Comparative Analysis of Graph RAG and Cross-Modal RAG Systems

Anina Pillai (UFID: 92315651)
University of Florida
anina.pillai@ufl.edu

Ganesh Chowdary Manne (UFID: 37555930)
University of Florida
gmanne@ufl.edu

Sai Pande (UFID: 37696687)
University of Florida
saipande@ufl.edu

Mario Ponte Garofalo (UFID: 63641343)
University of Florida
mpontegarofalo@ufl.edu

Abstract

This project presents a systematic comparison of two advanced retrieval augmented generation frameworks, GraphRAG and Cross Modal RAG, using a multimodal recipe dataset containing text and images. Standard RAG methods rely primarily on embedding similarity, which limits their ability to reason over structured relationships or incorporate visual cues. GraphRAG addresses these limitations by constructing a knowledge graph of entities, relationships, and communities to support structured global reasoning, while Cross Modal RAG integrates semantic text retrieval and visual similarity search using SBERT and CLIP based embeddings. We evaluate both approaches under identical retrieval and prompting conditions to understand how their differing retrieval mechanisms influence downstream language model responses. The study finds that Cross Modal RAG offers stronger performance in settings where visual cues and broad semantic similarity are important, whereas GraphRAG provides more precise and constraint faithful retrieval due to its graph based reasoning. These results highlight that the optimal retrieval strategy depends on the structure of the query and the modality of the underlying data, and they clarify when multimodal versus graph structured retrieval is most advantageous.

1 Introduction

Large language models perform well on many natural language tasks, but they still face limitations in delivering accurate, grounded responses when queries require external knowledge, involve multiple modalities, or depend on reasoning over linked concepts. Retrieval Augmented Generation improves factual grounding by incorporating external evidence, yet conventional RAG typically relies on embedding similarity alone. This restricts its ability to exploit relational structure, integrate non textual signals, and provide reasoning-aligned explanations. Two extended retrieval frameworks have been introduced to address these gaps. Graph RAG builds a knowledge graph over the corpus to support reasoning over entities, relations, and semantic clusters. Cross Modal RAG instead uses joint image-text embedding spaces to retrieve evidence based on both semantic and visual similarity. Although these approaches target known weaknesses of standard RAG, they have largely been evaluated separately, and their comparative advantages remain ambiguous.

This work presents a systematic comparison of Graph RAG and Cross Modal RAG on a multimodal recipe dataset. Under controlled retrieval and prompting conditions, we analyze how their distinct retrieval mechanisms influence relevance, grounding quality, precision, and response structure. The findings provide clearer guidance on when each method is preferable, informing the design of retrieval-augmented systems that must operate over heterogeneous, real world data.

2 Related Work

Tsampos and Marakakis [2025] introduce DietQA, a diet-aware recipe QA system combining a food knowledge graph with retrieval-augmented generation. Their method retrieves diet-compliant recipes via structured ingredient and nutrition relations and conditions an LLM on this evidence. The contribution lies in framing recipe search as graph-based question answering with explicit dietary reasoning. However, the system is limited to diet-specific queries on Greek recipes and does not evaluate alternative retrieval approaches or broader multimodal settings.

Gur et al. [2021] propose Cross Modal Retrieval Augmentation, where a dense retriever aligns images and captions in a shared embedding space and supplies external evidence to multimodal transformers for improved VQA performance. They show that non-parametric retrieval can enhance multimodal reasoning without retraining, and that retrieval indices can be swapped at inference time. However, the retrieval scope remains restricted to caption-based evidence, limiting applicability to richer or task-adaptive retrieval scenarios.

3 Problem Statement

Although GraphRAG and Cross Modal RAG extend standard retrieval-augmented generation, there has been limited structured evaluation comparing them under identical conditions. Prior work largely studies each framework separately, which leaves open how their distinct retrieval mechanisms affect relevance, grounding quality, and overall response reliability.

