

Query-Driven Multimodal GraphRAG: Dynamic Local Knowledge Graph Construction for Online Reasoning

Chenyang Bu ¹, Guojie Chang¹, Zihao Chen¹, Cunyuan Dang¹,
Zhize Wu², Yi He ³, Xindong Wu ¹

¹ Key Laboratory of Knowledge Engineering with Big Data of Ministry of Education,
School of Computer Science and Information Engineering,
Hefei University of Technology, China

² School of Artificial Intelligence and Big Data, Hefei University, Hefei, China

³ Department of Data Science, William & Mary, Williamsburg, VA, USA

Correspondence to: {chenyangbu, xwu}@hfut.edu.cn

Abstract

An increasing adoption of Large Language Models (LLMs) in complex reasoning tasks necessitates their interpretability and reliability. Recent advances to this end include retrieval-augmented generation (RAG) and knowledge graph-enhanced RAG (GraphRAG), whereas they are constrained by static knowledge bases and ineffective multimodal data integration. In response, we propose a Query-Driven Multimodal GraphRAG framework that *dynamically* constructs local knowledge graphs tailored to query semantics. Our approach 1) derives graph patterns from query semantics to guide knowledge extraction, 2) employs a multi-path retrieval strategy to pinpoint core knowledge, and 3) supplements missing multimodal information ad hoc. Experimental results on the MultimodalQA and WebQA datasets demonstrate that our framework achieves the state-of-the-art performance among unsupervised competitors, particularly excelling in cross-modal understanding of complex queries. The code is publicly available at <https://github.com/DMiC-Lab-HFUT/Query-Driven-Multimodal-GraphRAG>.

1 Introduction

The integration of knowledge graphs (KGs) with LLMs (LLMs) through retrieval-augmented generation (RAG), known as GraphRAG, has emerged as a promising paradigm for enhancing factual accuracy and reasoning capabilities (Pan et al., 2024; Edge et al., 2024). While traditional RAG systems leverage unstructured text retrieval, recent GraphRAG approaches attempt to combine structured knowledge graphs (KGs) with LLMs to capture complex entity relationships and enable multi-hop reasoning (Huang et al., 2025).

To do that, the existing methods construct KGs typically in a *bottom-up*, information-driven fashion (Cohen et al., 2023), which begins by identifying available data sources and integrating them

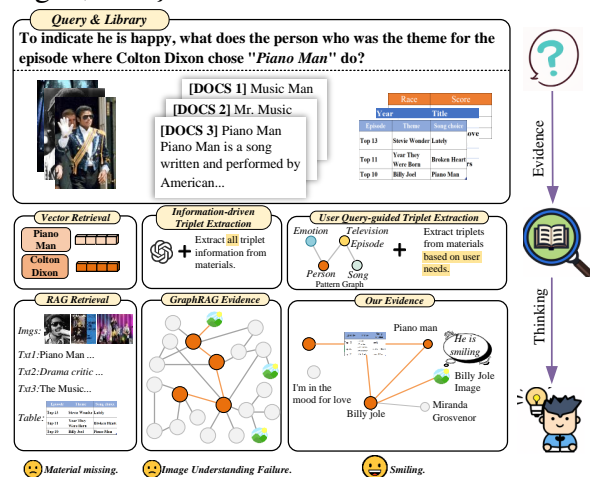


Figure 1: Challenges and Insights in Multimodal Multi-hop Query Answering: Integrating Image, Table, and Document Data for Online Reasoning. Answering the query requires identifying key details from a table to identify the theme person when Colton Dixon performed 'Piano Man'. Then, this information must be corroborated with an image to determine his emotional expression. This highlights the difficulty in building knowledge graphs that can fully cover all aspects of a query without knowing the exact question in advance.

into triples through entity extraction, relationship extraction, and knowledge fusion. Yet, this can introduce extra complexity and runtime overhead. Specifically, they lack awareness of user needs during KG construction (Cohen et al., 2023), leading to either overly granular graphs with high computational costs or overly coarse representations that lose critical semantic details. For example, in multimodal settings, parsing complex visual information without query context results in either excessive detail extraction or incomplete knowledge integration. This rigidity is particularly problematic for queries requiring cross-modal understanding, where existing frameworks struggle to effectively combine textual, visual, and tabular data into a unified knowledge representation. As illustrated in Fig. 1, answering a query typically requires extracting key details from multiple modalities—such as

identifying the theme person in a table and confirming his emotional expression through an image. This highlights the challenge of constructing KGs that can fully cover all aspects of a query without knowing the exact question in advance.

Such inefficiencies motivate us to pursue a more dynamic and query-driven approach for KG construction. In this paper, we draw insight from cognitive science. That is, human reasoning often employs *top-down*, goal-driven attention to prioritize task-relevant information, so as to filter and integrate the most pertinent knowledge (Theeuwes, 2010; Baluch and Itti, 2011). To instantiate this insight, we propose an *unsupervised Query-Driven Multimodal GraphRAG* framework that promotes efficient LLM-KG integration through three key components: 1) **Dynamic Graph Pattern Construction**: By analyzing query semantics, our framework derives entity-relationship patterns to guide knowledge extraction, ensuring focused and selective retrieval. 2) **Multimodal Selective Attention**: We implement a multi-path filtering strategy to identify the most relevant information across textual, visual, and tabular modalities. 3) **Iterative Knowledge Refinement**: Our framework iteratively supplements missing information through cross-modal reasoning, creating a comprehensive and adaptive knowledge representation.

Contributions of this paper are as follows:

- We propose an unsupervised, query-driven KG construction paradigm that adapts to user needs in an ad-hoc fashion, improving efficiency and relevancy of the retrieval.
- A multimodal selective attention is tailored to enable effective integration of textual, visual, and tabular data in LLM reasoning.
- Extensive experiments on the widely used MultimodalQA and WebQA benchmarks, show that our approach is on a par with the state-of-the-art supervised methods, while significantly outperforming its unsupervised competitors by 10.9% in F1 on MultimodalQA and 9.13% in QA score on WebQA. Further, it improves the exact match accuracy by 15.5% and 8.05% over the graph construction methods in KAG (Liang et al., 2024) and GraphRAG (Edge et al., 2024), respectively, while reducing the constructed nodes by 78.04 and 12.36 and edges by 179.66 and 7.99 per question for efficiency.

2 Related Work

RAG has emerged as a key paradigm for addressing hallucination issues in LLMs by integrating external knowledge bases into the generative process. Traditional RAG systems retrieve relevant documents based on user queries and use them as context for LLMs to generate grounded responses (Lewis et al., 2020). While effective, these systems frequently assume that relevant information is localized within specific text regions, which fails to capture cross-document or long-range contextual dependencies (Kuratov et al., 2024). Moreover, their reliance on text matching for retrieval commonly results in fragmented results, limiting their effectiveness for multi-hop reasoning tasks (Laskar et al., 2020; Yao et al., 2017).

The integration of KGs with RAG, known as GraphRAG, addresses many limitations of traditional RAG by leveraging structured knowledge representations. Unlike conventional RAG, which operates on raw text, GraphRAG constructs KGs from documents, explicitly modeling semantic relationships between entities to support multi-hop reasoning and improve interpretability (Edge et al., 2024). Recent advancements in GraphRAG have focused on enhancing both retrieval and reasoning capabilities: (1) **Graph-Based Retrieval**: Methods like (Edge et al., 2024) propose entity graphs and community summaries to improve diversity and comprehensiveness in large-scale text processing. (2) **Query Optimization**: Approaches such as (Xu et al., 2024) integrate historical query-based KGs to preserve structural relationships and optimize retrieval. (3) **Efficient Architectures**: LightRAG (Guo et al., 2024) introduces a dual-layer retrieval system that incorporates KGs into text indexing, balancing efficiency and retrieval quality. (4) **Reasoning Enhancement**: The KG-RAG framework (Sanmartin, 2024) employs the Chain of Explorations (CoE) algorithm to combine structured KGs with LLMs, improving performance on long-text and multi-hop reasoning tasks.

Despite these advancements, a critical challenge remains in effectively integrating multimodal data into KGs. Most GraphRAG frameworks focus primarily on textual knowledge, overlooking the structured incorporation of images, tables, and other modalities. While recent efforts like MuRAG (Chen et al., 2022) have begun exploring multimodal RAG, seamless fusion of heterogeneous data at the KG level remains an open

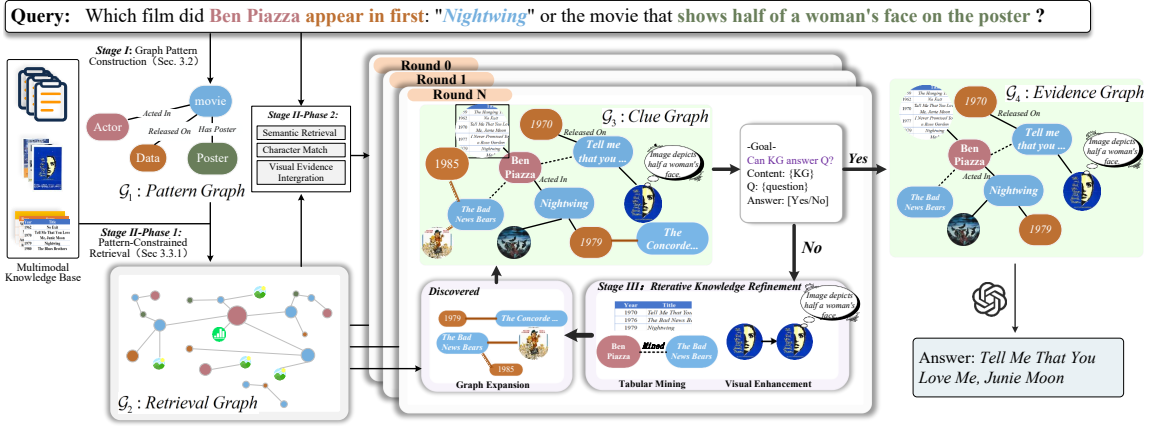


Figure 2: Overview of the Query-Driven Multimodal GraphRAG Framework. The framework consists of three stages: (1) Pattern Graph Construction, (2) Multimodal Selective Attention, and (3) Iterative Knowledge Refinement.

problem. This limitation is particularly evident in complex queries requiring cross-modal reasoning, where existing methods struggle to synthesize information from diverse sources. The lack of dynamic KG construction methods further exacerbates this issue, as static KGs cannot adapt to new knowledge or evolving query contexts.

