MMGraph-Agent: Agentic Multimodal RAG via Cache-Inspired Multimodal Knowledge HyperGraphs

Haoran Luo[1†], **Ziyue Zhu**[2*], **Shangyang Wu**[2], **Linhao Luo**[3], **Yikai Guo**[4], **Qika Lin**[5], **Fangzhi Xu**[6], **Jiapu Wang**[8], **Xiaobao Wu**[9], **Yifan Zhu**[2], **Luu Anh Tuan**[1,7*]
[1]Nanyang Technological University [2]Beijing University of Posts and Telecommunications
[3]Monash University [4]Beijing Institute of Computer Technology and Application
[5]National University of Singapore [6]Xi'an Jiaotong University [7]VinUniversity
[8]Nanjing University of Science and Technology [9]Shanghai Jiao Tong University
haoran.luo@ntu.edu.sg, zyzhu@bupt.edu.cn, anhtuan.luu@ntu.edu.sg

Abstract

Driven by increasingly complex real-world applications, Retrieval-Augmented Generation (RAG) has evolved from textual pipelines to multimodal settings, comprising two complementary dimensions: knowledge organization and retrieval enhancement. However, existing methods still face challenges of high construction cost, context window limitations, and the lack of unified training across offline and online knowledge sources. To address these challenges, we propose **MMGraph-Agent**, a unified multimodal RAG framework based on cache-inspired multimodal knowledge hypergraphs. MMGraph-Agent enables extremely lightweight hypergraph construction with zero API cost and dynamic memory-based retrieval that alleviates long-context issues. Experiments across offline, online, and joint retrieval settings demonstrate consistent improvements in performance, efficiency, and architectural unification. Our software and data are publicly available[1].

1 Introduction

Retrieval-Augmented Generation (RAG) [27] systems have evolved from text-only pipelines to multimodal settings [65, 50] spanning text, images, tables, and diagrams. Building on this paradigm, multimodal RAG thus decomposes into two dimensions: **(i) Knowledge Organization**, which structures multimodal knowledge into coherent representations [9, 35, 14], and **(ii) Retrieval Enhancement**, which governs how relevant knowledge is retrieved and then injected into the generation process [24, 60].

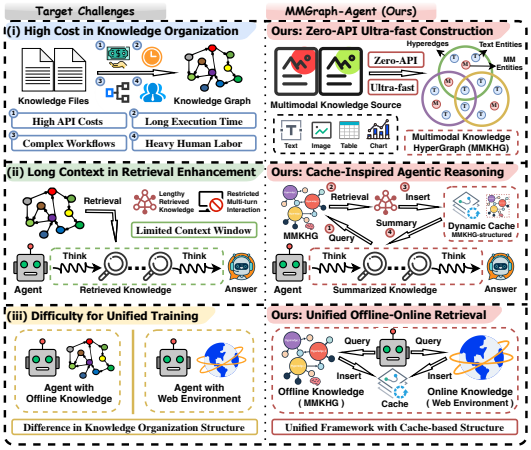


Figure 1: Target challenges and MMGraph-Agent.

Existing RAG approaches range from chunk-based text retrieval to their image-level extensions [65, 50]. As for knowledge organization, GraphRAG [9] and HyperGraphRAG [35] use entity-relation graphs to enhance knowledge structures, whereas MMGraphRAG [53] and RAG-Anything [14] extend this to multimodal knowledge graphs and add multimodal entities. For retrieval enhancement, Search-R1 [24] and Graph-R1 [34] employ agentic reinforcement learning for active retrieval, and MMSearch-R1 [60] further enables multimodal agentic search on the web.

[†]Project Lead. [*]Corresponding Authors. [1]https://github.com/LHRLAB/MMGraph-Agent

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

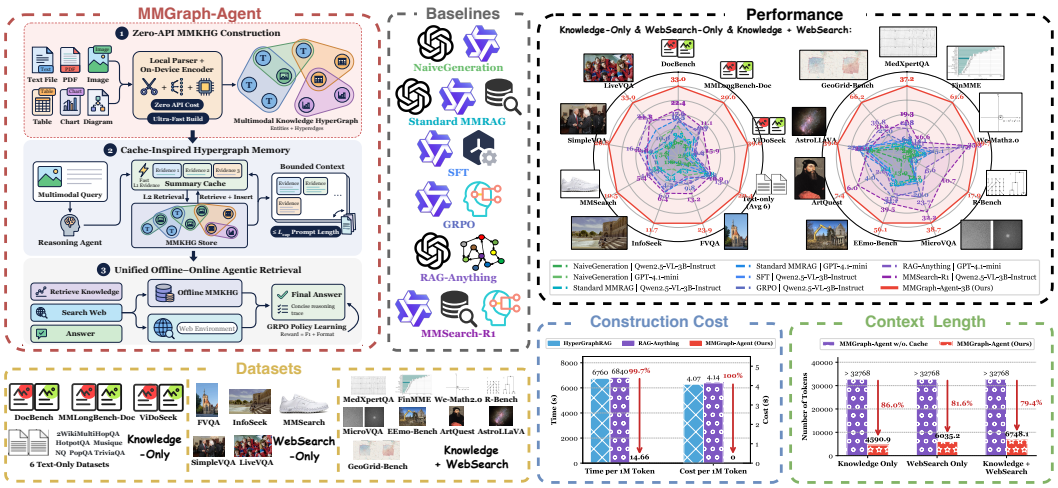


Figure 2: Overview of MMGraph-Agent: a multimodal RAG framework built on cache-inspired knowledge hypergraphs supporting unified joint retrieval with reduced cost and context overhead.

However, as shown in Figure 1, three main challenges remain. **(i) High Cost in Knowledge Organization.** Constructing multimodal knowledge graphs often incurs substantial costs [35, 14], relying on manual annotation or expensive API calls, which limits scalability. **(ii) Long Context in Retrieval Enhancement.** Multi-round retrieval is constrained by model context windows [2]; as retrieved evidence accumulates, context overflow becomes increasingly likely [34, 66], degrading reasoning quality. **(iii) Lack of a Unified Framework.** Unifying local private knowledge [14] and online web knowledge [60] within a single framework is challenging due to structural differences.

To address these challenges, we propose **MMGraph-Agent**, a unified multimodal RAG framework based on cache-inspired multimodal knowledge hypergraphs. As shown in Figure 2, we first propose a lightweight **multimodal hypergraph construction method with zero API cost**, substantially reducing both construction cost and build time while still preserving knowledge completeness. We further design a **cache-inspired multimodal hypergraph memory** that dynamically stores, updates, and serves retrieved multimodal evidence during iterative agentic reasoning, thereby alleviating context window limitations and improving multi-round retrieval efficiency. Furthermore, MMGraph-Agent also integrates multimodal reinforcement learning (RL) to enable **coordinated retrieval from offline knowledge corpora and online web environments** within a single unified RAG framework.

To validate its effectiveness, we conduct experiments in three settings: Knowledge-Only, WebSearch-Only, and Knowledge+WebSearch. Results demonstrate that MMGraph-Agent consistently outperforms a range of baselines spanning generation, RAG-based, and training-based methods in answer accuracy and retrieval efficiency, while substantially reducing construction cost and context token usage, thereby demonstrating its strong potential for real-world multimodal question answering.

2 Related Work

Multimodal RAG. RAG grounds LLMs with external knowledge [27] and has evolved from text-only pipelines to multimodal settings [65, 50]. For *knowledge organization*, prior work moves from flat or hierarchical chunk retrieval [46, 1] to entity-relation graphs and higher-order hyperedges [9, 61, 29, 15, 16, 32, 37, 35, 54, 56, 30, 38, 20], further extended to multimodal knowledge graphs and hybrid vector-graph designs [53, 14, 42, 19, 7, 67, 31]. In this work, MMGraph-Agent builds a multimodal hypergraph that preserves cross-modal relational structure at zero API cost and high build speed.

Agentic Retrieval. For *retrieval enhancement*, LLM-based agents reframe retrieval as sequential decision-making, letting the model decide *when* and *what* to retrieve [64, 55]. Recent methods drive iterative, reasoning-grounded retrieval over textual and graph corpora [24, 48, 34] and extend it to multimodal and open-web search [60, 18, 8, 49, 59, 28, 33, 62, 21, 51]. Building on this line of work, MMGraph-Agent introduces a cache-inspired hypergraph memory to keep context bounded across long retrieval trajectories, and a unified policy that jointly retrieves from offline and online sources.

Table 1: Comparison of representative baselines. StandardRAG/MMRAG: flat (multi-)modal vector RAG; HyperGraphRAG: n-ary entity hyperedges; RAG-Anything: paired structural and textual graphs; Graph-R1/MMSearch-R1: RL agents over a hypergraph or web; MMGraph-Agent (Ours): unifies a multimodal hypergraph with live web data via hierarchical cache/graph/web retrieval.

Method	**Knowledge Organization**	**Retrieval Enhancement**
StandardRAG [27]	$\mathcal{K} = \{c_i\}_{i=1}^{N},\ \mathbf{e}_i = \mathrm{Enc}(c_i)$	$\mathcal{R}(q) = \text{Top-}k_{c_i \in \mathcal{K}}\, \mathrm{sim}(\mathrm{Enc}(q), \mathbf{e}_i)$
StandardMMRAG [65]	$\mathcal{K} = \{I_i\}_{i=1}^{N},\ \mathbf{e}_i = \mathrm{MMEnc}(I_i)$	$\mathcal{R}(q) = \text{Top-}k_{I_i \in \mathcal{K}}\, \mathrm{sim}(\mathrm{MMEnc}(q), \mathbf{e}_i)$
HyperGraphRAG [35]	$\mathcal{G}_H = (\mathcal{V}, \mathcal{E}_H),\ e \in 2^{\mathcal{V}}$ n-ary relational entity hyperedges.	$\mathcal{V}_q \to \mathcal{E}_q = \{e :\ e \cap \mathcal{V}_q \neq \emptyset\}$ Staged entity-then-hyperedge retrieval.
RAG-Anything [14]	$\mathcal{G} = (\mathcal{G}_s, \mathcal{G}_t)$ Dual structural and textual graphs.	$\mathcal{R}(q) = \mathrm{Nav}(\mathcal{G}_s, q) \cup \mathrm{Sim}(\mathcal{G}_t, q)$ Graph navigation with semantic matching.
Graph-R1 [34]	$\mathcal{G}_H = (\mathcal{V}, \mathcal{E}_H)$ Lightweight higher-order hypergraph.	$a_t \sim \pi_\theta(\cdot \mid s_t)$ RL-driven multi-turn hyperedge search.
MMSearch-R1 [60]	$\mathcal{W},\ \mathcal{T} = \{\text{text}, \text{img}\}$ Dynamic multimodal text-image web.	$a_t \sim \pi_\theta(\cdot \mid s_t, \mathcal{T})$ RL-driven multi-turn web search agent.
MMGraph-Agent (Ours)	$\mathcal{G}_H^m \cup \mathcal{W}$ Multimodal hypergraph with web data.	$a_t \sim \pi_\theta(\cdot \mid s_t, \mathcal{M}_1, \mathcal{G}_H^m, \mathcal{W})$ Hierarchical cache/graph/web retrieval.

3 Methodology: MMGraph-Agent

Problem Formulation. We study multimodal retrieval-augmented question answering over a static private corpus and the live web; Table 1 contrasts how representative paradigms instantiate knowledge organization and retrieval enhancement. Formally, let $\mathcal{K} = \{d_i\}_{i=1}^{N}$ be a static corpus of N multimodal documents over modalities $\mathbb{M} = \{\text{text}, \text{image}, \text{table}, \text{diagram}\}$, and $\mathcal{W}$ a dynamic web corpus reachable only through a search interface $\mathcal{S}$. Given a query (q, v_q) with text q and an optional image v_q, the task is to learn a policy π_θ that, over a horizon T, emits a trajectory $\tau = (a_0, \ldots, a_T)$ scheduling offline retrieval $\mathcal{R}_{\text{off}}$ on an indexed view $\phi(\mathcal{K})$, online retrieval $\mathcal{R}_{\text{on}}$ on $\mathcal{W}$ through $\mathcal{S}$, and the final answer y_τ, jointly optimized by

$$\tau \sim \pi_\theta\big(\cdot \mid q, v_q; \phi(\mathcal{K}), \mathcal{S}(\mathcal{W})\big), \qquad \mathcal{J}(\pi_\theta) = \mathbb{E}_{(q, v_q, y^*) \sim \mathcal{D}}\big[R_{F_1}(y_\tau, y^*)\big], \tag{1}$$

where $\phi : \mathcal{K} \to \mathcal{Z}$ is the offline organizer producing the indexed knowledge store $\mathcal{Z}$, $\mathcal{D}$ the training distribution, y^* the gold answer, and R_{F_1} the token-level F_1 reward, maximized via the RL objective of §3.3. The three modules below address the cost, context, and unification challenges in turn.

3.1 Zero-API-Cost Multimodal Hypergraph Construction

We instantiate ϕ at *zero API cost*: our local parser–extractor replaces the paid LLM extractors [35, 14, 53] while achieving modality-inclusive construction and within-unit higher-order relations.

Modality-Aware Parsing and Local Entity Extraction. Multimodal documents in $\mathcal{K}$ interleave paragraphs, figures, tables and diagrams whose semantic units rarely align with token boundaries, so we realize ϕ as a parser–extractor pipeline that runs entirely on local components and never queries any paid LLM. A rule-based parser ψ first slices every document $d \in \mathcal{K}$ into modality-tagged units, i.e., spaCy-segmented sentences as textual units and attached figures as image units (tables and diagrams enter as flattened text or rendered images); a local extractor π_{ext} then applies a spaCy named-entity recognizer (NER) to every textual unit, while a frozen multimodal encoder $f : \mathcal{X} \to \mathbb{R}^D$ embeds the extracted entities and the image units into a shared space in one streaming pass,

$$\psi(d) = \big\{u_j^{\mu_j}\big\}_{j=1}^{J(d)}, \quad \mathcal{V}^m = \bigcup_{d \in \mathcal{K}} \bigcup_{u \in \psi_{\text{txt}}(d)} \pi_{\text{ext}}(u), \quad \mathbf{E}_{\mathcal{V}} = \big\{f(v) : v \in \mathcal{V}^m\big\} \subset \mathbb{R}^D, \tag{2}$$

where $u_j^{\mu_j}$ is the j-th unit tagged by modality $\mu_j \in \mathbb{M}$, $\psi_{\text{txt}}(d) \subseteq \psi(d)$ its textual units, $J(d)$ the unit count of d, $\mathcal{V}^m$ the entity set extracted from textual units (the superscript m marks multimodal-corpus constructs), $\mathcal{X}$ the joint space of entities, hyperedges, images and queries shared by f, and $\mathbf{E}_{\mathcal{V}}$ the D-dimensional entity embedding bank; image units, collected as $\mathcal{I} = \{u : \mu(u) = \text{image}\}$ with $\mu(u)$ the modality tag of unit u, bypass π_{ext} and are embedded directly by f into a parallel visual bank

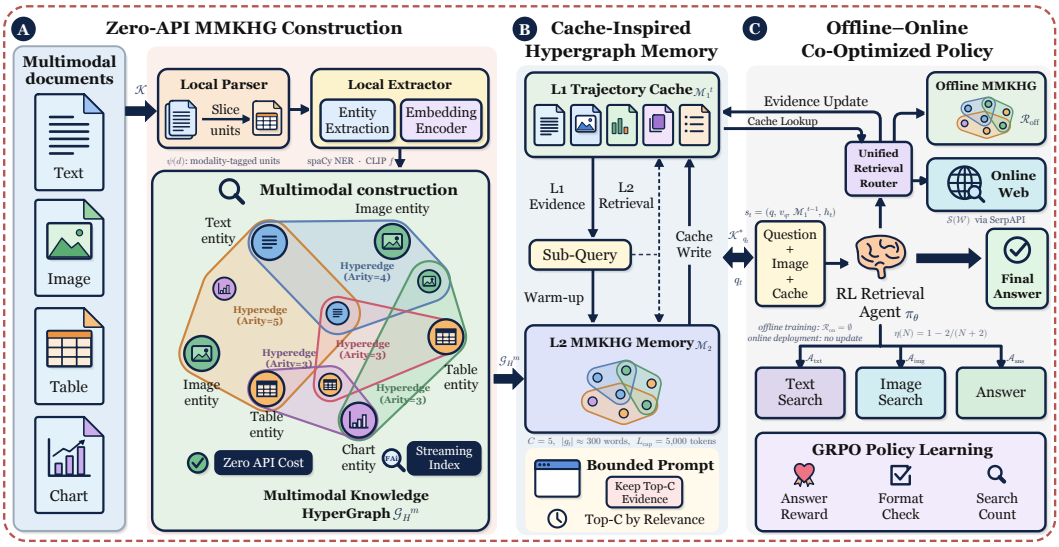


Figure 3: Methodology of MMGraph-Agent: zero-API multimodal hypergraph (MMKHG) wrapped by a two-tier cache memory and queried by an offline-trained, online-deployed retrieval policy.

