ADAMM-RAG: ADAPTIVE MULTI-SOURCE MULTI-MODAL RAG VIA PLANNER-GUIDED COORDINATION

Anonymous authors
Paper under double-blind review

ABSTRACT

RAG grounds LLMs in external knowledge, but existing multimodal multi-agent systems usually use a fixed topology that calls all sources in parallel and aligns only vision-language evidence. This causes two issues: irrelevant sources add noise and cost, and cross-source evidence is never unified pre-fusion. Adaptive agent methods remain text-only, and do not route across sources or modalities. We propose AdaMM-RAG, which enables source-level adaptive routing via a Planner Agent that selects per-query which sources to consult. A Cross Alignment Agent unifies selected heterogeneous evidence; a Decision Agent refines and verifies answers with CoT; a quality gate triggers fallback only when evidence is insufficient. On ScienceQA, AdaMM-RAG achieves 95.51% accuracy, surpassing HM-RAG (SOTA multi-agent RAG) by +1.78% and MAO-ARAG by +12.02%; on Crisis-MMD, it reaches 80.20% accuracy, outperforming HM-RAG by +21.65% and MAO-ARAG by +5.77%, while reducing retrieval cost by 55.5%.

1 INTRODUCTION

Retrieval-Augmented Generation (RAG) (Lewis et al., 2020) augments a large language model (LLM) with retrieved external evidence to mitigate outdated parametric knowledge and hallucinations, and has become the dominant paradigm for knowledge-intensive tasks (Gao et al., 2023). In practice, however, real queries often span heterogeneous data ecosystems (text, knowledge graphs, web) and multiple modalities (images, tables), and a single-agent retrieve-then-generate pass struggles to coordinate such complex reasoning. Research therefore has moved toward multi-agent collaboration for multimodal retrieval.

However, current multi-agent multimodal RAG systems suffer from two disjoint limitations: a fixed source-invocation topology and incomplete cross-modal alignment. Multimodal multi-agent RAG (Chang et al., 2025; Liu et al., 2025b) extends standard RAG (Lewis et al., 2020) by coordinating specialized agents across text, knowledge graphs, and web sources, but every query triggers the same fixed parallel retrieval regardless of its actual needs; irrelevant sources inject noise and waste computation. Meanwhile, adaptive routing (Jeong et al., 2024; Chen & Zhang, 2025) handles text-only queries without addressing multi-source or multimodal coordination: its adaptivity chooses *whether/how to retrieve* or *which text executors to compose*, but never *which data sources to invoke*. The two research lines are disjoint; one adapts the retrieval procedure inside a fixed corpus, the other fixes the invocation topology; leaving the coupled problem of multimodal, multi-source, adaptive coordination, in which the *source set itself* is the adaptive decision variable, unresolved.

Unresolved, this coupling creates two concrete limitations. First, fixed-topology systems such as HM-RAG (Liu et al., 2025b) invoke vector, graph, and web retrievers unconditionally; even a query answerable from a single source pays the full computational cost and risks contamination from irrelevant evidence. Echoing MAIN-RAG's finding that irrelevant retrieved documents degrade RAG quality (Chang et al., 2025), the problem here operates at the data-source level rather than the document level. Second, cross-source alignment in these systems is fragmentary. HM-RAG bridges only vision and language, while its graph abstraction discards fine-grained textual detail from vector and web retrievers (Liu et al., 2025b). Text-only adaptive routers (Jeong et al., 2024; Chen & Zhang, 2025) avoid these issues by design but cannot operate on multimodal inputs. The core difficulty is that coordination needs to be query-conditioned at both the source-selection and evidence-fusion levels, yet no existing framework provides this.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

To address this problem, we propose AdaMM-RAG (Adaptive Multi-source Multi-modal RAG), which conditions both retrieval routing and evidence fusion on the input query. The core insight is that coordination should be query-conditioned: retrieving only what a question needs and aligning evidence before fusion removes the noise and cost of indiscriminate full calls while improving fusion quality. AdaMM-RAG retains HM-RAG's hierarchical skeleton but replaces its fixed call graph with planner-guided on-demand coordination. A training-free Planner routing agent decides which data sources to query (vector, entity, relation, graph, web) per question instead of invoking all of them. A Cross Alignment Agent then projects heterogeneous evidence from every selected source into a unified semantic space before fusion, preserving fine-grained facts that graph-only abstraction would discard. A Decision Agent refines the aligned evidence into a verified answer, resolving conflicts via structured chain-of-thought reasoning, and a quality gate triggers fallback retrieval only when single-source evidence is insufficient, bounding overhead. We evaluate AdaMM-RAG on ScienceQA and CrisisMMD, where it surpasses SOTA methods in accuracy while reducing retrieval cost on simple queries (Section 4.2). Our contributions are:

- A Planner-guided routing mechanism that replaces the fixed parallel topology of prior multimodal multi-agent RAG with on-demand source selection conditioned on the input query.
- A cross-source, cross-modal Alignment Agent that unifies heterogeneous evidence into a shared semantic space before decision fusion, generalizing prior vision-only framework.
- A Decision Agent that refines and verifies the aligned evidence through structured chain-of-thought reasoning without simple voting, unifying multi-source refinement and single-source fact-checking in one module.
- A bounded quality gate that couples with planner routing to provide a controllable accuracy-cost trade-off rather than unconditional full-call overhead.

2 Related Work

2.1 Adaptive RAG

A line of work adds adaptivity to RAG while staying within a single agent and a purely textual setting. Self-RAG (Asai et al., 2024) trains a model to self-reflect on whether to retrieve and critique its own output; Corrective RAG (Yan et al., 2024) uses a lightweight critic to detect low-quality retrieval and trigger self-correction or web supplementation; Adaptive-RAG (Jeong et al., 2024) routes a query to a retrieval strategy of appropriate complexity by predicted difficulty. RAPTOR (Sarthi et al., 2024) and GraphRAG (Edge et al., 2024) organize passages into hierarchical tree or graph structures for structured access, while Modular RAG (Gao et al., 2024) reframes RAG as recombinable functional modules. Most relevant to our work, MAO-ARAG (Chen & Zhang, 2025) trains a planner to dynamically compose query-rewrite, retrieval, document-selection, and generation executors per query, the first method to couple adaptive routing with a cost objective. Yet the shared limitation is that adaptivity operates within the textual modality: routing decides "whether/how to retrieve" from a text corpus, never which data sources (vector, graph, web) to invoke or how to fuse multimodal evidence (Singh et al., 2025). Multimodal queries that span images, tables, and structured knowledge therefore fall outside the scope of this family entirely.

2.2 Multimodal or Multi-Agent RAG

A parallel line turns to multi-agent collaboration to handle heterogeneous sources, but adopts a fixed topology that cannot adapt per query. MAIN-RAG (Chang et al., 2025) assigns three LLM agents (Predictor, Judge, Aggregator) to filter noisy documents by consensus, but operates on text only. HM-RAG (Liu et al., 2025b) introduces the first hierarchical multimodal multi-agent RAG, with a decompose agent, parallel vector/graph/web retrieval agents, and a decision agent that votes, yet the retrieval invocation is hardcoded: all three sources are queried on every query regardless of relevance, and its cross-modal alignment bridges only vision and language, leaving graph abstractions and web evidence unaligned. mRAG (CIIR@LiveRAG, 2025) similarly composes fixed planning, retrieval, and reasoning agents without per-query routing. These systems contrast sharply with the adaptive-textual camp: they handle multimodal, multi-source input, but they never route or prune sources conditionally.

Fixed-topology methods never adapt across sources per query; adaptive-textual methods never touch multimodal evidence. Existing adaptive systems therefore decide at the document/strategy level (Adaptive-RAG, Self-RAG, Corrective-RAG: whether and how to retrieve within one text corpus) or the agent/executor level (MAO-ARAG: how to compose text-only executors), always with the source set fixed a priori. AdaMM-RAG instead adapts at the source × modality level: per query it chooses among vector, entity, relation, graph, and web retrievers spanning text, knowledge-graph, and web evidence, never invoking unselected sources. This is an orthogonal dimension; source-level routing can coexist with document- or executor-level adaptation inside each selected source, and no prior framework in either camp selects among data sources, the precise gap AdaMM-RAG targets.