3 Proposed Framework

3.1 Cognitive Foundations

Our proposed framework is rooted in cognitive science principles of human problem-solving in the following key aspects:

Definition 1 (Temporary Mental Patterns). *Temporary Mental Patterns (TMPs) (Johnson-Laird, 1983), are dynamic cognitive frameworks transiently constructed in working memory to support task-specific reasoning.*

These ephemeral structures exhibit four key characteristics, illustrated through the example shown in Fig. 1: (1) Transience - maintained only during active task processing; (2) Focus - organize knowledge by selectively activating relevant long-term memory fragments (e.g., retrieving “performer–song” mappings from tabular data); (3) Anticipation - prioritize information critical to current goals through contextual prediction (e.g., automatically retrieving a relevant image when the query involves emotional expression); and (4) Plasticity - iteratively restructure through evidence accumulation (e.g., dynamically incorporating missing contextual cues such as the identity of the performer).

The operational mechanism of TMPs can be analogized to a librarian managing limited cognitive resources: (1) Shelf Selection: Locate specific memory partitions aligned with task demands (e.g.,

selecting the table containing artist and song performance records); (2) Book Filtering: Rapidly discard irrelevant data streams (e.g., ignoring unrelated statistics such as song duration); (3) Passage Tagging: Cognitively highlight critical information fragments (e.g., emphasizing the row where Colton Dixon performed “Piano Man”); (4) Annotation Updates: Integrate newly discovered evidence into the active knowledge framework (e.g., incorporating the emotional expression inferred from an associated image).

Remark 1 (Iterative Refinement). *Initial patterns may be incomplete or contain inaccuracies (cognitive “hypotheses”), which are progressively refined through:*

$$\mathcal{G}_1^{(t+1)} = f_{\text{update}}(\mathcal{G}_1^{(t)}, \mathcal{F}_{\text{feedback}}) \quad (1)$$

where $\mathcal{F}_{\text{feedback}}$ incorporates evidence from downstream reasoning stages.

Definition 2 (Neural Efficiency Principle). *The neural efficiency principle (Neubauer and Fink, 2009) describes the brain’s ability to dynamically allocate cognitive resources based on task demands. Formally, this is implemented through the resource optimization equation:*

$$\text{Computational Cost} \propto \text{Uncertainty}(\mathcal{G}_1)^{-1} \quad (2)$$

where \mathcal{G}_1 denotes the initial pattern graph. This establishes an inverse relationship between system confidence and resource expenditure.

Based on this definition, the assumptions of our approach are as follows and shown in Fig. 2: (1) In Stage I, there is high uncertainty regarding the correctness of the Pattern Graph \mathcal{G}_1 generated by the LLM, thus in Section 3.3, two rapid retrieval methods are used: one extracts the graph from the original data based on \mathcal{G}_1 (yielding the retrieval graph),

and the other uses \mathcal{G}_2 and the Query-Focused Clue Refinement in Section 3.3.2 for fast matching. (2) Once the clue graph \mathcal{G}_3 is obtained and passed to the LLM for determining whether it can adequately answer the question, the uncertainty decreases significantly because the model has already assessed the completeness of the answer. Therefore, in Stage III, we iteratively expand the KG using high-confidence refinement, including augmentation operators for adding icon information and graph expansion operators for extending one-hop neighboring nodes. (3) The final query-focused KG containing all relevant multimodal evidence (denoted as \mathcal{G}_4) alongside the original query is provided as context to the LLM for reasoning.

3.2 Stage I: Graph Pattern Construction

The first stage constructs a *Pattern Graph* (denoted as \mathcal{G}_1) to guide knowledge acquisition based on query semantics. This mimics human cognitive processes by identifying key entity types and relationships relevant to the query.

Definition 3 (Pattern Graph, \mathcal{G}_1). *The Pattern Graph is a structured representation of query semantics, defined as $\mathcal{G}_1 = (T, \mathcal{R})$, where $T = \{t_1, t_2, \dots, t_n\}$ is the set of key entity types, and $\mathcal{R} = \{r_1, r_2, \dots, r_m\}$ is the set of semantic relationship types.*

For example, given the query “Which actors have appeared in Marvel movies?”, the system identifies entity types $T = \{\text{Actor}, \text{Movie}\}$ and relationship type $\mathcal{R} = \{\text{ActedIn}\}$, constructing the Pattern Graph $\mathcal{G}_1 = \{(\text{Actor}, \text{ActedIn}, \text{Movie})\}$.

We adopt LLMs to generate first-approximation patterns using their parametric knowledge acquired during pretraining. It is worth pointed out that while initial patterns may contain inaccuracies, our iterative refinement process in the following subsection aligns with human cognitive adaptation observed in problem-solving tasks (see Remark 1), where initial mental models are progressively refined through evidence accumulation.

3.3 Stage II: Multimodal Selective Attention

Our framework implements a neurocognitive-inspired selective attention mechanism through two progressive filtering phases:

3.3.1 Phase 1: Pattern-Constrained Retrieval

The initial phase constructs the *Retrieval Graph* (\mathcal{G}_2) through pattern-guided filtering:

Definition 4 (Retrieval Graph, \mathcal{G}_2). *Given a Pattern Graph $\mathcal{G}_1 = (T, \mathcal{R})$ and multimodal knowledge base \mathcal{D} , the Retrieval Graph is a constrained subgraph:*

$$\mathcal{G}_2 = \Psi_{\mathcal{G}_1}^1(\mathcal{D}) = (V, E) \quad (3)$$

subject to:

Type Filtering : $V = \{v \in \mathcal{D} \mid \phi_{\text{type}}(v) \in T\}$

Relation Pruning : $E = \{(v_i, r, v_j) \in \mathcal{D} \mid r \in \mathcal{R} \wedge \{\phi_{\text{type}}(v_i), \phi_{\text{type}}(v_j)\} \subseteq T\}$

Here, ϕ_{type} maps entities to types; and $\Psi_{\mathcal{G}_1}^1$ represents a resource-light graph construction operator, with pattern constraints of \mathcal{G}_1 :

(1) **Textual Extraction**:

$$V^T = \bigcup_{d \in \mathcal{D}_{\text{text}}} \text{LLM}(d, P_{\text{extract}}(T, \mathcal{R})) \quad (4)$$

where d refers to text document from corpus $\mathcal{D}_{\text{text}}$; prompt P_{extract} refers to a prompt template enforcing $\forall v \in V^T : \phi_{\text{type}}(v) \in T$ and $\forall e \in E^T : \phi_{\text{type}}(e) \in \mathcal{R}$.

(2) **Cross-Modal Node Anchoring**: The visual anchoring process establishes connections between image nodes and existing entity nodes through title-entity similarity analysis:

$$V^M = \{m \in \mathcal{D}_{\mathcal{M}} \mid \exists v \in V^T, \text{ED}(m, v) < \tau_{\mathcal{M}}\}$$

with edge creation constraint:

$$E^M = \{(v_j, \text{related_to}, v_i) \mid v_j \in V^M, v_i \in V\},$$

where ED denotes the edit distance computed between (1) modality-specific textual attributes (e.g., image and table titles), and (2) name attributes of existing nodes in V^T ; $\mathcal{M} \in \{\text{Image}, \text{Table}\}$ and $\tau_{\mathcal{M}}$ is a similarity threshold.

3.3.2 Phase 2: Query-Focused Clue Refinement

The subsequent phase constructs the *Clue Graph* (\mathcal{G}_3) through precision filtering:

Definition 5 (Clue Graph (\mathcal{G}_3)). *The Clue Graph is a subgraph of \mathcal{G}_2 , obtained by filtering based on the user query, retaining only entities and relationships highly relevant to the query:*

$$\mathcal{G}_3 = \Psi_{\mathcal{G}_2}^2(\mathcal{Q}) = (\mathcal{V}_C, \mathcal{E}_C)$$

where:

$$\begin{cases} \mathcal{V}_C = \mathcal{F}_{DPER}(V) \cup \mathcal{F}_{VLM}(V^I) \\ \mathcal{E}_C = \{e \in E \mid h(e), t(e) \in \mathcal{V}_C\} \end{cases}$$

Here, Q denotes the user query; $\mathcal{F}_{\text{DPER}}$ denotes the Dual-Path Entity Retrieval (DPER), and \mathcal{F}_{VLM} refers to the Visual Evidence Integration process, which are detailed as follows.

Dual-Path Entity Retrieval ($\mathcal{F}_{\text{DPER}}$). We propose a dual-path entity retrieval (DPER) method that combines lexical filtering and semantic filtering, formalized as follows.

(1) **Lexical Filtering:** we use a LLM to extract a set of keywords $W = \{w_1, w_2, \dots, w_n\}$ from the user query Q . Then, we calculate the edit distance between each keyword w and the name of each node v in the graph. Nodes with the smallest edit distance to the keyword are selected into the candidate set, which can be expressed as:

$$\mathcal{V}_{\text{lex}} = \text{Top}_k \left[\min_{w \in W} \text{ED}(v_{\text{name}}, w) \right] \quad (5)$$

(2) **Semantic Filtering:** we calculate the semantic similarity between the query Q and the node description as follows. The top- k nodes most relevant to the query are reserved.

$$\mathcal{V}_{\text{sem}} = \text{Top}_k \left[\text{sim}_{\text{cos}}(\text{BERT}(Q), \text{BERT}(v_{\text{desc}})) \right]_{v \in V}$$

where v_{desc} refers to the description of node $v \in V$.

Visual Evidence Integration (\mathcal{F}_{VLM}). We employ a multi-stage image matching strategy that combines visual information with the query content to enhance the matching accuracy of image nodes: (1) For a given user query Q , we first use an LLM to extract a set of image-related description information $D = \{d_1, d_2, \dots, d_n\}$. For example, for the query "Which film did Ben Piazza appear in first: 'Nightwing' or the movie that shows half of a woman's face on the poster?", the extracted description could be "half of a woman's face on the poster." (2) Next, we use the CLIP (Radford et al., 2021) model to select the top- k images related to each description d_i from the image node set of the KG. Specifically, for each description d_i , we calculate its similarity score $S(d_i, v_j)$ with each image node v_j and select the top- k images with the highest scores. (3) Finally, we input the candidate image set selected by the CLIP model into a multimodal large model (VLM) (Liu et al., 2024a) for deep semantic matching, where each description d_i is matched to the most relevant image.

3.4 Stage III: Iterative Knowledge Refinement

Definition 6 (Evidence Graph, \mathcal{G}_4). *The conclusive knowledge representation formed through multi-*

modal evidence accumulation:

$$\mathcal{G}_4 = \Psi_{\mathcal{G}_3}^3(Q) = (\mathcal{V}_E, \mathcal{E}_E)$$

where:

$$\begin{cases} \mathcal{V}_E = \mathcal{V}_C \cup \mathcal{V}_{\text{img}}^+ \cup \mathcal{V}_{\text{table}}^+ \\ \mathcal{E}_E = \mathcal{E}_C \cup \mathcal{E}_{\text{img}} \cup \mathcal{E}_{\text{table}} \cup \mathcal{E}_{\text{expand}} \end{cases}$$

with $\mathcal{V}_{\text{img}}^+/\mathcal{E}_{\text{img}}$ denoting augmented visual nodes/edges and $\mathcal{V}_{\text{table}}^+/\mathcal{E}_{\text{table}}$ representing enriched tabular components.