$\mathbf{E}_{\mathcal{I}} = \{f(u) : u \in \mathcal{I}\}$, sharing one retrieval space with textual items. As neither ψ nor π_{ext} calls a paid LLM, the entity layer of ϕ is built at *zero API cost* and re-runs locally on new documents.

Hyperedge Induction and Streaming Index. Pairwise edges collapse the higher-order context that grounds multimodal reasoning, so all entities co-occurring inside the same parsed sentence form an n-ary hyperedge e_H that carries the full sentence as fragment payload, and these hyperedges are embedded by the same encoder f to populate a streaming index alongside the entity and visual banks,

$$\mathcal{E}_H = \big\{(\pi_{\text{ext}}(u), u) : u \in \psi_{\text{txt}}(d),\ d \in \mathcal{K},\ |\pi_{\text{ext}}(u)| \geq 1\big\},\quad \mathbf{E}_{\mathcal{E}_H} = \{f(e_H)\}_{e_H \in \mathcal{E}_H},\quad \mathcal{G}_H^m = (\mathcal{V}^m, \mathcal{E}_H), \tag{3}$$

where each e_H has arity $n = |\pi_{\text{ext}}(u)| \geq 1$ and retains its host sentence u as payload, $\mathbf{E}_{\mathcal{E}_H} \subset \mathbb{R}^D$ is the hyperedge bank, and $\mathcal{G}_H^m$ together with the visual bank $\mathbf{E}_{\mathcal{I}}$ constitutes the multimodal hypergraph store. Cross-modal relevance then reduces to the inner product $\text{sim}(x, x') = \langle f(x), f(x') \rangle$ on $\mathcal{X}$, used uniformly by every downstream retrieval call: f is a frozen CLIP backbone with $D = 768$ and $\mathcal{G}_H^m$ is backed by flat FAISS inner-product indices, giving fast append-only insertion and single-call *exact* Top-k retrieval (for any requested depth k; §3.2 uses $k = C$) across all the modalities of the corpus.

Proposition 1. *$\mathcal{G}_H^m$ attains entity coverage comparable to an LLM extractor at zero API cost.*

Proof. We provide experimental results in §4.4 and formal analyses in Appendix B.1. □

3.2 Cache-Inspired Multimodal Hypergraph Memory

On top of $\mathcal{G}_H^m$, an L1/L2 cache memory dynamically stores and reuses retrieved evidence [34, 2, 71, 10] so the prompt stays bounded uniformly across the multi-round retrieval horizon.

Two-Tier Memory with Per-Round Warm-Up. Multi-round retrieval inflates the prompt and risks context overflow, so we mirror the CPU L1/L2 cache hierarchy and split $\mathcal{R}_{\text{off}}$ across two tiers: the L2 backend $\mathcal{M}_2$ persistently stores the full hypergraph store of §3.1, while the L1 per-trajectory cache $\mathcal{M}_1^t$ accumulates the evidence retrieved so far. At round t, given the agent state $s_t = (q, v_q, \mathcal{M}_1^{t-1}, h_t)$ with reasoning trace h_t, the policy emits the sub-query $q_t \in \mathcal{X}$ inside its retrieval action (§3.3); the controller consults L2 (and, when enabled, the web channel $\mathcal{R}_{\text{on}}$), admits the fresh evidence into L1, and serves the merged Top-C evidence from the unified L1 index,

$$\mathcal{R}_2(q_t) = \underset{x \in \mathcal{M}_2}{\text{Top-C}} \langle f(q_t), f(x) \rangle,\quad \mathcal{M}_1^t = \mathcal{M}_1^{t-1} \cup \mathcal{R}_2(q_t) \cup \mathcal{R}_{\text{on}}(q_t),\quad \mathcal{K}_{q_t}^* = \underset{x \in \mathcal{M}_1^t}{\text{Top-C}} \langle f(q_t), f(x) \rangle, \tag{4}$$

where C is the per-bank retrieval budget shared by both tiers, $\mathcal{R}_{\text{on}}(q_t)$ the web evidence (empty when disabled), and $\mathcal{K}_{q_t}^*$ the merged evidence returned to the agent. The empty initial cache $\mathcal{M}_1^0 = \varnothing$ warms the working set from $\mathcal{G}_H^m$ rather than policy priors, and later rounds re-serve evidence from L1.

You are a helpful assistant. Answer the given question based on the provided images; you can query the knowledge base as many times as you want. You must first conduct reasoning inside `<reason>...</reason>`. For image search, append `<search><img></search>` behind the reasoning, which runs Google Lens on the original image; for text search, output a concise query inside `<text_search>...</text_search>`, answered by KG/Memory retrieval (and Google Search when enabled). Retrieved evidence returns inside `<information>...</information>`; when you have the final answer, output it inside `<answer>...</answer>`. Here is the image and the question: `<image>` *question*. Assistant:

Table 2: Prompt template used by MMGraph-Agent during rollout. At inference time, `question` is replaced with the user query and `<image>` with the visual input; evidence returned from either tool is inserted inside `<information>...</information>` right after the corresponding tag.

Digest-Bounded Prompt Admission. The admission rule decides what enters the context: raw concatenation grows the prompt linearly in T, so each evidence block is truncated to $L_{\max}$ words, condensed by the policy into a digest g_t, and the digest is admitted, with the rollout halted at L_{cap},

$$\text{prompt}(s_{t+1}) = \text{prompt}(s_t) \oplus g_t, \quad |g_t| \le L_{\text{dig}}, \quad |\text{prompt}(s_t)| \le \min\big(|\text{prompt}(s_0)| + t \cdot L_{\text{dig}},\, L_{\text{cap}}\big), \tag{5}$$

where $\oplus$ appends one conversation turn, g_t summarizes $\mathcal{K}^*_{q_t}$, and $|\cdot|$ measures length in tokens ($L_{\max}$ is counted in words; digests fit the 2,048-token turn limit). We set $C = 5$, $L_{\max} = 1{,}200$ words, a $\approx$300-word digest, $L_{\text{cap}} = 5{,}000$ tokens; the prompt thus stays bounded uniformly in T while every retrieved item remains reachable in L1 and the full $\mathcal{G}^m_H$ in L2.

Proposition 2. *The two-tier memory bounds the prompt uniformly over T with $\mathcal{G}^m_H$ fully reachable.*

Proof. We provide experimental results in §4.5 and theoretical proofs in Appendix B.2. □

3.3 Offline-Trained, Online-Deployed Policy Learning

We fit π_θ with multimodal RL to schedule offline and online retrieval in one trajectory: training is offline-only, and the source-agnostic interface deploys the trained policy online without updates.

Unified Action Space and Hit-Aware Reward. Following Table 2, π_θ samples $a_t \sim \pi_\theta(\cdot \mid s_t)$ from text-query $\mathcal{A}_{\text{txt}}$, image-query $\mathcal{A}_{\text{img}}$, and termination $\mathcal{A}_{\text{ans}}$ channels in one autoregressive emission; both retrieval channels are routed through the cache (§3.2), so the policy emits a single retrieval token whether the hit comes from $\mathcal{G}^m_H$ or web search via $\mathcal{S}$. We shape the scalar return with a smooth call-count regulator $\eta(\cdot)$ that pays for the first retrieval and decays the marginal bonus of extra calls (no explicit per-call cost is imposed), plus a structural penalty on ill-formed or retrieval-free trajectories,

$$\mathcal{A} = \mathcal{A}_{\text{txt}} \cup \mathcal{A}_{\text{img}} \cup \mathcal{A}_{\text{ans}}, \quad \eta(N) = 1 - \tfrac{2}{N+2}, \quad r(\tau) = \mathbb{1}[N_c \ge 1]\, \eta(N_c)\, R_{F_1} - \lambda\big(1 - \mathbb{1}_{\text{fmt}}\, \mathbb{1}[N_c \ge 1]\big), \tag{6}$$

where τ ends in $\mathcal{A}_{\text{ans}}$, N_c counts retrievals, η is concave with $\eta(0) = 0, \eta(\infty) = 1$, R_{F_1} the token-level F_1 against y^* (0 if none), $\mathbb{1}_{\text{fmt}}$ flags schema compliance, and $\lambda = 0.5$; the penalty vanishes only on compliant trajectories with $N_c \ge 1$, pushing π_θ to consult the cache rather than parametric priors.

Cost-Efficient Offline-Only Training. As SerpAPI is costly per rollout, we keep $\mathcal{R}_{\text{on}}$ disabled during training and enable $\mathcal{W}$ at inference; the unified cache hides the source, so offline gradients transfer once $\mathcal{R}_{\text{on}}$ is enabled. We optimize a GRPO clipped surrogate [47] against a frozen π_{ref},

$$\hat{A}_i = \frac{r(\tau_i) - \mu_G}{\sigma_G + \delta}, \quad \mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}_{\tau_i}\left[\sum_{i,t} \tfrac{1}{G|\tau_i|} \min\big(\rho_t \hat{A}_i,\, \text{clip}(\rho_t, 1 \pm \epsilon) \hat{A}_i\big)\right] - \kappa\, \text{KL}(\pi_\theta \,\|\, \pi_{\text{ref}}), \tag{7}$$

with $\tau_i \sim \pi_{\theta_{\text{old}}}$ the G rollouts sampled per query, (μ_G, σ_G) their group reward mean and standard deviation, δ a small stability constant, ρ_t the importance ratio, ϵ the clip range, and κ the KL weight. We run 75 steps with $G = 8, \epsilon = 0.2, \kappa = 10^{-3}$ on 8×A100 and activate $\mathcal{R}_{\text{on}}$ at inference.

Proposition 3. *The source-agnostic interface deploys the offline-trained π_θ online without updates.*

Proof. We provide experimental results in §4.6 and formal analyses in Appendix B.3. □

Table 3: Main results on 23 multimodal QA datasets in Knowledge-Only, WebSearch-Only, and Knowledge+WebSearch. Token-level F_1 (%) with std. over three runs; **bold** marks the best per row, † flags out-of-distribution (OOD) rows, and green is the gain over the second-best baseline.

Datasets	NaiveGen		StdMMRAG		SFT	GRPO	RAG-Any	MMS-R1	MMGraph-Agent-3B
	Qwen2.5-3B	GPT-4.1	Qwen2.5-3B	GPT-4.1	Qwen2.5-3B	Qwen2.5-3B	GPT-4.1	Qwen2.5-3B	(Ours)
DocBench	6.74 ±0.3	14.43 ±0.8	9.54 ±0.4	17.79 ±0.3	9.88 ±0.5	16.50 ±0.7	17.57 ±0.9	22.39 ±1.4	**33.05** ±4.4 (+10.66)
MMLongBench-Doc	4.14 ±0.2	4.78 ±0.3	5.63 ±0.2	7.33 ±1.8	4.95 ±0.3	9.91 ±1.1	9.29 ±1.5	11.08 ±1.1	**20.64** ±2.3 (+9.56)
ViDoSeek†	2.64 ±0.1	6.49 ±0.4	6.32 ±0.5	7.58 ±0.8	3.08 ±0.6	9.37 ±0.6	8.58 ±0.4	15.89 ±0.8	**38.95** ±3.5 (+23.06)
2Wiki	4.11 ±0.3	15.89 ±2.0	12.51 ±1.3	15.94 ±1.5	12.42 ±0.7	14.42 ±0.7	15.30 ±1.1	15.36 ±1.1	**17.68** ±5.3 (+1.74)
HotpotQA	7.29 ±1.3	12.79 ±1.2	11.40 ±1.3	13.85 ±1.2	15.16 ±1.0	15.51 ±0.6	13.32 ±1.0	14.72 ±1.3	**20.70** ±4.2 (+5.19)
Musique	0.45 ±0.2	0.59 ±0.3	0.65 ±0.3	0.66 ±1.1	0.27 ±0.9	0.98 ±1.0	1.22 ±0.2	0.78 ±0.7	**5.67** ±1.3 (+4.45)
NQ	3.10 ±0.2	6.82 ±1.9	9.36 ±1.2	8.83 ±1.0	7.67 ±1.3	7.79 ±0.7	10.99 ±1.1	10.90 ±0.8	**24.42** ±2.9 (+13.43)
PopQA†	7.09 ±1.2	10.90 ±1.3	14.34 ±1.5	7.29 ±1.6	7.23 ±1.2	9.21 ±1.7	14.74 ±1.2	11.58 ±1.1	**16.80** ±2.7 (+2.06)
TriviaQA†	3.11 ±0.4	8.41 ±0.3	10.82 ±1.1	19.08 ±1.8	10.05 ±1.7	11.15 ±1.3	22.30 ±0.9	12.13 ±1.3	**37.37** ±5.5 (+15.07)
FVQA†	2.66 ±0.4	6.73 ±0.4	4.34 ±0.1	7.95 ±0.5	2.53 ±0.6	9.83 ±0.8	7.17 ±0.3	13.23 ±0.8	**23.87** ±3.9 (+10.64)
InfoSeek†	0.94 ±0.1	2.16 ±0.3	3.42 ±0.2	6.14 ±0.2	1.00 ±0.2	4.20 ±0.2	6.01 ±0.2	6.39 ±0.5	**11.66** ±3.5 (+5.27)
MMSearch†	1.61 ±0.1	4.16 ±0.4	3.31 ±0.3	4.69 ±0.4	1.60 ±0.3	3.46 ±0.2	4.82 ±0.1	5.37 ±0.2	**10.51** ±2.9 (+5.14)
SimpleVQA†	5.83 ±0.1	8.58 ±0.3	9.71 ±0.6	14.07 ±0.7	4.34 ±0.5	11.22 ±0.4	16.70 ±0.4	12.77 ±1.2	**28.82** ±4.0 (+12.12)
LiveVQA†	6.59 ±0.9	13.21 ±0.3	7.63 ±0.8	17.64 ±0.2	6.48 ±0.6	10.01 ±0.8	24.32 ±0.7	22.84 ±0.6	**35.89** ±2.7 (+11.57)
MedXpertQA	2.84 ±0.2	4.93 ±0.1	7.32 ±0.2	8.95 ±0.1	5.39 ±0.4	6.81 ±0.3	14.83 ±0.2	19.27 ±0.8	**37.17** ±4.5 (+17.90)
FinMME	5.27 ±0.1	8.14 ±0.5	9.91 ±0.9	10.18 ±0.4	9.53 ±0.8	14.12 ±0.9	10.08 ±0.5	16.63 ±0.9	**61.56** ±4.1 (+44.93)
We-Math2.0	3.95 ±1.9	9.19 ±2.1	6.79 ±0.6	14.17 ±1.8	10.92 ±1.8	15.11 ±1.5	29.32 ±1.8	35.29 ±3.2	**39.74** ±4.6 (+4.45)
R-Bench	0.07 ±0.0	0.61 ±0.1	1.26 ±0.4	2.62 ±0.1	0.76 ±0.1	1.14 ±0.1	5.56 ±0.1	10.66 ±0.1	**17.85** ±2.6 (+7.19)
MicroVQA	12.18 ±0.9	13.30 ±0.2	12.90 ±0.8	13.87 ±0.7	15.56 ±0.5	19.98 ±1.0	23.72 ±0.7	32.20 ±0.6	**38.72** ±5.1 (+6.52)
EEmo-Bench	13.87 ±0.9	23.64 ±0.2	21.97 ±0.2	25.48 ±0.1	24.46 ±0.2	34.08 ±1.0	31.31 ±0.4	39.50 ±1.8	**56.11** ±5.1 (+16.61)
ArtQuest	0.67 ±0.1	1.86 ±0.1	2.38 ±0.3	2.56 ±0.2	2.36 ±0.1	3.99 ±0.3	2.61 ±0.2	5.95 ±0.2	**7.38** ±2.3 (+1.43)
AstroLLaVA	19.25 ±0.6	21.78 ±0.7	22.81 ±0.7	30.46 ±1.0	23.26 ±0.8	27.52 ±0.8	25.08 ±0.6	21.97 ±1.7	**39.57** ±1.5 (+9.11)
GeoGrid-Bench	9.85 ±0.3	15.60 ±0.6	12.49 ±0.4	16.66 ±0.6	20.62 ±0.7	27.00 ±0.2	36.76 ±0.5	31.17 ±0.6	**66.23** ±4.7 (+29.47)

4 Experiments

We answer six RQs: **RQ1:** Does MMGraph-Agent outperform existing baselines across the three regimes? **RQ2:** Does each module of MMGraph-Agent work? **RQ3:** How cheap is the zero-API-cost multimodal hypergraph construction? **RQ4:** Does the cache-inspired memory bound the prompt under long-chain reasoning? **RQ5:** Does the offline-trained policy transfer to online retrieval without parameter updates? **RQ6:** How does MMGraph-Agent solve a hard representative multimodal case?

