriptions. A representative benchmark (Liu et al., 2019) introduces MMKGs that were constructed by linking overlapping enti-

ties via *sameAS* relations and annotating them with web-crawled images and numeric literals. MMKGs have demonstrated utility across tasks, including KG completion (Mousselly-Sergieh et al., 2018; Xie et al., 2017), recommendation systems (Sun et al., 2020), and image captioning (Zhao and Wu, 2023).

More recently, MMKGs have been integrated into RAG pipelines to support multimodal QA with LLMs. For instance, MR-MKG (Lee et al., 2024) utilizes manually constructed MMKGs that encode visual and factual knowledge, enabling LLMs to reason over structured multimodal inputs. Although effective, this approach depends on manually built, domain-specific MMKGs that are costly to scale. Query-driven MMKG (Bu et al., 2025) takes a step toward automatic MMKG construction by dynamically building query-specific local graphs for online multimodal reasoning. Nevertheless, such local graph construction is mainly designed for short-form, query-centered settings and does not explicitly model document-level structure or cross-page dependencies in long-form documents. Therefore, building scalable MMKGs that can be constructed automatically and support open-domain multimodal RAG over long, visually rich documents remains a key challenge.

### 3 Methodology

In this section, we present MegaRAG, covering the iterative construction process of MMKG, graph indexing and retrieval mechanisms, and the answer generation pipeline.

#### 3.1 MMKG Construction

We define our MMKG as  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ , where  $\mathcal{V}$  is the set of nodes representing entities, and  $\mathcal{E}$  is the set of edges denoting relations between entities. Given a document consisting of  $N$  pages, we extract four types of content from each page  $i$ : text content  $T_i$ , figure images  $F_i$ , table images  $B_i$ , and the full-page rendered image  $I_i$  (which captures the layout of the page). These elements are obtained using a document analysis tool. We define the input for page  $i$  as  $P_i = \{T_i, F_i, B_i, I_i\}$ , which serves as input to our graph construction pipeline.

**Initial Graph Construction.** As illustrated in Figure 1(a), the initial stage involves extracting entities and relations from each page in parallel using a graph generation function  $G(\cdot)$ , which leverages an MLLM guided by a task-specific prompt. The

prompt specifies the extraction goals, provides reasoning instructions, and enforces a constrained output format to ensure consistency across pages. In our implementation, GPT-4o-mini serves as the MLLM for the MMKG construction.

Given a multimodal input  $P_i$ , the graph generation function produces a set of page-level entities and relations  $(E, R)_i^0 = G(P_i)$ , extracted from both textual and visual content. The MLLM is guided to identify multiple entities within the text and to treat each figure or table as a single, standalone entity. For instance, a bar chart titled “Monthly Website Visitors” may be recognized as an entity and connected to surrounding text discussing user engagement trends. Decorative or non-informative visuals, such as background patterns or logos, are ignored. The full-page image  $I_i$  is used solely to support spatial reasoning and does not generate entity nodes. Each extracted entity includes a name, a predefined type (e.g., person, organization), and a description. Relations are defined by a source and target entity, a description, and a set of representative keywords.

After generating the set of page-level entities and relations (denoted as  $\{(E, R)_i^0\}_{i=1}^N$ ), we merge them into a unified MMKG  $\mathcal{G}^0$ . This involves consolidating entity nodes with the same name and merging relation edges with matching source, target, and relation types. During this process, different descriptions associated with the same entity or relation are aggregated to form a richer, more comprehensive representation. Similarly, keywords from multiple occurrences are accumulated.

**Graph Refinement and Enrichment.** The initial MMKG  $\mathcal{G}^0$  is often incomplete, as many cross-modal entities and relationships may be overlooked during the first-pass extraction. To bridge the gaps, we introduce a refinement stage that enhances graph  $\mathcal{G}^1$ , leveraging both the original multimodal inputs and the preliminary knowledge encoded in  $\mathcal{G}^0$ . The process is illustrated in Figure 1(b).

To efficiently refine MMKG under the MLLM’s limited context window, we focus on constructing lightweight, page-specific subgraphs rather than processing the entire graph. For each page  $i$ , we extract a context-specific subgraph  $\mathcal{G}_i^0$  from  $\mathcal{G}^0$ . In practice, we reuse entity names and relation keywords from the previously extracted page-level output  $(E, R)_i^0$  to retrieve relevant content in  $\mathcal{G}^0$ , reducing redundancy and simplifying subgraph construction. These entity names and relation keywords are encoded into semantic embeddings and

efficiently matched against dense vector representations of entities and relations built from initial MMKG. To enrich the local context, the selected nodes and edges are further expanded by including their one-hop neighbors, resulting in a compact yet informative subgraph. A detailed explanation of this graph indexing and retrieval process is provided in Section 3.2.

The refinement process is formalized as  $(E, R)_i^1 = R(P_i, \mathcal{G}_i^0)$ , where  $R(\cdot)$  is a refinement function that reuses the same MLLM from the initial stage, now guided by a KG-specific refinement prompt. Since the pages remain independent when extracting the entity relationship leveraging the subgraph, the benefit of parallelism is maintained for efficient graph construction. This function identifies missing knowledge in page  $P_i$  by examining the retrieved subgraph  $\mathcal{G}_i^0$ . Specifically, it detects entities mentioned in the input that are not yet present in the subgraph, as well as implicit relations between entities that are suggested by the content but missing from  $\mathcal{G}_i^0$ .

For example, consider a page where the text states “Electric vehicle sales increased significantly in 2023,” and a nearby figure titled “Annual Sales by Vehicle Type” presents a bar chart with a prominent “EV” bar (denoting Electric Vehicles). In the initial extraction, the text and the figure may be treated as independent entities. During refinement, the MLLM infers that the figure visually supports the textual claim and adds a relation such as *illustrates* or *supports* between the textual entity “Electric vehicle sales in 2023” and the visual entity “Annual Sales by Vehicle Type.”

These newly identified entities and relations are added to the refined set  $(E, R)_i^1$ . The updated page-level outputs  $\{(E, R)_i^1\}_{i=1}^N$  are then merged to form the enriched MMKG  $\mathcal{G}^1$ . Although we perform only a single refinement step, the process can be applied iteratively to further improve graph completeness. To balance effectiveness and efficiency, we adopt one round of refinement and provide the full prompt formats used for both the initial construction and refinement. More details can be found in Appendix B.

### 3.2 Indexing and Retrieval

We adopt a unified retrieval framework that integrates graph structure, represented by entities and relations, along with page images within a shared embedding space to enable seamless cross-modal retrieval. Specifically, we use GME (Zhang et al.,

2025), a multimodal encoder that jointly embeds textual and visual inputs. GME aligns all content types, including both textual and visual information, into a common vector space, supporting text-to-text and text-to-image retrieval through a unified representation.

**Indexing.** Our indexing process encompasses three content types, as illustrated in Figure 1(c): document page images, entities, and relations. Page images are directly encoded using GME without additional preprocessing. For each entity, we concatenate its name with its textual description to form a descriptive sentence, which is then embedded using GME. Relation embeddings are constructed similarly, by combining relation keywords, the names of the source and target entities, and a textual description. All embeddings are stored in separate dense vector stores by type.

**Graph Retrieval.** To retrieve relevant knowledge, we adopt a dual-level retrieval strategy (Guo et al., 2025) that targets both entities and relations. Given a user query, we first prompt the MLLM to extract two types of keywords: low-level keywords corresponding to specific entities, and high-level keywords that capture broader concepts. These keywords are then embedded by using the same GME model adopted during indexing. Both low-level and high-level keywords are combined into a single keyword list and used to query the entity vector store, retrieving the top- $k$  most relevant entities. In parallel, the top- $k$  most relevant relations, along with their associated source and target entities, are retrieved from the relation store. To further enrich the context, each retrieved entity is expanded by incorporating its one-hop neighbors from  $\mathcal{G}^1$ . The final set of entities and relations serves as input to the downstream reasoning module.

**Page Retrieval.** Complementary to graph retrieval, we also perform text-to-page(image) retrieval to capture fine-grained visual and layout cues that may be missed by symbolic representations alone. Given the same input query, we retrieve the top- $m$  relevant document pages by comparing text and image embeddings within the shared vector space.

### 3.3 MMKG-augmented Generation

When combined with visual content and MMKG in a single MLLM prompt, this integration can lead to modality bias. The model often disproportionately focuses on one modality, typically text, while underutilizing the other. To address this issue, we propose a two-stage answer generation approach

|                   | Agriculture |             |      | CS       |             |      | Legal    |             |      | Mix      |             |      |
|-------------------|-------------|-------------|------|----------|-------------|------|----------|-------------|------|----------|-------------|------|
|                   | NaiveRAG    | Ours        | Tie  | NaiveRAG | Ours        | Tie  | NaiveRAG | Ours        | Tie  | NaiveRAG | Ours        | Tie  |
| Comprehensiveness | 5.6         | <b>42.7</b> | 51.6 | 7.2      | <b>44.8</b> | 48.0 | 8.0      | <b>51.2</b> | 40.8 | 5.6      | <b>50.4</b> | 44.0 |
| Diversity         | 16.1        | <b>70.2</b> | 13.7 | 14.4     | <b>68.0</b> | 17.6 | 17.6     | <b>69.6</b> | 12.8 | 12.0     | <b>77.6</b> | 10.4 |
| Empowerment       | 12.9        | <b>66.9</b> | 20.2 | 22.4     | <b>47.2</b> | 30.4 | 20.8     | <b>61.6</b> | 17.6 | 20.0     | <b>62.4</b> | 17.6 |
| Overall           | 8.1         | <b>62.1</b> | 29.8 | 9.6      | <b>53.6</b> | 36.8 | 13.6     | <b>64.0</b> | 22.4 | 8.0      | <b>66.4</b> | 25.6 |
|                   | GraphRAG    | Ours        | Tie  | GraphRAG | Ours        | Tie  | GraphRAG | Ours        | Tie  | GraphRAG | Ours        | Tie  |
| Comprehensiveness | 5.6         | <b>64.8</b> | 29.6 | 4.0      | <b>68.0</b> | 28.0 | 11.2     | <b>60.8</b> | 28.0 | 4.0      | <b>59.2</b> | 36.8 |
| Diversity         | 14.4        | <b>76.8</b> | 8.8  | 14.4     | <b>72.0</b> | 13.6 | 12.8     | <b>75.2</b> | 12.0 | 22.4     | <b>59.2</b> | 18.4 |
| Empowerment       | 0.8         | <b>94.4</b> | 4.8  | 2.4      | <b>93.6</b> | 4.0  | 10.4     | <b>86.4</b> | 3.2  | 12.0     | <b>80.0</b> | 8.0  |
| Overall           | 5.6         | <b>82.4</b> | 12.0 | 4.0      | <b>79.2</b> | 16.8 | 11.2     | <b>80.0</b> | 8.8  | 7.2      | <b>70.4</b> | 22.4 |
|                   | LightRAG    | Ours        | Tie  | LightRAG | Ours        | Tie  | LightRAG | Ours        | Tie  | LightRAG | Ours        | Tie  |
| Comprehensiveness | 4.0         | <b>65.6</b> | 30.4 | 3.2      | <b>68.8</b> | 28.0 | 9.6      | <b>54.4</b> | 36.0 | 3.2      | <b>76.8</b> | 20.0 |
| Diversity         | 10.4        | <b>70.4</b> | 19.2 | 12.8     | <b>72.0</b> | 15.2 | 14.4     | <b>69.6</b> | 16.0 | 11.2     | <b>76.8</b> | 12.0 |
| Empowerment       | 4.8         | <b>76.0</b> | 19.2 | 10.4     | <b>75.2</b> | 14.4 | 12.0     | <b>73.6</b> | 14.4 | 4.0      | <b>80.8</b> | 15.2 |
| Overall           | 4.8         | <b>75.2</b> | 20.0 | 4.8      | <b>76.8</b> | 18.4 | 11.2     | <b>72.0</b> | 16.8 | 7.2      | <b>80.0</b> | 12.8 |

Table 1: Performance on the UltraDomain benchmark in terms of win rates (%).

that decouples the processing of textual and visual inputs. Given the retrieved subgraph and the relevant page images, the model first generates two intermediate responses in parallel: one based on the symbolic knowledge graph, and the other on the visual content. In the second stage, the MLLM synthesizes a final answer by integrating both intermediate outputs. Full prompt formats for each generation stage are provided in Appendix B.

## 4 Experiments

In this section, we outline the experimental setups and present the results for our MegaRAG method.