The Evidence Graph expands the knowledge content by analyzing images and tables in the Clue Graph, enabling a more comprehensive response to user queries. For example, given the query "Who are the main actors in Marvel movies?", the Evidence Graph may include movie posters and actor information. However, if the actor photos and names on the posters are not further processed, they may remain as static information and cannot be effectively utilized. During the construction of the final answer graph, analyzing images and tables allows the extraction of hidden information from multimodal data, leading to a more complete response to the query.

The evidence graph construction follows an **Evaluate-Augment-Expand** loop as shown in Algorithm 1. In Step 3, the information sufficiency evaluation is conducted to determine whether the current graph contains sufficient information to answer the query Q .

Algorithm 1 Iterative Knowledge Refinement

- 1: Initialize $\mathcal{G}_4^{(0)} \leftarrow \mathcal{G}_3$
 - 2: **for** $t \in 1..T_{\text{max}}$ **do**
 - 3: $\text{Sufficient}^{(t)} \leftarrow \text{LLM}_{\text{eval}}(Q, \mathcal{G}_4^{(t-1)})$
 - 4: **if** $\text{Sufficient}^{(t)}$ **then break**
 - 5: **end if**
 - 6: $\mathcal{G}_4^{(t)} \leftarrow \text{Augment}(\mathcal{G}_4^{(t-1)})$
 - 7: $\mathcal{G}_4^{(t)} \leftarrow \text{Expand}(\mathcal{G}_4^{(t)})$
 - 8: **end for**
-

Augmentation Operators in Step 6 of Algorithm

1. (1) Visual Evidence Enhancement: We employ a Vision-Language Model (VLM) to augment image nodes with on-demand information, extracting relevant visual details from images to overcome the limitation of missing fine-grained visual information in the current graph. Specifically, we use a LLM to analyze query Q , identify the image nodes v_k in \mathcal{G}_3 that are related to the query, and generate

an image-related answer q_k for each image node v_k , which can be formalized as follows:

$$\mathcal{V}_{\text{img}}^+ = \bigcup_{v_k \in \mathcal{V}_C^I} \text{VLM}(v_k, \text{LLM}_{\text{vis}}(Q, v_k)) \quad (6)$$

where \mathcal{V}_C^I denotes image nodes in \mathcal{G}_3 , and LLM_{vis} generates visual queries.

(2) Tabular Relationship Mining: Although table nodes contain rich relational information, statically pre-modeling all table relationships may lead to an overly complex graph structure. Therefore, we opt to dynamically extract query-relevant relationships from table data during the inference stage, based on the specific needs of the query. Specifically, we use a LLM to analyze the content of the table nodes in the clue graph \mathcal{G}_3 and identify a set of candidate entities V_{req} that may be relevant to the query. This process can be formalized as:

$$\mathcal{E}_{\text{table}} = \bigcup_{t \in \mathcal{V}_C^T} \text{Join}(t, \text{LLM}(t, Q)) \quad (7)$$

with LLM performing semantic table search.

Graph Expansion Operator in Step 7. Structural broadening through neighborhood inclusion:

$$\mathcal{E}_{\text{expand}} = \bigcup_{v \in \mathcal{V}_E} \{(v, r, u) \mid u \in N(v)\} \quad (8)$$

where $N(v)$ denotes one-hop neighbors in \mathcal{G}_2 .

4 Experiments

To evaluate the effectiveness, we conduct experiments on two multimodal multihop question-answering datasets: MultimodalQA (Gupta et al., 2018) and WebQA (Chang et al., 2022), employing evaluation metrics of EM and F1 for MultimodalQA, and QA-FL, QA-ACC, and QA-F1 for WebQA. We use BGE-M3 (Chen et al., 2024) for text embeddings and CLIP (Radford et al., 2021) for image embeddings. The Qwen 2.5 72B model (Yang et al., 2024) handles graph construction, while LLaVA 34B (Liu et al., 2024a) and Gemini-1.5-flash (Team et al., 2024) manage cross-modal reasoning. All experiments are conducted on NVIDIA A100 GPUs with 80GB memory.

We compare our method with two categories of approaches: (1) **Supervised:** AR (Gupta et al., 2018), ID (Gupta et al., 2018), MGT (He and Wang, 2023), MMHQA-ICL (Liu et al., 2023), PReasM-Large (Yoran et al., 2021), HPOPPO (Shi et al., 2024), SKURG (Yang et al., 2023), vlp-x101fnp (Chang et al., 2022), vlp-VinVL (Chang et al., 2022), MuRAG (Chen et al., 2022); and (2)

Unsupervised: Vicuna-7B (Chiang et al., 2023), Llama2Chat-13b (Touvron et al., 2023), OpenChat-v2-w-13b (Wang et al., 2024), MOQAGPT (Zhang et al., 2023), Binder (Cheng et al., 2023), OFA-Cap (Wang et al., 2022), and PROMPTCap (Hu et al., 2023). The results of the comparison methods are taken from the original papers.

4.1 Main Results

RQ1: *Does dynamic KG construction improve performance in multimodal question answering?*

Overall performance. Tables 1 and 2 show that our method demonstrates significant superiority over both supervised and unsupervised baselines on the MultimodalQA and WebQA datasets. On the MultimodalQA dataset, our method achieves 68.0% F1 and 60.3% EM without using labeled data, outperforming the supervised baseline SKURG, which achieves 64.0% F1 and 59.8% EM, and the unsupervised approach Binder, which achieves 57.1% F1 and 51.0% EM. These results highlight the effectiveness of our supervision-agnostic approach, which excels in multimodal reasoning tasks by efficiently constructing dynamic KGs and integrating multimodal data. This performance confirms that our method provides a more robust and accurate solution for handling complex multimodal queries, even in the absence of labeled data.

Efficiency gains. To validate the effectiveness of the proposed graph construction strategy, we compared different graph construction methods on the Multimodal dataset, while keeping the retrieval strategy constant. The results, presented in Table 3, demonstrate that our method significantly reduces graph complexity while maintaining high performance. Specifically, the graphs constructed by KAG contain an average of 137.99 nodes and 218.32 edges per question, whereas our method’s graphs have an average of only 59.95 nodes and 38.66 edges. This reduction in graph size highlights our method’s ability to filter out redundant information, focusing only on entities and relationships most relevant to the query, thanks to the Pattern Graph-guided construction. In terms of performance, our method achieves an EM score of 60.30, which outperforms KAG by 15.50 points and GraphRAG by 8.05 points. These results illustrate that the query-driven approach not only improves graph construction efficiency but also enhances reasoning accuracy, making it more suitable for handling large-scale knowledge bases.

Table 1: Performance Comparison on the MultimodalQA dataset. *F1* and *EM* measure the word-level matching accuracy and exact match rate between model predictions and ground truth answers, respectively. *Unimodal* indicates questions that only require a single modality to answer, and *Multimodal* indicates questions that require cross-modal reasoning, while *All* represents the results of all test questions.

| Type | Model | Unimodal | | Mutimodal | | All | |
|--------------|---------------------------------------|-------------|-------------|-----------|------|------|------|
| | | F1 | EM | F1 | EM | F1 | EM |
| Supervised | AR (ICLR, 2021) | 58.5 | 51.7 | 40.2 | 34.2 | 51.1 | 44.7 |
| | ID (ICLR, 2021) | 58.4 | 51.6 | 51.2 | 44.6 | 55.5 | 48.8 |
| | MGT (ACL, 2021) | - | - | - | - | 57.7 | 52.1 |
| | MMHQA-ICL (Liu et al., 2023) | 72.9 | 60.5 | 55.5 | 46.2 | 65.8 | 54.8 |
| | PREasM-Large (ACL, 2021) | - | - | - | - | 65.5 | 59.0 |
| | HPROPRO (ACL, 2024) | - | - | - | - | 66.7 | 59.0 |
| | SKURG (MM, 2023) | 69.7 | 66.1 | 57.2 | 52.5 | 64.0 | 59.8 |
| Unsupervised | Vicuna-7B (Chiang et al., 2023) | 20.3 | 17.1 | 16.4 | 11.9 | 18.6 | 14.9 |
| | Llama2Chat-13b (Touvron et al., 2023) | 24.6 | 21.3 | 17.3 | 13.0 | 21.5 | 17.7 |
| | OpenChat-v2-w-13b (ICLR, 2024) | 25.3 | 22.0 | 18.9 | 15.5 | 22.5 | 19.2 |
| | MOQAGPT (ACL, 2023) | 54.6 | 49.1 | 33.8 | 30.5 | 45.6 | 41.1 |
| | Binder (ICLR, 2023) | - | - | - | - | 57.1 | 51.0 |
| | Ours | 72.8 | 64.5 | 60.7 | 54.3 | 68.0 | 60.3 |

Table 2: Performance Comparison on the WebQA Dataset. *QA-FL* evaluates the fluency and semantic coherence between the generated and reference answers; *QA-ACC* measures the overlap of key entities; and the overall *QA* score is computed as the corpus-level average of the product of *QA-FL* and *QA-ACC*, serving as the primary metric for assessing overall performance. The symbol * indicates results reproduced under our unified experimental settings using the official model implementations.

| Type | Model | QA-FL | QA-ACC | QA |
|--------------|--|--------------|--------------|--------------|
| Supervised | vlp-x101fnp (CVPR, 2022) | 42.56 | 36.68 | 22.61 |
| | vlp-VinVL (CVPR, 2022) | 44.15 | 38.88 | 24.06 |
| | MuRAG (ACL, 2022) | 55.7 | 54.6 | 36.1 |
| | SKURG (MM, 2023) | 55.4 | 57.1 | 37.7 |
| Unsupervised | Vicuna-7B* (Chiang et al., 2023) | 20.64 | 38.06 | 12.14 |
| | Llama2Chat-13b* (Touvron et al., 2023) | 19.47 | 37.95 | 13.15 |
| | MOQAGPT* (ACL, 2022) | 25.48 | 44.78 | 17.08 |
| | OFA-Cap (GPT-3) (ICML, 2022) | 52.8 | 55.4 | 33.5 |
| | PROMPTCap (GPT-3) (ICCV, 2023) | 53.0 | 57.2 | 34.5 |
| | Ours | 58.43 | 63.14 | 43.63 |

Table 3: Comparison of Graph Construction Methods on the MultimodalQA Dataset, including the Recently Released GraphRAG (Han et al., 2024) by Microsoft and KAG (Liang et al., 2024) by Alibaba Corporation.

| Method | node nums | edge nums | EM |
|-----------------------------|-----------|-----------|-------|
| KAG (Liang et al., 2024) | 137.99 | 218.32 | 44.8 |
| GraphRAG (Han et al., 2024) | 72.31 | 46.65 | 52.25 |
| Ours | 59.95 | 38.66 | 60.3 |

It is worth noting that the EM results for KAG and GraphRAG do not represent the performance of their independent, complete algorithms in multimodal question answering. This is because the original algorithms of KAG and GraphRAG lack multimodal mechanisms and are primarily focused on unimodal information processing. Therefore, to evaluate the performance of different graph construction methods on the MultimodalQA dataset and simulate the application of KAG and GraphRAG in multimodal scenarios under as fair conditions as possible, the graph construction meth-

ods of KAG and GraphRAG were evaluated within our framework.