3 METHOD

We present AdaMM-RAG (Adaptive Multi-source Multi-modal Retrieval-Augmented Generation), a multi-agent framework that replaces the fixed topology of prior multimodal RAG systems with planner-guided, on-demand coordination (Figure 1). The central novelty is a shift in the granularity of coordination: instead of fixing which data sources to retrieve at design time, AdaMM-RAG treats the source set itself as a query-conditioned decision at the source × modality level (Section 2). Under this principle, three mechanisms cooperate: (i) source-level adaptivity, in which a Planner Agent (Section 3.2) selects the per-query subset $S \subseteq \{V, E, R, G, W\}$ (vector, entity, relation, graph, web) of data sources and only the selected sources are invoked (Section 3.3); (ii) cross-modal, cross-source alignment, in which a Cross Alignment Agent (Section 3.4) unifies heterogeneous evidence from the selected sources into one space before fusion; and (iii) bounded, decision-driven recovery, in which a Decision Agent (Section 3.5) refines the aligned evidence under a quality gate (Section 3.6) that triggers fallback only when single-source evidence is insufficient. These elements are not stacked patches: they are three facets of one coordination principle whose shared decision variable is the source set, and the ablation in Section 4.3 isolates their individual and coupled contributions.

3.1 OVERALL FRAMEWORK

Given a (possibly multi-modal) query q, AdaMM-RAG executes a five-stage pipeline, illustrated in Figure 1. First, the **Planner Agent** reads q and emits a structured plan $p = (S, d, v)$, where $S \subseteq \{V, E, R, G, W\}$ is the subset of sources to consult (vector, entity, relation, graph, web) and $d, v \in \{0, 1\}$ indicate query decomposition and verification. Second, the **on-demand retrieval pool** invokes only the retrievers in S, returning raw evidence E. Third, the **Cross Alignment Agent** maps every piece of E into a unified representation U; if all sources agree, it exits early with a consensus answer. Fourth, the **Decision Agent** refines U into a candidate answer A. Finally, the **quality gate** triggers fallback re-planning (adding skipped sources) only when single-source evidence is insufficient, re-running stages 2–4 up to a bounded number of attempts.

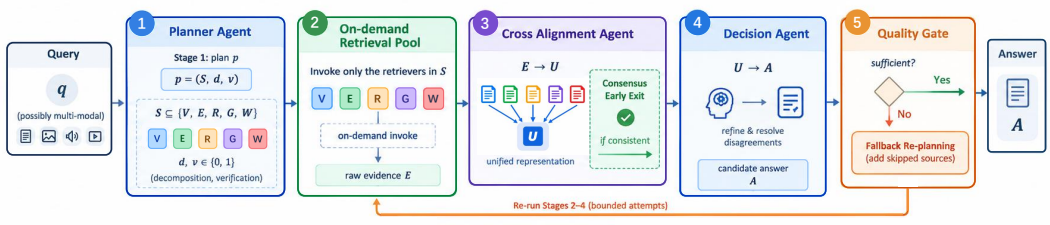


Figure 1: Overview of AdaMM-RAG. Stage 1: Planner selects source subset S. Stages 2–3: Retrieval and cross-alignment with early-exit consensus. Stage 4: Decision agent resolves disagreements into answer A. Stage 5: A quality gate ensures sufficiency or triggers fallback by adding skipped sources and re-running Stages 2–4.

3.2 PLANNER AGENT FOR ROUTING

Building on the on-demand retrieval pool, the Planner produces a structured plan for a query q

$$p = \text{Planner}(q) = (S, d, v), \qquad S \subseteq \{V, E, R, G, W\}, \tag{1}$$

where S is the chosen data-source subset, d indicates query decomposition, and v indicates need for multi-source verification. Concretely, we adopt a *training-free LLM router* that decodes a plan from a structured CoT:

$$\pi(p \mid q) = \text{LLM}_{\text{router}}\big(\text{CoT}(q)\big), \qquad \hat{p} = \text{Decode}(\pi(p \mid q)), \tag{2}$$

where $\text{Decode}(\cdot)$ applies the threshold-based decoding: it scores each source $s \in \{V, E, R, G, W\}$ from 0 to 3 by usefulness, keeps every source with score ≥ 1, enforces $|\hat{S}| \geq 2$ except for exact single static-fact lookups, and emits the decomposition and verification flags $(\hat{d}, \hat{v})$. Equation 2 thereby matches the structured JSON decoding in Appendix F.1.

Fixed-topology systems such as HM-RAG (Liu et al., 2025b) invoke all sources regardless of the question, and MAO-ARAG's planner (Chen & Zhang, 2025) composes only text executors. As MAIN-RAG (Chang et al., 2025) shows for documents, irrelevant evidence degrades RAG quality; at the data-source level, calling an irrelevant source injects noise, so a dedicated Planner must decide which sources and modalities a query needs.

By replacing a static call graph with a query-conditioned plan, the Planner localizes computation and noise, and decomposes the graph source into independently routable entity (E), relation (R), and graph (G) retrievers, extending routing from "text-only complexity" (Jeong et al., 2024) to the *source* $\times$ *modality* dimension prior multimodal RAG left fixed.

3.3 ON-DEMAND MULTI-SOURCE RETRIEVAL AGENT POOL

We instantiate five independent retrievers, denoted V (vector), E (entity), R (relation), G (graph), and W (web), each serving a distinct retrieval role (design details and cost model in Appendix C), and invoke *only* those selected in S. The vector retriever ranks candidates by dense cosine similarity:

$$h_q = E_{\text{text}}(q), \quad s_j = \frac{h_q^\top h_j}{\|h_q\| \, \|h_j\|}, \quad D_V = \text{topk}_j\{s_j\}, \tag{3}$$

where h_q is the query embedding, h_j the embedding of the j-th item, s_j their cosine similarity, and D_V the top-k evidence. The entity, relation, and graph retrievers operate at increasing knowledge scales (local entities $\rightarrow$ global relations $\rightarrow$ multi-hop graphs), and the web retriever queries a real-time search API. Sources absent from S are never called, so their cost and noise are avoided entirely.

Because the retrievers are *independently routable*, the Planner selects the granularity a query needs rather than a monolithic full-corpus index, realizing "retrieve what the question needs, not everything" at the source level, thereby reducing both retrieval cost and noise (Section 4.4).

3.4 CROSS ALIGNMENT AGENT

This agent serves as the cornerstone for unifying heterogeneous evidence. Its primary objective is to normalize heterogeneous evidence into a common text space so that a shared text encoder can compare across sources and modalities, removing representational mismatches that would otherwise inject noise into fusion.

Vision-Language Bridging. First, to incorporate visual evidence into the reasoning process, we bridge an image I_v to the textual space. Unlike simple retrieval methods, we employ a generative alignment strategy where the image is encoded and decoded into a descriptive caption T_v:

$$T_v = D_{\text{VLM}}\big(f_{\text{align}}(E_{\text{VLM}}(I_v))\big), \tag{4}$$

where E_{VLM} denotes the visual encoder extracting feature maps from I_v, f_{align} is a projection layer (e.g., a Q-Former or linear mapper) that maps visual features to the language embedding space, and D_{VLM} represents the frozen LLM decoder that generates the natural language description T_v.

Unified Evidence Projection. Subsequently, to ensure comparability across different knowledge sources, every evidence item e originating from a specific source m is projected into a unified text space:

$$u_e = E_{\text{text}}\big(\text{Project}_m(e)\big), \quad m \in \{V, I, G, W\}. \tag{5}$$

In this formulation, $\text{Project}_m(\cdot)$ acts as a modality-specific normalizer: for graph data, it serializes structural triples (h, r, t) into natural language sentences; for image data, it utilizes the generated caption T_v from Equation 4; and for web and text sources, it extracts and cleans the raw snippets. Following this normalization, the function E_{text} (a shared text encoder) maps these texts into dense vector representations u_e. This yields a unified evidence set $U = \{u_e \mid e \in E\}$, where all elements are embedded by the same encoder and are therefore directly comparable.

Consensus-Based Early Exit. Not every query requires full fusion for decision. When the aligned evidence from all selected sources points to the same conclusion with sufficient confidence, the downstream Decision Agent and quality gate introduce unnecessary latency and computation. We therefore add a consensus check immediately after alignment. Let $\mathcal{C}_m$ denote the answer candidate extracted from each source m based on its aligned evidence $U_m \subseteq U$. If the pairwise agreement $\text{Agree}(\mathcal{C}_m, \mathcal{C}_n)$ holds for every pair of sources (i.e., no contradiction exists and the conclusions are identical), the system skips the Decision Agent and directly returns the consensus answer. This bounded early exit reduces token cost and latency on straightforward queries while preserving accuracy, because the check itself is a lightweight string-matching or embedding-similarity operation over already-aligned representations.