### 4.1 Datasets

**Global QA.** To evaluate the global (book-level) QA capabilities of MegaRAG, we use two document collections: a textual corpus and a multimodal dataset. For the **textual benchmark**, we adopt the Ultradomain (Qian et al., 2024) dataset, which contains 428 college-level textbooks across 18 disciplines; we focus on four representative subsets: Agriculture (2,017,886 tokens), Legal (5,081,069 tokens), Computer Science (2,306,535 tokens) and Mixed-Domain (619,009 tokens). Since no standard benchmark exists for multimodal global QA, we curate a new **multimodal benchmark** comprising four documents: World History (a world history textbook, 788 pages), Environmental Report (a corporate environmental report slide deck, 422 pages), DLCV (an English lecture slide deck, 1,984 pages), and GenAI (a Chinese lecture slide deck, 594 pages).

As these datasets lack manually labeled global questions, we adopt the question generation strat-

egy from GraphRAG (Edge et al., 2024) and LightRAG (Guo et al., 2025). For each dataset, we use the document outline as input and prompt an LLM to create five synthetic RAG users, each with a profile describing their background and information needs. Each user is assigned five tasks representing distinct information-seeking goals, and each task is used to generate five questions that require a comprehensive understanding of the full document. This process yields 125 global questions per dataset.

**Local QA.** To evaluate local (slide- or page-level) QA, we use two benchmarks: SlideVQA (Yang et al., 2018) and RealMMBench (Wasserman et al., 2025). SlideVQA includes over 52,000 slides and 14,500 questions covering complex reasoning and numerical understanding, but its scale makes full evaluation computationally expensive. Instead, we construct a subset of 2,000 slides, referred to as SlideVQA (2k). RealMMBench assesses retrieval in multimodal RAG settings using visual-rich, table-heavy, and rephrased queries. RealMMBench consists of four sub-datasets: FinReport (2,687 pages), FinSlides (2,280 pages), TechReport (1,674 pages), and TechSlides (1,963 pages). Additional details are provided in Appendix A.

### 4.2 Baselines and Evaluation Metrics

We compare MegaRAG with standard RAG baselines, including NaiveRAG and two recent KG-based methods, GraphRAG (Edge et al., 2024) and LightRAG (Guo et al., 2025) (details in Appendix C). For fair comparison, we evaluate all methods on both the multimodal benchmark and the textual benchmark.

|                   | DLCV     |             |      | World History |             |      | Environmental Report |             |      | GenAI    |              |      |
|-------------------|----------|-------------|------|---------------|-------------|------|----------------------|-------------|------|----------|--------------|------|
|                   | NaiveRAG | Ours        | Tie  | NaiveRAG      | Ours        | Tie  | NaiveRAG             | Ours        | Tie  | NaiveRAG | Ours         | Tie  |
| Comprehensiveness | 2.4      | <b>67.2</b> | 30.4 | 0.0           | <b>81.5</b> | 18.5 | 0.0                  | <b>72.8</b> | 27.2 | 0.0      | <b>95.2</b>  | 4.8  |
| Diversity         | 6.4      | <b>84.8</b> | 8.8  | 0.0           | <b>96.8</b> | 3.2  | 2.4                  | <b>92.0</b> | 5.6  | 0.0      | <b>98.4</b>  | 1.6  |
| Empowerment       | 11.2     | <b>66.4</b> | 22.4 | 3.2           | <b>82.3</b> | 14.5 | 12.8                 | <b>64.0</b> | 23.2 | 0.8      | <b>88.0</b>  | 11.2 |
| Overall           | 4.8      | <b>75.2</b> | 20.0 | 0.0           | <b>89.5</b> | 10.5 | 1.6                  | <b>80.0</b> | 18.4 | 0.0      | <b>98.4</b>  | 1.6  |
|                   | GraphRAG | Ours        | Tie  | GraphRAG      | Ours        | Tie  | GraphRAG             | Ours        | Tie  | GraphRAG | Ours         | Tie  |
| Comprehensiveness | 0.0      | <b>88.8</b> | 11.2 | 0.0           | <b>92.0</b> | 8.0  | 0.8                  | <b>68.8</b> | 30.4 | 0.0      | <b>92.8</b>  | 7.2  |
| Diversity         | 3.2      | <b>92.8</b> | 4.0  | 1.6           | <b>97.6</b> | 0.8  | 7.2                  | <b>81.6</b> | 11.2 | 0.0      | <b>97.6</b>  | 2.4  |
| Empowerment       | 1.6      | <b>95.2</b> | 3.2  | 0.0           | <b>96.0</b> | 4.0  | 1.6                  | <b>93.6</b> | 4.8  | 0.0      | <b>100.0</b> | 0.0  |
| Overall           | 0.0      | <b>92.8</b> | 7.2  | 0.0           | <b>93.6</b> | 6.4  | 0.8                  | <b>84.8</b> | 14.4 | 0.0      | <b>99.2</b>  | 0.8  |
|                   | LightRAG | Ours        | Tie  | LightRAG      | Ours        | Tie  | LightRAG             | Ours        | Tie  | LightRAG | Ours         | Tie  |
| Comprehensiveness | 0.0      | <b>78.4</b> | 21.6 | 0.0           | <b>89.6</b> | 10.4 | 0.0                  | <b>80.8</b> | 19.2 | 0.0      | <b>92.0</b>  | 8.0  |
| Diversity         | 3.2      | <b>90.4</b> | 6.4  | 0.0           | <b>95.2</b> | 4.8  | 1.6                  | <b>92.0</b> | 6.4  | 1.6      | <b>92.8</b>  | 5.6  |
| Empowerment       | 11.2     | <b>74.4</b> | 14.4 | 3.2           | <b>86.4</b> | 10.4 | 4.8                  | <b>79.2</b> | 16.0 | 1.6      | <b>91.2</b>  | 7.2  |
| Overall           | 0.8      | <b>84.8</b> | 14.4 | 0.0           | <b>90.4</b> | 9.6  | 0.0                  | <b>90.4</b> | 9.6  | 0.0      | <b>94.4</b>  | 5.6  |

Table 2: Performance across four multimodal datasets in terms of win rates (%).

| Method         | SlideVQA (2k) | RealMMBench  |              |              |              |
|----------------|---------------|--------------|--------------|--------------|--------------|
|                |               | FinReport    | FinSlides    | TechReport   | TechSlides   |
| NaiveRAG       | 11.34         | 29.66        | 14.64        | 36.63        | 32.94        |
| GraphRAG (L)   | 6.80          | 24.50        | 11.98        | 29.60        | 26.81        |
| GraphRAG (G)   | 5.22          | 10.08        | 3.04         | 15.07        | 16.03        |
| LightRAG       | 27.66         | 31.30        | 13.02        | 42.74        | 31.39        |
| <b>MegaRAG</b> | <b>64.85</b>  | <b>39.51</b> | <b>58.37</b> | <b>51.51</b> | <b>60.86</b> |

Table 3: Performance on SlideVQA (2k) and RealMM-Bench datasets in terms of Accuracy (%). GraphRAG (L) and GraphRAG (G) denote its local and global search modes.

**Global QA.** Since global (book-level) questions lack gold answers, we adopt the pairwise LLM-judge protocol from GraphRAG/LightRAG (Edge et al., 2024; Guo et al., 2025). Responses are evaluated by win rate (ties included) on four dimensions: Comprehensiveness, Diversity, Empowerment, and Overall (Guo et al., 2025).

**Local QA.** For local (page/slide-level) QA with references, an LLM judges semantic equivalence between the generated and ground-truth answers, and we report accuracy. Additional evaluation details are provided in Appendix C.

### 4.3 Implementation Details

To ensure fair comparisons, we standardize the LLM/MLLM and prompting across all methods. Response generation and global question generation use GPT-4o-mini, and evaluation uses GPT-4.1-mini. All baselines (NaiveRAG, GraphRAG, LightRAG) use text-embedding-3-small for textual embeddings. Textual documents are chunked into 1,200 tokens with a 100-token overlap; for GraphRAG and LightRAG, we set gleaning to 1. We use temperature 0 for all tasks.

For multimodal documents, we use MinerU (Wang et al., 2024a) to extract text,

figures, and tables. MegaRAG encodes multimodal embeddings with GME-Qwen2-VL-2B (Zhang et al., 2025). We retrieve top- $k$  entities/relations with  $k = 60$  for graph retrieval and top- $m$  pages with  $m = 6$  for page retrieval (Section 3.2). For text-only baselines, we keep only extracted text and apply the same textual pipeline.

### 4.4 Main Results

**Textual Global QA.** Table 1 shows the results on the UltraDomain benchmark consisting of purely textual documents. As can be seen, across all domains and evaluation dimensions, MegaRAG consistently outperforms the baselines, achieving average win rates of 59.0% for Comprehensiveness, 71.4% for Diversity, 74.8% for Empowerment, and 71.8% Overall.

A key contributor to this performance is MegaRAG’s graph refinement process. Unlike GraphRAG and LightRAG, which employ gleaning per page, a form of local subgraph refinement, MegaRAG doesn’t employ gleaning but constructs and refines a global knowledge graph that captures broader contextual relationships between documents. This approach enhances the expressiveness and coverage of the graph, leading to superior performance.

**Multimodal Global QA.** A main characteristic of our method is that it can build MMKGs for RAG. In this experiment, we evaluate our MegaRAG on global QA tasks over multimodal documents. As shown in Table 2, MegaRAG outperforms all baselines on four visually rich datasets: World History, Environmental Report, DLCV, and GenAI. It achieves average win rates of 83.3% for Com-

|                   | DLCV    |             |      | World History |             |      | Environmental Report |             |      | GenAI   |             |      |
|-------------------|---------|-------------|------|---------------|-------------|------|----------------------|-------------|------|---------|-------------|------|
|                   | VisRAG  | Ours        | Tie  | VisRAG        | Ours        | Tie  | VisRAG               | Ours        | Tie  | VisRAG  | Ours        | Tie  |
| Comprehensiveness | 0.8     | <b>35.2</b> | 64.0 | 4.0           | <b>31.2</b> | 64.8 | 4.0                  | <b>27.2</b> | 68.8 | 0.0     | <b>87.2</b> | 12.8 |
| Diversity         | 26.4    | <b>52.0</b> | 21.6 | 15.2          | <b>64.8</b> | 20.0 | 15.2                 | <b>60.8</b> | 24.0 | 2.4     | <b>94.4</b> | 3.2  |
| Empowerment       | 12.0    | <b>69.6</b> | 18.4 | 16.0          | <b>56.8</b> | 27.2 | 32.8                 | <b>32.0</b> | 35.2 | 1.6     | <b>93.6</b> | 4.8  |
| Overall           | 8.0     | <b>53.6</b> | 38.4 | 4.0           | <b>56.0</b> | 40.0 | 9.6                  | <b>40.0</b> | 50.4 | 0.0     | <b>96.0</b> | 4.0  |
|                   | GME     | Ours        | Tie  | GME           | Ours        | Tie  | GME                  | Ours        | Tie  | GME     | Ours        | Tie  |
| Comprehensiveness | 6.4     | <b>49.6</b> | 44.0 | 0.0           | <b>72.0</b> | 28.0 | 2.4                  | <b>48.8</b> | 48.8 | 0.8     | <b>75.2</b> | 24.0 |
| Diversity         | 19.2    | <b>59.2</b> | 21.6 | 5.6           | <b>80.8</b> | 13.6 | 12.0                 | <b>75.2</b> | 12.8 | 4.8     | <b>86.4</b> | 8.8  |
| Empowerment       | 23.2    | <b>49.6</b> | 27.2 | 5.6           | <b>75.2</b> | 19.2 | 24.0                 | <b>50.4</b> | 25.6 | 6.4     | <b>72.0</b> | 21.6 |
| Overall           | 14.4    | <b>57.6</b> | 28.0 | 1.6           | <b>78.4</b> | 20.0 | 5.6                  | <b>64.0</b> | 30.4 | 0.8     | <b>86.4</b> | 12.8 |
|                   | ColQwen | Ours        | Tie  | ColQwen       | Ours        | Tie  | ColQwen              | Ours        | Tie  | ColQwen | Ours        | Tie  |
| Comprehensiveness | 2.4     | <b>39.5</b> | 58.1 | 1.6           | <b>28.8</b> | 69.6 | 2.4                  | <b>37.6</b> | 60.0 | 0.0     | <b>88.8</b> | 11.2 |
| Diversity         | 29.0    | <b>57.3</b> | 13.7 | 14.4          | <b>65.6</b> | 20.0 | 17.6                 | <b>57.6</b> | 24.8 | 1.6     | <b>94.4</b> | 4.0  |
| Empowerment       | 14.5    | <b>62.9</b> | 22.6 | 16.0          | <b>53.6</b> | 30.4 | 25.6                 | <b>44.8</b> | 29.6 | 1.6     | <b>90.4</b> | 8.0  |
| Overall           | 6.5     | <b>50.8</b> | 42.7 | 3.2           | <b>52.0</b> | 44.8 | 9.6                  | <b>50.4</b> | 40.0 | 0.0     | <b>94.4</b> | 5.6  |

Table 4: Pairwise LLM-judge win rates (%) of MegaRAG against MMRAG baselines on four multimodal datasets.

prehensiveness, 92.7% for Diversity, 84.7% for Empowerment, and 89.5% Overall. The advantage is particularly evident on slide-based datasets such as DLCV and GenAI, where much of the core content is visual rather than textual. Compared with NaiveRAG and LightRAG, relying primarily on text, MegaRAG delivers stronger results across all evaluation dimensions. These gains stem from MegaRAG’s ability to build KGs that jointly encode textual information and visual cues.