4.2 Ablation Study

RQ2: How does the multimodal selective attention mechanism integrate textual, visual, and tabular data?

Our ablation study on the MultimodalQA dataset shown in 3 reveals the significant contribution of each component in enhancing multimodal reasoning. Starting with the baseline model (F1: 56.07%, EM: 48.60%), we progressively added components to assess their impact. Visual evidence enhancement alone led to modest improvements (F1: 56.81%, EM: 50.04%), and similar enhancements were observed with tabular relationship mining (F1: 57.63%, EM: 50.55%). However, when both image and table supplementation were combined, the performance saw a substantial boost (F1: 62.63%, EM: 54.97%). The integration of the image matching mechanism within the visual evidence integration

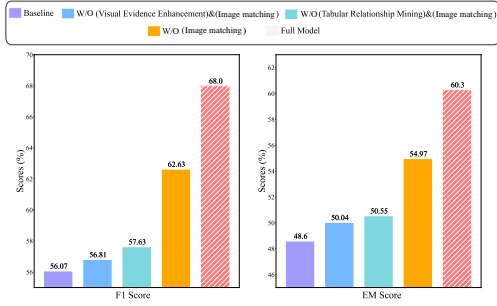


Figure 3: Progressive Ablation Study on MultimodalQA: Evaluating Framework Performance on F1 and EM Scores through Incremental Addition of (1) Visual Evidence Enhancement, (2) Tabular relationship Mining, (1) & (2) Combined, and (3) Image Matching.

(Full model) yielded the highest performance (F1: 68.00%, EM: 60.30%), demonstrating its critical role in improving the model’s accuracy. These results, illustrated in Fig. 3, provide empirical evidence for the effectiveness of the multimodal selective attention mechanism and highlight the synergistic effect of combining image and table supplementation with image matching. The ablation study confirms that each component contributes meaningfully to the overall system performance, with the image matching mechanism providing the most significant performance gains.

4.3 Case Study Analysis

RQ3: *How does iterative knowledge refinement enhance the answer accuracy?*

Fig. 4 demonstrates how our dynamic local KG construction enhances accuracy while reducing search scope. The reasoning process begins with the text-derived node "Ben Piazza" (upper left), which dynamically connects to tabular data containing film metadata. Through dashed-line connections to the "Movie Table" node, three candidate movie posters semantically matching the query description "half of a woman’s face on the poster" are retrieved. While CLIP embeddings alone fail to distinguish these visually similar candidates, our method resolves this ambiguity through iterative refinement: in Stage III, each poster undergoes dedicated multimodal analysis via VLM to extract discriminative semantic features, enabling precise identification of "Tell Me That You Love Me, Junie Moon" (1970).

This case highlights three key advantages of our framework: (1) Dynamic Multimodal Integration: The KG evolves from the initial text nodes to the correct evidence graph through tabular linking and

| Question | Which film did Ben Piazza appear in first: "Nightwing" or the movie that shows half of a woman's face on the poster? | |
|------------------|--|-------------|
| Method | Ours | MOQAGPT |
| Key Evidence | | |
| Predicted Answer | Tell Me That You Love Me, Junie Moon ✓ | Nightwing ✗ |

Figure 4: Comparative analysis of knowledge refinement strategies. Left: Our clue graph, including text-derived relationships (solid edges) and augmented tabular and visual connections (dashed edges). Right: MOQAGPT’s static retrieval results.

visual verification, achieving temporal relationship completeness. (2) Computational Efficiency: Localized processing reduces candidate nodes. As a comparison, MOQAGPT’s static approach suffers exhaustive embedding comparisons across all modalities. (3) Our implementation effectively operationalizes the Neural Efficiency Principle (Definition 2) through a dual-strategy approach: We first apply computationally efficient retrieval methods to rapidly narrow the local KG scope. Then for challenging cases (e.g., the three visually similar posters), we allocate substantial computational resources, employing VLMs for fine-grained semantic alignment. This hierarchical workflow achieves better accuracy while maintaining lower computational costs than exhaustive search methods like MOQAGPT. Thus, The neural efficiency paradigm enables balanced precision-efficiency tradeoffs.

5 Conclusion

This paper proposes a new approach to enhance LLM performance in complex multimodal question answering tasks. The proposed query-driven local multimodal KG construction approach is inspired by cognitive science modeling of goal-directed user behaviors and their selective attention. Our approach realizes efficient multimodal knowledge acquisition and integration through three key components, including graph pattern construction, multi-path retrieval, and dynamic reasoning. Experimental results on two challenging datasets, MultimodalQA and WebQA, demonstrate that our proposal is on a par with the state-of-the-art supervised methods while outperforming its unsupervised competitors by considerable margins.

Limitations

First, the current methodology exclusively adopts a query-driven (top-down) approach for KG construction. While effective for targeted question answering, this paradigm neglects potential synergies with existing data-driven (bottom-up) KGs that capture comprehensive domain relationships. In real-world scenarios where preconstructed KGs already exist (e.g., domain-specific graphs), integrating our query-driven method as a supplementary layer could enhance both precision and efficiency. For instance, hybrid approaches could first retrieve relevant subgraphs from existing KGs then apply dynamic refinement through query patterns, potentially reducing construction time.

Second, the framework demonstrates reduced effectiveness when processing vague or under-specified queries. While our pattern graph mechanism effectively handles focused questions, it lacks explicit mechanisms for query clarification or intent decomposition. Incorporating chain-of-thought prompting or question disentanglement techniques could help resolve ambiguities - preliminary tests show that adding a query refinement module improves EM scores on ambiguous WebQA questions. Future work should explore dynamic switching between direct answering and clarification dialogues based on query specificity.

Third, due to current dataset limitations, our experiments only consider table, image, and text modalities. In the future, we plan to explore experiments involving additional modalities, such as video (Liu et al., 2017, 2024b).

6 Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant No. 61806065, 62120106008, and 62406095), the Anhui Provincial Science and Technology Fortification Plan (Grant No. 202423k09020015), the Youth Talent Support Program of the Anhui Association for Science and Technology (Grant No. RCTJ202420), and the Natural Science Foundation of Anhui Province (Grant No. 2308085MF213). The authors also appreciate the partial support from Hefei AI Computing Center (Project Team). Y. He was not supported by any of these funds.

References

- Farhan Baluch and Laurent Itti. 2011. Mechanisms of top-down attention. *Trends in Neurosciences*, 34(4):210–224.
- Yingshan Chang, Mridu Narang, Hisami Suzuki, Guihong Cao, Jianfeng Gao, and Yonatan Bisk. 2022. Webqa: Multihop and multimodal qa. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16495–16504. IEEE.
- Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024. [Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation](#). *Preprint*, arXiv:2402.03216.
- Wenhu Chen, Hexiang Hu, Xi Chen, Pat Verga, and William W Cohen. 2022. Murag: Multimodal retrieval-augmented generator for open question answering over images and text. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5558–5570. ACL.
- Xin Cheng, Xun Wang, Xingxing Zhang, Tao Ge, Si-Qing Chen, Furu Wei, Huishuai Zhang, and Dongyan Zhao. 2024. xrag: Extreme context compression for retrieval-augmented generation with one token. In *Proceedings of the International Conference on Neural Information Processing Systems*, pages 109487–109516. NeurIPS.
- Zhoujun Cheng, Tianbao Xie, Peng Shi, Chengzu Li, Rahul Nadkarni, Yushi Hu, Caiming Xiong, Dragomir Radev, Mari Ostendorf, Luke Zettlemoyer, et al. 2023. Binding language models in symbolic languages. In *Proceedings of the International Conference on Learning Representations*. OpenReview.net.
- Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. [Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality](#).
- William W. Cohen, Wenhu Chen, Michiel de Jong, Nitish Gupta, Alessandro Presta, Pat Verga, and John Wieting. 2023. QA is the new kr: Question-answer pairs as knowledge bases. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 15385–15392. AAAI Press.
- 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*.
- Zirui Guo, Lianghao Xia, Yanhua Yu, Tu Ao, and Chao Huang. 2024. Lightrag: Simple and fast retrieval-augmented generation. *arXiv preprint arXiv:2410.05779*.