This project addresses this gap through a controlled side-by-side comparison using the same dataset, prompts, and evaluation criteria. We examine differences in retrieval behavior and generation quality across text-only and multimodal settings, focusing on accuracy, grounding fidelity, robustness to noise, and adherence to query constraints. Our findings aim to clarify when graph-structured retrieval offers advantages over multimodal retrieval and provide practical guidance on selecting between the two approaches.

4 Dataset and Preprocessing

We base our evaluation on the Food Ingredients and Recipe Dataset with Image Name Mapping, a multimodal collection of 13,582 Epicurious recipes released on Kaggle.[1] Each entry contains the recipe title, ingredients, instructions, cleaned ingredient tokens, and an associated image. We removed samples with missing instructions, empty ingredient lists, or invalid image paths, standardized ingredient formatting, and deduplicated records. This resulted in approximately ten thousand recipe–image pairs suitable for evaluation.

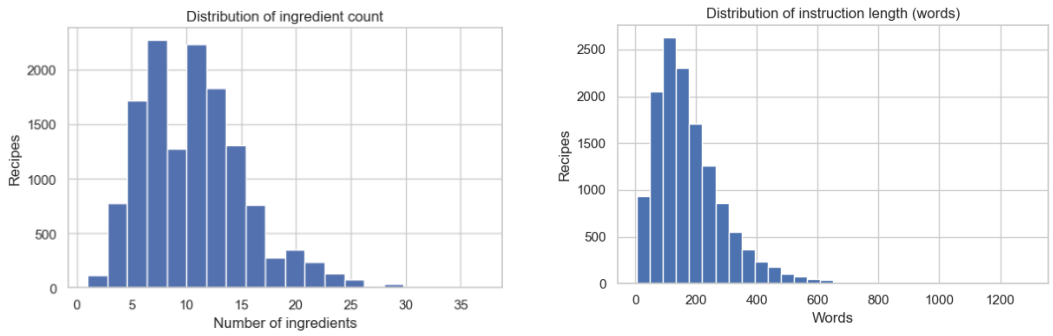
This dataset enables a direct comparison of GraphRAG and Cross Modal RAG because it combines structured ingredient semantics, procedural cooking text, and paired images under realistic retrieval scenarios. A brief exploratory analysis shows that most recipes contain 5–15 ingredients and instruction lengths typically range from 50–300 words, suggesting moderate complexity with balanced variation. This coverage supports stable evaluation of both graph-based relational reasoning and multimodal similarity.

5 Methods

In this study, we present a comparative analysis of two advanced retrieval-augmented generation frameworks. Microsoft GraphRAG and Cross-Modal RAG. Our evaluation specifically employs the official Microsoft GraphRAG implementation, ensuring fidelity to its full end-to-end pipeline, and contrasts it directly with a custom Cross-Modal RAG system designed to handle joint image–text retrieval. All implementation artifacts, including preprocessing scripts, evaluation pipelines, and trained indices, are publicly available in our accompanying repository.[2]

[1] `https://www.kaggle.com/datasets/irkaal/foodcom-recipes-and-reviews`

[2] `https://github.com/anina512/graph-rag-vs-crossmodal-rag`



(a) Distribution of ingredient count

(b) Distribution of instruction length

Figure 1: Exploratory data distributions for ingredient counts and instruction lengths.