Although all baselines in this comparison are text-only models, our ablation study, Section 4.6, further demonstrates that removing MMKG from MegaRAG leads to a substantial performance drop. Since our MegaRAG reduces to an MMRAG approach when its KG components are removed, this suggests that even vision-capable retrieval methods of MMRAG would struggle to match MegaRAG without multimodal global knowledge integration.

**Multimodal Local QA.** Table 3 shows the accuracy results on SlideVQA (2k) and the four RealMMBench subsets. Across all five test sets, MegaRAG performs more favorably. On SlideVQA (2k), which focuses on fine-grained slide-level reasoning, MegaRAG achieves 64.85% accuracy, higher than double the score of the strongest baseline. Similar trends are observed in RealMMBench. On FinSlides and TechSlides, which feature highly visual and table slide content, MegaRAG achieves 58.37% and 60.86%, outperforming the best baseline by 45 and 29 percents, respectively. Even in the more text-heavy FinReport and TechReport subsets, MegaRAG leads with 39.51% and 51.51%, surpassing LightRAG by 8 to 9%.

## 4.5 Comparison with Multimodal RAG Methods

Table 4 presents pairwise LLM-judge win rates between MegaRAG and three strong multimodal RAG methods, namely VisRAG, GME, and ColQwen, on four multimodal documents.

MegaRAG consistently outperforms all three baselines across all datasets and evaluation dimensions, often by a substantial margin. While VisRAG, GME, and ColQwen mainly rely on multimodal retrieval over raw page images or unified multimodal embeddings, they do not explicitly model corpus-level structure or cross-page relations. By contrast, MegaRAG constructs multimodal knowledge graphs automatically, enabling structured abstraction and cross-page, multi-hop reasoning. This design yields particularly strong improvements in *Comprehensiveness* and *Empowerment*, where capturing dispersed evidence are important.

## 4.6 Ablation Study

We study the impact of MegaRAG’s core components by ablating modules in three stages: MMKG construction, retrieval, and answer generation. Table 5 reports the ablation results. (A1) removes all visual inputs (figures, tables, and page images) during graph construction, yielding a text-only graph. (A2) disables MMKG retrieval and uses only page retrieval. (A3) replaces our two-stage generation with a single-pass setup that jointly conditions on the retrieved subgraph and page images.

**(A1) Text-only graph construction.** Removing visual cues consistently degrades performance. Without visual entities and cross-modal links, the graph

|                   | DLCV |              |      | World History |             |      |
|-------------------|------|--------------|------|---------------|-------------|------|
|                   | A1   | MegaRAG      | Tie  | A1            | MegaRAG     | Tie  |
| Comprehensiveness | 6.4  | <b>49.6</b>  | 44.0 | 0.0           | <b>72.0</b> | 28.0 |
| Diversity         | 19.2 | <b>59.2</b>  | 21.6 | 5.6           | <b>80.8</b> | 13.6 |
| Empowerment       | 23.2 | <b>49.6</b>  | 27.2 | 5.6           | <b>75.2</b> | 19.2 |
| Overall           | 14.4 | <b>57.6</b>  | 28.0 | 1.6           | <b>78.4</b> | 20.0 |
|                   | A2   | MegaRAG      | Tie  | A2            | MegaRAG     | Tie  |
| Comprehensiveness | 0.0  | <b>100.0</b> | 0.0  | 0.8           | <b>88.0</b> | 11.2 |
| Diversity         | 0.0  | <b>100.0</b> | 0.0  | 0.8           | <b>96.0</b> | 3.2  |
| Empowerment       | 0.0  | <b>100.0</b> | 0.0  | 1.6           | <b>86.4</b> | 12.0 |
| Overall           | 0.0  | <b>100.0</b> | 0.0  | 0.8           | <b>91.2</b> | 8.0  |
|                   | A3   | MegaRAG      | Tie  | A3            | MegaRAG     | Tie  |
| Comprehensiveness | 0.8  | <b>52.8</b>  | 46.4 | 0.0           | <b>67.2</b> | 32.8 |
| Diversity         | 12.0 | <b>72.8</b>  | 15.2 | 8.0           | <b>79.2</b> | 12.8 |
| Empowerment       | 5.6  | <b>70.4</b>  | 24.0 | 4.0           | <b>77.6</b> | 18.4 |
| Overall           | 1.6  | <b>61.6</b>  | 36.8 | 0.8           | <b>75.2</b> | 24.0 |

Table 5: Ablations on multimodal datasets (win rates, %). A1: text-only graph; A2: page-only retrieval; A3: single-pass (w/o two-stage) generation.

| Dataset (Pages)     | Input Tokens (Visual / Text) | Metric               | GraphRAG (Total) | MegaRAG |      |                      |
|---------------------|------------------------------|----------------------|------------------|---------|------|----------------------|
|                     |                              |                      |                  | Init.   | Ref. | Total (Init. + Ref.) |
| World History (788) | 17.0M / 0.4M                 | <i>Time (min)</i>    | 23.0             | 19.0    | 12.0 | 31.0                 |
|                     |                              | <i>KG Tokens (M)</i> | 1.2              | 22.9    | 15.3 | 38.2                 |
| Env. eport (422)    | 25.0M / 0.2M                 | <i>Time (min)</i>    | 10.0             | 8.5     | 5.5  | 14.0                 |
|                     |                              | <i>KG Tokens (M)</i> | 0.6              | 32.6    | 21.7 | 54.3                 |

Table 6: MMKG construction costs, where *Init.* and *Ref.* denote the Initial and Refinement stages.

loses critical grounding signals, which is particularly harmful for visually intensive documents. This confirms the necessity of incorporating visual elements when building the MMKG.

**(A2) Disable MMKG retrieval.** This variant yields the largest drop. Relying solely on page retrieval substantially weakens evidence aggregation and multi-hop reasoning, and MegaRAG strongly outperforms this setting across evaluation dimensions. The results indicate that structured retrieval over the MMKG is the primary driver of our gains. **(A3) Remove two-stage answer generation.** Using a single-pass generation causes moderate but consistent regressions. The largest decreases are typically observed in Diversity and Empowerment, suggesting that decoupling graph-based reasoning from visual grounding before fusion helps produce richer and more informative answers.

Overall, MMKG-based retrieval (A2) is the most critical component, while visual inputs in construction (A1) and the two-stage generation strategy (A3) provide complementary improvements.

#### 4.7 Computational Overhead

We analyze the computational overhead of MegaRAG in terms of both indexing cost and inference latency.

**Indexing Cost.** MegaRAG constructs the MMKG in a one-time offline process consisting of an *Initial*

*Stage* and a *Refinement Stage*. Table 6 reports the indexing costs on two representative datasets.

Importantly, graph construction is a one-time offline process; the resulting MMKG can be reused for subsequent queries without rebuilding the graph. Compared to text-only GraphRAG, MegaRAG incurs higher token usage and indexing latency (approximately  $1.4\times$ ) due to processing image content (17M to 25M visual tokens). This additional cost enables MegaRAG to extract visual entities (e.g., 473 to 538 figures) that text-only methods cannot capture and to recover global dependencies missed by single-pass approaches.

**Inference Latency.** MegaRAG adopts a two-stage generation strategy. We measure end-to-end latency on the *Environmental Report* dataset using a single RTX 3090 (24GB); the latency can be further reduced with more powerful GPUs. MMKG retrieval and page retrieval run concurrently ( $\sim 1.0s$  in total). In Stage 1, the MLLM generates answers from the retrieved subgraph ( $\sim 20.5s$ ) and from page images ( $\sim 26s$ ) in parallel. In Stage 2, a fusion step integrates both outputs ( $\sim 16s$ ), resulting in a total latency of  $\sim 42s$  per question.

For comparison, GME, a strong MMRAG method, completes retrieval ( $\sim 0.4s$ ) and generation ( $\sim 26s$ ), resulting in a total latency of  $\sim 26.4s$ . Although MegaRAG introduces an additional  $\sim 15.6s$  latency, it produces MMKG-augmented answers that combine textual and visual evidence for stronger cross-page and multi-hop reasoning.

## 5 Conclusion

In this paper, we introduced MegaRAG, a novel KG-based RAG method that leverages MLLMs to automatically construct MMKGs. MegaRAG improves MLLMs’ capabilities over complex, long-form documents by combining textual and visual information into a unified graph representation and refining it through iterative updates. MegaRAG needs no fine-tuning and is easy to use. To reduce modality bias, we adopt a two-stage answer generation process that separately reasons over textual and visual evidence before integrating the results, enabling more comprehensive and balanced responses. Through evaluations on both global and local QA tasks across textual and multimodal datasets, MegaRAG consistently outperforms other competitive RAG approaches. Our work highlights a promising new direction for scalable and interpretable multimodal reasoning in RAG systems.

## Limitations

Our experiments are primarily conducted on knowledge graphs constructed within a single book or report, and question answering across multiple books or reports remains limited. In addition, due to the high computational and preprocessing costs associated with figures and images, the scale of our multimodal dataset is still relatively limited.

Another limitation lies in our current treatment of visual content: each figure is represented as a single entity, which may overlook finer-grained elements or objects that are individually meaningful and worth representing as distinct entities. Moreover, since the MMKG is automatically constructed by MLLMs, hallucinations in entity or relation extraction may propagate to downstream reasoning. Understanding such hallucination behaviors and developing robust verification mechanisms are important directions for future work.

## Acknowledgements

This work was supported in part by NSTC under Grants NSTC 115-2634-F-002-012 and 112-2221-E-002-132-MY3, NTU under Grant 114L900902, and the Esun-NTU collaboration project under Grant 114HZA3S002. We thank the National Center for High Performance Computing (NCHC) of NARLabs in Taiwan for providing computational and storage resources.

## Ethical Considerations

All data used in this work are publicly available documents. No human subjects or user-generated data are involved, and we do not collect or infer personally identifying information beyond what is already contained in the original sources. Our approach relies on LLMs, which may occasionally produce incorrect or biased text. In our setting, LLM outputs are primarily used to assist knowledge graph construction, while final answers are grounded in retrieved subgraphs and source pages and can be verified against the underlying documents, which helps limit potential errors.

## References

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2024. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In *ICLR*.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibao Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, and 1 others. 2025. Qwen2. 5-v1 technical report. *arXiv preprint arXiv:2502.13923*.

Lucas Beyer, Andreas Steiner, André Susano Pinto, Alexander Kolesnikov, Xiao Wang, Daniel Salz, Maxim Neumann, Ibrahim Alabdulmohsin, Michael Tschannen, Emanuele Bugliarello, and 1 others. 2024. Paligemma: A versatile 3b vlm for transfer. *arXiv preprint arXiv:2407.07726*.

Chenyang Bu, Guojie Chang, Zihao Chen, CunYuan Dang, Zhize Wu, Yi He, and Xindong Wu. 2025. Query-driven multimodal graphrag: Dynamic local knowledge graph construction for online reasoning. In *Findings of ACL*.

Chi-Min Chan, Chunpu Xu, Ruibin Yuan, Hongyin Luo, Wei Xue, Yike Guo, and Jie Fu. 2024. RQ-RAG: Learning to refine queries for retrieval augmented generation. In *COLM*.

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

Manuel Faysse, Hugues Sibille, Tony Wu, Bilel Omrani, Gautier Viaud, Céline Hudelot, and Pierre Colombo. 2025. Colpali: Efficient document retrieval with vision language models. In *ICLR*.

Hanning Gao, Lingfei Wu, Po Hu, Zhihua Wei, Fangli Xu, and Bo Long. 2022. Graph-augmented learning to rank for querying large-scale knowledge graph. In *AAACL*.

Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. 2023. Precise zero-shot dense retrieval without relevance labels. In *ACL*.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Zirui Guo, Lianghao Xia, Yanhua Yu, Tu Ao, and Chao Huang. 2025. Lightrag: Simple and fast retrieval-augmented generation. In *Findings of EMNLP*.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval augmented language model pre-training. In *ICLR*.