- Deepak Gupta, Surabhi Kumari, Asif Ekbal, and Pushpak Bhattacharyya. 2018. Mmqa: A multi-domain multi-lingual question-answering framework for english and hindi. In *Proceedings of the International Conference on Language Resources and Evaluation*, pages 1–17. ELRA.
- Haoyu Han, Yu Wang, Harry Shomer, Kai Guo, Jiayuan Ding, Yongjia Lei, Mahantesh Halappanavar, Ryan A Rossi, Subhabrata Mukherjee, Xianfeng Tang, et al. 2024. [Retrieval-augmented generation with graphs \(graphrag\)](#). *arXiv preprint arXiv:2501.00309*.
- Xuehai He and Xin Eric Wang. 2023. Multimodal graph transformer for multimodal question answering. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics*, pages 189–200. ACL.
- Yushi Hu, Hang Hua, Zhengyuan Yang, Weijia Shi, Noah A. Smith, and Jiebo Luo. 2023. Promptcap: Prompt-guided image captioning for vqa with gpt-3. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2963–2975.
- Manzong Huang, Chenyang Bu, Yi He, and Xindong Wu. 2025. How to mitigate information loss in knowledge graphs for graphrag: Leveraging triple context restoration and query-driven feedback. In *Proceedings of the 34th International Joint Conference on Artificial Intelligence*. IJCAI.
- Philip Nicholas Johnson-Laird. 1983. *Mental models: Towards a cognitive science of language, inference, and consciousness*. 6. Harvard University Press.
- Yuri Kuratov, Aydar Bulatov, Petr Anokhin, Dmitry Sorokin, Artyom Sorokin, and Mikhail Burtsev. 2024. [In search of needles in a 10m haystack: Recurrent memory finds what llms miss](#). *arXiv preprint arXiv:2402.10790*.
- Md. Tahmid Rahman Laskar, Enamul Hoque, and Jimmy X. Huang. 2020. [Query focused abstractive summarization via incorporating query relevance and transfer learning with transformer models](#). In *Proceedings of the 33rd Canadian Conference on Artificial Intelligence (Canadian AI 2020)*, volume 12109 of *Lecture Notes in Computer Science*, pages 342–348. Springer.
- Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Proceedings of the International Conference on Neural Information Processing Systems*, pages 9459–9474. Curran Associates Inc.
- Lei Liang, Mengshu Sun, Zhengke Gui, Zhongshu Zhu, Zhouyu Jiang, Ling Zhong, Yuan Qu, Peilong Zhao, Zhongpu Bo, Jin Yang, et al. 2024. [Kag: Boosting llms in professional domains via knowledge augmented generation](#). *arXiv preprint arXiv:2409.13731*.
- Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2024a. Visual instruction tuning. In *Proceedings of the International Conference on Neural Information Processing Systems*, pages 34892–34916. Curran Associates Inc.
- Mengyuan Liu, Hong Liu, and Chen Chen. 2017. Enhanced skeleton visualization for view invariant human action recognition. *Pattern Recognition*, 68:346–362.
- Mengyuan Liu, Hong Liu, and Tianyu Guo. 2024b. Cross-model cross-stream learning for self-supervised human action recognition. *IEEE Transactions on Human-Machine Systems*.
- Weihao Liu, Fangyu Lei, Tongxu Luo, Jiahe Lei, Shizhu He, Jun Zhao, and Kang Liu. 2023. [Mmqh-icl: Multimodal in-context learning for hybrid question answering over text, tables and images](#). *arXiv preprint arXiv:2309.04790*.
- Aljoscha C Neubauer and Andreas Fink. 2009. Intelligence and neural efficiency. *Neuroscience & Biobehavioral Reviews*, 33(7):1004–1023.
- Shirui Pan, Linhao Luo, Yufei Wang, Chen Chen, Jiapu Wang, and Xindong Wu. 2024. Unifying large language models and knowledge graphs: A roadmap. *IEEE Transactions on Knowledge and Data Engineering*, 36(7):3580–3599.
- Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *Proceedings of the International Conference on Machine Learning*, pages 8748–8763. PMLR.
- Diego Sanmartin. 2024. [Kg-rag: Bridging the gap between knowledge and creativity](#). *arXiv preprint arXiv:2405.12035*.
- Qi Shi, Han Cui, Haofeng Wang, Qingfu Zhu, Wanxiang Che, and Ting Liu. 2024. Exploring hybrid question answering via program-based prompting. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11035–11046. ACL.
- Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, et al. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*.
- Jan Theeuwes. 2010. Top-down and bottom-up control of visual selection. *Acta Psychologica*, 135(2):77–99.
- Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. [Llama 2: Open foundation and fine-tuned chat models](#). *CoRR*, abs/2307.09288.

- Guan Wang, Sijie Cheng, Xianyuan Zhan, Xiangang Li, Sen Song, and Yang Liu. 2024. Openchat: Advancing open-source language models with mixed-quality data. In *Proceedings of the Twelfth International Conference on Learning Representations*. OpenReview.net.
- Peng Wang, An Yang, Rui Men, Junyang Lin, Shuai Bai, Zhikang Li, Jianxin Ma, Chang Zhou, Jingren Zhou, and Hongxia Yang. 2022. Ofa: Unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework. In *Proceedings of the International Conference on Machine Learning*, pages 23318–23340. PMLR.
- Jilan Xu, Yifei Huang, Junlin Hou, Guo Chen, Yuejie Zhang, Rui Feng, and Weidi Xie. 2024. Retrieval-augmented egocentric video captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13525–13536. IEEE.
- An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zhihao Fan. 2024. [Qwen2 technical report](#). *CoRR*, abs/2407.10671.
- Qian Yang, Qian Chen, Wen Wang, Baotian Hu, and Min Zhang. 2023. Enhancing multi-modal multi-hop question answering via structured knowledge and unified retrieval-generation. In *Proceedings of the ACM International Conference on Multimedia*, pages 5223–5234. ACM.
- Jin-ge Yao, Xiaojun Wan, and Jianguo Xiao. 2017. Recent advances in document summarization. *Knowledge and Information Systems*, 53(2):297–336.
- Ori Yoran, Alon Talmor, and Jonathan Berant. 2021. Turning tables: Generating examples from semi-structured tables for endowing language models with reasoning skills. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6016–6031. ACL.
- Le Zhang, Yihong Wu, Fengran Mo, Jian-Yun Nie, and Aishwarya Agrawal. 2023. Moqagpt: Zero-shot multi-modal open-domain question answering with large language model. In *Proceedings of Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 1195–1210. ACL.




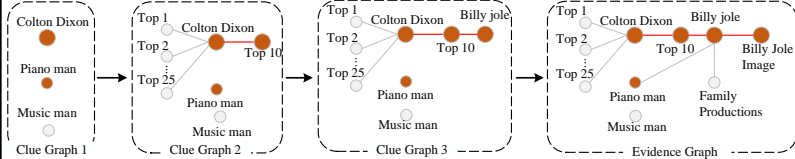
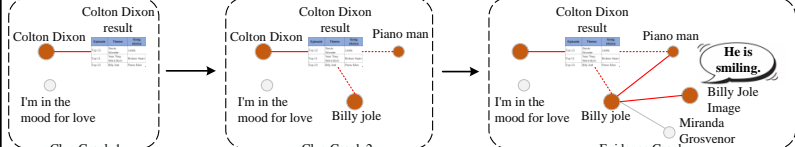
| Query | | To indicate he is happy, what does the person who was the theme for the episode where Colton Dixon chose "Piano Man" do? | | | | | | | | | | | | |
|-------------------|--|---|-------------|-------|-------------|--------|---------------|--------|--------|---------------------|--------------|--------|------------|-----------|
| Evidence & answer | RAG | <div style="display: flex; align-items: flex-start;"> <div style="border: 1px solid black; padding: 5px; margin-right: 10px;"> <p>[DOCS 1] Piano Man is a [DOCS 2] Drama critic</p> <p>[DOCS 3] The Music Man is a musical with book, music, and lyrics by...</p> </div> <div style="display: flex; gap: 10px;">    </div> <div style="margin-left: 10px;"> <table border="1" style="font-size: 8px;"> <thead> <tr> <th>Episode</th> <th>Theme</th> <th>Sung choice</th> </tr> </thead> <tbody> <tr> <td>Top 13</td> <td>Stevie Wonder</td> <td>Lately</td> </tr> <tr> <td>Top 11</td> <td>Year They Were Born</td> <td>Broken Heart</td> </tr> <tr> <td>Top 10</td> <td>Billy Joel</td> <td>Piano Man</td> </tr> </tbody> </table> </div> </div> <p style="text-align: right;">Material missing.</p> | Episode | Theme | Sung choice | Top 13 | Stevie Wonder | Lately | Top 11 | Year They Were Born | Broken Heart | Top 10 | Billy Joel | Piano Man |
| | Episode | Theme | Sung choice | | | | | | | | | | | |
| | Top 13 | Stevie Wonder | Lately | | | | | | | | | | | |
| Top 11 | Year They Were Born | Broken Heart | | | | | | | | | | | | |
| Top 10 | Billy Joel | Piano Man | | | | | | | | | | | | |
| GraphRAG |  <p style="text-align: right;">Image Understanding Failure</p> | | | | | | | | | | | | | |
| Ours |  <p style="text-align: right;">Smiling.</p> | | | | | | | | | | | | | |

Figure A1: Case Study Comparison with RAG and GraphRAG, illustrating how the KG construction methods impact reasoning efficiency and accuracy. The analysis reveals that 1) RAG struggles with connecting fragmented knowledge, leading to incomplete reasoning; 2) GraphRAG introduces redundancy and lacks effective image interpretation; 3) Our method, by dynamically refining the KG with image and table understanding, improves reasoning efficiency and accuracy, as shown in the experimental results.

Supplementary Material

Our appendix is divided into two sections: the first section presents additional experiments, while the second section contains the prompt templates.

A Additional Experiments

A.1 Case study Comparison with RAG and GraphRAG

Experimental Setup. To comprehensively evaluate the performance of our proposed Query-Driven Multimodal GraphRAG framework and ensure a fair comparison, we replicated two representative baseline methods: Traditional Retrieval-Augmented Generation (RAG) and Knowledge Graph-enhanced Retrieval-Augmented Generation (GraphRAG). For the Traditional RAG method, we endeavored to simulate its typical retrieval process. Initially, we employed the same BGE-M3 vector encoder (Chen et al., 2024) as our proposed method to encode document content, image captions, and table Markdown descriptions. This step aimed to map data from different modalities into a unified semantic space. Subsequently, we computed the cosine similarity between the query vector and these multimodal data vectors to assess their relevance to the query. Finally, based on the similarity scores, we performed Top-K retrieval (with K values of 3, 3, and 1 for documents, images, and tables, respectively) and concatenated the retrieved Top-K evidence materials as the final evidence input.

For the Knowledge Graph-enhanced Retrieval-Augmented Generation (GraphRAG) method, our objective was to faithfully reproduce its KG-enhanced characteristics. In the graph construction phase, we adopted the text prompting strategies proposed in the GraphRAG (Han et al., 2024) paper to guide a Large Language Model in extracting entities and relationships from document and table data, thereby constructing structured text and table KGs. Notably, for the image modality, to ensure fairness in comparison and to align as closely as possible with the GraphRAG approach to graph construction, we also utilized the same strategy as employed in Stage II (Retrieval Graph Formation) of our proposed method to construct a multimodal KG.

The most significant difference between the GraphRAG method and our proposed method lies in the reasoning process. GraphRAG still adopts an iterative approach to construct and optimize the evidence graph, but its reasoning and graph expansion are entirely dependent on the initially constructed static KG, lacking the ability to dynamically adapt to query demands and supplement fine-grained information during

Query: To indicate he is happy, what does the person who was the theme for the episode where Colton Dixon chose Piano Man do?

Title: A Piano in the House
Drama critic Fitzgerald Fortune (Barry Morse), a caustic and cruel man, goes to Throckmorton's Curio Shop to buy his wife Esther a player piano as a 26th birthday present. The grouchy owner (Philip Coolidge) demonstrates the piano by placing a roll of music inside. As it plays I'm in the Mood for Love, he begins speaking in a gentle, sentimental manner, even giving Fitzgerald a 20% discount because it is a gift. When the music stops, the owner resumes his ill-tempered sniping.

Title: Charlie Karp
Charles Karp (April 13, 1953 – March 10, 2019) was an American musician and Emmy Award-winning documentarian. A former student at Coleytown Middle School and Staples High School, both in Westport, Connecticut. He left school as a senior to pursue music. Karp had a professional career that stretched nearly 50 years. He was an Emmy producing music for films and television, and his biography credits him as working on jingles for such products as Twix candy bars, US Tobacco and Xerox. He died at age 65.

Title: His Musical Career
Charlie and his partner Mike work at a piano store, whose manager orders them to deliver a piano to Mr. Rich at 666 Prospect Street and reposes one from Mr. Poor at 999 Prospect Street. Hilarity ensues when they do exactly the opposite after mixing up the addresses of their customers.