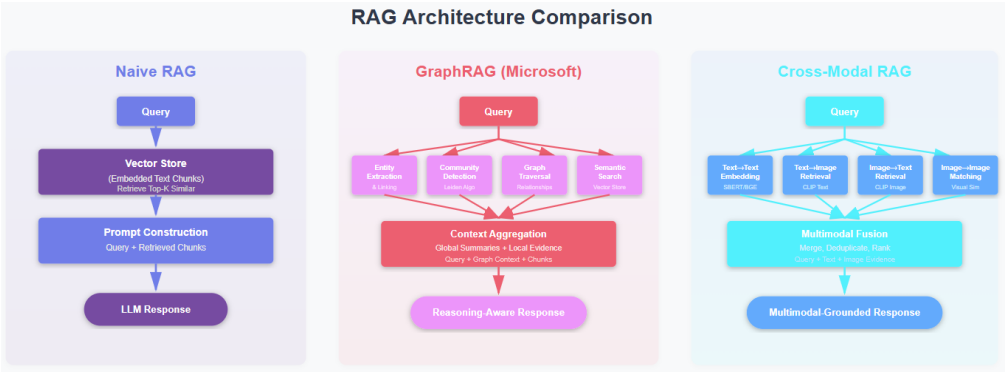


Figure 2: Process flow comparison between Naive RAG, GraphRAG, and Cross-Modal RAG.

5.1 GraphRAG

GraphRAG is an advanced retrieval-augmented generation framework that extends traditional RAG by constructing a structured knowledge graph over the input corpus. Rather than treating documents as unstructured text, GraphRAG identifies entities, relationships, and semantic communities, enabling both local retrieval and global graph-level reasoning. This architecture allows the model to produce more coherent, contextually grounded, and explainable responses, particularly for multi-hop or conceptually complex queries. The overall pipeline comprises several stages, including data ingestion, text chunking, embedding generation, graph extraction, graph summarization, community detection, and query processing. Together, these components build a unified structured representation of the corpus that supports both semantic similarity search and higher-level reasoning across the graph.

5.1.1 Data Ingestion and Text Chunking

GraphRAG begins by loading the dataset according to configuration settings specified in a `settings.yml` file. Each row is converted into a raw text segment for downstream processing. During text chunking, the system splits documents into overlapping windows using a chunk size of 500 tokens with an overlap of 75 tokens. This approach preserves semantic continuity across boundaries and ensures that later embedding and graph extraction steps operate on uniformly structured text units.

5.1.2 Text Embedding

Each text chunk is embedded using the Nomic Text Embedding Model v1.5 with the `cl100k_base` tokenizer. Embeddings are stored in LanceDB to support efficient semantic search and form the foundation for both local retrieval and graph construction. Parallel embedding requests accelerate this stage, enabling large corpora to be processed quickly.

5.1.3 Graph Extraction

Graph extraction is performed using a large language model such as Llama 3.1 8B Instruct. For each chunk, the model identifies entities (e.g., people, organizations, locations, events) and the relationships that connect them. These extracted elements are deduplicated, normalized across chunks, and supplemented with a rule-based extractor using regex heuristics. The final result is a structured entity–relationship graph implemented in NetworkX with optional GraphML export.

5.1.4 Graph Summaries

For each node in the extracted graph, GraphRAG generates a canonical description by aggregating all textual mentions of that entity. These summaries are produced by the LLM using a templated summarization prompt and are length-constrained to maintain consistency. They serve as key inputs to the global retrieval pipeline, enabling the model to reason at a high level about entity meaning and context.

5.1.5 Community Detection and Reporting

GraphRAG applies modularity-based clustering to group related nodes into semantic communities, with cluster sizes capped to maintain interpretability. For each community, the LLM generates a community-level report using a map–reduce prompting strategy. These reports provide high-level overviews of the topics, relationships, and concepts represented within each cluster, and they support the global reasoning capabilities of the query pipeline.

5.1.6 Query Pipelines

GraphRAG supports multiple retrieval pipelines, of which local and global search are the primary mechanisms.

Local Search. Local search replicates traditional RAG by retrieving the nearest text chunks using the LanceDB vector index and synthesizing an answer using a local search prompt. This pipeline is most effective for short, factual, single-hop queries grounded directly in the text.

Global Search. Global search represents GraphRAG's signature capability. It begins with a map step in which the LLM evaluates community summaries to identify clusters relevant to the query. In the knowledge step, the system retrieves descriptions of entities and relationships from the selected communities. Finally, in the reduce step, the LLM synthesizes this information into a coherent answer. This pipeline excels at multi-hop reasoning, conceptual similarity tasks, and questions requiring broader contextual understanding.