Comparison with Multi-Modal RAG. Existing multi-modal methods (Liu et al., 2025b) typically limit f_{align} to vision-language bridging, leaving graph and web evidence in disjoint spaces. Furthermore, their graph abstraction often sacrifices fine-grained textual fidelity. In contrast, our approach ensures that the Decision Agent operates on a unified text representation of every selected source. Our alignment mechanism is *cross-source* $\times$ *cross-modal*: vector, graph, web, and image evidence are normalized into one text space while preserving the fine-grained details of the original content. The resulting aligned set U is fusion-ready, significantly enhancing the signal-to-noise ratio for the subsequent decision and verification stages.

3.5 Decision Agent: Evidence Fusion and Verification

Operating on the aligned set U, the Decision Agent fuses evidence from all selected sources into a single answer. Unlike prior work that relies on majority voting (Liu et al., 2025b), we adopt a refinement-only strategy: voting assumes the majority candidate is correct, but this can fail when the truthful answer lies with the minority or when sources agree on a label while disagreeing on the underlying evidence. We therefore perform refinement over the unified evidence rather than voting.

Multi-Source Refinement. Given the query q and the aligned evidence set U, a chain-of-thought expert re-derives the answer by reasoning over U directly:

$$A = \text{CoT-Refine-LM}(q, U), \tag{6}$$

where CoT-Refine-LM is prompted to inspect each source's aligned evidence, weigh their reliability, and produce a single answer. Because U is already cross-modally aligned (Section 3.4) and the consensus-based early exit (Section 3.4) has already filtered out straightforward unanimous cases, the refinement here handles the genuinely challenging scenarios: conflicting evidence, complementary information, or partial coverage from individual sources.

Single-Source Verification. For single-source queries ($|S| = 1$), the agent performs fact-checking verification:

$$(A, \text{fallback}) = \begin{cases} (A_s,\ \text{False}), & \neg\text{Contradict}(A_s, U) \wedge \neg\text{Insufficient}(U), \\ (\text{CoT-Verify-LM}(q, A_s, U),\ \text{False}), & \text{Contradict}(A_s, U) \wedge \neg\text{Insufficient}(U), \\ (\text{CoT-Verify-LM}(q, A_s, U),\ \text{True}), & \text{Insufficient}(U), \end{cases} \tag{7}$$

where $\mathrm{Contradict}(\cdot)$ and $\mathrm{Insufficient}(\cdot)$ are as defined above. Contradiction is resolved by CoT-Verify-LM within the agent and does not require escalation. Insufficiency, however, cannot be fixed by reasoning alone, so the agent sets a fallback signal for the quality gate.

3.6 Quality Gate and Fallback

The quality gate triggers a controlled recovery only when the Decision Agent signals evidence insufficiency in a single-source query (Section 3.5, Equation 7). For multi-source queries ($|S| > 1$), cross-source comparison within the aligned space already provides inherent validation, and no additional gate is needed.

Formally, the gate checks:

$$\text{Fallback} = \begin{cases} \text{True}, & \text{if } |S| = 1 \ \wedge \ \text{fallback} = \text{True} \ \wedge \ n_{\text{retry}} < \max_{\text{replan}}, \\ \text{False}, & \text{otherwise}, \end{cases} \tag{8}$$

where fallback is the flag emitted by Equation 7, and n_{retry} denotes the number of re-planning attempts already performed. Note that contradiction is *not* a trigger: CoT-Verify-LM resolves conflicts within the Decision Agent, and escalating to fallback would be unnecessary overhead. Insufficiency is fundamentally different from contradiction; without additional data, reasoning alone cannot produce a missing answer, making it the only case that warrants re-planning.

When fallback is triggered, the Planner re-plans by adding sources it previously skipped (e.g., the graph G or the web W) and re-executes up to $\max_{\text{replan}}$ times. The incremental cost of each fallback round is bounded by the added source set $\mathcal{S}^{(k)}_{\text{added}} \subseteq \mathcal{S}_{\text{skipped}}$, where $\mathcal{S}_{\text{skipped}}$ denotes the sources omitted in the initial plan. The cumulative overhead satisfies:

$$\text{Cost}_{\text{total}} = \text{Cost}_{\text{first}} + \sum_{k=1}^{n_{\text{retry}}} \text{Cost}(\mathcal{S}^{(k)}_{\text{added}}), \quad \text{with } n_{\text{retry}} \leq \max_{\text{replan}}, \tag{9}$$

so that extra cost is incurred only when the single-source evidence is insufficient and the Decision Agent cannot compensate through reasoning alone.

4 Experiment

4.1 Experimental Setup

Datasets. ScienceQA (Lu et al., 2022) contains multimodal multiple-choice questions across NAT, SOC, and LAN subjects, requiring joint vision-language reasoning. We evaluate on a fixed stratified subset of test questions (NAT 217, SOC 100, LAN 107; TXT 184, IMG 220, NO 240; G1-6 271, G7-12 153). CrisisMMD (Alam et al., 2018) comprises crisis-related tweets with images, providing a noisy, real-world multimodal testbed. We evaluate on 1534 tweets for Task 1 and 955 for Task 2, using the official `train`/`test` partitions of the original release. All runs used fixed random seeds.

Baselines. We compare against three families. **(i) Zero-shot LLMs/VLMs:** GPT-3 (Brown et al., 2020), ChatGPT (OpenAI, 2023), CoT GPT-3 + Doc (Hsieh et al., 2023), DDCoT (Zheng et al., 2023), LaVIN-13B (Yang et al., 2023), LLaMA-SciTune (Horawalavithana et al., 2023), and LG-VQA (Ghosal et al., 2023). **(ii) Fine-tuned VLMs:** KAM-CoT (Mondal et al., 2024), Honeybee (Cha et al., 2024), MC-CoT-F-Large (Tan et al., 2023), and Enigma-COT-Large (Wei et al., 2023). **(iii) Multi-agent RAG:** HM-RAG (Liu et al., 2025b) (multimodal, fixed-topology) and MAO-ARAG (Chen & Zhang, 2025) (adaptive, text-only). Because MAO-ARAG operates purely on text, we augment it with our vision-language bridging module (Section 3.4) at its input stage only, so the comparison isolates adaptive coordination independent of modality coverage while preserving its published planner behavior.

Implementation. Built on PyTorch with an LLM backend (OpenAI or local Ollama), our framework encodes vision inputs using Qwen3-VL-4B and text via Qwen3-embedding-0.6B. To ensure a fair comparison with HM-RAG (Liu et al., 2025b), RAG's retrieval leverages the Qwen2.5-7B parameter adaptation framework. For decision refinement, we utilize Qwen3-8B.

Table 1: ScienceQA Top-1 accuracy (%). #P = trainable parameters. Categories: NAT (Natural Science); SOC (Social Science); LAN (Language Science); TXT/IMG/NO (Text/Image/No Context); G1-6/G7-12 (Grades 1-6/7-12). Compares SOTA results from the leaderboard[1].