Title: Piano Man (song)
"Piano Man" is a song written and performed by American singer-songwriter Billy Joel. His first single in North America, it was included on Joel's 1973 album of the same name and later released as a single on November 2, 1973. The song is sung from Joel's point-of-view as a piano player at a bar, reminiscing about his experiences there and the people he encountered. "Piano Man" is based on Joel's real-life experiences as a lounge musician in Los Angeles from 1972–73, which he had decided to pursue in an effort to escape his contracted New York-based record company at the time, Family Productions, following the poor commercial performance of the album "Cold Spring Harbor". Joel describes various characters, including a bartender named John and a "real-estate novelist," named Paul, all based on real-life individuals.


Title: Mr. Music
New York theater producer Alex Conway (Charles Coburn) travels with composer Paul Merrick (Bing Crosby) to Lawford College. Paul's alma mater, where one of his musicals is being revived by the students. The current campus hero is handsome athlete Jefferson Blake (Robert Stack), so Katherine Holbrook (Nancy Olson), class valedictorian and chairman of the welcoming committee for returning alumni, asks Paul to work in a phrase about Jeff in one of his songs. Paul balks at the suggestion, but Kate's matter-of-fact manner leaves no room for discussion.

Title: Mr. Tanner
The song tells the story of Martin Tanner, a local laundrer from Dayton, Ohio, who has a gift for singing. His friends try to talk him into becoming a singer because of his beautiful voice, until he finally agrees and uses most of his savings to travel to New York City and sing in a show. He holds a concert only to get panned by critics. He returns home and never sings again, except for only to himself when he sorts through the clothes at night.


Title: The Music Man
The Music Man is a musical with book, music, and lyrics by Meredith Willson, based on a story by Willson and Franklin Lacey. The plot concerns con man Harold Hill, who poses as a boys' band organizer and leader and sells band instruments and uniforms to naive Midwestern townsfolk, promising to train the members of the new band. Harold is no musician, however, and plans to skip town without giving any music lessons. Prim librarian and piano teacher Marian sees through him, but when Harold helps her younger brother overcome his limp and social awkwardness, Marian begins to fall in love. Harold risks being caught to win her.

Title: The Musical Man
In the episode, Cameron takes control of the spring musical at Lake and Manny's school, while Jay's brother pays him a visit, and Phil tries to get the family to be in his new real estate advertisement.

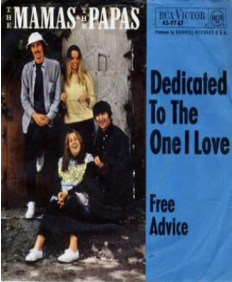
Title: Miranda Grosvenor
As "Miranda," Walton engaged in late-night telephone conversations with such stars as Billy Joel, Warren Beatty, Bob Dylan, Buck Henry, Eric Clapton, Michael Apted, Bono, Mike Nichols, Vitas Gerulaitis, Ted Kennedy, Johnny Carson, Art Garfunkel, Peter Gabriel, Robert De Niro, Rush Limbaugh, and Richard Gere, telling them that she was a blonde Tulane University student, wealthy socialite, and international model. According to "Vanity Fair", at least two of her telephone paramours, Quincy Jones and Richard Perry, proposed marriage. Billy Joel wrote songs which he sang on Miranda's answering machine, considered her at times his "only friend," and considered writing a musical about her. Many others bought her jewelry and sent her plane tickets. Novelist Kinky Friedman created a Miranda character in his detective novels.




Billy Joel



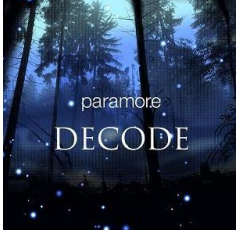
Stevie Wonder




Dedicated to the One I Love




September (Earth, Wind & Fire song)



Decode (song)



Paramore



1980s in music

| Episode | Theme | Song choice | Result |
|-------------------------|-----------------------|--|------------|
| Audition | Auditioner's Choice | "Permanent" | Advanced |
| Hollywood Round, Part 1 | First Solo | "Only Hope" | |
| Hollywood Round, Part 2 | Group Performance | Not aired | |
| Hollywood Round, Part 3 | Second Solo | "What About Now" | |
| Las Vegas Round | Songs from the 1950s | "Dedicated to the One I Love" | |
| Final Judgment | Final Solo | "Fix You" | Safe |
| Top 25 (13 Men) | Personal Choice | "Decode" | |
| Top 13 | Stevie Wonder | "Lately" | |
| Top 11 | Year They Were Born | "Broken Heart" | |
| Top 10 | Billy Joel | "Piano Man" | |
| Top 9 | Their Personal Idols | Solo "Everything" Trio "Landslide" "Edge of Seventeen" "Don't Stop" with Phillip Phillips & Elise Testone | |
| Top 8 | Songs from the 1980s | Duet "Islands in the Stream" with Skylar Laine Solo "Time After Time" | |
| Top 7 | Songs from the 2010s | Solo "Love the Way You Lie" | |
| | | Duet "Don't You Wanna Stay" with Skylar Laine | |
| | | Duet "Bad Romance" | |
| | Songs from Now & Then | "September" | Eliminated |

Colton Dixon Performances Table

Figure A2: The query and original resource database, including text, tables, and image materials. The red boxes highlight the key information critical for answering the query.

inference. To more clearly showcase the inherent reasoning capability of the GraphRAG baseline method and to facilitate effective comparison with our proposed approach, we specifically retained its original, static graph-based reasoning mechanism in the GraphRAG reasoning process, while removing the dynamic table and image information supplementation mechanisms unique to our method. Finally, similar to the

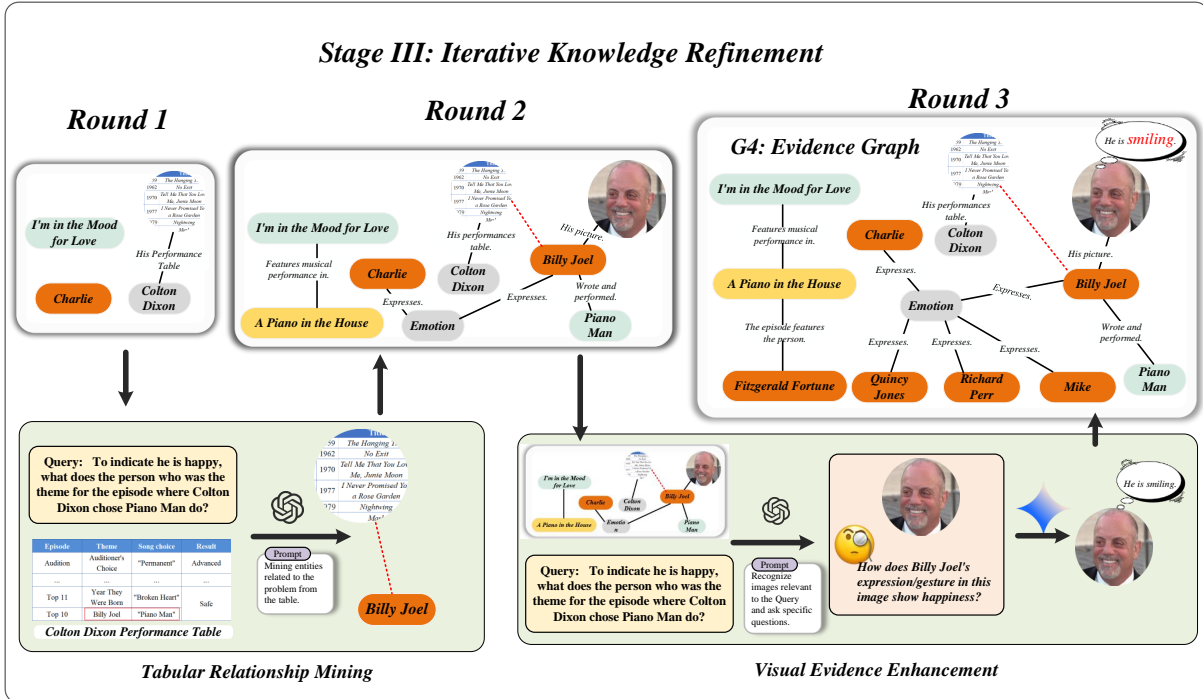
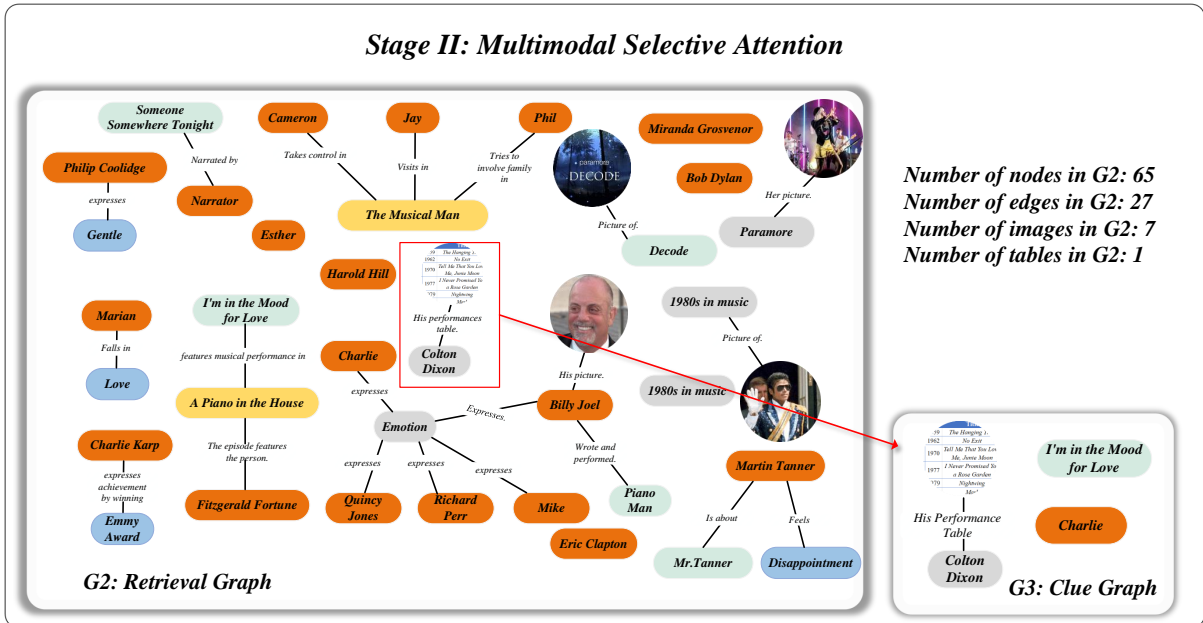
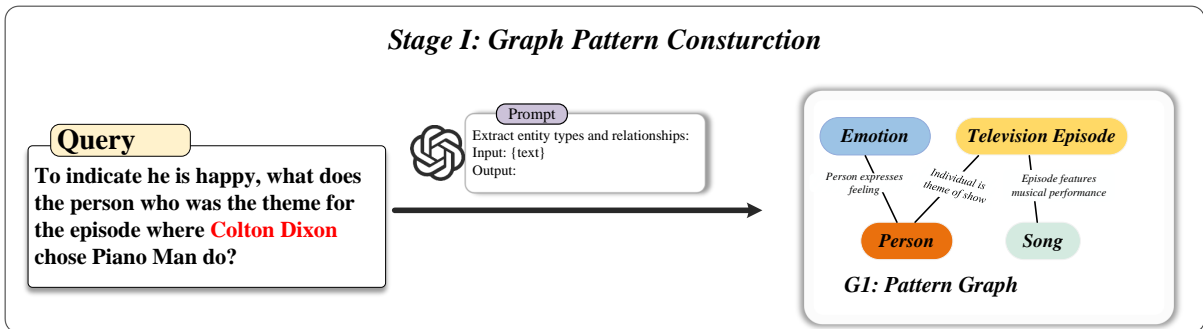


Figure A3: Complete reasoning pipeline of our proposed method for solving multimodal question answering, demonstrating the data flow through three stages: Pattern Graph Construction, Multimodal Selective Attention, and Iterative Knowledge Refinement.