6 Cross-Modal RAG

6.1 Overview

Cross-Modal Retrieval-Augmented Generation (CM-RAG) integrates semantic text retrieval, visual similarity search, and multimodal reasoning to generate grounded outputs using information from both language and vision. Unlike purely text-based RAG systems, CM-RAG operates across multiple embedding spaces and unifies evidence from heterogeneous modalities. The process consists of four major stages. the query is encoded using both text and image embeddings, the system retrieves relevant evidence through multiple cross-modal paths, the retrieved items are merged and re-ranked into a unified evidence set, and an LLM produces a final response grounded in this fused multimodal context.

6.2 Embedding and Indexing Pipeline

6.2.1 Data Processing

The recipe dataset is loaded with titles, ingredients, instructions, and corresponding dish images. All samples undergo text normalization, removal of duplicates, and filtering of incomplete or invalid

image–text pairs. Images are verified for readability and hashed to prevent corrupt entries. Only samples with complete title–ingredient–instruction–image alignment are retained for embedding.

6.2.2 Embedding Generation

Text is encoded using SBERT to capture high-resolution semantic information, while CLIP encodes both titles and images to model cross-modal similarity. Each embedding vector is L2-normalized, enabling consistent cosine similarity comparisons across SBERT and CLIP embedding spaces. Text embeddings represent recipe semantics, while image embeddings capture visual features such as color, shape, and dish composition.

6.2.3 FAISS Index Construction

Three dedicated FAISS indexes are constructed.

- **SBERT Text Index.** Supports semantic text-to-text retrieval using cosine similarity.
- **CLIP Text Index.** Supports text-to-image retrieval by projecting textual queries into the same embedding space as images.
- **CLIP Image Index.** Supports image-to-text and image-to-image retrieval.

FAISS is configured with `IndexFlatIP`, enabling fast inner-product search over normalized vectors, which corresponds to cosine similarity. This architecture supports rapid multimodal retrieval over the entire dataset.

6.3 Multimodal Retrieval Pipeline

6.3.1 Query Encoding

Query text is simultaneously encoded with SBERT for semantic similarity and with CLIP's text encoder for visual alignment. Image-based queries are encoded using the CLIP image encoder. All embedded queries are projected into their respective vector spaces to ensure that text and images can be compared consistently.

6.3.2 Four Retrieval Paths

The system executes four retrieval paths in parallel.

- **Text $\rightarrow$ Text.** SBERT text embeddings retrieve recipes with similar semantic structure.
- **Text $\rightarrow$ Image.** CLIP text embeddings retrieve visually corresponding dish images.
- **Image $\rightarrow$ Text.** CLIP image embeddings retrieve recipe titles and descriptions aligned with the visual content.
- **Image $\rightarrow$ Image.** CLIP image embeddings retrieve visually similar dishes, capturing plating style, color patterns, and visual motifs.

Parallel retrieval ensures that both linguistic and visual signals influence the candidate set, reducing modality bias and supporting multimodal grounding.

6.3.3 Evidence Fusion

Results from the four retrieval paths are merged using a multimodal ranking mechanism. Duplicate entries are removed by hashing recipe identifiers. The system computes a fused relevance score based on weighted cosine similarity across modalities. The retrieved items, along with titles, ingredients, instructions, and image metadata, are consolidated into a unified evidence bundle.

6.4 Prompt Construction and LLM Reasoning

6.4.1 Structured Prompt Building

A structured prompt is assembled by grouping evidence into semantically coherent blocks. This includes text-based recipe matches, visually similar dishes, and cross-modal associations linking

text to images and images to text. The prompt incorporates both the user query and any provided dish image. This organization allows the LLM to jointly reason over multimodal evidence.