4.1 Experimental Setup

Datasets and Evaluation Metrics. We evaluate on **23 multimodal QA datasets** grouped by regime (§D.1): **(i) Knowledge-Only (K-only)**: DocBench [72], MMLongBench-Doc [39], ViDoSeek [58], 2Wiki [17], HotpotQA [63], Musique [52], NQ [26], PopQA [40], TriviaQA [25]; **(ii) WebSearch-Only (Web-only)**: FVQA [57], InfoSeek [5], MMSearch [23], SimpleVQA [6], LiveVQA [11]; **(iii) Knowledge+WebSearch (K+Web)**: MedXpertQA [73], FinMME [36], We-Math2.0 [43], R-Bench [13], MicroVQA [4], EEmo-Bench [12], ArtQuest [3], AstroLLaVA [68], GeoGrid-Bench [22]; ViDoSeek, PopQA, TriviaQA and all Web-only datasets are held out as OOD to probe generalization. The only headline metric is token-level F_1 (%) against the gold answer [34, 35], which doubles as the per-rollout reward R_{F_1}; per-regime, per-dataset and efficiency breakdowns are in Appendix D.2.

Baselines and Ablations. All methods share one corpus and prompt (§E): **(a)** generation-only NaiveGeneration (NaiveGen) [69]; **(b)** offline RAG-based StandardMMRAG (StdMMRAG) [65] and RAG-Anything (RAG-Any) [14]; **(c)** training-based SFT [70], GRPO [47], and MMSearch-R1 [60] learning a web-search policy; plus *w/o Hypergraph*, *w/o Memory*, *w/o Policy*, and *w/o Retrieval* each isolating one module of §3. NaiveGen, StdMMRAG and RAG-Any have GPT-4.1-mini [41] variants.

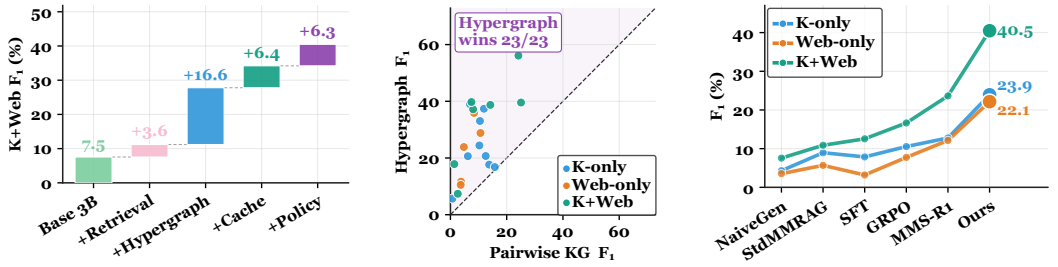
Implementation Details. Backbone: Qwen2.5-VL-3B [44] with a frozen CLIP [45] encoder f (D=768). Memory: per-bank retrieval budget C=5; evidence truncated to 1,200 words, condensed to an ≈300-word digest per round, context cap 5,000 tokens. GRPO: G=8, ϵ=0.2, κ=10^{-3} over 75 steps on 600 examples (233 from six K-only and 367 multimodal with 425 images from eight in-domain datasets; none OOD), on 8×A100 (80GB); F_1 is mean±std over three sampled evaluation runs on the same 640-item pool (temperature 1.0); full hyperparameters in Appendix E.

4.2 Main Results (RQ1)

As shown in Table 3, MMGraph-Agent-3B improves over the second-best baseline per dataset by $+11.6$ F_1 on average across all 23 datasets, with the largest gains on FinMME ($+44.93$), GeoGrid-Bench ($+29.47$), and ViDoSeek ($+23.06$); RAG-based and training-based baselines trail it on every regime, and the gap widens on long-document corpora and on the Knowledge+WebSearch regime where evidence must be combined across offline and online sources within one trajectory. On the eight held-out OOD datasets, it still leads the strongest published agent MMSearch-R1 by $+13.0$ F_1, indicating that the offline hypergraph and the shared retrieval schema generalize to distributions unseen during training, and it ranks first on all eight † (OOD) rows of Table 3.

4.3 Ablation Study (RQ2)

Figure 4 ablates MMGraph-Agent along three complementary axes: per-module contribution (**a**), multimodal hypergraph vs. pairwise KG (**b**), and per-regime gain decomposition across baselines (**c**).



(a) Components (b) Representations (c) Regimes

Figure 4: (a) cumulative component waterfall on K+Web; (b) multimodal hypergraph against a pairwise KG, one point per dataset; (c) per-regime F_1 for the main baseline families and for Ours.

The waterfall in Figure 4a decomposes the K+Web F_1 from Base 7.5 to Full 40.5 into $+3.6$ retrieval, $+16.6$ hypergraph, $+6.4$ cache memory, and $+6.3$ policy learning (gains rounded), so the hypergraph contributes the most (roughly half of the $\approx$33-point gain) and every §3 module helps. Against a pairwise-edge knowledge graph built from the same entities and corpus, the multimodal hypergraph wins on $\mathbf{23/23}$ datasets (Figure 4b, one point per dataset), suggesting that n-ary hyperedges are the more effective primitive for cross-modal reasoning. The regime breakdown in Figure 4c shows MMGraph-Agent reaching $23.9/22.1/40.5$ on K-only / Web-only / K+Web, beating MMSearch-R1 by $+11.2/+10.0/+16.9$ F_1, with the largest margin on K+Web, where offline and online evidence must be combined, consistent with the per-dataset gains on the K+Web rows of Table 3.

4.4 Analysis of Construction Cost (RQ3)

As shown in Figure 5, we compare the construction cost and downstream F_1 of several multimodal knowledge organizers on the same 92K-chunk, 74K-image corpus.

Construction cost. MMGraph-Agent reaches an overall F_1 of 30.0 at zero API cost and a build of only a few hours on commodity CPUs, whereas GraphRAG and RAG-Anything need paid LLM extraction and far longer builds. The reduction is enabled by replacing LLM-based extraction with a deterministic spaCy NER and a frozen CLIP encoder, which eliminates the API cost of LLM-based extraction without degrading downstream accuracy, as the Pareto frontier in Figure 5 shows.

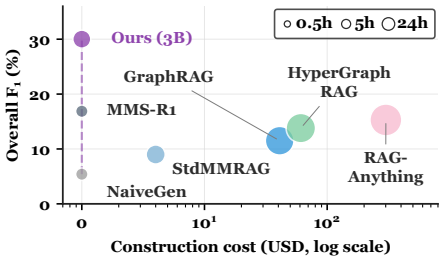
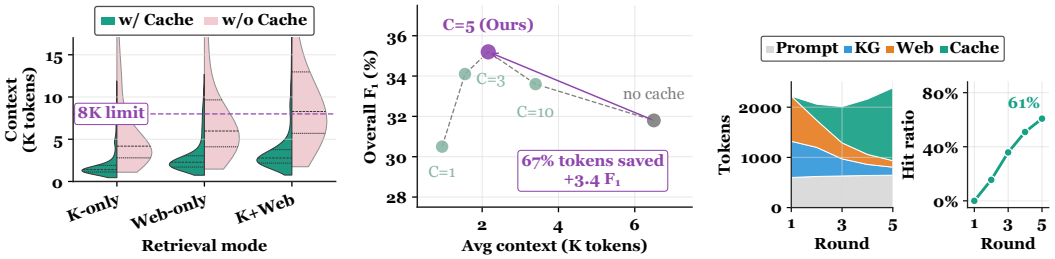


Figure 5: Cost–performance Pareto on the same corpus; bubble area $\propto$ build time.

Hypergraph scale and modality routing. The constructed hypergraph contains roughly 2M nodes and 0.92M hyperedges over 18 of the 23 datasets, while the 5 Web-only datasets are evaluated without an offline hypergraph by design. At inference, the policy issues image search for 35–39% of actions on K+Web data and 0–3% on K-only data, indicating that the policy relies on text queries over the offline hypergraph when queries are text-centric, with visual content indexed as image units (§3.1).

4.5 Cache-Inspired Memory on Long-Chain Episodes (RQ4)

As shown in Figure 6, we test whether the cache-inspired memory keeps the prompt bounded under long-chain reasoning of 3 to 5 rounds, where context overflow is the main failure mode. Long-chain trajectories are collected separately, since the 3B policy converges to one retrieval per query at the default budget (Figure 7b), too few rounds to stress the two-tier L1/L2 memory of §3.2.



(a) Context distribution (b) Cache-capacity Pareto (c) Per-round behavior

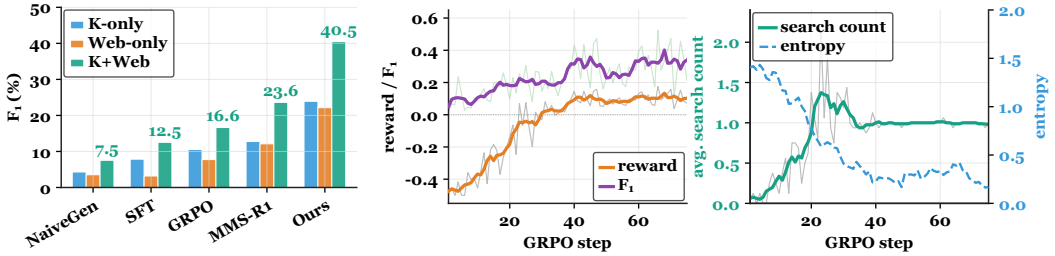
Figure 6: (a) context with/without L1 cache by regime; (b) cache-capacity Pareto across $C \in \{1, 3, 5, 10, \infty\}$ (∞: no cache); (c) per-round tokens and cache-hit ratio over rounds 1–5.

Bounded context and capacity sweep. Figure 6a shows that with the L1 cache per-query context stays mostly around 1K–3K tokens across all three retrieval regimes, whereas without the L1 cache, a non-trivial fraction of trajectories, mostly on K+Web, exceeds the 8K window. Sweeping the per-bank budget $C \in \{1, 3, 5, 10, \infty\}$ in Figure 6b, the default C=5 lies on the Pareto frontier, saving 67% tokens against the no-cache configuration while gaining +3.4 F_1, which motivates C=5 in §3.2.

Per-round token economy. Figure 6c decomposes per-round context into prompt, KG-evidence, web-evidence and cache-hit components. The cache-hit share grows from 0% at round 1 to roughly 61% at round 5, so later rounds reuse evidence already cached by earlier rounds rather than issuing new tool calls, which bounds both the per-round prompt and the cumulative latency per episode.

4.6 Analysis of Offline-to-Online Policy Transfer (RQ5)

As shown in Figure 7, we decompose the trained 3B policy by retrieval regime and inspect its training dynamics across the full 75 GRPO steps on 600 in-domain examples, with web search disabled.



(a) Per-regime unification gain (b) Training dynamics across 75 GRPO steps

Figure 7: (a) per-regime F_1 for four 3B baselines and Ours; (b) 75-step GRPO trajectory of reward, F_1, search count, and entropy (thin lines: raw values; bold lines: 5-step moving averages).

Per-regime gains and action mix. Figure 7a reports per-regime F_1 for four 3B baselines and Ours, and the largest gain over MMSearch-R1 of +16.9 F_1 falls on K+Web. On K+Web the policy combines ⟨`text_search`⟩ and ⟨`search⟨img⟩`⟩ actions, while on K-only it issues text search for 97 to 100% of actions, and on Web-only it issues text search for 63 to 68% of actions.

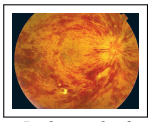
Training dynamics. Figure 7b shows reward improving from −0.45 at step 1 to roughly +0.07 at step 75, and per-batch F_1 rising from about 7 to about 33. The corresponding evaluation F_1 on the evaluation pool reaches 30.0, the overall F_1 plotted in Figure 5. The search count grows from near zero to roughly 1.0 by step 25 and stabilizes through step 75, while policy entropy decreases. The policy first learns when to emit a search tag, and then refines what to query in that call.

4.7 Case Study (RQ6)

Table 4: Case study on MedXpertQA sample 244. NaiveGen, MMSearch-R1 and MMGraph-Agent traces are all taken directly from the same evaluation logs on this sample. F_1 is reported per sample, with the dataset-level average shown in parentheses for reference.

Question (MedXpertQA #244). A 71-year-old female presents with acute monocular vision loss in her right eye on awakening; no pain. PMH: myocardial infarction, hypertension, osteoporosis. BP 145/86 mmHg, pupils symmetric and reactive. The fundus of the right eye is shown; the left fundus is normal. *Choices:* (A) Retinal detachment / (B) Central retinal artery occlusion / (C) Retinal vein occlusion / (D) Closed-angle glaucoma / (E) Papilledema.

Ground truth: (C) Retinal vein occlusion.



Right-eye fundus.

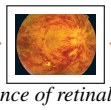
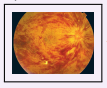
Method	Trace
NaiveGen Qwen2.5-VL-3B no retrieval $F_1 = 0.50$ (avg. 0.028)	<reason> 71-y/o female, acute monocular vision loss; normal pupillary reflexes consistent with vascular occlusion. **The fundus in Figure A shows a characteristic appearance of CRAO: a narrowed retinal artery with 'beading' and a pale optic disc. [hallucinated: absent from the actual fundus]** </reason> <answer> **(B) Central retinal artery occlusion** </answer> × wrong; $F_1 = 0.50$ (shared tokens "retinal", "occlusion").
MMSearch-R1 Qwen2.5-VL-3B RL-tuned web agent (text + Lens, no cache) $F_1 = 0.50$ (avg. 0.193)	<text_search> *CRAO vs RVO elderly painless monocular vision loss* </text_search> <information> *"CRAO classically presents with sudden, painless monocular vision loss in the elderly…"* </information> [anchors symptom-driven prior on CRAO] <search> </search> <information> *"[Lens] The image likely depicts a retina with evidence of retinal vein occlusion (RVO)…"* </information> [contradicts the text retrieval] <reason> Two retrievals conflict; the symptom triad is the canonical prior, weigh text more. </reason> <answer> **(B) Central retinal artery occlusion** </answer> × wrong; without cache-grounded reward to trust visual evidence, the policy anchors on the textbook prior.
MMGraph-Agent (Ours) Qwen2.5-VL-3B hypergraph + L1/L2 cache + graph/web policy **$F_1 = 1.00$** (avg. 0.372)	<reason> *Multiple vascular differentials (CRAO, RVO, detachment) match the symptoms; the fundus is the discriminating signal.* </reason> <search> </search> *cache routing:* L2 warm-up → **multimodal hypergraph** + web search → cache write <information> *"[Search Results] The images in the provided figure likely depict a retina with evidence of retinal vein occlusion (RVO)… Since the 71-year-old female presented with acute monocular vision loss in her right eye, combined with waking sensations without pain, the most probable diagnosis is (C) Retinal vein occlusion."* </information> [digest] <reason> *Visual classification matches (C); CRAO prior never enters the chain.* </reason> <answer> **Retinal vein occlusion** </answer> ✓ exact match; search_count = 1; one image-grounded round was both necessary and sufficient.