RAG method, we input the evidence graph constructed by GraphRAG at the end of iteration into GPT-4 to perform reasoning and generate the final answer. To ensure a fair comparison, all methods (including our proposed method, RAG, and GraphRAG) all utilized the same GPT-4 model in the reasoning and answer generation phase.

Results and Findings. Our experiments reveal fundamental limitations in traditional RAG and GraphRAG approaches for complex multimodal reasoning. As demonstrated in Fig. A1 (analyzing the query: "To indicate he is happy, what does the person who was the theme for the episode where Colton Dixon chose 'Piano Man' do?"), three critical insights emerge: (1) Traditional RAG Failure: Vector/keyword-based retrieval (top panel) fails to capture cross-modal relationships, missing crucial visual evidence - particularly the relevant facial expression in image data. (2) GraphRAG Limitations: While constructing KGs (middle panel) improves entity linking, its static approach creates oversized graphs (382 nodes shown) with redundant connections. More critically, it lacks visual understanding capabilities, unable to decode the smiling expression in the 'Billy Joel Image'. (3) Our Method's Advantage: The bottom panel demonstrates our framework's success through: dynamic graph construction, Tabular reasoning, and Visual evidence integration. This case study confirms our method's ability to synthesize tabular and visual evidence, correctly identifying the 'Smiling' response through multimodal alignment. The error reduction compared to GraphRAG highlights the effectiveness of query-driven graph construction and iterative visual verification.

Study Case. We present a case study to demonstrate the capability of our method in handling complex multimodal reasoning tasks. The query "To indicate he is happy, what does the person who was the theme for the episode where Colton Dixon chose Piano Man do?" requires integrating textual, visual, and tabular data, as shown in Fig. A2. To answer this question, our approach decomposes the reasoning process into three stages, as illustrated in Fig. A3.

In Stage I (Pattern Graph Construction), we first identify key entity types from the query, including emotion, TV show, person, and song. Based on these entities, we construct an initial pattern graph G1 to guide the KG construction from the database.

In Stage II (Multimodal Selective Attention), we use G1 to direct the LLM in constructing a retrieval graph G2, which consists of 65 nodes and 27 edges, including 7 images and 1 table. Through query-focused clue refinement, we retrieve the key starting nodes for answering the question: Colton Dixon and Colton Dixon's performance table. Based on this, we filter out redundant information from G2 and construct a refined clue graph G3, which retains only the most relevant details to the query.

In Stage III (Iterative Knowledge Refinement), we further enrich and expand the KG by analyzing tables and images. Specifically, through table mining, we identify relevant tabular information that reveals the connection between Piano Man and its theme artist Billy Joel, and we locate his related image through graph expansion operators. Next, we leverage visual evidence enhancement to analyze Billy Joel's image and generate a sub-query: "How does Billy Joel's expression/gesture in this image indicate happiness?" Finally, as shown in G4, our model concludes that Billy Joel expresses happiness through smiling.

The case study illustrates how our method effectively handles complex multimodal question-answering tasks through a structured three-stage reasoning process. First, pattern graph construction identifies key entities and their relationships, defining the problem framework. Next, multimodal selective attention integrates relevant information from multiple sources while filtering out redundancy. Finally, iterative knowledge refinement enhances reasoning by incorporating table mining and visual evidence. This structured approach improves answer accuracy and reliability while enhancing interpretability, offering a robust paradigm for complex multimodal reasoning tasks.

A.2 Additional Experiments on Single-modal QA Datasets

We compared our method with the recently proposed retrieval-augmented generation benchmark Xrag (Cheng et al., 2024) on two single-modality datasets, i.e., HotpotQA and Drop. Xrag retrieves relevant data from the knowledge base by calculating the similarity between the query and the data and then generates an answer. In contrast, our method employs the same retrieval strategy but constructs a graph from the retrieved content before generating the answer. As shown in Table A1, our proposed method

Table A1: Performance Comparison on Drop and HotpotQA Datasets. Character-level n-gram metrics ChrF and ChrF++ capture subword-level similarities. METEOR evaluates semantic adequacy through flexible matching mechanisms. R1, R2, and RL represent ROUGE scores measuring unigram overlap, bigram overlap, and longest common subsequence, respectively. EM denotes the Exact Match accuracy rate. PPL indicates the language model perplexity score. CER and WER measure error rates at character and word levels, respectively.

| Data | Methods | ConG | | | | | | | | | |
|----------|---------------------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|---------------|---------------|---------------|
| | | ChrF | ChrF++ | METEOR | R1 | R2 | RL | EM | PPL↓ | CER↓ | WER↓ |
| HotpotQA | XRAG[2024](GPT-3.5 Turbo) | 15.49 | 18.54 | 16.35 | 27.71 | 14.44 | 21.57 | 14.66 | 133.73 | 0.8714 | 0.9570 |
| | XRAG(Gemini) | 15.36 | 18.47 | 15.91 | 25.57 | 14.29 | 21.00 | 14.14 | 129.25 | 0.8753 | 0.9692 |
| | XRAG(Llama3.1-8B) | 14.60 | 17.52 | 15.30 | 25.02 | 13.42 | 20.56 | 15.71 | 136.37 | 0.8689 | 0.9554 |
| | XRAG(GPT-4o Mini) | 15.52 | 18.65 | 16.37 | 25.68 | 14.81 | 21.21 | 17.80 | 131.28 | 0.8670 | 0.9540 |
| | Ours(Qwen 2.5-72B) | 31.20 | 30.82 | 23.66 | 33.26 | 16.57 | 33.24 | 26.53 | 354.53 | 0.9895 | 0.9959 |
| Drop | XRAG(GPT-3.5 Turbo) | 25.78 | 23.85 | 15.34 | 21.74 | 05.28 | 21.74 | 21.04 | 322.77 | 1.5747 | 1.1400 |
| | XRAG(Gemini) | 26.35 | 24.40 | 15.39 | 22.50 | 05.11 | 22.50 | 22.24 | 322.86 | 1.4352 | 1.0743 |
| | XRAG(Llama3.1-8B) | 26.03 | 24.06 | 15.24 | 21.62 | 05.02 | 21.62 | 21.44 | 314.41 | 1.4849 | 1.0905 |
| | XRAG(GPT-4o Mini) | 26.36 | 24.36 | 15.41 | 22.57 | 05.39 | 22.57 | 24.25 | 316.52 | 1.5361 | 1.1397 |
| | Ours(Qwen 2.5-72B) | 33.20 | 32.96 | 17.14 | 29.91 | 08.64 | 29.83 | 26.67 | 301.38 | 0.9922 | 0.9980 |

significantly outperforms Xrag in most metrics. For example, on the HotpotQA dataset, our method achieves a ChrF score of 31.20, compared to Xrag’s best ChrF score of 15.62 (using GPT-4o Mini). Similarly, for the Drop dataset, our method obtains a ChrF score of 33.20, while Xrag’s best ChrF score is 26.36 (using GPT-4o Mini). These results demonstrate that, our method, originally designed for multimodal scenarios, also demonstrates superior performance in single-modal settings compared to Xrag, which was specifically designed for single-modal tasks. This indicates that the graph construction approach in our method effectively enhances the reasoning capabilities of the model, even when only a single modality is involved. The significant improvements across various metrics on both HotpotQA and Drop datasets highlight the robustness and adaptability of our proposed framework.

B Prompt Templates

This section documents the prompt templates used on MultimodalQA dataset in our method.

Prompt Template For Pattern Graph Construction

```
\begin{prompt}
-Goal-
Entity and Relation Extraction from Questions: Given a question, identify the relevant entity types
and the relationships between these entities. Then, return this information in a standardized
format using specific delimiters. Ensure that all identified entities and relations are
directly relevant to answering the given question, avoiding extraneous information.
-Steps-
1. Read the given question carefully.
2. Identify all relevant entity types mentioned or implied in the question. Derive them from the
context of the question.
3. Identify all potential relationships between the identified entity types. These relationships
should be directional and specific.
4. Use specific and descriptive relationship descriptions.
5. Use ## as the list delimiter.
6. Format the output as follows:
Entity types: [entity\_type1, entity\_type2, ...]
\texttt{"relationship" <|> [source\_entity] <|> [target\_entity] <|> [relationship\_description]}
#####
-Examples-
#####
Example 1:
Input: "For which film did Ben Piazza play the role of Mr. Simms?"
#####
Output:
Entity types: [performer, film, role]##
("relationship"<|>performer<|>film<|>actor performs in movie)##
("relationship"<|>film<|>role<|>movie contains character)##
("relationship"<|>performer<|>role<|>actor portrays character)<|COMPLETE|>
#####
```



```

Example 2:
Input: "Who directed the movie 'Inception' and when was it released?"
#####
Output:
Entity types: [person, movie, date]##
("relationship"<|>person<|>movie<|>director creates film)##
("relationship"<|>movie<|>date<|>film has release date)<|COMPLETE|>
#####
-Real Data-
#####
Input: {question}
#####
Output:
\end{prompt}

```

Textual Extraction Prompt Template

```

-Goal-
Entity and Relation Extraction from Documents: Given a text document and a graph pattern (which
includes entity types and relationship types), identify and extract only the entities and
relationships that are explicitly mentioned in the text and match the given pattern.

-Steps-
1. Carefully read the provided input text.
2. Analyze the given graph pattern to understand the required entity types and relationships.
3. Extract only the entities from the input text that are explicitly mentioned and match the
specified entity types in the graph pattern. For each extracted entity, provide a description
based solely on the information given in the text.
4. Identify only the relationships between the extracted entities that are explicitly stated in the
text and match the relationship types specified in the graph pattern.

Important: Only extract entities and relationships that are explicitly stated in the text. Do not
infer, assume, or create any entities or relationships that are not directly mentioned, even if
they seem logical or likely.