Learning Models	#P	NAT	SOC	LAN	TXT	IMG	NO	G1-6	G7-12	Average
Baseline										
Human	-	90.23	84.97	87.48	89.60	87.50	88.10	91.59	82.42	88.40
Zero-shot LLMs										
ChatGPT (OpenAI, 2023)	-	-	-	-	-	-	-	-	-	69.41
GPT-3 (0-shot) (Brown et al., 2020)	173B	75.04	66.59	78.00	74.24	65.74	79.58	76.36	69.87	74.04
DDCoT (GPT-3) (Zheng et al., 2023)	175B	78.60	73.90	80.45	77.27	69.96	82.93	80.65	73.50	78.09
CoT GPT-3 + Doc (Hsieh et al., 2023)	173B	-	-	-	-	-	-	-	-	79.91
DDCoT (ChatGPT) (Zheng et al., 2023)	175B	80.15	76.72	82.82	78.89	72.53	85.02	82.86	75.21	80.15
Zero-shot VLMs										
LaVIN-13B (Yang et al., 2023)	-	-	-	-	-	-	-	-	-	77.54
LLaMA-SciTune (Horawalavithana et al., 2023)	7B	84.50	94.15	82.91	88.35	83.64	88.74	85.05	85.60	86.11
LG-VQA (BLIP-2) (Ghosal et al., 2023)	-	-	-	-	-	-	-	-	-	86.32
LG-VQA (CLIP) (Ghosal et al., 2023)	-	-	-	-	-	-	-	-	-	87.22
LLaMA-SciTune (Horawalavithana et al., 2023)	13B	89.30	95.61	87.00	93.08	86.67	91.75	84.37	91.30	90.03
Fine-tuned VLMs										
KAM-CoT (Mondal et al., 2024)	280M	94.76	92.24	93.36	94.53	93.16	94.15	94.24	93.21	93.87
Honeybee (Vicuna-13B) (Cha et al., 2024)	13B	95.20	96.29	91.18	94.48	93.75	93.17	95.04	93.21	94.39
MC-CoT_F-Large (Tan et al., 2023)	783M	97.47	90.44	93.18	96.97	93.75	94.49	95.30	94.13	94.88
Enigma-COT_Large (Wei et al., 2023)	793M	97.51	84.70	94.73	96.68	91.37	95.89	94.46	93.47	94.11
Single-agent RAG										
Vector RAG (Liu et al., 2025a)	7B	84.54	74.24	86.91	82.74	72.53	90.03	84.51	80.28	82.98
Graph RAG (Liu et al., 2025a)	7B	84.15	75.14	87.64	82.99	73.18	89.69	84.40	80.95	83.16
Web RAG	7B	83.79	72.89	91.82	81.09	70.55	94.01	85.98	79.30	83.59
GPT-4o (Hurst et al., 2024)	-	92.72	93.48	86.09	92.67	90.88	87.60	92.91	88.00	91.16
Multi-agent RAG										
MAO-ARAG (Chen & Zhang, 2025)	-	84.79	90.00	74.77	81.52	86.36	85.00	85.24	80.39	83.49
HM-RAG (Liu et al., 2025b)	-	94.36	90.66	**94.91**	93.79	89.94	96.03	94.42	92.49	93.73
AdaMM-RAG (Ours)	-	**96.31**	**99.00**	90.65	**94.57**	**96.36**	**96.25**	**97.05**	**92.81**	**95.51**

4.2 Main Results

In this section, we conduct an evaluation of AdaMM-RAG against state-of-the-art zero-shot LLMs, VLMs, and single-agent or multi-agent RAG-enhanced approaches across multiple benchmarks. Reported numbers reflect the best of three runs of our method; the full per-run results and means are provided in Appendix D.

Result on ScienceQA. Table 1 reports the ScienceQA comparison. AdaMM-RAG achieves 95.51% average accuracy, outperforming the strongest multi-agent baselines HM-RAG by +1.78% (94.36→96.31 on NAT; 90.66→99.00 on SOC) and MAO-ARAG by +12.02%. The gain over HM-RAG is concentrated in SOC (+8.34) and IMG (+6.42), confirming that planner-guided routing and cross-modal alignment improve multi-source reasoning beyond fixed-topology retrieval. Against fine-tuned VLMs (e.g., MC-CoT-F-Large at 94.88%), AdaMM-RAG is competitive without task-specific parameter updates, demonstrating that retrieval-side coordination alone can rival full fine-tuning on knowledge-grounded QA.

Result on CrisisMMD. Table 2 presents the Top-1 classification accuracy on the CrisisMMD dataset. **AdaMM-RAG** achieves state-of-the-art results with an average accuracy of **80.20%**, outperforming the multi-agent baseline MAO-ARAG by **5.77%** and HM-RAG by **21.65%**. Our method demonstrates robustness across tasks, securing the highest scores in Binary Classification (Task 1: **80.70%**) and Multi-class Classification (Task 2: **81.15%**, Task 2 Merged: **78.74%**). These results highlight the necessity of multi-agent collaboration, as AdaMM-RAG significantly surpasses single-agent retrieval methods (e.g., Graph Retrieval at 56.54%). Furthermore, compared to generalist models like GPT-4o (55.11%) and Qwen2-VL (53.76%), the framework narrows the domain gap on this noisy multimodal benchmark.

[1] https://scienceqa.github.io/leaderboard.html

Table 2: CrisisMMD Top-1 classification accuracy (%). -I = instruction-tuned; Bold = best. Task 1: binary classification; Tasks 2/2 Merged: multi-classification. For compactness we report each family's strongest variant; the complete size sweep is given in Appendix E.

Method	#P	Task 1	Task 2	Task 2 Merged	Average
Single-modal LLMs					
LLaMA-2 (Touvron et al., 2023)	70B	63.15	28.87	36.89	42.97
Qwen2.5 (Yang et al., 2024)	72B	67.95	50.51	50.29	56.25
GPT-4 (Achiam et al., 2023)	-	66.83	47.25	49.44	54.51
Multimodal VLMs					
Qwen2-VL (Wang et al., 2024)	72B-I	65.80	47.21	48.28	53.76
LLaVA (Liu et al., 2023)	13B	60.58	20.14	23.44	34.72
CLIP (Radford et al., 2021)	-	43.36	17.88	20.79	27.34
GPT-4o (Hurst et al., 2024)	-	68.20	47.58	49.55	55.11
Single-agent RAG					
Vector RAG (Liu et al., 2025a)	7B	67.49	45.11	45.94	52.85
Graph RAG (Liu et al., 2025a)	7B	68.90	50.02	50.69	56.54
Multi-agent RAG					
HM-RAG (Liu et al., 2025b)	-	72.06	51.50	52.09	58.55
MAO-ARAG (Chen & Zhang, 2025)	-	79.73	71.62	71.94	74.43
AdaMM-RAG (Ours)	-	**80.70**	**81.15**	**78.74**	**80.20**

4.3 Ablation Study

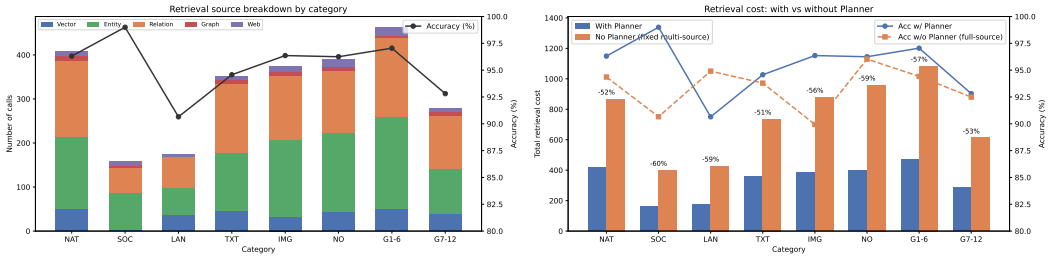
We ablate each core component, Planner Agent (P), Alignment Agent (A), and Decision Agent (D), to measure their individual and coupled contributions on ScienceQA. Note that the w/o-Planner ablation removes routing entirely and invokes all five sources on every query, whereas the fixed multi-source cost baseline in Section 4.4 mirrors HM-RAG's actual topology (vector, graph, web). Table 3 reports per-category and overall accuracy alongside average end-to-end query time.

Table 3: ScienceQA ablation study. ✓ / × : component present/removed. ↑ / ↓: higher/lower is better. Accuracy in %; Time: avg. end-to-end latency per query (s).

P	A	D	NAT ↑	SOC ↑	LAN ↑	Avg ↑	Time (s) ↓
✓	✓	✓	**96.31**	**99.00**	90.65	**95.51**	9.41
×	✓	✓	94.93	**99.00**	**92.52**	95.28	14.27
✓	✓	×	93.55	98.00	86.92	92.92	7.38
✓	×	×	93.55	95.00	85.05	91.75	**2.05**

Planner Agent (P). Removing the Planner Agent (row 2) forces all sources to be queried on every input. Average accuracy changes by only $+0.23$ points (95.51→95.28), within the run-to-run standard deviation in Appendix D, while latency increases by 51.6% (9.41 s→14.27 s). The per-category effect is mixed: NAT drops (96.31→94.93), consistent with indiscriminate retrieval injecting noise, while LAN improves (90.65→92.52) with extra information from web source. This is expected for a mechanism whose value is primarily source-level selectivity rather than accuracy: the Planner invokes only a subset of the five sources (e.g., web only 93 times across 1,492 queries; Figure 2, Section 4.4), realizing the new adaptive dimension, adapting the invoked source set rather than the retrieval procedure inside a fixed corpus, while alignment and refinement account for the larger gains (rows 3–4). We therefore report routing as an efficiency-and-noise mechanism.