Xiaoxin He, Yijun Tian, Yifei Sun, Nitesh Chawla, Thomas Laurent, Yann LeCun, Xavier Bresson, and Bryan Hooi. 2024. G-retriever: Retrieval-augmented generation for textual graph understanding and question answering. In *NeurIPS*.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, and 1

- others. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.
- Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick SH Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *EMNLP*.
- Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *SIGIR*.
- Junlin Lee, Yequan Wang, Jing Li, and Min Zhang. 2024. Multimodal reasoning with multimodal knowledge graph. In *ACL*.
- 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. In *NeurIPS*.
- Mufei Li, Siqi Miao, and Pan Li. 2025. Simple is effective: The roles of graphs and large language models in knowledge-graph-based retrieval-augmented generation. In *ICLR*.
- Ye Liu, Hui Li, Alberto Garcia-Duran, Mathias Niepert, Daniel Onoro-Rubio, and David S. Rosenblum. 2019. Mmkg: Multi-modal knowledge graphs. In *ESWC*.
- Shengjie Ma, Chengjin Xu, Xuhui Jiang, Muzhi Li, Huaren Qu, Cehao Yang, Jiaxin Mao, and Jian Guo. 2025. Think-on-graph 2.0: Deep and faithful large language model reasoning with knowledge-guided retrieval augmented generation. In *ICLR*.
- Xueguang Ma, Sheng-Chieh Lin, Minghan Li, Wenhui Chen, and Jimmy Lin. 2024. Unifying multimodal retrieval via document screenshot embedding. In *EMNLP*.
- Hatem Mousselly-Sergieh, Teresa Botschen, Iryna Gurevych, and Stefan Roth. 2018. A multimodal translation-based approach for knowledge graph representation learning. In *SEM*.
- Tyler Thomas Procko and Omar Ochoa. 2024. Graph retrieval-augmented generation for large language models: A survey. In *AIxSET*.
- Hongjin Qian, Peitian Zhang, Zheng Liu, Kelong Mao, and Zhicheng Dou. 2024. Memorag: Moving towards next-gen rag via memory-inspired knowledge discovery. *arXiv preprint arXiv:2409.05591*.
- Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: Bm25 and beyond. *Found. Trends Inf. Retr.*
- G. Salton, A. Wong, and C. S. Yang. 1975. A vector space model for automatic indexing. *ACM*.
- Keshav Santhanam, Omar Khattab, Jon Saad-Falcon, Christopher Potts, and Matei Zaharia. 2022. Colbertv2: Effective and efficient retrieval via lightweight late interaction. In *ACL*.
- Rui Sun, Xuezhi Cao, Yan Zhao, Junchen Wan, Kun Zhou, Fuzheng Zhang, Zhongyuan Wang, and Kai Zheng. 2020. Multi-modal knowledge graphs for recommender systems. In *CIKM*.
- Ryota Tanaka, Kyosuke Nishida, Kosuke Nishida, Taku Hasegawa, Itsumi Saito, and Kuniko Saito. 2023. Slidevqa: A dataset for document visual question answering on multiple images. In *AAAI*.
- Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, and 1 others. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.
- Vincent A Traag, Ludo Waltman, and Nees Jan Van Eck. 2019. From louvain to leiden: guaranteeing well-connected communities. *Scientific reports*.
- Bin Wang, Chao Xu, Xiaomeng Zhao, Linke Ouyang, Fan Wu, Zhiyuan Zhao, Rui Xu, Kaiwen Liu, Yuan Qu, Fukai Shang, and 1 others. 2024a. Mineru: An open-source solution for precise document content extraction. *arXiv preprint arXiv:2409.18839*.
- Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, and 1 others. 2024b. Qwen2-vl: Enhancing vision-language model’s perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*.
- Navve Wasserman, Roi Pony, Oshri Naparstek, Adi Raz Goldfarb, Eli Schwartz, Udi Barzelay, and Leonid Karlinsky. 2025. Real-mm-rag: A real-world multimodal retrieval benchmark. In *ACL*.
- Ruobing Xie, Zhiyuan Liu, Huanbo Luan, and Maosong Sun. 2017. Image-embodied knowledge representation learning. In *IJCAI*.
- 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 *EMNLP*.
- Shi Yu, Chaoyue Tang, Bokai Xu, Junbo Cui, Junhao Ran, Yukun Yan, Zhenghao Liu, Shuo Wang, Xu Han, Zhiyuan Liu, and 1 others. 2025. Vis-rag: Vision-based retrieval-augmented generation on multi-modality documents. In *ICLR*.
- Ningyu Zhang, Lei Li, Xiang Chen, Xiaozhuan Liang, Shumin Deng, and Huajun Chen. 2023. Multimodal analogical reasoning over knowledge graphs. In *ICLR*.

Xin Zhang, Yanzhao Zhang, Wen Xie, Mingxin Li, Ziqi Dai, Dingkun Long, Pengjun Xie, Meishan Zhang, Wenjie Li, and Min Zhang. 2025. Gme: Improving universal multimodal retrieval by multimodal llms. In *CVPR*.

Wentian Zhao and Xinxiao Wu. 2023. Boosting entity-aware image captioning with multi-modal knowledge graph. *IEEE Transactions on Multimedia*.

## Appendix

In the Appendix, we present the Datasets, Implementation Details, and Baselines & Evaluations in Appendices A, B, and C, respectively.

| Dataset                     | Documents | Pages | Figures | Tables | Text Tokens |
|-----------------------------|-----------|-------|---------|--------|-------------|
| <b>Ultradomain</b>          |           |       |         |        |             |
| Agriculture                 | 12        | -     | -       | -      | 2,017,886   |
| Computer Science (CS)       | 10        | -     | -       | -      | 2,306,535   |
| Legal                       | 94        | -     | -       | -      | 5,081,069   |
| Mix                         | 61        | -     | -       | -      | 619,009     |
| <b>Multimodal Documents</b> |           |       |         |        |             |
| DLCV                        | 18        | 1,984 | 2,018   | 75     | 136,032     |
| Environmental Report        | 5         | 422   | 416     | 122    | 229,014     |
| GenAI                       | 20        | 594   | 686     | 33     | 55,913      |
| World History               | 1         | 788   | 468     | 5      | 441,764     |
| <b>SlideVQA</b>             |           |       |         |        |             |
| SlideVQA (2k)               | 100       | 2,000 | 1,581   | 139    | 119,776     |
| <b>RealMMBench</b>          |           |       |         |        |             |
| FinReport                   | 19        | 2,687 | 411     | 2,963  | 1,583,640   |
| FinSlides                   | 65        | 2,280 | 730     | 1,842  | 123,891     |
| TechReport                  | 17        | 1,674 | 928     | 337    | 535,415     |
| TechSlides                  | 62        | 1,963 | 2,254   | 119    | 138,766     |

Table 7: Datasets statistics used in our experiments. The Ultradomain benchmark is purely textual documents; hence, entries for pages, figures, and tables are marked with a dash (–) to indicate not applicable.

## A Datasets

We provide an overview of the datasets in our experiments and dataset statistics in Table 7.

### A.1 Dataset Statistics

The Ultradomain benchmark (Qian et al., 2024) comprises 428 college-level textbooks spanning 18 academic disciplines. For this study, we focus on the four representative subsets:

**Agriculture dataset.** Consisting of 12 textbooks and 2.02 million text tokens, this subset covers topics such as beekeeping, hive management, crop cultivation, and disease prevention in modern agriculture. **Computer Science (CS) dataset.** Containing 10 textbooks and 2.31 million tokens, the CS subset emphasizes key topics in algorithms, data structures, artificial intelligence, machine learning, and real-time data analytics. **Legal dataset.** Comprising 94 textbooks and totaling 5.08 million tokens. It spans a wide range of legal topics, including corporate restructuring, regulatory compliance, financial governance, and case law analysis. **Mixed-Domain (Mix) dataset.** A diverse collection of 61 textbooks totaling 620,000 tokens. This subset includes literary works, philosophical essays, biographies, and cultural-historical studies.

The global QA multimodal datasets are derived from publicly available documents:

**Deep Learning for Computer Vision (DLCV) dataset.** Comprising 18 slide decks, this dataset<sup>1</sup> includes 1,984 pages, 2,018 figures, 75 tables, and 136,000 tokens. The content is drawn from a deep learning and computer vision course, covering image classification, object detection, and societal impacts of AI. **Environmental Report dataset.** Consisting of 5 corporate sustainability reports, this dataset includes 422 pages, 416 figures, 122 tables, and 229,000 tokens. It documents environmental strategies from Google<sup>2</sup>, Apple<sup>3</sup>, Microsoft<sup>4</sup>, Meta<sup>5</sup>, and NVIDIA<sup>6</sup> (FY24 Sustainability Report), including goals for carbon reduction and renewable energy. **Generative AI (GenAI) dataset.** This dataset comprises 20 lecture slide decks<sup>7</sup> (in Chinese), with 594 pages, 686 figures, 33 tables, and 55,900 tokens. Topics focus on generative AI, including transformer architectures, generation techniques, cross-modal applications, and ethical considerations in large-scale AI systems. **World History dataset.** A textbook<sup>8</sup> comprising 788 pages, 468 figures, 5 tables, and 442,000 tokens. It traces global developments from prehistory to 1500 CE, covering early civilizations, empires, religious movements, and intercultural exchanges. **SlideVQA (2k).** SlideVQA (Tanaka et al., 2023) includes over 52,000 slides and 14,500 questions covering complex reasoning and numerical understanding, but its scale makes full evaluation computationally expensive. Instead, we construct a subset of SlideVQA, which consists of 2,000 educational slides, featuring 1,581 figures, 139 tables, and 120,000 tokens.

The RealMMBench (Wasserman et al., 2025) is designed to evaluate retrieval performance in realistic multi-modal RAG scenarios, and contains four subsets:

**FinReport.** This subset includes 19 long-form table-heavy financial reports from IBM, totaling 2,687 pages, 411 figures, 2,963 tables, and 1.58 mil-

<sup>1</sup><https://cs231n.stanford.edu/slides/2024/>

<sup>2</sup><https://sustainability.google/reports/google-2024-environmental-report/>

<sup>3</sup>[https://www.apple.com/environment/pdf/Apple\\_Environmental\\_Progress\\_Report\\_2024.pdf](https://www.apple.com/environment/pdf/Apple_Environmental_Progress_Report_2024.pdf)

<sup>4</sup><https://www.microsoft.com/en-us/corporate-responsibility/sustainability/report>

<sup>5</sup><https://sustainability.atmeta.com/2024-sustainability-report/>

<sup>6</sup><https://www.nvidia.com/en-us/sustainability/>

<sup>7</sup><https://speech.ee.ntu.edu.tw/~hylee/genai/2024-spring.php>

<sup>8</sup><https://open.umn.edu/opentextbooks/textbooks/1418>

lion tokens. **FinSlides.** Comprising 65 corporate financial slide decks, this subset spans 2,280 pages, 730 figures, 1,842 tables, and 124,000 tokens. It presents a more visual but still data-rich format for financial information, including quarterly earnings briefings, strategic outlooks, and KPI dashboards. **TechReport.** This collection includes 17 technical reports with 1,674 pages, 928 figures, 337 tables, and 535,000 tokens. Documents are sourced from specialized domains such as enterprise hardware and storage systems. **TechSlides.** Featuring 62 technical presentation slide decks, this subset comprises 1,963 pages, 2,254 figures, 119 tables, and 139,000 tokens. It has the highest figure density across RealMMBench, which conveys technical concepts through diagrams and flowcharts.

## A.2 Global Question Generation

To generate global questions, we utilize the prompt shown in Figure 5. This prompt guides the MLLM (GPT-4o-mini) to first identify representative user profiles and their associated tasks, then generate questions that require a comprehensive understanding of the dataset.

## B Implementation Details

### B.1 Prompts Used in MegaRAG

**MMKG construction.** For MMKG construction in Section 3.1, we use prompts to guide GPT-4o-mini in extracting structured knowledge from multimodal document inputs. The prompt used in the initial graph construction stage is shown in Figure 2. For graph refinement, we employ a separate prompt designed to identify missing or implicit connections. This prompt, illustrated in Figure 3.

**MMKG-augmented Answer Generation.** For MMKG-augmented answer generation (Section 3.3), we adopt a two-stage prompting strategy. In the first stage, GPT-4o-mini is guided to generate intermediate answers separately: one based on the visual page (Figure 4(a)) and another based on the retrieved subgraph (Figure 4(b)). In the second stage, a follow-up prompt combines these intermediate responses to produce the final answer (Figure 4(c)).