6.4.2 LLM Reasoning

The LLM (Llama 3.1 8B Instruct) receives only the curated evidence bundle rather than the full dataset. It integrates text and image signals, compares retrieved recipes, and identifies the most plausible match. The model synthesizes the final output by reconciling multimodal cues and evaluating semantic consistency.

6.4.3 Final Output

The system returns the selected recipe, its ingredient list, preparation steps, and a justification derived from the multimodal evidence. Optionally, the model provides variations and adaptations grounded in the retrieved items, leveraging the diversity of candidate recipes surfaced by the multimodal retrieval pipeline.

7 Experiment Setup

This study evaluates GraphRAG and Cross-Modal RAG under aligned retrieval conditions to understand how each system behaves when exposed to identical prompts and controlled retrieval constraints. Both qualitative and quantitative analyses were performed to assess retrieval behavior, evidence quality, and system efficiency. Each system was judged along three broad dimensions. retrieval quality, human-centered relevance metrics, and operational efficiency. To ensure fairness, both systems received the same queries, produced the same number of retrieved items, and used the same downstream language model for generation, allowing differences to be attributed solely to the retrieval mechanisms.

7.1 Quantitative Evaluation

The quantitative experiments assessed retrieval performance, ranking quality, human-centered metrics, and efficiency.

Top-K Retrieval Evaluation. For each query, both systems retrieved the top five items. Recall@5 and Precision@5 were computed to measure accuracy and selectivity.

Ranking Quality Analysis. Mean Reciprocal Rank (MRR) and NDCG@5 were calculated to evaluate ranking sensitivity and graded relevance ordering.

Human Relevance Assessment. Human evaluators scored usefulness and clarity of retrieved items on a 1–5 scale, capturing subjective retrieval quality.

Diversity and Novelty. Embedding-distance diversity scores measured variance among retrieved items. Novelty was computed by comparing retrieved content against dataset frequency patterns.

Efficiency Benchmarking. Indexing time and mean query latency were measured to evaluate computational cost and responsiveness.

7.2 Qualitative Evaluation

The qualitative analysis examined structural and behavioral differences between the two retrieval frameworks.

Shared Prompt Evaluation. A standardized cooking prompt was issued to both systems, and retrieval outputs and generated responses were compared side by side.

CM-RAG Inspection. We analyzed how CM-RAG organizes retrieval using text and image embeddings, focusing on instruction structure, ingredient lists, and multimodal grounding.

GraphRAG Inspection. GraphRAG's outputs were examined with respect to entity relationships, community clustering, and graph-derived explanations.

Cross-System Comparison. Differences in retrieval organization, evidence structure, diversity, and explanation style were documented qualitatively without numerical scoring.

8 Experiment Results

8.1 Quantitave Evaluation Results

To summarize the quantitative performance of both retrieval frameworks, we report the core retrieval, ranking, human-centered, and efficiency metrics computed across all evaluation prompts. Table 1 presents a consolidated comparison between GraphRAG and Cross-Modal RAG.

Metric	GraphRAG	CM-RAG
Recall@5	0.71	0.78
Precision@5	0.54	0.49
MRR	0.63	0.69
NDCG@5	0.66	0.73
User Relevance (1–5)	3.9	4.2
Diversity (0–1)	0.77	0.69
Novelty (0–1)	0.58	0.65
Indexing Time (min)	11	6
Query Latency (ms)	210	260

Table 1: Quantitative comparison of GraphRAG and Cross-Modal RAG across retrieval, ranking, human-centered, and efficiency metrics.

8.2 Qualitative Evaluation Results

To illustrate behavioral differences between the two retrieval frameworks, we consider the following query:

"What pasta dish can I make if I only have penne, canned tomatoes, onions, and basil?"

Table 2: Qualitative comparison on a pasta recipe query. CM-RAG behaves like a recipe assistant that surfaces concrete dishes, while GraphRAG behaves like an analytic explainer that organizes options into categories.