Table 4 reports the trace. NaiveGen hallucinates fundus features and outputs (B) CRAO. MMSearch-R1 retrieves both the textbook CRAO passage and the correct Lens RVO classification but anchors on the textual evidence and gives the same wrong answer. MMGraph-Agent queries the image first; the cache routes through the multimodal hypergraph and the augmenting web channel, and the resulting evidence digest yields the correct diagnosis in one round, raising F_1 from 0.50 to 1.00.

5 Conclusion

In this work, we introduce MMGraph-Agent, a unified multimodal RAG framework that addresses three problems of current systems: costly knowledge organization, long-context retrieval, and the structural split between offline and online sources. It pairs a zero-API-cost multimodal hypergraph, a cache-inspired L1/L2 memory, and a GRPO-trained policy emitting retrieval and answer tokens over one shared action space. On 23 multimodal QA datasets MMGraph-Agent outperforms generation-only, RAG-based, and training-based baselines, and ablations show that the three modules play complementary roles. Multilingual encoders and learned memory constants are left for future work.

Acknowledgments

This work is supported by the National Research Foundation, Singapore, under its Frontier Competitive Research Programme (NRF-F-CRP-2024-0005). Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not reflect the views of National Research Foundation, Singapore. The authors declare no competing interests.

References

[1] Akari Asai, Zeqiu Wu, Yizhong Wang, Avi Sil, and Hannaneh Hajishirzi. Self-rag: Learning to retrieve, generate, and critique through self-reflection. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun, editors, *International Conference on Learning Representations*, volume 2024, pages 9112–9141, 2024.

[2] Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. LongBench: A bilingual, multitask benchmark for long context understanding. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3119–3137, Bangkok, Thailand, August 2024. Association for Computational Linguistics.

[3] Tibor Bleidt, Sedigheh Eslami, and Gerard de Melo. Artquest: Countering hidden language biases in artvqa. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 7326–7335, January 2024.

[4] James Burgess, Jeffrey J Nirschl, Laura Bravo-Sánchez, Alejandro Lozano, Sanket Rajan Gupte, Jesus G. Galaz-Montoya, Yuhui Zhang, Yuchang Su, Disha Bhowmik, Zachary Coman, Sarina M Hasan, Alexandra Johannesson, William D. Leineweber, Malvika G Nair, Ridhi Yarlagadda, Connor Zuraski, Wah Chiu, Sarah Cohen, Jan N. Hansen, Manuel D Leonetti, Chad Liu, Emma Lundberg, and Serena Yeung-Levy. Microvqa: A multimodal reasoning benchmark for microscopy-based scientific research. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19552–19564, June 2025.

[5] Yang Chen, Hexiang Hu, Yi Luan, Haitian Sun, Soravit Changpinyo, Alan Ritter, and Ming-Wei Chang. Can pre-trained vision and language models answer visual information-seeking questions? In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 14948–14968, Singapore, December 2023. Association for Computational Linguistics.

[6] Xianfu Cheng, Wei Zhang, Shiwei Zhang, Jian Yang, Xiangyuan Guan, Xianjie Wu, Xiang Li, Ge Zhang, Jiaheng Liu, Yuying Mai, Yutao Zeng, Zhoufutu Wen, Ke Jin, Baorui Wang, Weixiao Zhou, Yunhong Lu, Hangyuan Ji, Tongliang Li, Wenhao Huang, and Zhoujun Li. Simplevqa: Multimodal factuality evaluation for multimodal large language models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4637–4646, October 2025.

[7] Sijun Dai, Qiang Huang, Xiaoxing You, and Jun Yu. MG^2-RAG: Multi-granularity graph for multimodal retrieval-augmented generation. In Paolo Favaro, Zuzana Kukelova, Atsuto Maki, Anna Rohrbach, Konrad Schindler, and Federico Tombari, editors, *Computer Vision – ECCV 2026*, pages 554–577, Cham, 2026. Springer Nature Switzerland.

[8] Chenlong Deng, Mengjie Deng, Junjie Wu, Dun Zeng, Teng Wang, Qingsong Xie, Jiadeng Huang, Shengjie Ma, Changwang Zhang, Zhaoxiang Wang, Jun Wang, Yutao Zhu, and Zhicheng Dou. DeepImageSearch: Benchmarking multimodal agents for context-aware image retrieval in visual histories. In Tong Zhang, Miroslav Dudik, Martin Jaggi, Alekh Agarwal, Sharon Li, Dale Schuurmans, Jerry Zhu, Felix Berkenkamp, Hanze Dong, and Alberto Bietti, editors, *Proceedings of the 43rd International Conference on Machine Learning*, volume 306 of *Proceedings of Machine Learning Research*, pages 23935–23952. PMLR, 06–11 Jul 2026.

[9] Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, Dasha Metropolitansky, Robert Osazuwa Ness, and Jonathan Larson. From local to global: A graph rag approach to query-focused summarization, 2025.

[10] Yichao Feng, Ran Zhang, Haoran Luo, Zhenghong Lin, Carl Yang, and Anh Tuan Luu. Diachronic hypergraphs for orchestrated multi-agent multimodal memory curation, 2026.

[11] Mingyang Fu, Yuyang Peng, Dongping Chen, Zetong Zhou, Benlin Liu, Yao Wan, Zhou Zhao, Philip S Yu, and Ranjay Krishna. Seeking and updating with live visual knowledge. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen, editors, *Advances in Neural Information Processing Systems*, volume 38, Main Conference. Curran Associates, Inc., 2025.

[12] Lancheng Gao, Ziheng Jia, Yunhao Zeng, Wei Sun, Yiming Zhang, Wei Zhou, Guangtao Zhai, and Xiongkuo Min. Eemo-bench: A benchmark for multi-modal large language models on image evoked emotion assessment. In *Proceedings of the 33rd ACM International Conference on Multimedia*, MM '25, page 7064–7073. ACM, October 2025.

[13] Meng-Hao Guo, Jiajun Xu, Yi Zhang, Jiaxi Song, Haoyang Peng, Yi-Xuan Deng, Xinzhi Dong, Kiyohiro Nakayama, Zhengyang Geng, Chen Wang, Bolin Ni, Guo-Wei Yang, Yongming Rao, Houwen Peng, Han Hu, Gordon Wetzstein, and Shi-Min Hu. RBench: Graduate-level multidisciplinary benchmarks for LLM & MLLM complex reasoning evaluation. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu, editors, *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 21184–21201. PMLR, 13–19 Jul 2025.

[14] Zirui Guo, Xubin Ren, Lingrui Xu, Jiahao Zhang, and Chao Huang. Rag-anything: All-in-one rag framework, 2025.

[15] Zirui Guo, Lianghao Xia, Yanhua Yu, Tu Ao, and Chao Huang. LightRAG: Simple and fast retrieval-augmented generation. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng, editors, *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 10746–10761, Suzhou, China, November 2025. Association for Computational Linguistics.

[16] Bernal Jiménez Gutiérrez, Yiheng Shu, Weijian Qi, Sizhe Zhou, and Yu Su. From RAG to memory: Non-parametric continual learning for large language models. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu, editors, *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 21497–21515. PMLR, 13–19 Jul 2025.

[17] Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps. In Donia Scott, Nuria Bel, and Chengqing Zong, editors, *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6609–6625, Barcelona, Spain (Online), December 2020. International Committee on Computational Linguistics.

[18] Jack Hong, Chenxiao Zhao, ChengLIn Zhu, Weiheng Lu, Guohai Xu, and XingYu. Deepeyesv2: Toward agentic multimodal model. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust, editors, *International Conference on Learning Representations*, volume 2026, pages 114851–114872, 2026.

[19] Chi-Hsiang Hsiao, Yi-Cheng Wang, Tzung-Sheng Lin, Yi-Ren Yeh, and Chu-song Chen. MegaRAG: Multimodal knowledge graph-based retrieval augmented generation. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens, editors, *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 48031–48059, San Diego, California, United States, July 2026. Association for Computational Linguistics.

[20] Hao Hu, Yifan Feng, Ruoxue Li, Rundong Xue, Xingliang Hou, Zhiqiang Tian, Yue Gao, and Shaoyi Du. Cog-rag: Cognitive-inspired dual-hypergraph with theme alignment retrieval-augmented generation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(37):31032–31040, March 2026.

[21] Yusong Hu, Runmin Ma, Yue Fan, Jinxin Shi, Zongsheng Cao, Yuhao Zhou, Jiakang Yuan, Shuaiyu Zhang, Shiyang Feng, Xiangchao Yan, Shufei Zhang, Wenlong Zhang, Lei Bai, and Bo Zhang. FlowSearch: Advancing deep research with dynamic structured knowledge flow. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens, editors, *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 21212–21245, San Diego, California, United States, July 2026. Association for Computational Linguistics.

[22] Bowen Jiang, Yangxinyu Xie, Xiaomeng Wang, Jiashu He, Joshua Bergerson, John K Hutchison, Jordan Branham, Camillo J Taylor, and Tanwi Mallick. Geogrid-bench: Can foundation models understand multimodal gridded geo-spatial data?, 2025.

[23] Dongzhi Jiang, Renrui Zhang, Ziyu Guo, Yanmin Wu, jiayi lei, Pengshuo Qiu, Pan Lu, Zehui Chen, Guanglu Song, Gao Peng, Yu Liu, Chunyuan Li, and Hongsheng Li. Mmsearch: Unveiling the potential of large models as multi-modal search engines. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu, editors, *International Conference on Learning Representations*, volume 2025, pages 1662–1699, 2025.

[24] Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. In *Conference On Language Models*, volume 2, pages 1073–1103, 2025.

[25] Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In Regina Barzilay and Min-Yen Kan, editors, *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada, July 2017. Association for Computational Linguistics.

[26] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466, 2019.

[27] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc., 2020.

[28] Kuan Li, Zhongwang Zhang, Huifeng Yin, Liwen Zhang, Litu Ou, Jialong Wu, Wenbiao Yin, Baixuan Li, Zhengwei Tao, Xinyu Wang, Weizhou Shen, Junkai Zhang, Dingchu Zhang, Xixi Wu, Yong Jiang, Ming Yan, Pengjun Xie, Fei Huang, and Jingren Zhou. Websailor: Navigating super-human reasoning for web agent, 2025.

[29] Lei Liang, Zhongpu Bo, Zhengke Gui, Zhongshu Zhu, Ling Zhong, Peilong Zhao, Mengshu Sun, Zhiqiang Zhang, Jun Zhou, Wenguang Chen, Wen Zhang, and Huajun Chen. Kag: Boosting llms in professional domains via knowledge augmented generation. In *Companion Proceedings of the ACM on Web Conference 2025*, WWW '25, page 334–343. ACM, May 2025.

[30] Wen-Sheng Lien, Yu-Kai Chan, Hao-Lung Hsiao, Bo-Kai Ruan, Meng-Fen Chiang, Chien-An Chen, Yi-Ren Yeh, and Hong-Han Shuai. Hyperrag: Reasoning n-ary facts over hypergraphs for retrieval augmented generation. In *Proceedings of the ACM Web Conference 2026*, page 2465–2476. ACM, April 2026.

[31] Jiashi Lin, Changhong Jiang, Xiangru Lin, Ruifei Zhang, Xinyi Zhu, Jiyao Liu, Cheng Tang, Ye Du, Shujian Gao, Junzhi Ning, Lihao Liu, Ziyan Huang, Tianbin Li, Jin Ye, and Junjun He. Evograph-r1: Self-evolving multimodal knowledge hypergraphs for agentic retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 756–765, June 2026.

[32] Hao Liu, Zhengren Wang, Xi Chen, Zhiyu Li, Feiyu Xiong, Qinhan Yu, and Wentao Zhang. HopRAG: Multi-hop reasoning for logic-aware retrieval-augmented generation. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Findings of the Association for Computational Linguistics: ACL 2025*, pages 1897–1913, Vienna, Austria, July 2025. Association for Computational Linguistics.

[33] Rui Lu, Zhenyu Hou, Zihan Wang, Hanchen Zhang, Xiao Liu, Yujiang Li, Shi Feng, Jie Tang, and Yuxiao Dong. Deepdive: Advancing deep search agents with knowledge graphs and multi-turn rl, 2025.

[34] Haoran Luo, Haihong E, Guanting Chen, Qika Lin, Yikai Guo, Fangzhi Xu, Zemin Kuang, Meina Song, Xiaobao Wu, Yifan Zhu, and Anh Tuan Luu. Graph-r1: Towards agentic GraphRAG framework via end-to-end reinforcement learning. In Tong Zhang, Miroslav Dudik, Martin Jaggi, Alekh Agarwal, Sharon Li, Dale Schuurmans, Jerry Zhu, Felix Berkenkamp, Hanze Dong, and Alberto Bietti, editors, *Proceedings of the 43rd International Conference on Machine Learning*, volume 306 of *Proceedings of Machine Learning Research*, pages 82220–82240. PMLR, 06–11 Jul 2026.

[35] Haoran Luo, Haihong E, Guanting Chen, Yandan Zheng, Xiaobao Wu, Yikai Guo, Qika Lin, Yu Feng, Zemin Kuang, Meina Song, Yifan Zhu, and Anh Tuan Luu. Hypergraphrag: Retrieval-augmented generation via hypergraph-structured knowledge representation. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen, editors, *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pages 152206–152234. Curran Associates, Inc., 2025.

[36] Junyu Luo, Zhizhuo Kou, Liming Yang, Xiao Luo, Jinsheng Huang, Zhiping Xiao, Jingshu Peng, Chengzhong Liu, Jiaming Ji, Xuanzhe Liu, Sirui Han, Ming Zhang, and Yike Guo. FinMME: Benchmark dataset for financial multi-modal reasoning evaluation. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 29465–29489, Vienna, Austria, July 2025. Association for Computational Linguistics.

[37] Linhao Luo, Zicheng Zhao, Reza Haffari, Dinh Phung, Chen Gong, and Shirui Pan. Gfm-rag: Graph foundation model for retrieval augmented generation. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen, editors, *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pages 36371–36405. Curran Associates, Inc., 2025.

[38] Linhao Luo, Zicheng Zhao, Junnan Liu, Zhangchi Qiu, Junnan Dong, Serge Panev, Chen Gong, Thuy-Trang Vu, Gholamreza Haffari, Dinh Phung, Alan Liew, and Shirui Pan. G-reasoner: Foundation models for unified reasoning over graph-structured knowledge. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust, editors, *International Conference on Learning Representations*, volume 2026, pages 63769–63793, 2026.

[39] Yubo Ma, Yuhang Zang, Liangyu Chen, Meiqi Chen, Yizhu Jiao, Xinze Li, Xinyuan Lu, Ziyu Liu, Yan Ma, Xiaoyi Dong, Pan Zhang, Liangming Pan, Yu-Gang Jiang, Jiaqi Wang, Yixin Cao, and Aixin Sun. Mmlongbench-doc: Benchmarking long-context document understanding with visualizations. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 95963–96010. Curran Associates, Inc., 2024.

[40] Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*

(Volume 1: Long Papers), pages 9802–9822, Toronto, Canada, July 2023. Association for Computational Linguistics.

[41] OpenAI. Introducing GPT-4.1 in the API. `https://openai.com/index/gpt-4-1/`, April 2025.

[42] Hyeongcheol Park, Jiyoung Seo, Jaewon Mun, Hogun Park, Wonmin Byeon, Sung June Kim, Hyeonsoo Im, JeungSub Lee, and Sangpil Kim. M^3kg-rag: Multi-hop multimodal knowledge graph-enhanced retrieval-augmented generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14968–14978, June 2026.