### B.2 Retrieval and Generation Details

MegaRAG leverages the General Multimodal Embedder (GME) (Zhang et al., 2025) to encode entities, relations, and page images within a unified embedding space. GME is built upon the Qwen2-

VL architecture, a MLLM capable of processing text, images, or combined text-image inputs. It supports a broad range of retrieval tasks, including single-modality retrieval (e.g., text-to-text, image-to-image), cross-modality retrieval (e.g., text-to-image, image-to-text), and fused-modality retrieval (e.g., text with image to text with image). To generate embeddings, GME uses the final hidden state of the last token as the representation of the input. GME’s strength lies in its flexibility and generalization capability, making it well-suited for MegaRAG, which requires seamless integration of both text-to-text and text-to-page (image) retrieval tasks.

**GME Encoding Time.** In our pipeline, the GME-Qwen2-VL-2B encoder is executed locally to process both text and image inputs. All encoding is performed on a single NVIDIA RTX 3090 GPU with 24GB of VRAM. Due to memory constraints, we limit GME to encoding two page images concurrently, with an average processing time of approximately 0.97 seconds per image.

During graph retrieval in the MMKG refinement stage, as described in Section 3.1, we retrieve the top 120 entities and relations from the initial MMKG and concatenate them into a single string (as illustrated in Figure 3, subgraph). We then truncate this string to a maximum of 32,000 tokens. The truncated string is then used to prompt the MLLM to identify missing entity-relation pairs that were not captured in the initial stage. We experimented with both larger and smaller retrieval sizes and found that retrieving 120 entities and relations provides the best balance between global coverage of the MMKG and input length constraints.

One-shot Exemplar

**Entity types:** [person, technology]  
 Primary Image Document text content:  
 "Alex clenched his jaw in frustration as Taylor asserted control. Jordan's drive for discovery clashed with Cruz's desire for order. Later, Taylor examined a device with reverence, hinting at its transformative power."  
**Additional Layout Detection Images:**  
 - Image 1: (An image file showing a handwritten note on a whiteboard)  
 - Image 2: (An image file showing a decorative background pattern with no meaningful information)  
**Output:**  
 ("entity", "Taylor", "person", "Taylor is portrayed with strong authority and later shows respect toward a powerful device.")  
 ("entity", "The Device", "technology", "The device is treated as a transformative object with great potential.")  
 ("relationship", "Taylor", "The Device", "Taylor's reverence for the device emphasizes its significance.", "technological importance", 8)  
 ("content\_keywords", "authority, technology, significance")

---

**Given a primary image document with its text content that is potentially relevant to this activity, along with additional images obtained from layout detection (if available), and a list of entity types, identify all entities of those types from the text content and from any additional images that contain meaningful content. Note that:**

- The first image is always the primary image document.
- The remaining 0 to many images are results from layout detection.
- For each additional image, analyze whether it contains meaningful content (e.g., tables, charts, images of important persons, events, etc.). In making this determination, also reference the primary image document and its text content to understand the context. If the additional image is meaningful, treat it as an entity by extracting its relevant details. If the image is merely decorative or irrelevant (e.g., decorative patterns, unrelated photos), then ignore it.
- The input images are provided by appending them directly after the text (with the primary image document guaranteed to be the first image).

Use {language} as output language.

Input Definition

---

**-Steps-**

1. Process the Input:
  - a. The primary image document and its text content.
  - b. Additional images from layout detection (if any), appended after the prompt.
2. Identify all entities from the text content and from any additional images that contain meaningful content. For each identified entity, extract the following information:
  - entity\_name: Name of the entity, using the same language as the input text (capitalize the name if it is in English).
  - entity\_type: One of the following types: {[entity\_types]}
  - entity\_description: A comprehensive description of the entity's attributes and activities.
 Format each entity as ("entity"<tuple\_delimiter>entity\_name<tuple\_delimiter>entity\_type<tuple\_delimiter>entity\_description<tuple\_delimiter>entity\_description)
  - \*For additional images that are deemed meaningful (for example, a table showing financial data, a chart representing trends, an image of an important person or event, or in the one of the following types: {[entity\_types]}), create an entity with an appropriate name and description indicating the content and significance of the image. When evaluating these images, also refer to the primary image document and its text content for context.
3. From the entities identified in step 2, identify all pairs of (source\_entity, target\_entity) that are clearly related to each other. For each pair, extract the following information:
  - source\_entity: Name of the source entity, as identified in step 2.
  - target\_entity: Name of the target entity, as identified in step 2.
  - relationship\_description: Explanation of why the source entity and the target entity are related.
  - relationship\_strength: A numeric score indicating the strength of the relationship between the source and target entities.
  - relationship\_keywords: One or more high-level keywords that summarize the overarching nature of the relationship, focusing on concepts or themes rather than specific details.
 Format each relationship as ("relationship"<tuple\_delimiter>source\_entity<tuple\_delimiter>target\_entity<tuple\_delimiter>relationship\_description<tuple\_delimiter>relationship\_keywords<tuple\_delimiter>relationship\_strength)
  - 4. Identify high-level keywords that summarize the main concepts, themes, or topics of the entire text and images. Format these as ("content\_keywords"<tuple\_delimiter>high\_level\_keywords)
5. Return the output in {language} as a single list of all the entities and relationships identified in steps 2 and 3. Use \*\*{record\_delimiter}\*\* as the list delimiter.
6. When finished, output {completion\_delimiter}

Chain-of-Thought

---

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

In-context One-shot Exemplar

---

#####  
**-Real Data-**  
 #####  
 Entity types: {entity\_types}  
 Primary Image Document text content: {input\_text}  
 Additional Layout Detection Images:  
 (The images are provided by appending them directly after this prompt, with the primary image document as the first image.)

Inputs


---

Text Content


Best Practices for Enterprise Gen AI Solutions  
**A proven, scalable platform | Vela cloud-native supercomputer**  
 Deployed on Vela Cloud-native supercomputer, IBM Cloud Full stack running across thousands of GPUs with OpenShift (each node with 8 x A100 GPUs) Covering entire life cycle of foundation model, from data preprocessing, training, inference and workbench Jobs requiring anywhere between single to hundreds of GPUs Support for priorities and pre-emption, improving utiliton and user experience ...

Entity Type

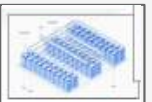
"person", "organization", "job\_title",  
 "concept\_or\_framework", "quote\_or\_statement",  
 "challenge\_or\_problem", "question\_or\_use\_case",  
 "technology\_investment\_area",  
 "business\_goal\_or\_value", "audience\_or\_stakeholder"




**Page Image**



**Image 1**



**Image 2**



**Image 3**

Figure 2: Prompt for extracting entities and relations during the initial construction of the MMKG.

**Entity\_types:** [person, technology]  
Primary Image Document text content:  
"Alex clenched his jaw in frustration as Taylor asserted control. Jordan's drive for discovery clashed with Cruz's desire for order. Later, Taylor examined a device with reverence, hinting at its transformative power."  
**Additional Layout Detection Images:**  
- Image 1: (An image file showing a handwritten note on a whiteboard)  
- Image 2: (An image file showing a decorative background pattern with no meaningful information)  
**Output:**  
("entity", "Taylor", "person", "Taylor is portrayed with strong authority and later shows respect toward a powerful device.")  
("entity", "The Device", "technology", "The device is treated as a transformative object with great potential.")  
("relationship", "Taylor", "The Device", "Taylor's reverence for the device emphasizes its significance.", "technological importance", 8)  
("content\_keywords", "authority, technology, significance")

**One-shot Exemplar**

---

Given a primary image document with its text content that is potentially relevant to this activity, along with additional images obtained from layout detection (if available), and a list of entity types, identify all entities of those types from the text content and from any additional images that contain meaningful content. Additionally, use the provided Knowledge Graph Data to enhance entity extraction by leveraging prior knowledge, ensuring that:

- Entities and relationships already present in the Knowledge Graph Data should **not be re-extracted** from the text content or images.
- If a new entity is found in the text content or images that is **not present in the Knowledge Graph Data**, it should be extracted.
- If an entity from the text content or images is related to an existing entity in the Knowledge Graph Data, establish a **new relationship** between them.
- If two existing entities from the Knowledge Graph Data have a **new relationship** within the given text content or images, this relationship should also be extracted.
- If a previously known entity appears in the current text content or image with **new descriptive attributes not found in the Knowledge Graph Data**, those descriptions should be added to the entity.
- If a new entity is mentioned multiple times across text or images with different complementary attributes, the extracted description should integrate all such information.

Note that:

- The first image is always the primary image document.
- The remaining 0 to many images are results from layout detection.

- For each additional image, analyze whether it contains meaningful content (e.g., tables, charts, images of important persons, events, etc.). In making this determination, also reference the primary image document, its text content, and the Knowledge Graph Data to understand the context. If the additional image is meaningful, treat it as an entity by extracting its relevant details. If the image is merely decorative or irrelevant (e.g., decorative patterns, unrelated photos), then ignore it.

- The input images are provided by appending them directly after the text (with the primary image document guaranteed to be the first image).

- Use {language} as the output language.

*Input Definition*

---

**Steps**

1. **Process the Input:**
  - a. The primary image document and its text content.
  - b. Additional images from layout detection (if any), appended after the prompt.
  - c. The Knowledge Graph Data, which provides structured relationships and prior knowledge that can help with entity identification.
2. **Identify all new entities** from the text content and additional images containing meaningful content.
  - **Do not extract entities that already exist in the Knowledge Graph Data.**
  - **If a new entity is found, extract the following:**
    - **entity\_name:** Name of the entity, using the same language as the input text (capitalize the name if it is in English).
    - **entity\_type:** One of the following types: {entity\_types}
    - **entity\_description:** A comprehensive description of the entity's attributes and activities. If found in multiple locations, integrate all details into one complete description.
  - **Format:** {entity}{tuple\_delimiter}<entity\_name>{tuple\_delimiter}<entity\_type>{tuple\_delimiter}<entity\_description>
3. **Identify relationships between entities**, ensuring that:
  - **If a new entity (from text content or images) is related to an entity already in the Knowledge Graph Data, establish a new relationship.**
  - **If two existing Knowledge Graph Data entities have a new relationship within this document, extract that relationship.**
  - **Format each relationship as:**
    - **source\_entity:** Name of the source entity, as identified in step 2 or the Knowledge Graph Data.
    - **target\_entity:** Name of the target entity, as identified in step 2 or the Knowledge Graph Data.
    - **relationship\_description:** Explanation of why the source entity and the target entity are related.
    - **relationship\_strength:** A numeric score indicating the strength of the relationship between the source and target entities.
    - **relationship\_keywords:** One or more high-level keywords that summarize the overarching nature of the relationship, focusing on concepts or themes rather than specific details.
  - **Format:** {relationship}{tuple\_delimiter}<source\_entity>{tuple\_delimiter}<target\_entity>{tuple\_delimiter}<relationship\_description>{tuple\_delimiter}<relationship\_keywords>{tuple\_delimiter}<relationship\_strength>
4. **Extract high-level content keywords** summarizing the main concepts, themes, or topics from the text and meaningful images, but excluding the Knowledge Graph Data.
  - **Format:** {content\_keywords}{tuple\_delimiter}<high\_level\_keywords>
5. **Return the output** in {language} as a single list of all the entities and relationships identified in steps 2 and 3. Use {record\_delimiter} as the list delimiter.
6. **When finished, output {completion\_delimiter}.**

*Chain-of-Thought*

---

#####  
-Examples-  
#####  
{examples}  
#####  
-Real Data-  
#####

*In-context One-shot Exemplar*

---


Entity\_types: {entity\_types}  
Primary Image Document text content: {input\_text}  
Additional Layout Detection Images: (The images are provided by appending them directly after this prompt, with the primary image document as the first image).  
Knowledge Graph Data:  
{kg\_context}

*Inputs*

---

Best Practices for Enterprise Gen AI Solutions  
A proven, scalable platform | Vela cloud-native supercomputer  
Deployed on Vela Cloud-native supercomputers, IBM Cloud Full stack running across thousands of GPUs with OpenShift (each node with 8 x A100 GPUs) Covering entire life cycle of ...


**Text Content**



**Page Image**

("entity", "Vela Cloud-native Supercomputer", "platform", "A scalable platform deployed on IBM Cloud, supporting large-scale AI workloads.") ... ("relationship", "Vela Cloud-native Supercomputer", "IBM Cloud", "Vela is deployed on IBM Cloud, leveraging its infrastructure.", "deployment, cloud integration", 9) ...

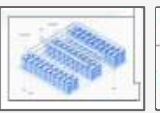
**Subgraph**




**Image 1**

"person", "organization", "job title", "concept\_or\_framework", "quote\_or\_statement", "challenge\_or\_problem", "question\_or\_use\_case", "technology\_investment\_area", "business\_goal\_or\_value", "audience\_or\_stakeholder"

**Entity Type**



**Image 2**



**Image 3**

Figure 3: Prompt for MMKG refinement stage.