Decision Agent (D). Removing D while keeping A (row 3) reduces Avg accuracy from 95.51 to 92.92 (2.59 points), with the largest per-category drop on LAN (90.65 → 86.92). Without the CoT refinement step, the system cannot resolve conflicts or weigh evidence across sources, leading to reliable degradation on linguistically nuanced queries. Latency drops from 9.41 s to 7.38 s (-2.03



(a) (b)

Figure 2: (a) Breakdown of retrieval source calls by category. Stacked bars show Vector, Graph, Entity, Relation and Web calls; the overlaid line reports accuracy. (b) Per-category retrieval cost comparison with vs. without Planner.

s) when D is removed; the larger runtime swing comes from the Alignment Agent, so A dominates compute time. The accuracy loss (-2.59 points, roughly ten times the run-to-run SD of 0.27) nonetheless shows that D is essential for robust fusion.

Alignment Agent (A). Because the Decision Agent operates on the unified space produced by A, A cannot be removed while keeping D functional; we therefore compare rows 3 and 4 (both without D) to isolate A's standalone contribution. Adding A (row 3) improves Avg accuracy over the no-A, no-D baseline (row 4) by +1.17 points (92.92 vs. 91.75), with gains on SOC (+3.00) and LAN (+1.87); at roughly four times the run-to-run SD (0.27) this estimate exceeds the system's stochastic spread. The coupled gain confirms that A and D form a complementary pair.

4.4 Planner Routing and Cost Analysis

Figure 2 analyzes the Planner's per-category source selection and cost on ScienceQA. Entity (1,107) and Relation (1,041) calls dominate across all categories, reflecting the Planner's preference for fine-grained graph primitives over coarse document-level retrieval; Vector calls are comparatively modest (305 total). Graph-traversal calls are rare (54 total, $\leq$10 per category) and Web calls minimal (93), confirming the Planner invokes the expensive or noisy sources only when a query demands them, e.g., Web on G1-6 (20) where up-to-date evidence is needed.

We define retrieval cost as $V = E = R = W = 1$, $G = 2$ (graph covers entity and relation extraction). The fixed multi-source baseline (vector, graph, web) costs 1 + 2 + 1 = 4 per query. This is a normalized operation-count metric counting retrieval calls, not tokens, API cost, or wall-clock time; latency is reported separately in Table 3. The Planner reduces total cost from 5,968 to 2,654 (55.5% reduction), with per-category savings of 51–60%. Compared with the fixed multi-source baseline, Planner routing improves accuracy on 7 of 8 categories while halving cost; LAN degrades. Categories with the largest savings remain among the most accurate (SOC). Combined with the ablation results (Section 4.3), this indicates the Planner removes noise on most categories, with a small and category-dependent accuracy effect. We therefore characterize routing as an efficiency-oriented mechanism with a modest, predominantly non-negative accuracy impact.

5 Conclusion

This paper addresses the coupled problem of multimodal, multi-source, and adaptive coordination in RAG by proposing AdaMM-RAG. The key idea is *query-conditioned coordination*: a training-free Planner routes retrieval on demand across five sources, an alignment module unifies the selected evidence into one semantic space before fusion, and a quality gate triggers fallback only when single-source evidence is insufficient. Experiments on ScienceQA and CrisisMMD show consistent gains over fixed-topology multimodal baselines (HM-RAG) and adaptive baselines (MAO-ARAG) in both accuracy and retrieval cost.

AI USE STATEMENT

In this work, generative AI tools were used only for recommended-disclosure tasks: drafting/editing text, suggesting organization, formatting references, and modifying figures. No required-disclosure tasks were performed with AI (e.g., no synthetic data generation, theoretical development, hypothesis formulation, experiment design, method implementation, or result interpretation). All figures derive from authors' experimental data and were manually verified. All AI-assisted content was reviewed for factual accuracy, plagiarism, and scientific integrity through manual literature survey and author verification. We take full responsibility for the final content of this work.

ETHICS STATEMENT

Our work uses two publicly available datasets: ScienceQA (Lu et al., 2022), a benchmark of science multiple-choice questions, and CrisisMMD (Alam et al., 2018), a collection of crisis-related social media posts. Both are established research benchmarks with no personally identifying information introduced by our processing. The models used in this work (Qwen3-VL, Qwen3-Embedding, and Qwen3-8B) are publicly released open-weight models; we do not train on or distribute any user data, and no human participants or sensitive personal data are involved. We acknowledge potential biases in social media corpora used for crisis analysis and in the underlying large language models; results reflect benchmark performance and should not be interpreted as operational guidance. All stakeholders of the datasets retain licensing terms; please refer to the original sources for attribution. We present results transparently, including a stability analysis across multiple runs (Appendix D), to avoid over-claiming.

REPRODUCIBILITY STATEMENT

We describe the complete method in Section 3, including the Planner's design and source-selection protocol (Section 3.2), the five retrievers and their design (Section 3.3 and Appendix C), the alignment and decision modules, and the quality gate with its bounded fallback (Section 3.5, 3.6). All models are publicly available open-weight checkpoints and all datasets are public benchmarks; experimental settings (model names, embedding choices, retrieval cost model, and query counts) are specified in Section 4.1 and the appendix. To support reproducibility we report full multi-run results (mean and standard deviation) in Appendix D, and provide source code at `https://anonymous.4open.science/r/RAG-Adamm-FC83`. We encourage the community to reproduce and extend our results under the stated settings.

REFERENCES

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Firoj Alam, Ferda Ofli, and Muhammad Imran. Crisismmd: Multimodal twitter datasets from natural disasters. In *Proceedings of the 12th International AAAI Conference on Web and Social Media (ICWSM)*, 2018.

Akari Asai et al. Self-rag: Learning to retrieve, generate, and critique through self-reflection. *ICLR*, 2024. arXiv:2310.11511.

Tom Brown, Benjamin Mann, Nick Ryder, et al. Language models are few-shot learners. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Junbum Cha, Wooyoung Kang, Jonghwan Mun, and Byungseok Roh. Honeybee: Locality-enhanced projector for multimodal llm. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Jiang Chang et al. Main-rag: Multi-agent filtering retrieval-augmented generation. *ACL*, 2025. arXiv:2501.00332.

X. Chen and Y. Zhang. Mao-arag: Multi-agent orchestration for adaptive retrieval-augmented generation. *arXiv preprint*, 2025. arXiv:2508.01005.

CIIR@LiveRAG. Optimizing multi-agent retrieval augmented generation through self-training. *arXiv preprint*, 2025. arXiv:2506.10844.

Darren Edge et al. From local to global: A graph rag approach to query-focused summarization. *arXiv preprint*, 2024. arXiv:2404.16130.

Yunfan Gao et al. Retrieval-augmented generation for large language models: A survey. *arXiv preprint*, 2023. arXiv:2312.10997.

Yunfan Gao et al. Modular rag: Transforming rag systems into lego-like reconfigurable frameworks. *arXiv preprint*, 2024. arXiv:2407.21059.

Deepanway Ghosal, Navonil Majumder, Roy Ka-Wei Lee, Rada Mihalcea, and Soujanya Poria. Language guided visual question answering: Elevate your multimodal language model using knowledge-enriched prompts. *arXiv preprint arXiv:2310.20159*, 2023.

Sameera Horawalavithana, Sai Munikoti, Ian Stewart, and Henry Kvinge. Scitune: Aligning large language models with scientific multimodal instructions. *arXiv preprint arXiv:2307.01139*, 2023.

Cheng-Yu Hsieh, Sibei Chen, Chun-Liang Li, Yasuhisa Fujii, Alexander Ratner, Chen-Yu Lee, Ranjay Krishna, and Tomas Pfister. Tool documentation enables zero-shot tool-usage with large language models. *arXiv preprint arXiv:2308.00675*, 2023.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

Soyeong Jeong et al. Adaptive-rag: Learning to adapt retrieval-augmented llms through question complexity. *NAACL*, 2024. arXiv:2403.14403.

Patrick Lewis et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *NeurIPS*, 2020. arXiv:2005.11401.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in Neural Information Processing Systems*, 36:34892–34916, 2023.

Junming Liu, Siyuan Meng, Yanting Gao, Song Mao, Pinlong Cai, Guohang Yan, Yirong Chen, Zilin Bian, Botian Shi, and Ding Wang. Aligning vision to language: Text-free multimodal knowledge graph construction for enhanced llms reasoning. *arXiv preprint arXiv:2503.12972*, 2025a.