[43] Runqi Qiao, Qiuna qiuna, Peiqing Yang, Yanzi Wang, Xiaowan Wang, Enhui Wan, Guanting Dong, Shiqiang Lang, Sitong Zhou, Yida Xu, Yuchen Zeng, Jie Wang, Chong Sun, Chen Li, and Honggang Zhang. We-math 2.0: A versatile mathbook system for incentivizing visual mathematical reasoning. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust, editors, *International Conference on Learning Representations*, volume 2026, pages 150171–150218, 2026.

[44] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-VL technical report, 2025.

[45] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR, 18–24 Jul 2021.

[46] Parth Sarthi, Salman Abdullah, Aditi Tuli, Shubh Khanna, Anna Goldie, and Christopher Manning. Raptor: Recursive abstractive processing for tree-organized retrieval. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun, editors, *International Conference on Learning Representations*, volume 2024, pages 32628–32649, 2024.

[47] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024.

[48] Huatong Song, Jinhao Jiang, Yingqian Min, Jie Chen, Zhipeng Chen, Wayne Xin Zhao, Lei Fang, and Ji-Rong Wen. R1-searcher: Incentivizing the search capability in llms via reinforcement learning, 2025.

[49] Hao Sun, Zile Qiao, Jiayan Guo, Xuanbo Fan, Yingyan Hou, Yong Jiang, Pengjun Xie, Yan Zhang, Fei Huang, and Jingren Zhou. Zerosearch: Incentivize the search capability of llms without searching, 2026.

[50] Yubo Sun, Chunyi Peng, Yukun Yan, Shi Yu, Zhenghao Liu, Sen Mei, Chi Chen, and Maosong Sun. Visrag2.0: Mitigating visual hallucinations via evidence-guided multi-image reasoning in visual retrieval-augmented generation, 2026.

[51] Tongyi DeepResearch Team, Baixuan Li, Bo Zhang, Dingchu Zhang, Fei Huang, Guangyu Li, Guoxin Chen, Huifeng Yin, Jialong Wu, Jingren Zhou, Kuan Li, Liangcai Su, Litu Ou, Liwen Zhang, Pengjun Xie, Rui Ye, Wenbiao Yin, Xinmiao Yu, Xinyu Wang, Xixi Wu, Xuanzhong Chen, Yida Zhao, Zhen Zhang, Zhengwei Tao, Zhongwang Zhang, Zile Qiao, Chenxi Wang, Donglei Yu, Gang Fu, Haiyang Shen, Jiayin Yang, Jun Lin, Junkai Zhang, Kui Zeng, Li Yang, Hailong Yin, Maojia Song, Ming Yan, Minpeng Liao, Peng Xia, Qian Xiao, Rui Min, Ruixue Ding, Runnan Fang, Shaowei Chen, Shen Huang, Shihang Wang, Shihao Cai, Weizhou Shen, Xiaobin Wang, Xin Guan, Xinyu Geng, Yingcheng Shi, Yuning Wu, Zhuo Chen, Zijian Li, and Yong Jiang. Tongyi deepresearch technical report, 2026.

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

[53] Xueyao Wan and Hang Yu. Mmgraphrag: Bridging vision and language with interpretable multimodal knowledge graphs, 2026.

[54] Jingjin Wang and Jiawei Han. PropRAG: Guiding retrieval with beam search over proposition paths. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng, editors, *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 6212–6227, Suzhou, China, November 2025. Association for Computational Linguistics.

[55] Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, Wayne Xin Zhao, Zhewei Wei, and Jirong Wen. A survey on large language model based autonomous agents. *Frontiers of Computer Science*, 18(6):186345, Mar 2024.

[56] Nengbo Wang, Xiaotian Han, Jagdip Singh, Jing Ma, and Vipin Chaudhary. CausalRAG: Integrating causal graphs into retrieval-augmented generation. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Findings of the Association for Computational Linguistics: ACL 2025*, pages 22680–22693, Vienna, Austria, July 2025. Association for Computational Linguistics.

[57] Peng Wang, Qi Wu, Chunhua Shen, Anthony Dick, and Anton van den Hengel. Fvqa: Fact-based visual question answering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(10):2413–2427, October 2018.

[58] Qiuchen Wang, Ruixue Ding, Zehui Chen, Weiqi Wu, Shihang Wang, Pengjun Xie, and Feng Zhao. ViDoRAG: Visual document retrieval-augmented generation via dynamic iterative reasoning agents. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng, editors, *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 9113–9134, Suzhou, China, November 2025. Association for Computational Linguistics.

[59] Jialong Wu, Baixuan Li, Runnan Fang, Wenbiao Yin, Liwen Zhang, Zhenglin Wang, Zhengwei Tao, Ding-Chu Zhang, Zekun Xi, Robert Tang, Yong Jiang, Pengjun Xie, Fei Huang, and Jingren Zhou. Webdancer: Towards autonomous information seeking agency. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen, editors, *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pages 120957–120985. Curran Associates, Inc., 2025.

[60] Jinming Wu, Zihao Deng, Wei Li, Yiding Liu, Bo You, Bo Li, Zejun MA, and Ziwei Liu. MMSearch-R1: Incentivizing LMMs to search. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens, editors, *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2456–2487, San Diego, California, United States, July 2026. Association for Computational Linguistics.

[61] Junde Wu, Jiayuan Zhu, Yunli Qi, Jingkun Chen, Min Xu, Filippo Menolascina, Yueming Jin, and Vicente Grau. Medical graph RAG: Evidence-based medical large language model via graph retrieval-augmented generation. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 28443–28467, Vienna, Austria, July 2025. Association for Computational Linguistics.

[62] Cehao Yang, Xiaojun Wu, Xueyuan Lin, Chengjin Xu, Xuhui Jiang, Yuanliang Sun, Jia Li, Hui Xiong, and Jian Guo. Graphsearch: An agentic deep searching workflow for graph retrieval-augmented generation, 2025.

[63] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. HotpotQA: A dataset for diverse, explainable multi-hop question

answering. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium, October-November 2018. Association for Computational Linguistics.

[64] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations*, 2023.

[65] Shi Yu, Chaoyue Tang, Bokai Xu, Junbo Cui, Junhao Ran, Yukun Yan, Zhenghao Liu, Shuo Wang, Xu Han, Zhiyuan Liu, and Maosong Sun. Visrag: Vision-based retrieval-augmented generation on multi-modality documents. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu, editors, *International Conference on Learning Representations*, volume 2025, pages 21074–21098, 2025.

[66] Qianhao Yuan, Jie Lou, Zichao Li, Jiawei Chen, Yaojie Lu, Hongyu Lin, Le Sun, Debing Zhang, and Xianpei Han. MemSearcher: Iterative memory integration for search agent via end-to-end reinforcement learning. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens, editors, *Findings of the Association for Computational Linguistics: ACL 2026*, pages 14965–14977, San Diego, California, United States, July 2026. Association for Computational Linguistics.

[67] Xu Yuan, Liangbo Ning, Qingqing Ye, Wenqi Fan, and Qing Li. mkg-rag: Leveraging multimodal knowledge graphs in retrieval-augmented generation for knowledge-intensive vqa. In *Proceedings of the 49th International ACM SIGIR Conference on Research and Development in Information Retrieval*, page 2274–2285. ACM, 2026.

[68] Sharaf Zaman, Michael J. Smith, Pranav Khetarpal, Rishabh Chakrabarty, Michele Ginolfi, Marc Huertas-Company, Maja Jabłońska, Sandor Kruk, Matthieu Le Lain, Sergio José Rodríguez Méndez, and Dimitrios Tanoglidis. Astrollava: towards the unification of astronomical data and natural language, 2025.

[69] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Zican Dong, Yupeng Hou, Beichen Zhang, Yingqian Min, Junjie Zhang, Peiyu Liu, Xiaolei Wang, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Yiwen Hu, Jian-Yun Nie, and Ji-Rong Wen. A survey of large language models. *Frontiers of Computer Science*, 20(12):2012627, May 2026.

[70] Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, and Zheyan Luo. LlamaFactory: Unified efficient fine-tuning of 100+ language models. In Yixin Cao, Yang Feng, and Deyi Xiong, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pages 400–410, Bangkok, Thailand, August 2024. Association for Computational Linguistics.

[71] Chulun Zhou, Chunkang Zhang, Guoxin Yu, Fandong Meng, Jie Zhou, Wai Lam, and Mo Yu. HGMem: Hypergraph-based working memory to improve multi-step RAG for long-context complex relational modeling. In Tong Zhang, Miroslav Dudik, Martin Jaggi, Alekh Agarwal, Sharon Li, Dale Schuurmans, Jerry Zhu, Felix Berkenkamp, Hanze Dong, and Alberto Bietti, editors, *Proceedings of the 43rd International Conference on Machine Learning*, volume 306 of *Proceedings of Machine Learning Research*, pages 166312–166333. PMLR, 06–11 Jul 2026.

[72] Anni Zou, Wenhao Yu, Hongming Zhang, Kaixin Ma, Deng Cai, Zhuosheng Zhang, Hai Zhao, and Dong Yu. Docbench: A benchmark for evaluating llm-based document reading systems, 2024.

[73] Yuxin Zuo, Shang Qu, Yifei Li, Zhang-Ren Chen, Xuekai Zhu, Ermo Hua, Kaiyan Zhang, Ning Ding, and Bowen Zhou. MedXpertQA: Benchmarking expert-level medical reasoning and understanding. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu, editors, *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 80961–80990. PMLR, 13–19 Jul 2025.

Appendix

A MMGraph-Agent Operators and Prompts

This appendix lists every deterministic operator and LLM prompt of MMGraph-Agent, reproduced verbatim from the released code ({x} marks a runtime placeholder) in the prompt-card style of HyperGraphRAG [35] and grouped by the three modules of §3. Modules 1 and 2 invoke no external LLM, so we show their core operators; only Module 3 prompts the unified GRPO-trained policy π_θ.

A.1 Zero-API-Cost Multimodal Hypergraph Construction (Module 1)

Operator O1: Sentence-Level Multimodal Extractor (π_{ext}, §3.1). spaCy splits each parsed chunk into sentences and turns every sentence into an n-ary hyperedge over its named entities; no LLM is invoked, so extraction incurs zero API cost and runs entirely on local CPU hardware.

```
# operate.py: zero-API-cost extractor running under ProcessPoolExecutor
def _mmextractor_worker(text: str):
    global _NLP
    doc = _NLP(text)                        # spaCy en_core_web_lg, frozen
    results = []
    for sent in doc.sents:
        ents = [ent.text for ent in sent.ents]
        if ents:
            results.append({
                "sentence": sent.text.strip(),
                "entities": list(dict.fromkeys(ents))
            })
    return results
```

Operator O2: CLIP Embedding and FAISS Flat Index (f, $\mathcal{G}_H^m$, §3.1). Entities, hyperedges, images and queries share frozen CLIP (D=768) and flat FAISS inner-product indices: simple insertion, single-call exact Top-k search over each bank, free of the recall loss of approximate indices.

```
# mmgraph.py: shared CLIP embedding bank for entity / hyperedge / image
embedding_model = SentenceTransformer('clip-ViT-L-14')        # d = 768
embeddings = model.encode(corpus_entity_content[corpus_source], batch_size=batch_size)
index = faiss.index_factory(dim, 'Flat', faiss.METRIC_INNER_PRODUCT)
index.add(corpus_numpy)                                       # append-only insertion
# search_utils.py: single exact Top-k call
_, ids = index.search(query_embedding, top_k)
```

A.2 Cache-Inspired Multimodal Hypergraph Memory (Module 2)

Operator O3: Two-Tier Warm-Up Retrieval (Eq. (4) of §3.2). Every round consults L2 (the hypergraph store, plus the web when enabled), admits new evidence into the per-trajectory L1 memory and serves the merged Top-C from L1, so evidence retrieved in earlier rounds stays reusable.

```
# mem_server.py: L2 warm-up -> L1 serve on every round
kg_results = http_requests.post(f"http://localhost:{KG_PORT}/search",
    json={"queries": query_str}).json()       # L2: persistent hypergraph store
rag_memory.insert(unique_contexts)            # admit evidence into L1 memory
entity_match    = faiss_search(index_entity[traj_hash],
                               corpus_entity[traj_hash],    embeddings, 5)
hyperedge_match = faiss_search(index_hyperedge[traj_hash],
                               corpus_hyperedge[traj_hash], embeddings, 5)
result = rag_memory.query(query,              # merged Top-k served from L1
    param=QueryParam(only_need_context=True, top_k=5),
    entity_match=entity_match, hyperedge_match=hyperedge_match, ...)
```

Operator O4: Digest-Bounded Prompt Admission (Eq. (5) of §3.2). Evidence is truncated to $L_{\max}$ words and condensed into a digest of about 300 words; the digest is admitted and rollout halts once the prompt reaches L_{cap} tokens, so the context stays bounded regardless of the number of rounds.

```
# rollout.py: bounded per-round admission into the prompt
retrieved_data = [item['<knowledge>']
                  for item in json.loads(res[index])['results'][:1]]
if len(corpus_text.split()) > 1200:           # truncate evidence payload
    corpus_text = ' '.join(corpus_text.split()[:1200]) + '...'
# ... the policy condenses <information> into an ~300-word digest ...
if len(vllm_inputs[i_gen]['prompt_token_ids']) >= 5000:
    idx_to_remove.append(i_gen)               # hard context cap: stop generating
```

A.3 Offline-Trained, Online-Deployed Policy Learning (Module 3)

Prompt P0: Digest Request (O4, `rollout.py`). Sent after every retrieval; the reply is the digest g_t.

```
Searched results: <information>{retrieved evidence}</information>
Original user's question: {question}
Please provide an approximately 300-word summary of the current information related
to current query in English.
```

Prompt P1: Initial Reasoning Prompt (§3.3, Table 2). Issued at round $t{=}0$ (`dataset.py`); the policy reasons inside `<reason>` and then chooses image search, text search or the final answer, with the same prompt for every retrieval regime and identically at both training and inference time.

```
You are a helpful assistant. Answer the given question based on the provided images. You can
query from knowledge base provided to you to answer the question. You can query knowledge as
many times as you want. You must first conduct reasoning inside <reason>...</reason>. You
have two kinds of query tools, image search and text search. If you need to use image
search, you can append the string <search><img></search> behind reasoning. Format:
<reason>.....generate your reasoning here...</reason> <search><img></search>. This will
trigger a Google Lens search using the original image to retrieve relevant information that
can help you confirm the visual content. If you need to use text search, you can set a
concise and specific query, and output it in the format <text_search>your query
here</text_search> behind reasoning. Format: <reason>...generate your reasoning
here...</reason> <text_search>...generate your query here...</text_search>. The text search
tool will then use Google Search to return relevant information based on your query. When
you have the final answer, you can output the answer inside <answer>...</answer> behind
reasoning. Format: <reason>...generate your reasoning here...</reason> <answer>...generate
your answer here...</answer>. Please generate in English.
Here is the image and the question:
<image> {question}
```

Prompt P2: Continuation with Image Input (§3.3, `rollout.py`). Appended to every retrieval result when the query carries an image: the evidence (or its digest) is wrapped in `<information>`, the question is restated, and all three actions remain available to the policy in the next round.

```
Searched results: <information>{evidence or digest}</information>
Original user's question: {question}
Based on information above, you need to determine whether one more Google image or text
search is needed to retrieve additional information, or whether to answer the
question directly. You must first conduct reasoning inside <reason>...</reason>. You have
three choices: image search, text search, or answer. If you need to use image search, you
can append the string <search><img></search> behind reasoning. Format: <reason>...generate
your reasoning here...</reason> <search><img></search>. If you need to use text search, you
can set a concise and specific query, and output it in the format <text_search>your query
here</text_search> behind reasoning. Format: <reason>...generate your reasoning
here...</reason> <text_search>...generate your query here...</text_search>. If you have the
final answer, you can output the answer inside <answer>...</answer> behind reasoning.
Format: <reason>...generate your reasoning here...</reason> <answer>...generate your answer
```

```
here...</answer>. Please generate in English.
```