You are a helpful assistant responding to user query about Document Images provided below.

**---Goal---**

Generate a concise response based on Document Images and follow Response Rules, considering both the conversation history and the current query. Summarize all information in the provided Document Images, and incorporating general knowledge relevant to the Document Images. Do not include information not provided by Document Images.

When handling content with timestamps:

1. Each piece of content has a "created\_at" timestamp indicating when we acquired this knowledge
2. When encountering conflicting information, consider both the content and the timestamp
3. Don't automatically prefer the most recent content - use judgment based on the context
4. For time-specific queries, prioritize temporal information in the content before considering creation timestamps

**---Response Rules---**

- Target format and length: Multiple Paragraphs
- Use markdown formatting with appropriate section headings
- Please respond in English.
- Ensure the response maintains continuity with the conversation history.
- If you don't know the answer, just say so.
- Do not include information not provided by the Document Images.

**(a) Page Image Intermediate Answer Generation**

You are a helpful assistant responding to user query about Knowledge Base provided below.

**---Goal---**

Generate a concise response based on Knowledge Base and follow Response Rules, considering both the conversation history and the current query. Summarize all information in the provided Knowledge Base, and incorporating general knowledge relevant to the Knowledge Base. Do not include information not provided by Knowledge Base.

When handling relationships with timestamps:

1. Each relationship has a "created\_at" timestamp indicating when we acquired this knowledge
2. When encountering conflicting relationships, consider both the semantic content and the timestamp
3. Don't automatically prefer the most recently created relationships - use judgment based on the context
4. For time-specific queries, prioritize temporal information in the content before considering creation timestamps

**---Knowledge Base---**  
{context\_data}

**---Response Rules---**

- Target format and length: Multiple Paragraphs
- Use markdown formatting with appropriate section headings
- Please respond in English.
- Ensure the response maintains continuity with the conversation history.
- If you don't know the answer, just say so.
- Do not make anything up. Do not include information not provided by the Knowledge Base.

**(b) Knowledge Graph Intermediate Answer Generation**

You are a professional assistant responsible for answering questions based on both a knowledge graph and visual information extracted from document images containing relevant textual and visual content (e.g., scanned pages, slides, charts, or forms).

You are provided with a user query and two independent answers:

1. An answer based on the knowledge graph.
2. An answer based on the document images.

Your task is to analyze the user's query and integrate the two provided answers into a single comprehensive response. Do not omit any relevant points from either source. When the answers conflict or provide complementary insights, use grounded reasoning to reconcile them. If the knowledge graph provides explicit facts, do not override them unless contradicted by strong visual evidence.

Please respond in English.

**---Query---**  
{query}

**---Input Answers---**

- **\*\*Answer from Knowledge Graph\*\*:**  
{kg\_answer}
- **\*\*Answer from Document Images\*\*:**  
{image\_answer}

**---Goal---**

Generate a concise response to the query that incorporates all relevant information from both Answers from the Knowledge Graph and the Document Images. If you don't know the answer, just say so. Do not make anything up or include information where the supporting evidence is not provided.

When handling information with timestamps:

1. Each piece of information (both relationships and content) has a "created\_at" timestamp indicating when we acquired this knowledge.
2. When encountering conflicting information, consider both the content/relationship and the timestamp.
3. Don't automatically prefer the most recent information - use judgment based on the context.
4. For time-specific queries, prioritize temporal information in the content before considering creation timestamps.

**---Response Rules---**

- Target format and length: Multiple Paragraphs
- Generate a final answer that integrates both inputs.
- Use markdown formatting with appropriate section headings.
- Organize answer in sections focusing on one main point or aspect of the answer
- List up to 5 most important reference sources at the end under a "References" section. Clearly indicate whether each source is from Knowledge Graph (KG) or Document Content (DC), using this format: [KG/DC] Source content.
- Ensure the response maintains continuity with the conversation history.
- If you don't know the answer, just say so. Do not make anything up.
- Do not include information not provided by the inputs.

**(c) Final Answer Generation**

Figure 4: Prompts for MMKG-augmented answer generation. (a) Generates an intermediate answer from the retrieved pages. (b) Generates an intermediate answer from the retrieved MMKG subgraph. (c) The final answer is produced by combining both intermediate responses.

## C Baselines and Evaluation

### C.1 Baselines

We evaluate MegaRAG against two widely used graph-based RAG baselines: GraphRAG and LightRAG, as well as a commonly adopted non-graph baseline, NaiveRAG. To ensure a fair comparison, we set the generation temperature to 0 across all models. Below, we provide a detailed overview of each method along with its specific settings for reference.

**NaiveRAG.** Serving as a standard baseline among RAG systems, NaiveRAG divides the input document into multiple text chunks, which are then encoded into a vector space using text embeddings. At query time, relevant chunks are retrieved based on the similarity between their embeddings and the query representation.

**GraphRAG.** GraphRAG begins by segmenting the input text into chunks and extracting entities and relationships to construct a graph. This graph is subsequently partitioned into communities at multiple levels. During retrieval, GraphRAG identifies entities mentioned in the query and synthesizes answers by referencing summaries of the corresponding communities. Compared to traditional RAG approaches, GraphRAG offers a more structured and high-level understanding of the document.

**LightRAG.** LightRAG is a variant of GraphRAG. It is designed to reduce computational overhead while enhancing retrieval quality through a dual-level retrieval mechanism. This design improves both efficiency and effectiveness, offering a better balance between performance and resource usage compared to GraphRAG.

### C.2 Evaluation

**Global QA.** To evaluate model performance on global (book-level) questions, where no gold-standard answers are available, we conduct pairwise comparative evaluations between MegaRAG and baseline models. Responses are assessed along three qualitative dimensions: **Comprehensiveness**, **Diversity**, and **Empowerment**, as well as an **overall** rating that reflects performance across all criteria.

Each evaluation instance presents a question alongside two competing answers, one from a baseline model and one from MegaRAG. We employ GPT-4.1-mini as the evaluator to compare the two responses, select a winner for each dimension, and provide brief justifications. Comprehensiveness

measures how thoroughly the answer addresses all aspects of the question. Diversity evaluates the richness and variety of perspectives presented. Empowerment assesses how effectively the answer enhances user understanding and supports informed decision-making. The full evaluation prompt used in this process is shown in Figure 6 (a).

**Local QA.** For local (slide- or page-level) QA, where reference answers are available, we use GPT-4.1-mini to assess answer correctness. Each instance includes a question, the model’s response, and the corresponding ground truth. The LLM judge evaluates whether the response is semantically consistent with the reference, regardless of surface phrasing. The output is a binary label (yes or no) accompanied by a brief explanation. Accuracy is calculated as the proportion of responses judged correct. The evaluation prompt is shown in Figure 6 (b).

### C.3 Analysis of Evaluation Biases

We further analyze two potential sources of bias in our LLM-based evaluation, namely *positional bias* and *single-model bias*.

**Positional Bias.** To mitigate positional bias, we conduct pairwise evaluation twice by swapping the order of the two candidate answers. If the judge selects the first answer in both rounds, we treat the outcome as a tie, since this behavior likely reflects a positional preference rather than a genuine content-based preference.

**Single-Model Bias.** To reduce dependence on a single judge model, we additionally use Gemini-3-Flash alongside GPT-4.1-mini. The results show that Gemini-3-Flash is more conservative, yielding a tie rate that is approximately 15 percentage points higher on average. Nevertheless, the relative performance rankings remain consistent across the two judges. Moreover, the inter-judge agreement between GPT-4.1-mini and Gemini-3-Flash on winning samples reaches a Cohen’s kappa of 0.72, suggesting that MegaRAG’s advantages are stable across evaluators.

Table 9 reports the pairwise win-rate comparison. Each entry is formatted as *Other Method Win / MegaRAG Win / Tie*, where the main number denotes the GPT-4.1-mini result and the value in parentheses denotes the Gemini-3-Flash result.

|                   | DLCV    |             |      | World History |             |     | Environmental Report |             |     | GenAI   |             |     |
|-------------------|---------|-------------|------|---------------|-------------|-----|----------------------|-------------|-----|---------|-------------|-----|
|                   | 4o-mini | MegaRAG     | Tie  | 4o-mini       | MegaRAG     | Tie | 4o-mini              | MegaRAG     | Tie | 4o-mini | MegaRAG     | Tie |
| Comprehensiveness | 0       | <b>94.4</b> | 5.6  | 0             | <b>98.4</b> | 1.6 | 0                    | <b>96.8</b> | 3.2 | 0       | <b>99.2</b> | 0.8 |
| Diversity         | 0       | <b>95.2</b> | 4.8  | 0             | <b>99.2</b> | 0.8 | 3.2                  | <b>92.8</b> | 4   | 0.8     | <b>97.6</b> | 1.6 |
| Empowerment       | 7.2     | <b>78.4</b> | 14.4 | 0             | <b>93.6</b> | 6.4 | 1.6                  | <b>90.4</b> | 8   | 1.6     | <b>95.2</b> | 3.2 |
| Overall           | 0       | <b>96.0</b> | 4    | 0             | <b>99.2</b> | 0.8 | 2.4                  | <b>97.6</b> | 2.4 | 0       | <b>99.2</b> | 0.8 |

Table 8: Compare MegaRAG with using only GPT-4o-mini in terms of win rates (%).

| Method vs. MegaRAG | Metric                   | DLCV                                     | World History                           | Env. eport                                | GenAI                                   |
|--------------------|--------------------------|--|---|---|---|
| NaiveRAG           | <i>Comprehensiveness</i> | 2.4/67.2/30.4<br><b>(1.9/52.7/45.4)</b>  | 0.0/81.5/18.5<br><b>(0.0/66.5/33.5)</b> | 0.0/72.8/27.2<br><b>(0.0/57.8/42.2)</b>   | 0.0/95.2/4.8<br><b>(0.0/80.2/19.8)</b>  |
|                    | <i>Diversity</i>         | 6.4/84.8/8.8<br><b>(5.3/70.9/23.8)</b>   | 0.0/96.8/3.2<br><b>(0.0/81.8/18.2)</b>  | 2.4/92.0/5.6<br><b>(1.9/77.5/20.6)</b>    | 0.0/98.4/1.6<br><b>(0.0/83.4/16.6)</b>  |
|                    | <i>Empowerment</i>       | 11.2/66.4/22.4<br><b>(9.0/53.6/37.4)</b> | 3.2/82.3/14.5<br><b>(2.6/67.9/29.5)</b> | 12.8/64.0/23.2<br><b>(10.2/51.6/38.2)</b> | 0.8/88.0/11.2<br><b>(0.6/73.2/26.2)</b> |
|                    | <i>Overall</i>           | 4.8/75.2/20.0<br><b>(3.9/61.1/35.0)</b>  | 0.0/89.5/10.5<br><b>(0.0/74.5/25.5)</b> | 1.6/80.0/18.4<br><b>(1.3/65.3/33.4)</b>   | 0.0/98.4/1.6<br><b>(0.0/83.4/16.6)</b>  |
| GraphRAG           | <i>Comprehensiveness</i> | 0.0/88.8/11.2<br><b>(0.0/73.8/26.2)</b>  | 0.0/92.0/8.0<br><b>(0.0/77.0/23.0)</b>  | 0.8/68.8/30.4<br><b>(0.6/54.0/45.4)</b>   | 0.0/92.8/7.2<br><b>(0.0/77.8/22.2)</b>  |
|                    | <i>Diversity</i>         | 3.2/92.8/4.0<br><b>(2.7/78.3/19.0)</b>   | 1.6/97.6/0.8<br><b>(1.4/82.8/15.8)</b>  | 7.2/81.6/11.2<br><b>(6.0/67.8/26.2)</b>   | 0.0/97.6/2.4<br><b>(0.0/82.6/17.4)</b>  |
|                    | <i>Empowerment</i>       | 1.6/95.2/3.2<br><b>(1.3/80.5/18.2)</b>   | 0.0/81.0/19.0<br><b>(0.0/81.0/19.0)</b> | 1.6/93.6/4.8<br><b>(1.3/78.9/19.8)</b>    | 0.0/100.0/0.0<br><b>(0.0/85.0/15.0)</b> |
|                    | <i>Overall</i>           | 0.0/92.8/7.2<br><b>(0.0/77.8/22.2)</b>   | 0.0/93.6/6.4<br><b>(0.0/78.6/21.4)</b>  | 0.8/84.8/14.4<br><b>(0.7/69.9/29.4)</b>   | 0.0/99.2/0.8<br><b>(0.0/84.2/15.8)</b>  |
| LightRAG           | <i>Comprehensiveness</i> | 0.0/78.4/21.6<br><b>(0.0/63.4/36.6)</b>  | 0.0/89.6/10.4<br><b>(0.0/74.6/25.4)</b> | 0.0/80.8/19.2<br><b>(0.0/65.8/34.2)</b>   | 0.0/92.0/8.0<br><b>(0.0/77.0/23.0)</b>  |
|                    | <i>Diversity</i>         | 3.2/90.4/6.4<br><b>(2.8/75.8/21.4)</b>   | 0.0/95.2/4.8<br><b>(0.0/80.2/19.8)</b>  | 1.6/92.0/6.4<br><b>(1.3/77.3/21.4)</b>    | 1.6/92.8/5.6<br><b>(1.4/78.0/20.6)</b>  |
|                    | <i>Empowerment</i>       | 11.2/74.4/14.4<br><b>(9.3/61.3/29.4)</b> | 3.2/86.4/10.4<br><b>(2.7/71.9/25.4)</b> | 4.8/79.2/16.0<br><b>(4.0/65.0/31.0)</b>   | 1.6/91.2/7.2<br><b>(1.4/76.4/22.2)</b>  |
|                    | <i>Overall</i>           | 0.8/84.8/14.4<br><b>(0.7/69.9/29.4)</b>  | 0.0/90.4/9.6<br><b>(0.0/75.4/24.6)</b>  | 0.0/90.4/9.6<br><b>(0.0/75.4/24.6)</b>    | 0.0/94.4/5.6<br><b>(0.0/79.4/20.6)</b>  |