Pei Liu et al. Hm-rag: Hierarchical multi-agent multimodal retrieval augmented generation. *ACM Multimedia (ACM MM)*, 2025b. arXiv:2504.12330.

Pan Lu, Swaroop Mishra, Tony Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. In *The 36th Conference on Neural Information Processing Systems (NeurIPS)*, 2022.

Debjyoti Mondal, Suraj Modi, Subhadarshi Panda, Rituraj Singh, and Godawari Sudhakar Rao. Kam-cot: Knowledge augmented multimodal chain-of-thoughts reasoning. In *AAAI Conference on Artificial Intelligence*, 2024.

OpenAI. Chatgpt: Optimizing language models for dialogue. `https://openai.com/blog/chatgpt/`, 2023.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pp. 8748–8763. PMLR, 2021.

Parth Sarthi et al. Raptor: Recursive abstractive processing for tree-organized retrieval. *ICLR*, 2024. arXiv:2401.18059.

Tianjun Singh et al. Agentic retrieval-augmented generation: A survey on agentic rag. *arXiv preprint*, 2025. arXiv:2501.09136.

Cheng Tan, Jingxuan Wei, Zhangyang Gao, Linzhuang Sun, Siyuan Li, Xihong Yang, and Stan Z Li. Boosting the power of small multimodal reasoning models to match larger models with self-consistency training. *arXiv preprint arXiv:2311.14109*, 2023.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024.

Jingxuan Wei, Cheng Tan, Zhangyang Gao, Linzhuang Sun, Siyuan Li, Bihui Yu, Rui (Ami) Guo, and Stan Z. Li. Enhancing human-like multimodal reasoning: a new challenging dataset and comprehensive framework. *Neural Computing and Applications*, 36:20849 – 20861, 2023.

Shi-Qi Yan et al. Corrective retrieval augmented generation. *arXiv preprint*, 2024. arXiv:2401.15884.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

Xiaocui Yang, Wenfang Wu, Shi Feng, Ming Wang, Daling Wang, Yang Li, Qi Sun, Yifei Zhang, Xiaoming Fu, and Soujanya Poria. Mm-bigbench: Evaluating multimodal models on multimodal content comprehension tasks. *arXiv preprint arXiv:2310.09036*, 2023.

Ge Zheng, Bin Yang, Jiajin Tang, Hong-Yu Zhou, and Sibei Yang. Ddcot: Duty-distinct chain-of-thought prompting for multimodal reasoning in language models. *Advances in Neural Information Processing Systems*, 36:5168–5191, 2023.

A Query Examples

Figure 3 and Figure 4 illustrate two representative ScienceQA queries and how the Planner routes them across sources. These examples show that routing decisions are not arbitrary: the chosen sources directly match the knowledge scale demanded by the query, and omitting irrelevant sources neither reduces answer quality nor introduces noise, consistent with the aggregate gains in Table 1 and the cost analysis in Section 4.4.

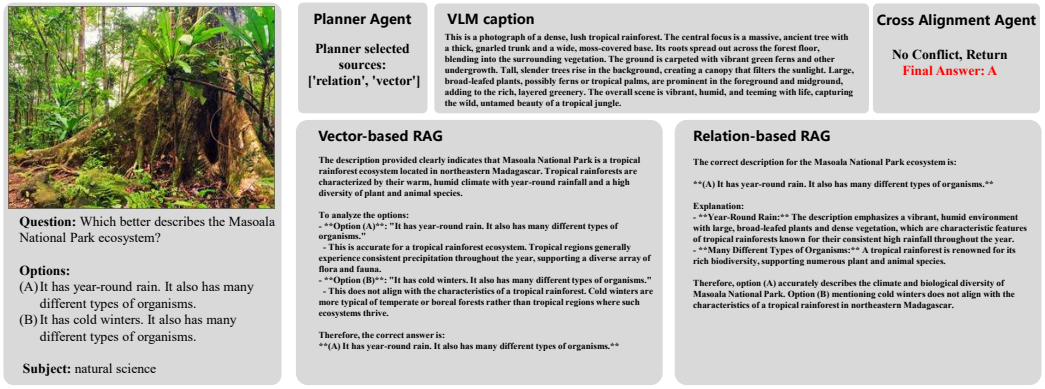


Figure 3: Representative ScienceQA query and the sources selected by the Planner, with no conflict from different sources.



Figure 4: Representative ScienceQA query and the sources selected by the Planner, with conflict from different sources and refinement in decision agent.

B Source Distribution and Fallback Analysis

Figure 5 reports how many sources the Planner selects per query and how often fallback triggers. Across all categories, the vast majority of queries use exactly two sources (57.9–89.1%), and 3+ source plans are rare (0–3.2%), confirming that the Planner's on-demand routing rarely invokes the full source set, in contrast to the fixed multi-source baseline, which always pays the cost of all sources. Single-source queries account for 9.8–41.0% of each category and are concentrated where one source suffices (e.g., IMG 30.5%, NO 39.6%).

For single-source queries, the quality gate triggers fallback only when the Decision Agent deems the evidence insufficient. The overall fallback rate is 4.2%, i.e., the vast majority of single-source queries are answered on the first pass. The rate varies by category (0.0–16.7%), being highest on G7-12, where complex reasoning questions often require evidence beyond a single source, and zero on SOC and G1-6. The low fallback rate is consistent with (i) the Planner's source selection being accurate enough that a single source usually suffices, and (ii) the fallback mechanism being a bounded safety net rather than a frequent, costly re-planning path; it could also reflect conservative routing or a lenient insufficiency check, so we do not treat it as a verification.

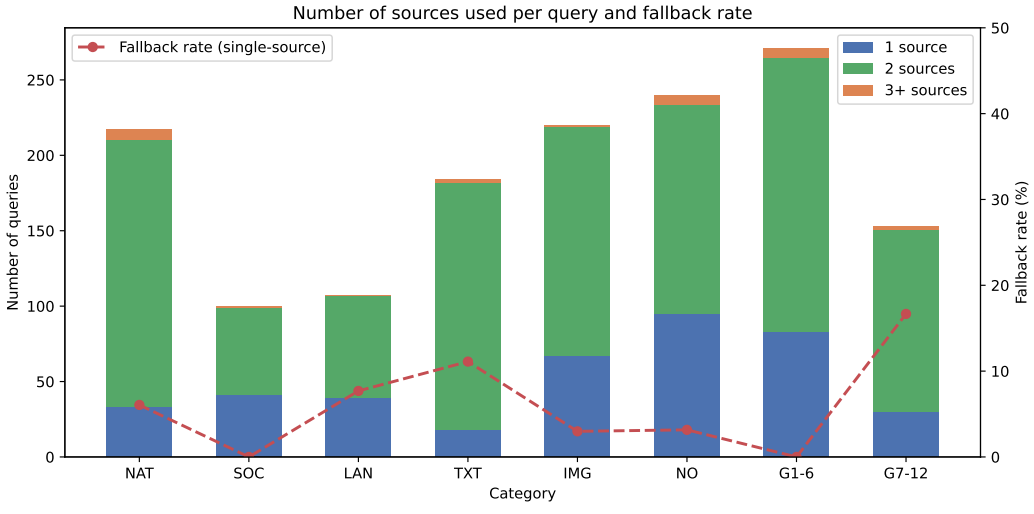


Figure 5: Proportion of queries by number of sources used (stacked bars) and fallback rate on single-source queries (line).

C Retrieval Source Design and Cost Justification

C.1 Source Selection Rationale

The five retrievers are chosen to cover distinct knowledge scales and information granularities, rather than as redundant alternatives (Table 4). The *vector* source performs pure semantic similarity search over raw text chunks, serving simple fact lookup at the fastest speed. The *entity* source is entity-centric and expands one-hop relations, retrieving low-level attributes of a specific object (e.g., the founding year of a company). The *relation* source is relation-centric and recovers both endpoints, supporting high-level abstraction and cross-document reasoning such as macro trends or comparisons. The *graph* source runs entity and relation retrieval in parallel and merges the two, trading speed for multi-hop reasoning capacity on complex questions. Finally, the *web* source queries a live search API (Exa) for time-sensitive information that no static knowledge base can hold. Because each source answers a different type of question, combining a subset of them rather than always calling all is both necessary for coverage and sufficient for correctness, which motivates the Planner's per-query selection in Section 3.2.