Prompt P3: Continuation for Text-Only Input (§3.3, `rollout.py`). Used instead of P2 when the query has no image, so the image-search branch is dropped, leaving text search and the answer.

```
Searched results: <information>{evidence or digest}</information>
Original user's question: {question}
Based on information above, you need to determine whether one more Google text search is
needed to retrieve additional information to answer the question or to answer the question
directly. You must first conduct reasoning inside <reason>...</reason>. You have two
choices: text search, or answer. If you need to use text search, you can set a concise and
specific query, and output it in the format <text_search>your query here</text_search>
behind reasoning. Format: <reason>...generate your reasoning here...</reason>
<text_search>...generate your query here...</text_search>. If you have the final answer, you
can output the answer inside <answer>...</answer> behind reasoning. Format:
<reason>...generate your reasoning here...</reason> <answer>...generate your answer
here...</answer>. Please generate in English.
```

A.4 Format-Reward Grammar (Module 3, training-time)

Operator O5: Schema Compliance Regex ($\mathbb{1}_{\text{fmt}}$, Eq. (6) of §3.3). Each segment must match one anchored pattern with exactly one open/close pair per active tag; only fully compliant trajectories get $\mathbb{1}_{\text{fmt}}=1$ and avoid the penalty $-\lambda$, which steers the policy toward well-formed, parseable trajectories.

```
# score.py: anchored regex grammar checked per response segment
direct_answer_format     = r'^<reason>.*</reason>.*<answer>.*</answer>$'
call_image_search_format = r'^<reason>.*</reason>.*<search><img></search>$'
call_text_search_format  = r'^<reason>.*</reason>.*<text_search>.*</text_search>$'

def is_valid_text_search(response, call_text_search_format) -> bool:
    if not re.match(call_text_search_format, response, re.DOTALL):
        return False
    if response.count('<reason>') != 1 or response.count('</reason>') != 1:
        return False
    if response.count('<text_search>') != 1 or response.count('</text_search>') != 1:
        return False
    ...                       # likewise rejects <answer> and <search><img></search> tags
    return True
```

Operator O6: Hit-Aware Trajectory Reward ($r(\tau)$, Eq. (6) of §3.3). Implements $r(\tau) = \mathbb{1}[N_c{\geq}1]\,\eta(N_c)R_{F_1} - \lambda(1 - \mathbb{1}_{\text{fmt}}\mathbb{1}[N_c{\geq}1])$ with $\eta(N){=}1{-}\frac{2}{N+2}$ and $\lambda{=}0.5$.

```
# score.py: per-trajectory reward, lambda = 0.5 (deepcopy omitted)
def compute_score(prediction, ground_truth, raw_response_str):
    answer = extract_solution(prediction=prediction[-1])
    score = 0
    if answer is not None:
        score = cal_f1([ground_truth], [answer])
    format_score, search_count = format_reward(prediction)
    if format_score == 1 and search_count >= 1:
        overall_score = score * (1 - 2 / (search_count + 2))
    elif format_score == 0 and search_count >= 1:
        overall_score = -0.5 + score * (1 - 2 / (search_count + 2))
    else:
        overall_score = -0.5
```

Reproducibility. Operators O1–O6, prompts P1–P3, the rollout schema of Table 2 and the GRPO objective of Eq. (7) fully specify the runtime. All generations sample with temperature 1.0 and top-$p = 1.0$ (vLLM) on Qwen2.5-VL-3B [44], with at most $2{,}048$ new tokens per turn and $4{,}096$ per trajectory; O1–O2 run once over $\mathcal{K}$ (a few CPU hours for the 92K-chunk suite at zero API cost) and O3–O4 run in the policy process on every rollout step, without any external LLM service.

B Proofs and Supporting Analyses

This appendix supports Propositions 1–3 stated in §3: Proposition 1 (zero-API-cost construction with comparable aggregate coverage, §3.1), Proposition 2 (bounded prompt with full reachability, §3.2), and Proposition 3 (offline-trained policy deployed online, §3.3). Proposition 2 receives a self-contained proof; the cost and interface halves of Propositions 1 and 3 are proved formally, while their coverage and transfer halves are empirical claims supported by §4.4 and §4.6. Notation follows §3: $\mathcal{G}_H^m = (\mathcal{V}^m, \mathcal{E}_H)$ is the multimodal hypergraph populated by the parser–extractor pair (ψ, π_{ext}) under the frozen encoder $f : \mathcal{X} \to \mathbb{R}^D$, $\mathcal{M}_1^t$ and $\mathcal{M}_2$ are the L1/L2 memory tiers queried under the shared retrieval budget C, and π_θ is the GRPO-trained policy that emits trajectories $\tau = (a_0, \ldots, a_T)$ over the unified action space $\mathcal{A} = \mathcal{A}_{\text{txt}} \cup \mathcal{A}_{\text{img}} \cup \mathcal{A}_{\text{ans}}$ shared by training and inference.

B.1 Proof of Proposition 1

Proposition 1. *$\mathcal{G}_H^m$ attains entity coverage comparable to an LLM extractor at zero API cost.*

Proof. Let $\pi_{\text{ext}}^{\text{LLM}}$ denote any LLM-based entity extractor returning canonical surface forms over the type alphabet $\mathcal{Y} = \{\textit{person}, \textit{org}, \textit{geo}, \textit{event}, \textit{category}\}$ used in HyperGraphRAG [35], and let $\nu : \Sigma^* \to \Sigma_{\mathcal{Y}}^*$ be the deterministic UPPER-CASED canonicalizer shared with §3.1, where $\Sigma_{\mathcal{Y}}^*$ is the type-agnostic upper-cased canonical string space. For a corpus $\mathcal{K}$ and any extractor π, define the global coverage operator and the per-call monetary cost as

$$\text{Cov}(\pi) \;=\! \bigcup_{d \in \mathcal{K}} \bigcup_{u \in \psi(d)} \nu\big(\pi(u)\big) \;\subseteq\; \Sigma_{\mathcal{Y}}^*, \qquad \Gamma(\pi, \mathcal{K}) \;=\; \sum_{d \in \mathcal{K}} \sum_{u \in \psi(d)} c(\pi, u) \;\in\; \mathbb{R}_{\geq 0}, \tag{8}$$

where $c(\pi, u) \in \mathbb{R}_{\geq 0}$ is the dollar charge billed by π on the call u. The local extractor of §3.1 factorizes through the frozen NER head and frozen CLIP embedder,

$$\pi_{\text{ext}}(u) \;=\; \text{NER}_{\text{spaCy}}(u), \qquad \mathbf{E}_{\mathcal{V}}\big(\pi_{\text{ext}}(u)\big) \;=\; \big\{ f(v) : v \in \pi_{\text{ext}}(u) \big\} \;\subset\; \mathbb{R}^D, \tag{9}$$

both of which run on local hardware with parameters frozen at load time and issue no network call, so $c(\pi_{\text{ext}}, u) = 0$ for every u, and the corpus-level cost telescopes to

$$\Gamma(\pi_{\text{ext}}, \mathcal{K}) \;=\; \sum_{d \in \mathcal{K}} \sum_{u \in \psi(d)} 0 \;=\; 0, \qquad \Gamma(\pi_{\text{ext}}^{\text{LLM}}, \mathcal{K}) \;=\; \Theta\big(|\mathcal{K}| \cdot \bar{J} \cdot c_{\text{LLM}}\big) \;>\; 0, \tag{10}$$

where $\bar{J} = \mathbb{E}_{d \sim \mathcal{K}}[J(d)]$ and $c_{\text{LLM}} > 0$ is the strictly positive per-call API price, so the LLM baseline pays a dollar charge that scales linearly with corpus size while π_{ext} pays nothing. The OntoNotes classes of spaCy `en_core_web_lg` cover the named-entity types of $\mathcal{Y}$, and both extractors feed Eq. (8) through the same type-agnostic canonicalizer ν, so the only remaining discrepancy between the two coverages is the per-unit recall gap

$$\Delta(\mathcal{K}) \;=\; \frac{\big|\text{Cov}(\pi_{\text{ext}}^{\text{LLM}}) \setminus \text{Cov}(\pi_{\text{ext}})\big|}{\big|\text{Cov}(\pi_{\text{ext}}^{\text{LLM}})\big|} \;\in\; [0, 1], \qquad \text{Cov}(\pi_{\text{ext}}) \;\xrightarrow{\Delta \to 0}\; \text{Cov}(\pi_{\text{ext}}^{\text{LLM}}), \tag{11}$$

The cost side of Proposition 1 is exact: Eq. (10) holds by construction. The coverage side is an empirical claim rather than a formal result: we do not measure $\Delta(\mathcal{K})$ directly, and general-domain NER may differ from an LLM extractor on specialized vocabulary; §4.4 supports comparable aggregate coverage indirectly through the resulting graph scale and the downstream F_1 of Table 3, which is the precise sense in which "comparable" is used in the statement of Proposition 1. □

B.2 Proof of Proposition 2

Proposition 2. *The two-tier memory bounds the prompt uniformly over T with $\mathcal{G}_H^m$ fully reachable.*

Proof. The L1 memory and the prompt evolve under Eqs. (4)–(5): each round admits the fresh L2 (and web) evidence into L1 and appends one bounded digest to the conversation,

$$\mathcal{M}_1^t \;=\; \mathcal{M}_1^{t-1} \cup \mathcal{R}_2(q_t) \cup \mathcal{R}_{\text{on}}(q_t), \qquad \mathcal{M}_1^0 \;=\; \varnothing, \qquad \text{prompt}(s_{t+1}) \;=\; \text{prompt}(s_t) \oplus g_t, \tag{12}$$

where the digest g_t condenses the merged evidence block $\mathcal{K}_{q_t}^*$ (truncated to $L_{\max}$ words) and is itself bounded,

$$|g_t| \;\leq\; L_{\text{dig}}, \qquad \forall\, t \in \{0, \ldots, T-1\}. \tag{13}$$

Writing $s_t = (q, v_q, \mathcal{M}_1^{t-1}, h_t)$ for the agent state, an immediate induction on t, combined with the context threshold L_{cap} that halts the agent rollout once reached, yields the prompt bound

$$|\mathrm{prompt}(s_t)| \;\le\; \min\big(|\mathrm{prompt}(s_0)| + t \cdot L_{\mathrm{dig}},\; L_{\mathrm{cap}}\big) \;\le\; L_{\mathrm{cap}}, \qquad \forall\, t \in \{0, \dots, T\}, \tag{14}$$

independent of the trajectory horizon T and dependent on the admission constants $(L_{\mathrm{dig}}, L_{\mathrm{cap}})$ alone. For reachability, the L2 backend stores the entire constructed hypergraph store and is queried by exact inner-product Top-C under the flat FAISS index on every round,

$$\mathcal{M}_2 \;=\; \mathcal{V}^m \cup \mathcal{E}_H \cup \mathcal{I}, \qquad \mathcal{R}_2(q_t) \;=\; \operatorname*{Top\text{-}C}_{x \in \mathcal{M}_2} \langle f(q_t), f(x) \rangle, \tag{15}$$

unconditionally, so that no gating step can mask any part of $\mathcal{M}_2$ from a retrieval action,

$$\mathcal{R}_2(q_t) \;\subseteq\; \mathcal{M}_1^t, \qquad \mathcal{K}_{q_t}^* \;=\; \operatorname*{Top\text{-}C}_{x \in \mathcal{M}_1^t} \langle f(q_t), f(x) \rangle. \tag{16}$$

Fix any node $x^* \in \mathcal{G}_H^m$ and choose $q_t \in \mathcal{X}$ whose CLIP encoding aligns with $f(x^*)$ (the surface form for $x^* \in \mathcal{V}^m$, or the host sentence for $x^* \in \mathcal{E}_H$): the inner product $\langle f(q_t), f(x^*) \rangle$ then ranks x^* within the exact Top-C of Eq. (15), so $x^* \in \mathcal{R}_2(q_t) \subseteq \mathcal{M}_1^t$, and a quantifier sweep over $q_t \in \mathcal{X}$ shows

$$\bigcup_{q_t \in \mathcal{X}} \mathcal{R}_2(q_t) \;=\; \mathcal{V}^m \cup \mathcal{E}_H \cup \mathcal{I} \;\supseteq\; \mathcal{V}^m \cup \mathcal{E}_H, \tag{17}$$

so every stored item remains recoverable on demand through L2 with no loss of accessibility, while the prompt never ingests more than one bounded digest per round. Combining Eq. (14) with Eq. (17) establishes Proposition 2, with both guarantees holding for every trajectory horizon T. □

B.3 Proof of Proposition 3

Proposition 3. *The source-agnostic interface deploys the offline-trained π_θ online without updates.*

Proof. Denote the training and inference distributions by $\mathcal{D}_{\mathrm{off}}$ and $\mathcal{D}_{\mathrm{on}}$ respectively, sharing the query marginal of $\mathcal{D}$ but differing in the cached evidence source,

$$\mathcal{K}_{q_t}^* \subseteq \mathcal{V}^m \cup \mathcal{E}_H \cup \mathcal{I} \;\text{ under } \mathcal{D}_{\mathrm{off}}\; (\mathcal{R}_{\mathrm{on}} = \varnothing), \qquad \mathcal{K}_{q_t}^* \subseteq \mathcal{V}^m \cup \mathcal{E}_H \cup \mathcal{I} \cup \mathcal{S}(\mathcal{W}) \;\text{ under } \mathcal{D}_{\mathrm{on}}\; (\mathcal{R}_{\mathrm{on}} = \mathcal{S}(\mathcal{W})). \tag{18}$$

Every evidence element enters the prompt only through the per-round digests of Eq. (12), and the templating functions depend on the retrieved evidence only through the digest tokens, giving the source-agnostic prompt

$$\mathrm{prompt}(s_t) \;=\; T_{\mathrm{seed}}(q, v_q, h_t) \;\oplus\; \bigoplus_{i<t} T_{\mathrm{slot}}\big(g_i\big), \tag{19}$$

where neither T_{seed} nor T_{slot} branches on the provenance of the evidence underlying g_i. Because the same encoder f embeds entities, hyperedges and SerpAPI evidence, the inference embedding set lies in the training image $f(\mathcal{X})$, and the action set $\mathcal{A}$ is identical at both stages, so for every $s_t \in \mathrm{supp}(\mathcal{D}_{\mathrm{on}})$ there exists $s_t' \in \mathrm{supp}(\mathcal{D}_{\mathrm{off}})$ with $\mathrm{prompt}(s_t) = \mathrm{prompt}(s_t')$, yielding the policy invariance

$$\pi_\theta\big(a \,|\, s_t\big) \;=\; \pi_\theta\big(a \,|\, s_t'\big), \qquad \forall\, a \in \mathcal{A},\; \forall\, s_t \in \mathrm{supp}(\mathcal{D}_{\mathrm{on}}). \tag{20}$$

Letting $\theta^* = \arg\max_\theta \mathcal{J}_{\mathrm{GRPO}}(\theta; \mathcal{D}_{\mathrm{off}})$ be the GRPO-optimal parameters trained under $\mathcal{D}_{\mathrm{off}}$ via Eq. (7), Eqs. (19)–(20) show that enabling $\mathcal{R}_{\mathrm{on}}$ changes neither the action space, the prompt template, nor the embedding interface seen by π_{θ^*}: the web extension is invisible to the policy's input–output contract. This is a structural statement about the evidence interface, not a distributional one: $\mathcal{D}_{\mathrm{on}}$ may shift the evidence distribution, no gradient ever flows from the online channel, and we make no invariance claim about $\mathcal{J}_{\mathrm{GRPO}}(\theta; \mathcal{D}_{\mathrm{on}})$. The transfer itself is therefore an empirical finding: §4.6 shows the same checkpoint scoring all three retrieval regimes unchanged, which is the precise content of Proposition 3, obtained with no online gradient step and no SerpAPI call during training. □

Summary. The three results separate what holds by construction from what is measured. The zero-cost side of Proposition 1 (Eq. (10)), the prompt bound and full reachability of Proposition 2 (Eqs. (14) and (17)), and the interface invariance of Proposition 3 (Eq. (20)) are exact; the coverage side of Proposition 1 and the online transfer of Proposition 3 are empirical, supported by the graph scale and downstream F_1 of §4.4 and the unchanged checkpoint of §4.6, respectively.