Table 9: Pairwise win-rate comparison under two judge models. Each entry is formatted as *Other Method Win / MegaRAG Win / Tie*. The main value denotes the GPT-4.1-mini result, and the value in parentheses denotes the Gemini-3-Flash result.

| Metric            | World History      | Env. Report        |
|-------------------|--------------------|--------------------|
| Comprehensiveness | 9.2 / 72.8 / 18.0  | 6.7 / 84.5 / 8.8   |
| Diversity         | 15.6 / 70.8 / 13.6 | 12.4 / 74.8 / 12.8 |
| Empowerment       | 18.3 / 62.5 / 19.2 | 14.7 / 69.7 / 15.6 |
| Overall           | 8.8 / 71.2 / 20.0  | 5.5 / 84.1 / 10.4  |

Table 10: Pairwise LLM-judge win rates (%) between Query-driven MMKG and MegaRAG. Each entry is formatted as *Query-driven Win / MegaRAG Win / Tie*.

#### C.4 Ablation Study on Using GPT-4o-mini Only (without MMRAG)

To ensure that GPT-4o-mini has not been exposed to our evaluation datasets during pretraining, and to confirm that it cannot answer questions solely by relying on its internal knowledge, we conduct an additional ablation study. Specifically, we compare MegaRAG against a retrieval-free baseline where

answers are generated using GPT-4o-mini without access to any external context or retrieved information. As shown in Table 8, MegaRAG consistently outperforms the retrieval-free baseline, highlighting the value of combining retrieval with multimodal knowledge to enhance answer quality.

#### C.5 Recent Multimodal GraphRAG

We additionally compare MegaRAG with a recent multimodal GraphRAG method, *Query-driven MMKG* (Bu et al., 2025). Query-driven MMKG constructs query-specific local multimodal knowledge graphs dynamically at inference time, whereas MegaRAG builds a document-level multimodal knowledge graph offline and further refines it to capture cross-page and cross-modal dependencies. As a result, the two methods are designed for different operating regimes: Query-driven is

tailored to short-form, query-centered reasoning, while MegaRAG is designed for long-form document understanding and global reasoning.

We attempted to reproduce Query-driven MMKG based on the published paper and released code. However, some implementation details, particularly the prompts used for entity-relation extraction, were not fully specified. Therefore, the following comparison should be interpreted as a best-effort reproduction rather than an exact replication.

Table 10 reports the pairwise LLM-judge win rates between Query-driven and MegaRAG on two representative long-document benchmarks. Each entry is formatted as *Query-driven Win / MegaRAG Win / Tie*. MegaRAG consistently outperforms Query-driven across all evaluation dimensions on both datasets. We attribute this gap mainly to the difference in design focus: MegaRAG explicitly models document-level structure and performs global refinement, which is particularly beneficial for long-form multimodal documents requiring cross-page evidence aggregation and multi-hop reasoning.

What Will This Course Cover? *Description Exemplar*

**Deep Learning Basics:** Fundamentals of image classification, linear classifiers, regularization, optimization, neural networks, and MLPs.

**Visual Perception and Understanding:** Object detection, segmentation (semantic & instance), video understanding; core architectures including CNNs, RNNs, attention, and transformers.

**Generative & Interactive Visual Intelligence:** Self-supervised learning, generative models (e.g., style transfer, DALL·E), vision-language models (e.g., CLIP), 3D vision, and embodied intelligence.

**Human-Centered Applications & Ethics:** Applications in real-world, human-centered contexts; ethical and societal implications of computer vision.

**Learning Objectives:** 1. Formalize vision problems and assess data/model needs. 2. Implement, train, and debug deep vision models (PyTorch/TensorFlow). 3. Explore current research and ongoing challenges in computer vision.

Given the following description of a dataset:

{description} *Dataset Description*

*Key Definition*

Please identify 5 potential users who would engage with this dataset. For each user, list 5 tasks they would perform with this dataset. Then, for each (user, task) combination, generate 5 questions that require a high-level understanding of the entire dataset.

*Output Format*

Output the results in the following structure:

- User 1: [user description]
  - Task 1: [task description]
    - Question 1:
    - Question 2:
    - Question 3:
    - Question 4:
    - Question 5:
  - Task 2: [task description]
  - ...
  - Task 5: [task description]
- User 2: [user description]
- ...
- User 5: [user description]
- ...

**(a) Global Question Generation Prompt**

*Example Output*

- User 1: Data Scientist specializing in Computer Vision
- Task 1: Develop a state-of-the-art image classification model
- Question 1: How can we optimize the training process for better image classification accuracy?
- Question 2: What are the advantages of using CNNs over linear classifiers for image classification?
- Question 3: How do architectures impact the performance of a MLP in image classification tasks?
- Question 4: How can generative models be used to augment image classification datasets?
- Question 5: What are the recent advancements in image classification using PyTorch? ...

**(b) Example Global Questions**

Figure 5: (a) Prompt used for global question generation. (b) Example global questions.

You will evaluate two answers to the same question based on three criteria: **Comprehensiveness**, **Diversity**, and **Empowerment**.

- **Comprehensiveness**: How much detail does the answer provide to cover all aspects and details of the question?

- **Diversity**: How varied and rich is the answer in providing different perspectives and insights on the question?

- **Empowerment**: How well does the answer help the reader understand and make informed judgments about the topic?

For each criterion, choose the better answer (either Answer 1 or Answer 2) and explain why. Then, select an overall winner based on these three categories.

Here is the question:

{query}

Here are the two answers:

**Answer 1 Start:**

{answer1}

**Answer 1 End**

**Answer 2 Start:**

{answer2}

**Answer 2 End**

Evaluate both answers using the three criteria listed above and provide detailed explanations for each criterion.

Output your evaluation in the following JSON format:

```
{
  "Comprehensiveness": {
    "Winner": "[Answer 1 or Answer 2]",
    "Explanation": "[Provide explanation here]"
  },
  "Diversity": {
    "Winner": "[Answer 1 or Answer 2]",
    "Explanation": "[Provide explanation here]"
  },
  "Empowerment": {
    "Winner": "[Answer 1 or Answer 2]",
    "Explanation": "[Provide explanation here]"
  },
  "Overall Winner": {
    "Winner": "[Answer 1 or Answer 2]",
    "Explanation": "[Summarize why this answer is the overall winner based on the three criteria]"
  }
}
```

#### (a) Prompt for Pairwise Global QA Evaluation

You are given a question, the model's response, and the correct answer.

Your task is to evaluate whether the model's response correctly answers the question based on the correct answer provided.

Please follow this format in your output:

```
{
  "is_correct": "yes" or "no",
  "reason": "Your explanation of why the response is correct or incorrect."
}
```

Make sure your judgment is based only on the given answer, and explain your reasoning clearly and concisely.

Here is the input: Question: {query}

Model's Response: {result}

Correct Answer: {answer}

#### (b) Prompt for Local QA Evaluation

Figure 6: Overview of the global and local QA evaluation prompts.

## C.6 Case Studies

We present two case studies demonstrating the benefits of our MMKG refinement stage in improving knowledge extraction from visually rich documents. These examples show how refinement enhances multimodal grounding and enables the recovery of global, cross-page relations.

### **Example of enhanced multimodal relations.**

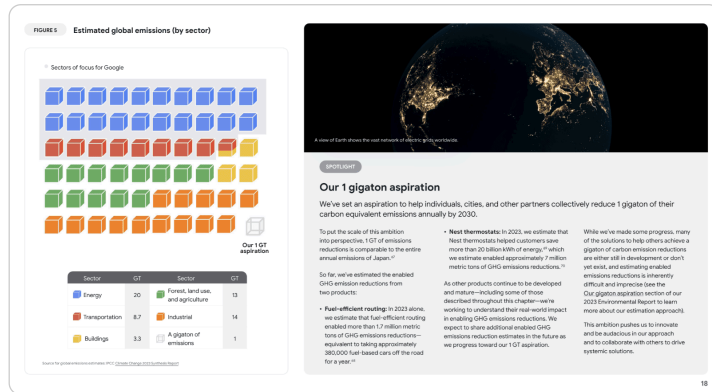
In the initial MMKG stage shown in Figure 7, entities such as *Estimated Global Emissions* and *Earth Network of Electric Grids* are extracted from figure images, but their connections to textual entities are missing. After refinement, these visual entities are correctly linked to the *1 Gigaton Aspiration*.

### **Example of enhanced cross-page relations.**

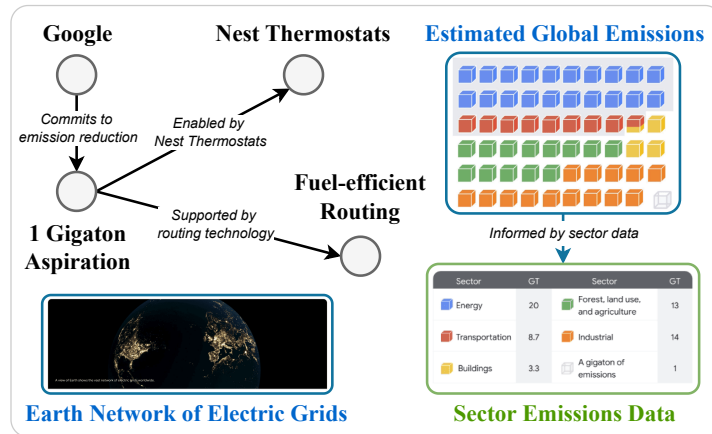
We demonstrate that cross-page relations can be recovered after the refinement stage in the example shown in Figure 8. By leveraging the provided MMKG subgraph, our method successfully links the visual entity *Renewable Energy Purchasing vs. Total Electricity* to the cross-page entity *Total Electricity Consumption*.

### **Comparative Analysis.**

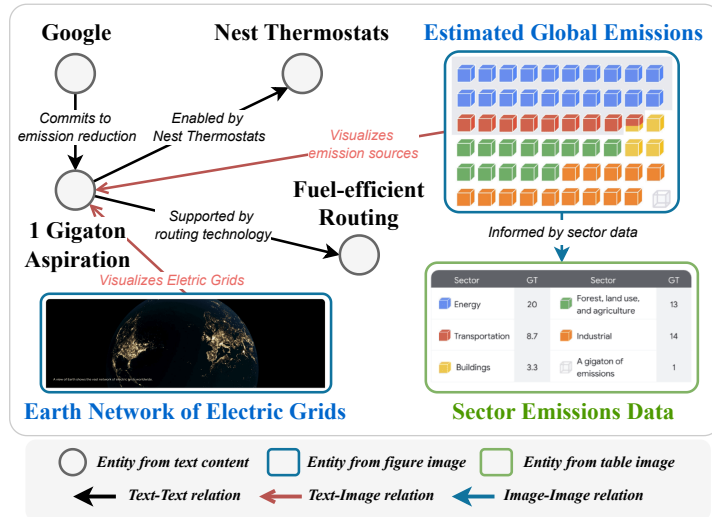
Further examples are provided in Tables 11, 12, 13, 14 to compare our MegaRAG with GraphRAG and LightRAG. As shown in the respective LLM judgement, our approach consistently outperforms the baselines across four evaluation metrics: comprehensiveness, diversity, empowerment, and overall.