Table 4: Design and cost model of the five retrieval sources.

Source	Core mechanism	Cost
vector	pure vector similarity over text chunks	1
entity	entity-centric, 1-hop relation expansion	1
relation	relation-centric, recovers both endpoints	1
graph	entity + relation parallel merge	2
web	live search API (Exa)	1

C.2 Cost Estimation

The cost model assigns unit cost 1 to each single-operation retrieval (vector, entity, relation, web) and cost 2 to the graph source, because a graph call internally combines entity and relation extraction plus a merge step, twice the work of any single-operation source. The fixed multi-source baseline therefore costs $1 + 1 + 2 = 4$ per query (invoking vector, graph, and web unconditionally), and no-Planner cost is computed as $4\times$ (number of queries per category) using the actual query counts (Figure 2). This per-operation weighting makes the comparison conservative: it charges the graph source at its real double cost, and does not inflate the Planner's savings by counting any uninvoked source. The measured 55.5% overall reduction therefore reflects genuine per-query routing decisions rather than an artifact of the cost definition.

D Stability Across Multiple Runs

To address concerns about single-run variance, we repeated both evaluations three times under identical settings and report per-run accuracy in Tables 5 and 6. For ScienceQA, the overall average accuracy is $95.20\pm0.27\%$ across runs (95.05, 95.51, 95.05), with per-category standard deviations ranging from 0.27 (NAT) to 1.47 (NO). Every run outperforms the strongest fixed-topology baseline HM-RAG (93.73%) by at least $+1.32$ points, and the worst-case run (95.05%) still exceeds it with a margin greater than one standard deviation of the run-to-run variation.

For CrisisMMD, the average accuracy across runs is $80.00\pm0.21\%$ (80.20, 80.00, 79.79). Every run outperforms HM-RAG (58.55%) by at least $+21.24$ points and MAO-ARAG (74.43%) by at least $+5.36$ points; the mean margins are $+21.45$ and $+5.57$, respectively. These results confirm that the reported gains on both benchmarks reflect genuine design improvements rather than favorable stochastic draws.

E Results for Full Comparison

Table 7 reports the complete CrisisMMD comparison used in Section 4.2, including all model sizes per family that the main-text Table 2 abbreviates for space. The baselines span four

Table 5: Three runs of the ScienceQA evaluation (Accuracy %). HM-RAG baseline: 93.73%; MAO-ARAG baseline: 83.49%.

Run	NAT	SOC	LAN	TXT	IMG	NO	G1-6	G7-12	Avg
1	95.85	98.00	90.65	95.11	98.18	95.00	96.68	92.16	95.05
2	96.31	99.00	90.65	94.57	96.36	96.25	97.05	92.81	95.51
3	95.85	97.00	91.59	97.28	97.27	93.33	96.31	92.81	95.05
Mean±SD	96.00±0.27	98.00±1.00	90.96±0.54	95.65±1.43	97.27±0.91	94.86±1.47	96.68±0.37	92.59±0.38	95.20±0.27

Table 6: Three runs of the CrisisMMD evaluation (Classification Accuracy %). HM-RAG baseline: 58.55%; MAO-ARAG baseline: 74.43%.

Run	Task 1	Task 2	Task 2 Merged	Average
1	80.70	81.15	78.74	80.20
2	80.83	80.63	78.53	80.00
3	80.51	81.26	77.59	79.79
Mean±SD	80.68±0.16	81.01±0.34	78.29±0.61	80.00±0.21

groups. (i) Single-modal LLMs: LLaMA-2 (7B/13B/70B) (Touvron et al., 2023) and Qwen2.5 (7B/32B/72B) (Yang et al., 2024) as text-only zero-shot references, together with GPT-4 (Achiam et al., 2023). (ii) Multimodal VLMs: Qwen2-VL (2B-I/7B-I/72B-I) (Wang et al., 2024), LLaVA (7B/13B/34B) (Liu et al., 2023), CLIP (Radford et al., 2021), and GPT-4o (Hurst et al., 2024), which condition on images but do not retrieve external evidence. (iii) Single-agent RAG: Vector RAG and Graph RAG (Liu et al., 2025a), which retrieve over one fixed corpus. (iv) Multi-agent RAG: HM-RAG (Liu et al., 2025b), a fixed-topology multimodal multi-agent baseline, and MAO-ARAG (Chen & Zhang, 2025), an adaptive text-only baseline (augmented with our vision-language bridging at its input stage, as in Section 3.4). All numbers are quoted from the corresponding papers except HM-RAG, MAO-ARAG, and AdaMM-RAG, which we re-run under our protocol (best of three runs; Appendix D reports per-run results and means).

F Prompt Design of the Chain-of-Thought Agents

We describe the system prompts of the Agents. Each prompt enforces a structured chain-of-thought (CoT) reasoning protocol and a strict output schema, which together ensure that the LLM's routing and verification decisions are transparent, deterministic, and auditable.

F.1 Planner Agent Prompt

The Planner prompt (Listing 1) casts routing as a structured decision task. Its design encodes three components that are critical for reliable, training-free routing:

(i) Per-source semantics. Each of the five sources is defined by the *knowledge scale* it targets, not by a generic description: `entity` for instance-level facts and one-hop attributes, `relation` for macro-level patterns and cross-entity statistics, `graph` for multi-hop causal chains, `vector` for exploratory semantic search over raw chunks, and `web` for real-time information beyond a static knowledge base. Anchoring each definition to query patterns ("compare", "over time", "how does A affect B") makes the routing decision grounded rather than heuristic.

(ii) Selection rules. The prompt mandates at least two sources except for exact single-static-fact lookups, and instructs the model to keep sources complementary rather than mutually exclusive. This prevents the two failure modes of naive routing: over-selection (invoking irrelevant sources, hence noise) and under-selection (dropping a necessary source, hence insufficient evidence).

(iii) Four-step CoT protocol. The model must (1) classify the query type along five dimensions, (2) score each source from 0 to 3 with justification, (3) apply the threshold (≥ 1) and coverage checks, and (4) confirm the final set with a summary. This explicit reasoning chain decouples the *decision* from the *output*: even when the final source set is small, the analysis field records why borderline sources were rejected, which supports the interpretability analysis in Section 4.4.

Table 7: CrisisMMD Top-1 classification accuracy (%). -I = instruction-tuned; Bold = best. Task 1: binary classification; Tasks 2/2 Merged: multi-classification. Full comparison used in Section 4.2.

Method	#P	Task 1	Task 2	Task 2 Merged	Average
Single-modal LLMs					
LLaMA-2 (Touvron et al., 2023)	7B	62.32	18.32	21.45	34.03
	13B	63.80	21.82	33.15	39.59
	70B	63.15	28.87	36.89	42.97
Qwen2.5 (Yang et al., 2024)	7B	65.04	44.52	45.33	51.63
	32B	67.28	46.94	47.07	53.76
	72B	67.95	50.51	50.29	56.25
GPT-4 (Achiam et al., 2023)	-	66.83	47.25	49.44	54.51
Multimodal VLMs					
Qwen2-VL (Wang et al., 2024)	2B-I	47.56	7.60	7.42	20.86
	7B-I	62.45	32.68	34.20	43.11
	72B-I	65.80	47.21	48.28	53.76
LLaVA (Liu et al., 2023)	7B	54.00	28.01	30.61	37.54
	13B	60.58	20.14	23.44	34.72
	34B	56.44	25.15	25.07	35.55
CLIP (Radford et al., 2021)	-	43.36	17.88	20.79	27.34
GPT-4o (Hurst et al., 2024)	-	68.20	47.58	49.55	55.11
Single-agent RAG					
Vector RAG (Liu et al., 2025a)	7B	67.49	45.11	45.94	52.85
Graph RAG (Liu et al., 2025a)	7B	68.90	50.02	50.69	56.54
Multi-agent RAG					
HM-RAG (Liu et al., 2025b)	-	**72.06**	**51.50**	**52.09**	**58.55**
MAO-ARAG (Chen & Zhang, 2025)	-	**79.73**	**71.62**	**71.94**	**74.43**
AdaMM-RAG (Ours)	-	**80.70**	**81.15**	**78.74**	**80.20**

The output is constrained to a strict JSON schema with exactly three fields (`analysis`, `sources`, and `reason`) so that parsing is unambiguous and each field can be logged for downstream analysis (e.g., recovering the CoT trace for the routing study).