C MMGraph-Agent Algorithm Details

This appendix expands §3 into executable pseudocode for MMGraph-Agent's three contributions, following the algorithm-card style of HyperGraphRAG [35]. Algorithm 1 populates the multimodal hypergraph $\mathcal{G}_H^m = (\mathcal{V}^m, \mathcal{E}_H)$ of §3.1 from the static corpus $\mathcal{K}$ through the local parser–extractor pair (ψ, π_{ext}) under the frozen CLIP encoder $f : \mathcal{X} \to \mathbb{R}^D$; Algorithm 2 realizes the cache-inspired two-tier memory of §3.2 as a per-round controller that exposes the same Top-C primitive on both tiers $\mathcal{M}_1^t$ and $\mathcal{M}_2$ and keeps the prompt bounded via digest admission; and Algorithm 3 fits the unified policy π_θ via the GRPO clipped surrogate of §3.3 on offline-only rollouts, leaving the inference-time activation of $\mathcal{R}_{\text{on}} = \mathcal{S}(\mathcal{W})$ as a free switch requiring no retraining or prompt change.

Module 1: Zero-API-Cost Multimodal Hypergraph Construction. Following §3.1, every document $d \in \mathcal{K}$ is first sliced by the rule-based parser ψ into modality-tagged units $\{u_j^{\mu_j}\}$, and each textual unit is fed to the local extractor $\pi_{\text{ext}} = \text{NER}_{\text{spaCy}}$ that emits the multiset of entities and the host sentence as the corresponding n-ary hyperedge, while image units are embedded directly by f into the shared retrieval space. The frozen multimodal encoder f embeds entities, hyperedges, images and queries into a single D=768 space, and a streaming flat FAISS inner-product index serves exact cross-modal Top-k with fast append-only insertion. No paid LLM is invoked at any stage, so the entire entity layer of ϕ is built at *zero API cost* and can be re-run locally on every newly arriving batch, extending the searchable index on ordinary commodity hardware at no API cost. Each hyperedge keeps its full host sentence as payload, so higher-order context stays within one retrievable unit.

Algorithm 1 Module 1: Zero-API-Cost Multimodal Hypergraph Construction

Require: Static corpus $\mathcal{K} = \{d_i\}_{i=1}^N$, modality alphabet $\mathbb{M}$, frozen CLIP encoder f
Ensure: Multimodal hypergraph $\mathcal{G}_H^m = (\mathcal{V}^m, \mathcal{E}_H)$ with embedding banks $(\mathbf{E}_\mathcal{V}, \mathbf{E}_{\mathcal{E}_H}, \mathbf{E}_\mathcal{I})$

Initialize $\mathcal{V}^m \leftarrow \varnothing,\ \mathcal{E}_H \leftarrow \varnothing,\ \mathbf{E}_\mathcal{V} \leftarrow \varnothing,\ \mathbf{E}_{\mathcal{E}_H} \leftarrow \varnothing,\ \mathbf{E}_\mathcal{I} \leftarrow \varnothing$
for each document $d \in \mathcal{K}$ in parallel **do**
  $\{u_j^{\mu_j}\}_{j=1}^{J(d)} \leftarrow \psi(d)$ {rule-based parser tags each unit by $\mu_j \in \mathbb{M}$}
  for each textual unit $u \in \psi_{\text{txt}}(d)$ **do**
    $V_u \leftarrow \pi_{\text{ext}}(u) = \text{NER}_{\text{spaCy}}(u)$ {local entities, no LLM call}
    $\mathcal{V}^m \leftarrow \mathcal{V}^m \cup V_u$
    if $|V_u| \geq 1$ **then**
      $e_H \leftarrow \text{sentence}(u)$ as n-ary hyperedge over V_u with the sentence as payload
      $\mathcal{E}_H \leftarrow \mathcal{E}_H \cup \{e_H\}$
    end if
  end for
  $\mathbf{E}_\mathcal{I} \leftarrow \mathbf{E}_\mathcal{I} \cup \{f(u) : u \in \psi(d),\ \mu(u) = \text{image}\}$ {image units embedded directly}
end for
$\mathbf{E}_\mathcal{V} \leftarrow \{f(v) : v \in \mathcal{V}^m\},\ \mathbf{E}_{\mathcal{E}_H} \leftarrow \{f(e_H) : e_H \in \mathcal{E}_H\}$ {frozen CLIP, D=768}
Insert $\mathbf{E}_\mathcal{V} \cup \mathbf{E}_{\mathcal{E}_H} \cup \mathbf{E}_\mathcal{I}$ into flat FAISS inner-product indices keyed on $\text{sim}(x, x') = \langle f(x), f(x') \rangle$
return $\mathcal{G}_H^m = (\mathcal{V}^m, \mathcal{E}_H)$ with $(\mathbf{E}_\mathcal{V}, \mathbf{E}_{\mathcal{E}_H}, \mathbf{E}_\mathcal{I})$

Complexity. With $|\mathcal{K}| = N$ documents, $\bar{J} = \mathbb{E}_d[J(d)]$ units per document, and $\bar{n}$ entities per unit, the entity scan costs $\mathcal{O}(N \cdot \bar{J} \cdot \bar{n})$ CPU operations on spaCy that parallelize trivially across $\mathcal{K}$. Embedding $\mathcal{V}^m \cup \mathcal{E}_H$ is one frozen-CLIP forward pass per element, and flat FAISS insertion is lightweight per node, so the wall-clock for the 92K-chunk evaluation corpus stays within a few CPU hours at zero API cost (§4.4), strictly dominating the API-bound profile of LLM-based construction baselines.

Module 2: Cache-Inspired Multimodal Hypergraph Memory. Following §3.2, the offline retrieval channel $\mathcal{R}_{\text{off}}$ is split across two tiers: the L2 backend $\mathcal{M}_2$ persistently retains every entity, hyperedge, and visual item, the L1 per-trajectory cache $\mathcal{M}_1^t$ accumulates the evidence retrieved so far, and both are queried by the same Top-C inner-product primitive on $\mathcal{X}$. The policy π_θ itself emits the sub-query $q_t \in \mathcal{X}$ inside its retrieval action; the controller consults L2 (and, when enabled, the web channel) on every round, admits the fresh evidence into L1, and serves the merged evidence $\mathcal{K}_{q_t}^*$ from the unified L1 index. The evidence block is truncated to $L_{\max}$ words and condensed into a bounded digest before entering the conversation until the halting threshold L_{cap}, so the prompt length is bounded uniformly in the horizon T, while every stored item remains fully reachable through the L2 backend. Starting from an empty L1 cache, the working set is warmed from the hypergraph rather than from policy priors, and later rounds re-serve the evidence already held in L1, with one budget for both tiers.

Algorithm 2 Module 2: Cache-Inspired Two-Tier Memory at Round t

Require: State $s_t = (q, v_q, \mathcal{M}_1^{t-1}, h_t)$, backend $\mathcal{M}_2 = \mathcal{V}^m \cup \mathcal{E}_H \cup \mathcal{I}$, encoder f, budget C, bounds $(L_{\max}, L_{\mathrm{dig}}, L_{\mathrm{cap}})$
Ensure: Round-t cache $\mathcal{M}_1^t$ and digest g_t appended to the prompt

$q_t \leftarrow \pi_\theta$'s retrieval action {sub-query emitted autoregressively, $q_t \in \mathcal{X}$}
$\mathcal{R}_2(q_t) \leftarrow \text{Top-}C_{x \in \mathcal{M}_2} \langle f(q_t), f(x) \rangle$ {L2 warm-up via flat FAISS index, every round}
$\mathcal{R}_{\mathrm{on}}(q_t) \leftarrow \mathcal{S}(\mathcal{W}, q_t)$ if web search enabled, else $\varnothing$
$\mathcal{M}_1^t \leftarrow \mathcal{M}_1^{t-1} \cup \mathcal{R}_2(q_t) \cup \mathcal{R}_{\mathrm{on}}(q_t)$ {admit fresh evidence into the L1 memory}
$\mathcal{K}_{q_t}^* \leftarrow \text{Top-}C_{x \in \mathcal{M}_1^t} \langle f(q_t), f(x) \rangle$ {merged evidence served from the unified L1 index}
Truncate $\mathcal{K}_{q_t}^*$ to $L_{\max}$ words; $g_t \leftarrow$ policy digest of $\mathcal{K}_{q_t}^*$, $|g_t| \le L_{\mathrm{dig}}$
$\mathrm{prompt}(s_{t+1}) \leftarrow \mathrm{prompt}(s_t) \oplus g_t$; halt the trajectory if $|\mathrm{prompt}| \ge L_{\mathrm{cap}}$
return $(\mathcal{M}_1^t, g_t)$

Complexity. Each round costs one exact L2 Top-C over the flat FAISS index, one insertion of the fresh evidence into the L1 memory, and one Top-C serve from the L1 index, giving a per-round controller cost that does not grow with the prompt across the horizon T. Only the bounded digests enter the language-model prompt until the halting threshold L_{cap}, so the prompt length stays bounded across the entire trajectory while the full $\mathcal{G}_H^m$ remains reachable through L2, matching the bound proved in Proposition 2 for every trajectory, every corpus size and all three retrieval regimes alike.

Module 3: Offline-Trained, Online-Deployed Policy Learning. Following §3.3, the policy π_θ samples actions $a_t \sim \pi_\theta(\cdot|s_t)$ from the unified action space $\mathcal{A} = \mathcal{A}_{\mathrm{txt}} \cup \mathcal{A}_{\mathrm{img}} \cup \mathcal{A}_{\mathrm{ans}}$ in one autoregressive emission, with both retrieval channels routed through the cache so the policy emits a single retrieval token agnostic to its source. Training samples G rollouts per query under web augmentation disabled, computes the hit-aware reward $r(\tau) = \mathbb{1}[N_c \ge 1]\eta(N_c)R_{F_1} - \lambda(1 - \mathbb{1}_{\mathrm{fmt}}\mathbb{1}[N_c \ge 1])$ with the smooth call-count regulator $\eta(N) = 1 - \frac{2}{N+2}$, and updates θ via the GRPO clipped surrogate anchored to a frozen reference π_{ref}. At inference $\mathcal{R}_{\mathrm{on}} = \mathcal{S}(\mathcal{W})$ is enabled and the unified evidence interface lets the trained policy serve online retrieval without any further parameter updates or retraining.

Algorithm 3 Module 3: Offline-Only GRPO Training of π_θ

Require: Training set $\mathcal{D}$, hypergraph $\mathcal{G}_H^m$, frozen reference π_{ref}, group size G, clip width ϵ, KL coefficient κ
Ensure: GRPO-optimal parameters θ^*

Initialize $\theta \leftarrow \theta_0$ from the Qwen2.5-VL-3B backbone
for step $= 1, \ldots, 75$ **do**
  Sample query $(q, v_q, y^*) \sim \mathcal{D}$ with $\mathcal{R}_{\mathrm{on}} = \varnothing$ (offline only)
  for $i = 1, \ldots, G$ in parallel **do**
    Roll out $\tau_i \sim \pi_{\theta_{\mathrm{old}}}(\cdot|q, v_q)$ via Algorithm 2 per round
    Compute $r(\tau_i) = \mathbb{1}[N_c^{(i)} \ge 1]\eta(N_c^{(i)})R_{F_1}^{(i)} - \lambda\big(1 - \mathbb{1}_{\mathrm{fmt}}^{(i)} \cdot \mathbb{1}[N_c^{(i)} \ge 1]\big)$
  end for
  $\mu_G \leftarrow \frac{1}{G}\sum_i r(\tau_i)$, $\sigma_G \leftarrow \mathrm{std}_i\, r(\tau_i)$, $\hat{A}_i \leftarrow (r(\tau_i) - \mu_G)/(\sigma_G + \delta)$
  $\rho_t \leftarrow \pi_\theta(a_t|s_t)/\pi_{\theta_{\mathrm{old}}}(a_t|s_t)$ for every token of every τ_i
  $\theta \leftarrow \theta + \alpha_{\mathrm{lr}} \cdot \nabla_\theta \Big[\sum_{i,t} \frac{1}{G|\tau_i|} \min\big(\rho_t \hat{A}_i,\ \mathrm{clip}(\rho_t, 1 \pm \epsilon)\hat{A}_i\big) - \kappa\, \mathrm{KL}(\pi_\theta \,\|\, \pi_{\mathrm{ref}})\Big]$
end for
return $\theta^* = \theta$ {at inference, activate $\mathcal{R}_{\mathrm{on}} = \mathcal{S}(\mathcal{W})$ via the same cache interface}

Complexity. Each step issues G rollouts whose cost is dominated by Algorithm 2 times the autoregressive decode of π_θ; the gradient pass is one forward-backward over the GRPO surrogate at $\mathcal{O}(G \cdot \bar{|\tau|} \cdot |\theta|)$ per step, executed on 8×A100 (80GB) for 75 steps with $G = 8, \epsilon = 0.2, \kappa = 10^{-3}$. Because training is offline-only and $\mathcal{S}$ is not invoked during rollouts, the SerpAPI bill scales with the inference budget alone; combined with the source-agnostic evidence interface of Proposition 3, θ^* requires no further parameter update once $\mathcal{R}_{\mathrm{on}}$ is enabled, so one checkpoint serves all three retrieval regimes.

End-to-end pipeline. Algorithms 1–3 compose as $\mathcal{C}_1 \to \mathcal{C}_2^* \to \mathcal{C}_3$ on the same FAISS-backed hypergraph: $\mathcal{C}_1$ builds $\mathcal{G}_H^m$ at zero API cost (Prop. 1), $\mathcal{C}_2$ keeps the prompt bounded uniformly per round (Prop. 2), and $\mathcal{C}_3$ trains π_{θ^*} on offline rollouts before $\mathcal{R}_{\mathrm{on}} = \mathcal{S}(\mathcal{W})$ activates at inference (Prop. 3), so each guarantee is preserved when the three modules are chained end to end.

D Benchmark Detail

This appendix expands §4 into a self-contained reference for the evaluation suite. We do not propose a new benchmark: every dataset is publicly released by its authors and reused as is, re-grouped by retrieval regime to match MMGraph-Agent's joint offline-online setting. Throughout, $\mathcal{K}$ is the static multimodal corpus, $\mathcal{W}$ the dynamic web reachable via $\mathcal{S}$, and (q, v_q) a query with gold answer y^*.

D.1 Dataset Details

The 23 datasets split into three retrieval regimes; the eight †-marked rows in Table 3 are OOD datasets held out from training so that §4.2 can probe out-of-distribution generalization. We sample 16–32 items per dataset for a 640-item evaluation pool, following MMSearch-R1 [60]; the reported std. reflects three repeated runs on this same fixed pool, as decoding samples at temperature 1.0. The 600 training examples comprise 233 questions from six Knowledge-Only datasets and 367 multimodal questions (425 images) from eight in-domain datasets; none is from the eight held-out OOD datasets, and with $\mathcal{R}_{\text{on}}$ disabled in training every retrieval targets the offline hypergraph.

(i) Knowledge-Only (9 datasets). The gold answer is recoverable from $\mathcal{K}$, so MMGraph-Agent serves them through $\mathcal{G}_H^m$ and rarely escalates to the web. *DocBench* [72], *MMLongBench-Doc* [39] require evidence from long PDFs with figures and tables; *ViDoSeek*† [58] adds visually rich pages. *2Wiki* [17], *HotpotQA* [63], *Musique* [52] test multi-hop composition over Wikipedia, while *NQ* [26], *PopQA*† [40], *TriviaQA*† [25] probe single-hop factoid recall over open-domain Wikipedia.