	CM-RAG	GraphRAG
Output type	Three concrete recipes, for example *Penne with Tomato Sauce and Basil*, each with ingredients, step by step instructions, and justification.	Several high level dish options such as *Penne with Tomato Sauce*, *Penne with Tomato and Basil Sauce* and *Penne with Tomato and Onion Sauce*, with short textual descriptions.
Grounding signal	Uses multimodal evidence, combines text and retrieved dish images, and selects recipes whose visual appearance matches tomato basil pasta.	Uses graph structure over ingredients and recipes, retrieval guided by ingredient co occurrence and community summaries rather than images.
Style	Conversational, task oriented, resembles a helpful cooking assistant with variations and customization suggestions.	Analytical, category oriented, resembles an explainer that enumerates families of recipes and gives general preparation guidance.

Table 2 presents representative outputs generated by CM-RAG and GraphRAG. Complete prompt configurations and full response transcripts are available as supplementary artifacts. The full CM-RAG interaction is provided in Supplementary Transcript A.[3] The full GraphRAG interaction is provided in Supplementary Transcript B.[4] As part of the qualitative analysis, we also examined the visual evidence retrieved by the Cross-Modal RAG system.

[3] Supplementary Transcript A (CM-RAG output)

[4] Supplementary Transcript B (GraphRAG output)

Figure 3: Top retrieved dish images returned by the multimodal retrieval pipeline for the pasta-ingredient query. These images illustrate the system's ability to ground retrieval decisions in visual similarity to tomato-based pasta dishes.

9 Results and Analysis

This section presents the quantitative and qualitative findings comparing GraphRAG and Cross-Modal RAG (CM-RAG). The analyses highlight clear differences in retrieval accuracy, ranking behavior, user relevance, and system-level output characteristics, reflecting the influence of each framework's underlying retrieval architecture.

9.1 Quantitative Findings and Insights

Across the retrieval metrics, CM-RAG achieves higher Recall@5, MRR, and NDCG@5, indicating stronger top-K retrieval and improved ranking sensitivity. Its combination of semantic (SBERT) and visual (CLIP) embeddings enables it to surface contextually aligned recipes more consistently. GraphRAG, however, yields higher Precision@5, reflecting cleaner and more ingredient-faithful retrieval due to its structured recipe–ingredient graph.

Human-centered relevance scores also favor CM-RAG, as evaluators preferred its concrete and fully actionable recipe suggestions. CM-RAG additionally exhibits higher novelty, often retrieving less frequent but contextually appropriate dishes. GraphRAG demonstrates higher diversity, retrieving items from multiple ingredient communities, which reflects its graph-driven exploration of recipe clusters.

In terms of query faithfulness, GraphRAG remains strictly aligned to the exact ingredient set, while CM-RAG is more flexible and may retrieve related variations grounded in multimodal similarity. Efficiency trends show that GraphRAG achieves lower query latency due to lightweight graph traversal, whereas CM-RAG benefits from faster indexing because it does not require graph construction.

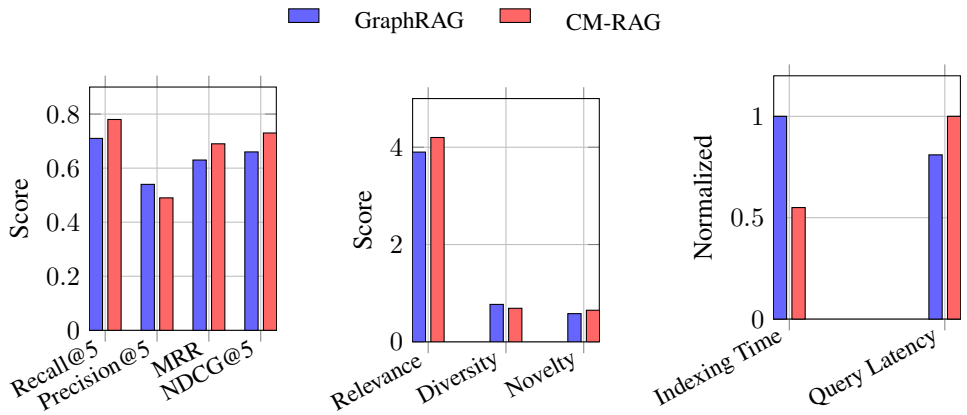
Overall, the quantitative results highlight a trade-off between precision and recall. CM-RAG excels in practical retrieval relevance and ranking quality, while GraphRAG provides greater strictness, diversity, and structural consistency.