F.2 Cross Alignment Agent Prompt

The Cross Alignment Agent prompt (Listing 2) performs four structured steps before fusion: relevance filtering, conflict detection, conflict resolution, and context generation. Its key design choices are as follows.

(i) Conflict detection distinguishes two conflict types. The prompt separates *cross-source* contradictions (two sources make directly opposing claims) from *intra-source* contradictions (a single snippet is self-inconsistent), while explicitly excluding complementary information from the definition of conflict. This distinction matters for the downstream Decision Agent: intra-source contradictions are a signal to discard the snippet, whereas cross-source conflicts may be resolved or surfaced.

(ii) Resolution follows a strict priority order. Conflicts are resolved by applying, in sequence, self-consistency (discard internally inconsistent snippets), concreteness (prefer specific numerical values, named entities, or verifiable facts), freshness (prefer recent timestamps when available), and logical soundness. If none of the rules resolve a conflict, the prompt *keeps both pieces* and marks the disagreement rather than silently dropping evidence.

(iii) Two booleans gate output. The JSON output exposes `conflict_in_evidence` and `conflict_in_conclusion`. The former indicates intra-source or cross-source contradictions before resolution; the latter indicates whether unresolved contradictory conclusions remain after resolution. These flags feed directly into the consensus early exit (Section 3.4) and the Decision Agent: if `conflict_in_conclusion` is `false` and all sources reach the same answer, the

Listing 1 Planner Agent prompt for routing (abridged). Full is provided in the released code.

```
SYSTEM: You are a query routing engine. Select ALL data
sources relevant to the query; most queries need 2-4.

SOURCE DEFINITIONS
1. entity    - named entity core description, static
               attributes, or one-hop relationships
2. relation  - connections/patterns/trends/aggregates
               across multiple entities ("compare",
               "top", "over time", "between X and Y")
3. graph     - entity details AND multi-hop relational
               context (causal chains, influence paths)
4. vector    - exploratory/open-ended/vague queries,
               semantic chunk search over documents
5. web       - real-time/external/current information
               (recent dates, news, live prices)

SELECTION RULES
- At least 2 sources, unless an exact single static-fact
  lookup (then `entity` only).
- Sources are complementary, not mutually exclusive.

CHAIN-OF-THOUGHT PROTOCOL (write all 4 steps in analysis)
Step 1  Identify query type (named entity? comparison?
        causal multi-hop? exploratory? real-time?)
Step 2  Score each source 0-3 with justification
Step 3  Select score>=1; verify >=2 sources and coverage
Step 4  Confirm final set; summarize why it fits

OUTPUT (raw JSON only, no code fences)
{ "analysis": "...", "sources": [...], "reason": "..." }
```

pipeline exits early; otherwise the Decision Agent refines the marked conflicts. The `context` field is set to "Insufficient evidence to form a consistent context." when all evidence is discarded, which triggers the single-source insufficiency signal for the quality gate (Section 3.6).

F.3 Decision Agent Prompts

The Decision Agent uses two prompts: one for multi-source refinement and one for single-source verification, that correspond exactly to the two branches of Section 3.5.

Multi-Source Refinement Prompt. The multi-source prompt (Listing 3) implements Equation 6. It performs a four-step process: analyze the candidate answers from different sources and identify core contradictions, cross-reference these contradictions against the aligned evidence to judge which source or deduction is most reliable, select the single best answer, and then independently check the answer, re-deducing it if it is flawed. The final answer is constrained to a valid option character. The explicit "Check Answer" step realizes the voting-free design: rather than trusting a majority candidate, the model must actively falsify its own choice against the evidence.

Single-Source Verification Prompt. The single-source prompt (Listing 4) implements the three-case equation (Equation 7). It runs three steps: analyze the evidence, cross-reference the single candidate answer for direct contradictions while evaluating whether the evidence is sufficient to verify or correct it, and then emit a verdict. Its output exposes two booleans that map directly onto the Decision Agent's branches: `is_contradiction` (whether the candidate conflicts with the evidence) and `is_sufficient` (whether the evidence fully supports the verdict). The tuple $(\neg$`is_contradiction`$,$ `is_sufficient`$)$ yields the clean / contradiction / insufficiency cases, and a `false` `is_sufficient` value raises the fallback signal consumed by the quality gate (Section 3.6).

Listing 2 Cross Alignment Agent prompt (abridged).

```
SYSTEM: You are a multi-source context alignment module.
Align multi-source evidence into one coherent context.

Step 1  Relevance Filtering
        Discard irrelevant evidence.

Step 2  Conflict Detection
        - Cross-source: sources make contradictory claims.
        - Intra-source: one snippet self-contradicts.
        Complementary info is NOT conflict.

Step 3  Conflict Resolution (strict priority)
        1. Self-consistency: discard self-contradictory
           snippets entirely.
        2. Concreteness: prefer specific numbers, named
           entities, verifiable facts.
        3. Freshness: prefer more recent evidence.
        4. Logic: retain more logically sound reasoning.
        5. If unresolvable, KEEP both and mark conflict.

Step 4  Context Generation
        Synthesize validated evidence into one coherent,
        concise context. If all evidence is discarded,
        set context to "Insufficient evidence to form a
        consistent context."

OUTPUT (raw JSON only)
{
  "reasoning": "Step 1: ... Step 2: ... Step 3: ... Step 4: ...",
  "conflict_in_evidence": true/false,
  "conflict_in_conclusion": true/false,
  "context": "..."
}
```

Listing 3 Decision Agent multi-source refinement prompt (abridged).

```
SYSTEM: You are an expert decision-maker. Using the
evidence, pick the best answer to the question.

Step 1  Analyze Candidates
        Review candidate answers from different sources;
        identify core contradictions.
Step 2  Evaluate Evidence
        Cross-reference contradictions with the evidence;
        judge which source or logical deduction is most
        reliable.
Step 3  Final Decision
        Select the single best answer from the options.
Step 4  Check Answer
        If the answer is flawed or incorrect, independently
        deduce and provide the correct answer.

OUTPUT (raw JSON only)
{ "reasoning": "Step 1: ... Step 2: ... Step 3: ... Step 4: ...",
  "final_answer": "correct option character" }
```

G Cost Model Sensitivity

Section 4.4 assigns unit cost 1 to vector, entity, relation, and web calls and cost 2 to graph calls (a graph call internally runs entity and relation extraction). Here we vary the graph weight w and re-

Listing 4 Decision Agent single-source verification prompt (abridged).

```
SYSTEM: You are a strict fact-checker and answer corrector.
Verify whether the answer contradicts the evidence, and if
so, provide the correct answer.

Step 1  Analyze Evidence
        Understand the core facts in the evidence.
Step 2  Cross-Reference & Sufficiency Check
        Compare the single candidate against the aligned
        evidence; check for direct contradictions or factual
        errors; evaluate whether the evidence is sufficient
        to verify or correct the candidate.
Step 3  Verdict & Correction
        - No contradiction: is_contradiction=false.
        - Contradiction: is_contradiction=true; deduce the
          correct answer (option character if options given).
        - is_sufficient=true if evidence is complete and
          directly supports the verdict; else false.

OUTPUT (raw JSON only)
{ "reasoning": "Step 1: ... Step 2: ... Step 3: ...",
  "is_contradiction": true/false,
  "correct_answer": "correct option character",
  "is_sufficient": true/false }
```

compute the overall and per-category savings, keeping the fixed multi-source baseline vector, graph, web at cost $w + 2$ per query. All counts are the actual per-category calls reported in Appendix C.

Table 8: Cost savings under alternative graph weights w. Planner cost and fixed baseline cost are summed over all 1,492 queries.

Graph weight w	Planner cost	Fixed baseline cost	Savings
$w = 1$ (all sources equal)	2,600	4,476	41.9%
$w = 2$ (reported)	2,654	5,968	55.5%
$w = 3$ (graph double-premium)	2,708	7,460	63.7%

The savings figure is robust: even pricing every source identically ($w = 1$, which removes the graph premium entirely) the Planner still cuts retrieval cost by 41.9%, and the largest weighting yields 63.7%. The swing is driven mainly by the fixed baseline's composition (which sources it invokes) rather than by the planner side: graph calls constitute only 54 of the 2,654 weighted calls, so the graph weight has a small effect on planner cost. We report the normalized operation-count metric as the primary figure; end-to-end wall-clock latency is measured separately in Table 3 and is not modeled here.