(ii) WebSearch-Only (5 datasets). The gold answer concerns entities absent from $\mathcal{K}$, so MMGraph-Agent must escalate to $\mathcal{W}$ via SerpAPI. *FVQA*† [57] requires commonsense facts about visual entities; *InfoSeek*† [5], *MMSearch*† [23], *SimpleVQA*† [6] ask encyclopedic questions whose answers depend on visual subject identification; *LiveVQA*† [11] adds time-sensitive after-cutoff news queries. We build no offline KG for them, so the WebSearch-Only column isolates the online-retrieval skill of π_θ.

(iii) Knowledge+WebSearch (9 datasets). The gold answer requires combining offline grounding with up-to-date or domain-specific web evidence within one trajectory. *MedXpertQA* [73] and *MicroVQA* [4] pose clinical and microscopy questions; *FinMME* [36] and *R-Bench* [13] pair financial or scientific reports with current snippets; *We-Math2.0* [43] and *GeoGrid-Bench* [22] couple math/geo evidence with web explanations; and *EEmo-Bench* [12], *ArtQuest* [3], *AstroLLaVA* [68] require both visual grounding and external cultural or scientific commentary retrieved from the web.

D.2 Evaluation Details

Every metric is computed from the rollout JSONL and dataset gold with no LLM judge. We write $\text{tok}(\cdot)$ for the whitespace-tokenizer applied after the Graph-R1 normalizer [34] (lower-case, articles removed, punctuation stripped). For predicted answer $\hat{y}$ and gold y^*, the token-level F_1 is the SQuAD-style metric used by Graph-R1 [34] and HyperGraphRAG [35],

$$F_1(\hat{y}, y^*) \;=\; \frac{2\,|\text{tok}(\hat{y}) \,\cap\, \text{tok}(y^*)|}{|\text{tok}(\hat{y})| + |\text{tok}(y^*)|} \;\in\; [0,1], \tag{21}$$

which doubles as the per-rollout reward R_{F_1} inside the GRPO reward of Eq. (6) and is the only headline number used across all the main and ablation tables. We additionally macro-average F_1 over each of the three regimes of §D.1, so the K+Web regime average isolates the value of jointly scheduling $\mathcal{R}_{\text{off}}$ and $\mathcal{R}_{\text{on}}$ within one trajectory, a contrast invisible to the per-dataset numbers alone. For every rollout we further log the search count N_c, the round count, and the prompt length in tokens; the cache statistics in §4.5 and the cost–latency numbers in §4.4 are derived from these logs without extra inference, with API calls priced at the public `gpt-4o`, `gpt-4o-mini`, and SerpAPI rates. Predictions are sampled with temperature 1.0 and top-$p = 1.0$ (at most 2,048 new tokens per turn, 4,096 per trajectory) under a MAX_GEN_ROUND $= 8$ budget; the same decoding configuration is shared across all baselines and ablations of Table 3 so that retrieval-side gains cannot be confounded with sampling stochasticity. Each evaluation run samples independently and is repeated three times; the macro-F_1 standard deviation is reported after the $\pm$ sign in every table cell, while † in Table 3 marks the eight held-out OOD datasets, none of which is seen in training. All datasets retain their public licenses; we release the processed corpus, code and evaluation harness, but not the raw SerpAPI pages, for independent evaluation on the same 640-item pool, so every reported F_1 can be re-derived end-to-end with the released pipeline and its token-level scoring scripts, without an LLM judge.

E Baseline Detail

Each method below is given as *(i) Overview* (purpose and substrate on $\mathcal{K}$), *(ii) Per-regime implementation* (an operator for each regime of §D.1, or why it is unsupported) and *(iii) Hyperparameters*. $\mathcal{K}=\{d_i\}_{i=1}^{N}$ is the static corpus, $\mathcal{W}$ the web reachable via SerpAPI $\mathcal{S}$, f the frozen CLIP encoder (D=768), $\mathrm{Chunk}(\cdot)$ a 300-token splitter, $\mathrm{LLM}(\cdot)$ a Qwen2.5-VL-3B [44] model (GPT-4.1-mini [41] for the closed-source columns of Table 3) and $\mathrm{TopK}_k\langle\cdot,\cdot\rangle$ inner-product Top-k on f. Unsupported regimes fall back to direct generation from (q, v_q); all retrieval-based methods share the embedder and the rollout prompt of Table 2, so the gaps in Table 3 reflect retrieval and policy design alone.

E.1 Generation-only

NaiveGen [69]. *(i) Overview.* Single-pass VLM decoding that exposes the parametric prior; substrate: none, only (q, v_q) is read. *(ii) Per-regime implementation.* $\hat{y}=\mathrm{LLM}(q, v_q)$ in all three regimes. *(iii) Hyperparameters.* temperature 1.0, $\mathrm{top}_p = 1.0$, 2,048 new tokens; run with both Qwen2.5-VL-3B and GPT-4.1-mini as backbones (both NaiveGen columns of Table 3). As it retrieves nothing, NaiveGen is the no-retrieval reference for every retrieval-based method described below.

E.2 RAG-based

RAG-based baselines add a one-shot offline index over $\mathcal{K}$ but no policy and no online channel.

StandardMMRAG [65]. *(i) Overview.* Vision-language RAG: text chunks and images of $\mathcal{K}$ are encoded by f into a flat inner-product index and the Top-k hits are concatenated to the prompt, with no graph layer or reranker; substrate $\mathcal{B}=\{(c_i, f(c_i))\}_{c_i\in\mathrm{Chunk}(\mathcal{K})\,\cup\,\mathrm{Imgs}(\mathcal{K})}$. *(ii) Per-regime implementation. Knowledge-Only*: $R=\mathrm{TopK}_k\langle f(q, v_q), f(c_i)\rangle$, $\hat{y}=\mathrm{LLM}(q, v_q, R)$; *WebSearch-Only*: fallback; *Knowledge+WebSearch*: offline half only. *(iii) Hyperparameters.* chunk size 300, top-$k=5$, the same frozen CLIP f as MMGraph-Agent, so the embedder is held fixed. It therefore realizes one-shot flat retrieval with no hypergraph, cache memory, policy or online web channel.

RAG-Anything [14]. *(i) Overview.* Multimodal GraphRAG: a paid VLM extracts entities, relations and modality tags into a structural graph $\mathcal{G}_s$ and a textual graph $\mathcal{G}_t$, built once at substantial API cost (§4.4); queries combine navigation on $\mathcal{G}_s$ with semantic matching on $\mathcal{G}_t$. *(ii) Per-regime implementation. Knowledge-Only*: $R=\mathrm{Nav}(\mathcal{G}_s, q)\cup\mathrm{Sim}(\mathcal{G}_t, q)$, $\hat{y}=\mathrm{LLM}(q, v_q, R)$; *WebSearch-Only*: fallback; *Knowledge+WebSearch*: offline half only (updates need a costly rebuild). *(iii) Hyperparameters.* extractor GPT-4o, navigation depth $\ell=2$, top-$k=5$ on $\mathcal{G}_t$; the GPT-4.1-mini column of Table 3.

E.3 Training-based

SFT and GRPO internalize the answer; MMSearch-R1 drives a single online channel; none uses an offline hypergraph or a bounded cross-round memory, both of which MMGraph-Agent adds.

SFT [70]. *(i) Overview.* Supervised fine-tuning on the same 600 training examples with a single-turn cross-entropy loss on the gold answer; no retrieval, reasoning trace or policy. *(ii) Per-regime implementation.* $\hat{y}=\mathrm{LLM}_{\theta_{\mathrm{SFT}}}(q, v_q)$ in all regimes; with no test-time retrieval, evidence outside the parametric memory (notably the eight OOD datasets) cannot reach the answer. *(iii) Hyperparameters.* learning rate 5×10^{-6}, batch 32, 3 epochs, AdamW; same backbone as MMGraph-Agent.

GRPO [47]. *(i) Overview.* GRPO on the same backbone and clipped objective (Eq. (7)) with both the hypergraph and the online channel disabled, so the policy can only refine its parametric reasoning over (q, v_q). *(ii) Per-regime implementation.* $\hat{y}=\mathrm{LLM}_{\theta_{\mathrm{GRPO}}}(q, v_q)$ in all regimes after 75 steps with $N_c\equiv0$, yielding a reasoning rewrite of the prior, not a retrieval policy. *(iii) Hyperparameters.* $G=8, \epsilon=0.2, \kappa=10^{-3}$, 75 steps as in MMGraph-Agent; reward $r(\tau)=R_{F_1}-\lambda\mathbb{1}[\neg\mathbb{1}_{\mathrm{fmt}}]$ without retrieval terms, so its gap to MMGraph-Agent isolates the contribution of the retrieval channels.

MMSearch-R1 [60]. *(i) Overview.* Multimodal RL search agent: GRPO over text search and Google Lens image search, with no offline hypergraph and no cache, so each round writes raw evidence into the prompt; substrate $\mathcal{W}\cup\mathcal{S}_{\mathrm{lens}}$. *(ii) Per-regime implementation. Knowledge-Only*: partial, answered by web search since no offline hypergraph exists; *WebSearch-Only*: native, $a_t\sim\pi_\theta(\cdot\,|\,s_t, \mathcal{W})$ with the prompt of Table 2; *Knowledge+WebSearch*: web-only coverage. *(iii) Hyperparameters.* GRPO as in MMGraph-Agent; SerpAPI returns 3 URLs per text or image search; at most 8 rounds.

E.4 MMGraph-Agent Ablations

Each ablation removes one component of the full system (§E.5) and keeps its pipeline and prompt unchanged, so the deltas of §4.3 reflect the contribution of each component to F_1. Unless stated otherwise, every variant inherits the backbone, tools and prompt (Table 2) of the full system in §E.5.

w/o Hypergraph. *(i) Overview.* Skips the construction of §3.1 (the local parser, spaCy NER and frozen CLIP encoder that build the entity, hyperedge and visual banks), leaving $\mathcal{G}_H^m$ empty so every retrieval escalates to the web; substrate $\mathcal{W}$ only. *(ii) Per-regime implementation. Knowledge-Only*: former hypergraph hits go to web search and the cache hit rate collapses without an L2 backend; *WebSearch-Only*: unchanged; *Knowledge+WebSearch*: web-only, as MMSearch-R1. *(iii) Hyperparameters.* the memory runs as a pass-through cache for SerpAPI hits; all else as MMGraph-Agent.

w/o Memory. *(i) Overview.* Keeps $\mathcal{G}_H^m$ and the policy but concatenates every retrieved chunk raw into the next prompt instead of passing it through the L1 cache of §3.2; substrate $\mathcal{G}_H^m \cup \mathcal{W}$ without the L1/L2 split. *(ii) Per-regime implementation.* In all regimes $\mathcal{K}_{q_t}^* = \mathcal{R}_2(q_t)$ every round and the digest admission of Eq. (5) becomes raw concatenation, so the prompt grows linearly in t and K+Web overflows the 8K window on $\sim 12\%$ of rollouts (§4.5). *(iii) Hyperparameters.* No L1 Top-C or 5K cap; truncation and digesting disabled; backbone, tools, prompt and checkpoint as MMGraph-Agent.

w/o Policy. *(i) Overview.* Hypergraph, memory and tools wired as in the full system, but the policy is the base Qwen2.5-VL-3B without any RL update; substrate $\mathcal{G}_H^m \cup \mathcal{W}$ with $\theta = \theta_{\text{base}}$. *(ii) Per-regime implementation.* Same routing as MMGraph-Agent in all regimes; the base model emits $\langle$`text_search`$\rangle$/$\langle$`search`$\langle$`img`$\rangle\rangle$ only when its zero-shot prior favors retrieval, so its calls are not calibrated to the evidence need and it under-searches K+Web, isolating the effect of the offline-only GRPO training of §3.3 and of its hit-aware reward as specified in Eq. (6). *(iii) Hyperparameters.* same prompt, tools, hypergraph and memory constants as MMGraph-Agent; no SFT or GRPO step.

w/o Retrieval. *(i) Overview.* All retrieval channels removed: a NaiveGen-style single-pass decode on the trained checkpoint, isolating the residual reasoning gain of GRPO; substrate: none. *(ii) Per-regime implementation.* $\hat{y} = \text{LLM}_{\theta_{\text{MMGA}}}(q, v_q)$ with $N_c \equiv 0$ in all regimes; format compliance is kept. *(iii) Hyperparameters.* as MMGraph-Agent, with retrieval calls short-circuited at the rollout layer, so neither $\mathcal{G}_H^m$, the L1/L2 memory nor the web channel is consulted during its single-pass decode.

E.5 MMGraph-Agent

MMGraph-Agent. *(i) Overview.* A frozen CLIP encoder and spaCy NER build $\mathcal{G}_H^m$ at zero API cost; an L1/L2 cache with digest-bounded admission bounds the prompt; a GRPO-trained π_θ emits one retrieval token, so offline and online retrieval share one action space; training is offline-only, and the source-agnostic interface deploys the trained policy online without updates. *(ii) Per-regime implementation.* $\mathcal{G}_H^m$ is partitioned per dataset, identically for all methods. *Knowledge-Only*: each round warms L1 from $\mathcal{G}_H^m$, with image search on $\leq 3\%$ of actions; *WebSearch-Only*: $\mathcal{G}_H^m$ is empty and L1 is fed by $\mathcal{S}$ alone; *Knowledge+WebSearch*: the same token is routed to $\mathcal{G}_H^m$ or $\mathcal{W}$, and π_θ alternates $\langle$`text_search`$\rangle$ and $\langle$`search`$\langle$`img`$\rangle\rangle$ (§4.6). *(iii) Hyperparameters.* f = CLIP-ViT-L/14 (D=768); $C = 5$, evidence truncation 1,200 words, digest $\approx$300 words, context cap 5,000 tokens; GRPO $G = 8, \epsilon = 0.2, \kappa = 10^{-3}$, AdamW at 2×10^{-6} on 8×A100 (80GB), 75 steps on 600 in-domain examples; SerpAPI 3 URLs per text or image search; at most 8 rounds; one checkpoint for all regimes.

F Future Work

Several promising directions extend the framework. First, multilingual and broader-modality encoders in place of CLIP+spaCy would let $\mathcal{G}_H^m$ absorb a wider range of corpora and modalities (e.g., audio, video and point clouds) while keeping the zero-API-cost guarantee of §3.1. Second, learning the memory constants (C and the truncation, digest and context-cap lengths) inside the GRPO loop would adapt the cache to each regime's trajectory length without the manual sweep of §4.5. Third, adding process-supervision terms (citation faithfulness, retrieval cost, latency) to the reward of Eq. (6) would let π_θ trade F_1 against deployment constraints while keeping the same offline-to-online interface. Finally, scaling π_θ to larger backbones with more training and building $\mathcal{G}_H^m$ over more domains and languages are natural extensions, as both reuse the zero-API-cost construction of §3.1 and the offline-to-online GRPO recipe of §3.3, with the unified L1/L2 cache keeping the prompt bounded.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and §1 state contributions formalized in §3 and validated in §4.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss limitations here: the offline hypergraph incurs a one-time construction cost before online serving; Web-only datasets depend on an external search API; and all experiments use only a single backbone scale, with other scales left to future work.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: Propositions 1–3 are stated in §3, with proofs and analyses in Appendix B.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: §4 and Appendices A–E specify prompts, pseudocode, data and settings.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: Code, data and harness are released; all datasets are public (Appendix D).

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Optimizer, learning rate, GRPO and cache settings are in §4 and Appendix E.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Table 3 reports $\pm$ std. over three evaluation runs on every cell (Appendix D.2).

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)

- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Hardware (8×A100, 80GB) and construction time are reported in §4 and §4.4.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We use only public datasets, with no human subjects or any personal data.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Positive: multimodal grounding can reduce hallucination. Risks: web results may be wrong or biased, and image retrieval may raise privacy or copyright issues. Mitigation: we use only public data and APIs, and recommend filtering and human oversight.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: Our assets build on public models and data and pose no elevated misuse risk.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All assets are cited in §4 and Appendix D.1 and used under their licenses.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The released code and data are documented in our GitHub README.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: This work involves no crowdsourcing and no research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: No research with human subjects is conducted, so no IRB approval is needed.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: The policy is a GRPO-trained Qwen2.5-VL-3B (§3.3); indexing uses no LLM.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.