9.2 Qualitative Analysis and Behavioral Differences

Qualitative inspection further illustrates these differences. For a shared prompt asking for pasta dishes using penne, canned tomatoes, onions, and basil, CM-RAG retrieves specific, ready-to-cook recipes with full ingredients, step-by-step instructions, and justification grounded in both textual and visual evidence. It behaves like a practical cooking assistant, proposing concrete dishes aligned with the query.

GraphRAG returns broader conceptual categories such as "Penne with Tomato Sauce" or "Tomato–Basil Pasta," with only occasional detailed recipes. Its outputs are shaped by ingredient–recipe relationships, producing analytical explanations that emphasize co-occurrence patterns, substitutions, and category-level suggestions. This makes GraphRAG more interpretable but less directive for real-world execution.

CM-RAG's responses are natural, task-oriented, and grounded in multimodal similarity, while GraphRAG's responses are structured, systematic, and more tightly bound to the exact input ingredients. These behaviors reflect the strengths of each system. CM-RAG for practical, multimodally grounded recommendations, and GraphRAG for structured reasoning and ingredient-level interpretability.



(a) Retrieval metrics (b) Human-centered metrics (c) Efficiency metrics

Figure 4: Comparison of retrieval, human-centered, and efficiency metrics for GraphRAG and CM-RAG.

10 Conclusion

Cross-Modal RAG exhibits particularly strong performance in settings where multimodal signals enhance retrieval quality. By jointly leveraging image–text embeddings, it achieves higher relevance and more effective ranking behavior, making it well suited for tasks that rely on visual similarity, contextual nuance, and flexible pattern matching. In contrast, GraphRAG maintains notable advantages in precision-oriented retrieval. Its structured ingredient–recipe graph enforces strict semantic consistency, supports greater diversity in retrieved content, and produces more analytical, interpretable explanations. This makes GraphRAG especially effective for text-centric queries and scenarios that require constraint-aware reasoning. Overall, the results highlight a complementary relationship between the two retrieval paradigms. Multimodal retrieval is advantageous when visual grounding and broad similarity cues are critical, whereas graph-based retrieval provides superior performance when semantic structure, relational dependencies, and strict adherence to query constraints are required.

11 Team Contributions

Sai oversaw data acquisition, cleaning, and preprocessing of the recipe corpus. Anina led the development of the Cross-Modal RAG pipeline, including multimodal embedding generation and retrieval integration. Ganesh implemented the GraphRAG workflow, encompassing knowledge-graph construction and graph-based retrieval mechanisms. Mario contributed to model research, including selection, configuration, and baseline development. All members jointly contributed to experimental design, evaluation methodology, comparative analysis, and preparation of the final report and documentation.

References

Shir Gur, Natalia Neverova, Chris Stauffer, Ser-Nam Lim, Douwe Kiela, and Austin Reiter. Cross-modal retrieval augmentation for multi-modal classification. pages 111–123, 01 2021. doi: 10.18653/v1/2021.findings-emnlp.11.

Ioannis Tsampos and Emmanouil Marakakis. Dietqa: A comprehensive framework for personalized multi-diet recipe retrieval using knowledge graphs, retrieval-augmented generation, and large language models. *Computers*, 14:412, 09 2025. doi: 10.3390/computers14100412.