```

```

Output Format
For each entity:
("entity" <|> <entity_name> <|> <entity_type> <|> <entity_description>)
For each relationship:
("relationship" <|> <source_entity> <|> <target_entity> <|> <relationship_description>)
Use ## as the list delimiter between entries.
Always end the output with <|COMPLETE|>.

```

```

If no matches are found, the output should be:
<|COMPLETE|>
No entities or relationships matching the graph pattern were found in the given text. [Your
explanation here]
#####

```

```

-Examples-
#####
Example 1:

```

```

Input text:
G.E.M. performed "Tornado" at the concert, and Xue Zhiqian sang "Ugly" at the concert.
Graph pattern:
Entity types: [singer, song]
("relationship"<|>singer<|>song<|>singer performs song)
#####
Output:
("entity"<|>"Tornado"<|>"song"<|>"Tornado is a song originally created by Jay Chou and covered by
G.E.M.")##
("entity"<|>"Ugly"<|>"song"<|>"Ugly is a song by Xue Zhiqian")##
("entity"<|>"Xue Zhiqian"<|>"singer"<|>"Xue Zhiqian is a well-known singer from mainland China")##
("entity"<|>"G.E.M."<|>"singer"<|>"G.E.M. is a well-known singer from mainland China")##
("relationship"<|>"Xue Zhiqian"<|>"Ugly"<|>"Xue Zhiqian performed 'Ugly' at a certain concert on a
certain date")##
("relationship"<|>"G.E.M."<|>"Tornado"<|>"G.E.M. performed 'Tornado' at a certain concert on a
certain date")<|COMPLETE|>
#####
-Real Data-
#####

```

Text: {input_text}
Graph pattern:
{Graph_pattern}

Output:

Image Description Extraction Prompt Template

-Goal-

You are an AI assistant tasked with extracting explicit image descriptions and visual characteristics from given questions or tasks.

-Steps-

1. Carefully read and analyze the provided question or task.
2. Identify any explicit descriptions of images, visual elements, or distinctive visual characteristics that could be used to identify elements in an image.
3. Return a list of strings, where each string is an extracted image description or visual characteristic. If no relevant descriptions are present, return an empty list.
4. Wrap the final output list with <|Answer|> and <|\Answer|> tags.
5. The list conforms to Python syntax, using double quotation marks to wrap each item.

Important:

1. Extract descriptions that are clearly referring to visual elements, images, or distinctive visual characteristics of people or objects, include physical descriptions that could be used to identify someone or something in an image.
2. Do not infer or generate new descriptions; use only what is explicitly stated in the text, include the full phrase or sentence that describes the visual element.
3. Do not modify or paraphrase the extracted descriptions.
4. If multiple relevant descriptions are present, extract each one separately.
5. Remove question words (what, which, how, etc.) and their associated terms (color, size, etc.) from the extracted descriptions, only keep the concrete visual elements.
6. Do not include any explanations or commentary in your output, only the list of extracted descriptions wrapped in <|Answer|> and <|\Answer|> tags.

#####

-Examples-

#####

Example 1:

Input: What is the main architectural style of the Eiffel Tower?

#####

Output: <|Answer|>[<|\Answer|>

#####

Example 2:

Input: Which film did Emma Stone appear in first: the one where she's a woman wearing a red cape and flying over a city skyline, or the one where she has long wavy hair standing next to a man in a tuxedo? Also, is she the one with short blonde hair holding a bouquet of flowers in another movie?

#####

Output: <|Answer|>["a woman wearing a red cape and flying over a city skyline",
"long wavy hair standing next to a man in a tuxedo",
"short blonde hair holding a bouquet of flowers"]<|\Answer|>

#####

-Real Data-

#####

Input: {question}

#####

Output: {Your final output}

Visual-Semantic Matching Prompt Template

-Goal-

You are given three pictures (labeled 1, 2 and 3) and a description in English. Analyze the content of all three pictures carefully. It is possible that multiple pictures fit the description well. You will need to look closely at the given description in English and determine which one fits the given description the best.

Steps:

1. Look closely at the details of all three pictures.
2. Compare the content of each picture with the given description.

3. Choose the picture that best matches the description among all three options.
4. After "Answer:", write only "1", "2" or "3" without any additional text.
5. Explain your choice succinctly in the "Reason:" section.
6. Do not include any comments or information other than these two required lines.
7. Remember to keep your answers concise and strictly adhere to the required two-line format.
8. Your answer should be consistent with your reasoning.

Output Format:

Reason: [explain your choice]

Answer: [1, 2 or 3]

Description: {description}

Output:{Returns in the format}

Information Sufficiency Evaluation Prompt Template

-Goal-

Given a question and a knowledge graph, determine if the question can be answered based on the current knowledge graph. Pay close attention to the specific details requested in the question.

Steps

1. Analyze the Question: Thoroughly examine the provided question to understand the information needs.
2. Carefully analyze the given knowledge map and determine if you can answer the question based on the information in the map.
3. If the question can be answered based on the knowledge graph and give the correct answer in an explanation that is as concise as possible.
4. If you can't answer, or can't answer very accurately, please return to No with an explanation, by think step by step.

Important:

You should carefully read the constraints in the question and each node in the given knowledge graph. You should list the constraints and find the corresponding information step by step before answering.

Output Format

Reason: {Reason for inability to answer Or final answer, think step by step}

Answer: [Yes/No]

#####

-Real Data-

#####

Input:

Knowledge Graph:

{GraphML}

#####

Question: {question}

Output:{Strictly formatted returns}

Image Question Generation Prompt Template

-Goal-

Image Query Assistant: Your task is to determine whether image information is needed to answer a given query. If it is needed, the relevant image modal entity must be identified and a specific question about the image must be asked to that image entity. Results are returned in a format that does not contain any additional information. Only return the list, don't give any other explanation.

Steps:

1. Read the Query: Carefully analyze the question to identify any implicit or explicit need for image information.
2. Analyze the Knowledge Graph: Review entities and relationships in the knowledge graph to determine their relevance to the query.
3. Determine Image Necessity: Decide if image information would help answer the question.
4. Formulate Image Queries: If image information is required, list the node names(d0) for this image modality and specific questions related to those images.

Output Format:

<|Answer|>[{{entity_name:question}}]<|\Answer|>

If there are no questions to ask, return an empty list, for example:

<|Answer|>[]<|\Answer|>

Important:

1. The entity_name MUST be the name of the entity in the image modal and match the actual name of the entity shown in the d0 field, and MUST NOT contain any identifiers or additional type information. The question shall be specific and directly related to the image associated with the entity.
2. Multiple queries should be included in the list if necessary.
3. If no image information is required to answer the query, return an empty list.
4. Return the results in a format that does not contain any extra information, entity_name that return nodes of other types are considered inexcusable.

#####

-Examples-

#####

Example 1:

Input:

Question: What is the main architectural style of the Eiffel Tower?

Knowledge Graph:

===== BEGIN: TEXT NODES BLOCK =====

Name: Eiffel Tower

Type: Structure

Description: Iconic wrought-iron lattice tower on the Champ de Mars in Paris, France. Completed in 1889, it stands 324 meters (1,063 ft) tall and is the most-visited paid monument in the world.

Name: Paris

Type: City

Description: Capital city of France, known for its art, culture, and historical landmarks including the Eiffel Tower.

===== END: TEXT NODES BLOCK =====

===== BEGIN: IMAGE NODES BLOCK =====

Name: Eiffel Tower

Type: image

Description: Photograph showing the full view of the Eiffel Tower, clearly displaying its intricate ironwork and overall structure.

===== END: IMAGE NODES BLOCK =====

===== BEGIN: TABLE NODES BLOCK =====

===== END: TABLE NODES BLOCK =====

===== BEGIN: RELATIONSHIPS BLOCK =====

Node 1 Name: Eiffel Tower

Node 1 Type: Structure

Node 2 Name: Paris

Node 2 Type: City

Relationship between Node 1 and Node 2: Located in

===== END: RELATIONSHIPS BLOCK =====

#####

Output:

<|Answer|>[{"Eiffel Tower": "What architectural style or design features are prominently visible in this image of the Eiffel Tower?"}]<|\Answer|>

#####

-Real Data-

#####

Input:

Question: {question}

Knowledge Graph:

{KG}

#####

Output:{Your final output}

Visual Information Extraction Prompt Template

-Goal-

Provide extremely concise and factual answers about the content of the image and the image captions

-Steps

1. First describe the image in the context of the image name and image content.

2. If the answer to the question cannot be found in the picture, briefly state that the picture does not contain the content of the question.
3. Use declarative sentences, do not offer additional information or invite further questions.

Important:

1. Use both the image content and the provided image Image_title as factual information.
2. Image_title is the name of that image and is perfectly correct

#####

-Real Data-

#####

Input:

Question: {question}

Image_title: {title}

#####

Output:

Table Entity Extraction Prompt Template

-Goal-

Given a question and a table in markdown format (some questions may require multiple entity jumps to get the answer), identify candidate entities from the description field of the knowledge graph table node that may help answer the question. And return the answer in the specified format.

Step:

1. Analyze the question and infer the information needed to answer the question.
2. Analyze the given Markdown-formatted table.
3. Extract entities from the table that are relevant to the question, even though they do not answer the question.

Important:

1. The names of the extracted entities must appear in the content of the table.
2. Don't ignore any entities that are relevant to the question, even if they don't answer it.
3. Returns the name of the entity associated with the question, or the name of the entity associated with some information in the question
4. If you believe that there are no task entities in the table that are relevant to the question, return an empty list, but think about it carefully.
5. Return to list only
6. Wrap the final output in <|Answer|> and <|\Answer|> tags.
7. The list conforms to Python syntax, using double quotation marks to wrap each item.

Output Format:

<|Answer|>[List of entity names extracted from descriptions, or [] if none]<|\Answer|>

#####

-Examples-

#####

Example 1

Input:

Question: Where has Liu Xiang run in the 13.50s in the Games?

Table name: Liu Xiang International competition record

Table content(markdown):

| Year | Competition | Position | Event | Notes |
|------|----------------------------|-------------|---------------|------------------------|
| 2000 | World Junior Championships | 4th | 110 m hurdles | 13.87 (wind: -0.1 m/s) |
| 2001 | World University Games | 1st | 110 m hurdles | 13.33 seconds |
| 2001 | World Championships | 4th (semis) | 110 m hurdles | 13.51 |
| 2001 | Chinese National Games | 1st | 110 m hurdles | 13.36 |
| 2001 | East Asian Games | 1st | 110 m hurdles | 13.42 seconds |

#####

Output:

<|Answer|>["World University Games","Chinese National Games","East Asian Games"]<|\Answer|>

#####

-Real Data-

#####

Question: {Question}

Table name: {Table name}

Table content(markdown):

{Table content}

#####

Output:{Your final output}