(a) Example Slide Page



(b) Page-level MMKG from Initial Stage

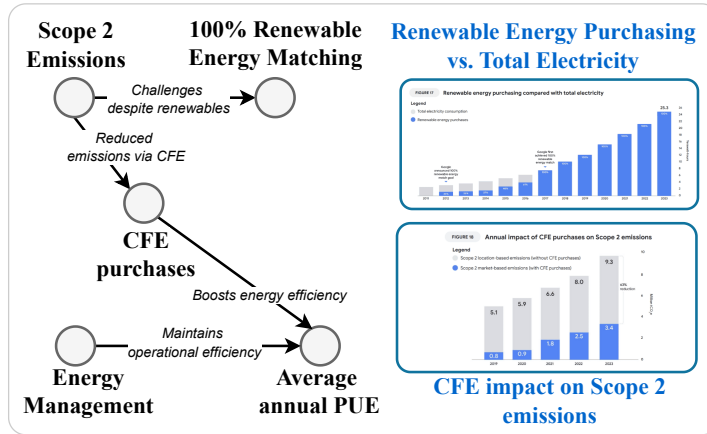


(c) Page-level MMKG from Refinement Stage

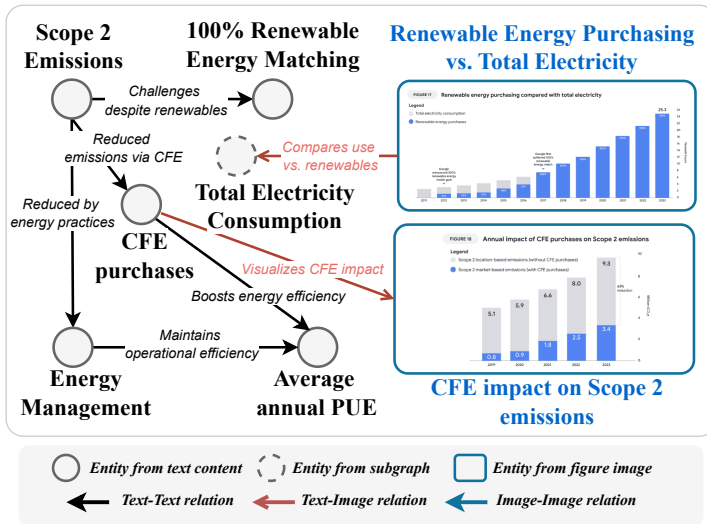
Figure 7: Example of enhanced multimodal relations. (a) A slide page from an environmental report. (b) Page-level MMKG generated in the initial stage. (c) Page-level MMKG from the refinement stage.



(a) Example Slide Page



(b) Page-level MMKG from Initial Stage



(c) Page-level MMKG from Refinement Stage

Figure 8: Example of enhanced cross-page relations. (a) A slide page from an environmental report. (b) Page-level MMKG generated in the initial stage. (c) Page-level MMKG from the refinement stage.

---

**Question 1:** What specific carbon reduction strategies are most effective across companies?

---

### ***MegaRAG***

**Effective Carbon Reduction Strategies Across Companies:** The pursuit of effective carbon reduction strategies is critical for companies aiming to mitigate their environmental impact and achieve sustainability goals. Both knowledge from the knowledge graph and insights derived from document images reveal a comprehensive landscape of strategies being implemented by various organizations, highlighting key themes such as emissions reduction, renewable energy initiatives, supply chain management, and investment in carbon removal technologies.

**Decarbonizing Energy Sources:** Leading companies like Meta, Microsoft, and Google are shifting to renewable energy. For instance, Meta has contracted over 11,700 MW of renewable energy to match 100% of its electricity use. Google has matched its energy use with 100% renewables for seven years, collaborating with the First Movers Coalition.

**Emissions Reduction Focus:** Meta targets a 42% reduction in Scope 1 and 2 emissions by 2031. Apple has cut product energy use by over 70% since 2008. Google and LinkedIn are also working on operational emissions.

**Supplier Engagement:** Microsoft promotes sustainability in its supply chain, encouraging local sourcing and renewable energy use by vendors.

**Investment in Carbon Removal:** Companies like Google and Meta are investing in direct air capture and forest restoration. Apple and Google focus on high-quality carbon credits. Microsoft leads collaborative efforts like Carbon Call to improve reporting standards.

**Conclusion:** Effective strategies span emissions reduction, renewable energy, supplier engagement, and carbon removal investments.

### ***GraphRAG***

**Effective Carbon Reduction Strategies Across Companies:** Companies are pursuing a range of strategies:

**Renewable Energy:** NVIDIA aims for 100% renewable electricity by FY2025. Apple is committed to carbon neutrality by 2030.

**Scope 3 Emissions:** Google, Apple, and Meta focus on tracking and reducing value-chain emissions.

**Climate Innovation:** Microsoft's Climate Innovation Fund invested \$761M in 2023 to support new technologies.

**Circular Economy:** Apple incorporates recycled materials into products; Meta operates on 100% renewable energy since 2020.

**Supplier Programs:** Apple's Supplier Energy Efficiency Program helps reduce emissions across the supply chain.

**Summary:** Key strategies include renewables, Scope 3 management, innovation funding, circularity, and supplier engagement.

### ***LLM Judgement***

#### **Comprehensiveness – Winner: MegaRAG**

Answer 1 gives a broader, more detailed view, linking specific strategies, company actions, and goals. Answer 2, while informative, is more concise.

#### **Diversity – Winner: MegaRAG**

Answer 1 presents a wider variety of strategies, including both direct and indirect emissions, and a broader range of carbon removal methods.

#### **Empowerment – Winner: MegaRAG**

Answer 1 empowers readers with clear explanations, real examples, and reasoning behind each strategy. Answer 2 lacks the same level of depth.

#### **Overall – Winner: MegaRAG**

Answer 1 is the most comprehensive, diverse, and empowering of the two answers.

---

---

**Question 1:** What specific carbon reduction strategies are most effective across companies?

---

### ***MegaRAG***

Due to space constraints, the detailed responses are presented in Table 11 (MegaRAG).

### ***LightRAG***

*Effective Carbon Reduction Strategies Across Companies:* Many companies, particularly in the technology sector, are implementing various carbon reduction strategies to achieve sustainability and mitigate climate change. Here are some of the most effective strategies identified across major organizations like Apple, Google, and Meta:

**Commitment to Renewable Energy:** Companies like Apple have achieved 100% renewable electricity for operations and are extending this to their supply chains. Meta and Google also match their energy use with renewables to reduce operational emissions.

**Carbon Removals and Offsetting:** Apple and Meta invest in high-quality carbon removal projects to handle residual emissions and support their net-zero goals.

**Efficiency Improvements:** Microsoft improves energy efficiency in data centers. Apple implements operational efficiency practices to lower emissions.

**Supply Chain Engagement:** Apple's programs help suppliers adopt low-carbon technologies. Meta's Net Zero Supplier Engagement Program supports suppliers in meeting emission reduction targets.

**Innovative Technologies and Practices:** Companies are adopting low-carbon technologies and nature-based solutions to reduce emissions in production and processes.

**Transparency and Reporting:** Apple and Meta publicly report their emissions and progress, increasing accountability and stakeholder trust.

**Conclusion:** These strategies show a unified effort among tech companies to reduce emissions through renewable energy, efficiency, supplier collaboration, and innovation.

### ***LLM Judgement***

#### **Comprehensiveness – Winner: MegaRAG**

Answer 1 delivers a more detailed view of carbon reduction strategies. It covers decarbonization, emissions targets, supplier engagement, carbon removal, and collaborative efforts, all backed by specific examples and data. Answer 2 provides a solid overview but lacks the same level of depth and specificity.

#### **Diversity – Winner: MegaRAG**

Answer 1 explores a wider range of approaches, from operational reductions to nature- and tech-based carbon removal, and cross-company collaborations. Answer 2 touches on similar themes but offers fewer perspectives and less nuance.

#### **Empowerment – Winner: MegaRAG**

By including concrete commitments, rationales, and cross-sector initiatives, Answer 1 helps readers better understand and assess real-world strategies. Answer 2 is clear but more general, offering less actionable insight.

#### **Overall – Winner: MegaRAG**

Answer 1 stands out for its depth, breadth, and practical value. It consistently outperforms Answer 2 across all criteria, making it a more comprehensive and informative response for understanding corporate carbon reduction efforts.

---

Table 12: Case (1) Study: Comparison between MegaRAG and LightRAG.

---

**Question 2:** How can generative models be used to augment image classification datasets?

---

### ***MegaRAG***

#### **Utilizing Generative Models for Augmenting Image Classification Datasets**

Generative models, such as Generative Adversarial Networks (GANs) and Variational Autoencoders (VAEs), enhance image classification datasets by increasing both size and quality.

**Data Generation:** These models create synthetic images that resemble real data. GANs can alter image attributes without extra labeling, while VAEs sample new images from the original data distribution—crucial for small datasets.

**Enhancing Feature Learning:** Variations created by generative models help classifiers recognize subtle differences and learn more discriminative features, improving accuracy under different conditions like lighting or orientation.

**Insight Extraction:** Generative models reveal hidden data structures in high-dimensional spaces—beneficial in fields like medical imaging—helping researchers understand data with limited labeled samples.

**Robust Augmentation:** Advanced augmentation techniques, including super-resolution and colorization, not only create new data but also enhance robustness against overfitting by exposing models to more diverse representations.

**Simulation and Adaptation:** They can simulate diverse environments for applications like robotics and reinforcement learning, allowing broader scenario training for better generalization.

### ***GraphRAG***

Generative models such as GANs and VAEs augment image classification datasets by generating synthetic images, improving dataset size and diversity. This process addresses class imbalance and boosts model robustness.

GANs use a generator and discriminator to produce realistic images, which are helpful when the dataset is small or imbalanced. VAEs encode input data into a latent space and sample from it to create new images, increasing dataset variation.

Generative models can also generate labeled data, reducing the need for expensive manual annotation. Basic augmentation like color shifts, noise addition, and rotations enhances generalization.

By integrating generative models early in the data pipeline, models learn from both real and synthetic images, leading to higher accuracy and reduced overfitting.

### ***LLM Judgement***

#### **Comprehensiveness – Winner: MegaRAG**

Answer 1 offers a more complete explanation, discussing synthetic generation, feature learning, data structure analysis, augmentation robustness, and simulation. Answer 2 focuses mainly on synthetic data and standard augmentation.

#### **Diversity – Winner: MegaRAG**

Answer 1 explores multiple dimensions—from feature learning to simulation—while Answer 2 concentrates on class imbalance and dataset enlargement.

#### **Empowerment – Winner: MegaRAG**

Answer 1 gives readers a clearer picture of how and why generative models are effective, with practical examples and diverse use cases. Answer 2 is helpful but more limited in depth.

#### **Overall – Winner: MegaRAG**

Answer 1 outperforms Answer 2 in all aspects, providing broader insights and more actionable information for leveraging generative models in dataset augmentation.

---

Table 13: Case (2) Study: Comparison between MegaRAG and GraphRAG.

---

**Question 2:** How can generative models be used to augment image classification datasets?

---

***MegaRAG***

Due to space constraints, the detailed responses are presented in Table 13 (MegaRAG).

***LightRAG***

**Augmenting Image Classification Datasets with Generative Models**

Generative models enhance image classification datasets by producing synthetic images that resemble original data, which is crucial for improving model robustness and generalization when data is limited.

**Data Generation:** GANs and VAEs can learn the data distribution and sample new images, expanding dataset size and diversity.

**Creating Diverse Samples:** Through transformations like rotation, scaling, color shifts, and noise, these models generate varied images, helping prevent overfitting.

**Enhancing Underrepresented Classes:** Generative models can target low-sample classes to balance the dataset.

**Improving Model Stability:** Adding synthetic data introduces broader scenarios, improving training stability and real-world performance.

**Applications:** Augmented datasets aid CNNs (e.g., AlexNet) and fine-tuning in transfer learning tasks for better classification results.

***LLM Judgement***

**Comprehensiveness – Winner: MegaRAG**

Answer 1 provides broader and more detailed coverage, including feature learning, high-dimensional insight extraction, advanced augmentation (e.g., super-resolution), and simulation. Answer 2 covers core concepts well but lacks these deeper applications.

**Diversity – Winner: MegaRAG**

Answer 1 discusses a wider array of technical and application perspectives—ranging from data generation to domain-specific use. Answer 2 focuses more narrowly on basic augmentation and dataset balance.

**Empowerment – Winner: MegaRAG**

Answer 1 better equips readers by showing \*how and why\* generative models enhance data. It includes multiple use cases and explains strategic benefits. Answer 2 is more concise, with fewer actionable insights.

**Overall – Winner: MegaRAG**

Answer 1 wins across all criteria. Its comprehensive scope, nuanced techniques, and practical guidance make it more informative and valuable overall.

---

Table 14: Case (2) Study: Comparison between MegaRAG and LightRAG.