GATOR: GRANULARITY-AWARE TOKEN-OPTIMIZED ROUTING FOR EFFICIENT MULTIMODAL RETRIEVAL

Anonymous authors
Paper under double-blind review

ABSTRACT

Multimodal agents rely on external memory to access stored evidence beyond a finite context window, yet reading retrieved evidence remains costly. A document page may be represented as a caption, full text, or an image, with costs differing by more than an order of magnitude. Retaining rich representations can waste budget, whereas uniformly choosing cheap ones can discard answer-critical evidence. We therefore propose GATOR, a cost-aware router that complements relevance ranking with query-dependent granularity selection. Under an explicit token budget, its greedy selector balances relevance, redundancy, and cost, retaining at most one representation per page while promoting coverage across pages. A lightweight scorer learns representation preferences from automatically derived weak supervision and on-policy distillation, without LLM calls during training. The same selector accepts host retrieval scores, enabling integration with existing memory systems without retraining or re-indexing them. On UniDoc and MMDocRAG, standalone GATOR reduces token-equivalent evidence cost by 41.3% and 59.4% relative to Vanilla RAG while improving answer F1. Mounted on Mem0 and A-Mem, it reduces evidence cost by up to 88.5%, with observed F1 changes ranging from -0.011 to $+0.042$. Code & data: https://anonymous.4open.science/r/GATOR-1A30.

1 INTRODUCTION

Agents increasingly tackle multimodal tasks that require understanding documents, interpreting figures, and reasoning over textual and visual evidence (Yu et al., 2025; Cho et al., 2024). The textual and visual evidence accumulated during these tasks can exceed the model's finite context window, limiting access to information needed for subsequent reasoning. **Multimodal agent memory** addresses this limitation by storing textual and visual information outside the active context and retrieving relevant evidence when needed (Zhang et al., 2025; Du et al., 2025; Huang et al., 2026). This allows agents to retain access to earlier observations and external knowledge while bringing a relevant subset into the model's input. Memory systems thus supports reasoning across queries and long-term interactions without requiring the entire accumulated history to remain in context.

Although external memory reduces context pressure, reading retrieved multimodal evidence can still incur substantial inference cost. Figure 1 highlights successive challenges faced in efficient multimodal memory: ❶ **Unspecified reading cost:** retrieval identifies relevant pages without explicitly choosing among their captions, full text, and images, whose token costs vary substantially. ❷ **Overpreserving detail:** without knowing which representation suffices, retaining the richest available evidence offers a conservative default but incurs the full cost of potentially unnecessary detail. ❸ **Overcorrecting for cost:** replacing this default with uniformly cheap representations reduces expenditure but can discard answer-critical information, sacrificing the quality it sought to preserve. Existing memory systems primarily focus on memory organization and retrieval relevance (Chhikara et al., 2025; Xu et al., 2025). Efficiency methods compress retrieved context (Jiang et al., 2023; Pan et al., 2024; Xu et al., 2024) or adapt whether and how much to retrieve (Asai et al., 2024; Jeong et al., 2024). Neither strategy directly determines how cheaply each retrieved page can be read while balancing retrieval cost and preserving sufficient evidence for the query.

The failures of both high-fidelity defaults and uniformly cheap selection reveal a common limitation: neither page relevance nor representation cost alone determines whether the evidence is sufficient

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

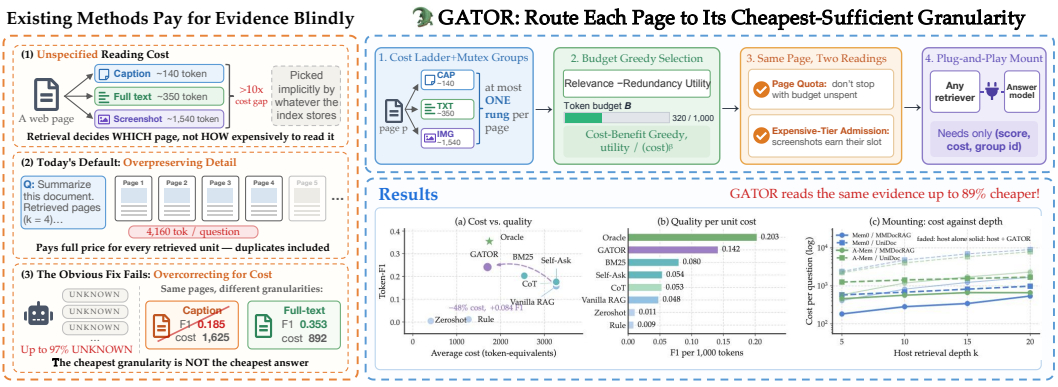


Figure 1: Motivation and preview of the proposed GATOR with result demonstration.

for a query. A relevant page may require detailed visual information for one query but only a brief summary for another. Our insight is therefore to treat *query-dependent granularity selection* as an explicit decision: balance the information retained for the query against the cost of its representation. This decision complements relevance ranking and can operate on its output, allowing the cost of reading evidence to be optimized without modifying the underlying retriever.

Building on this insight, we propose GATOR, a cost-aware router between retrieval and answer generation. GATOR groups alternative representations by source page and selects at most one representation per page under an explicit token budget. A budgeted greedy policy balances relevance and redundancy, while a coverage constraint preserves evidence from distinct pages. A lightweight scorer learns query-dependent representation preferences from weak supervision derived from evidence annotations and answer matches. On-policy distillation further aligns its sequential selections with a cost-aware teacher, without LLM calls during training. The selector requires only a relevance score, a token cost, and a group identifier per candidate, enabling integration with existing memory systems without retraining or re-indexing the host. On UniDoc and MMDocRAG, standalone GATOR reduces token-equivalent cost by 41.3% and 59.4% relative to Vanilla RAG. When mounted on Mem0 and A-Mem, GATOR reduces token-equivalent evidence cost by 21.4–88.5% without modifying host retrieval, with up to an 854% increase in F1 per token relative to the hosts. Our contributions are therefore threefold:

- **Granularity-aware selection:** We propose GATOR, a retriever-agnostic router that selects at most one representation per page under a token budget. A budgeted greedy policy balances relevance and redundancy, while a coverage constraint preserves evidence from distinct pages (Section 2.2).
- **Lightweight policy learning:** We train a compact scorer with relevance labels and weak granularity targets, augmented by on-policy distillation for sequential selection. Training requires no LLM calls, and deployment requires no host retraining or re-indexing (Sections 2.3 and 2.4).
- **Extensive benchmarking:** Comparisons with baselines on UniDoc and MMDocRAG, two multimodal document QA benchmarks, demonstrate superior answer quality and efficiency. Mounting experiments and ablations validate portability and component effectiveness (Sections 3.2–3.4).

2 METHOD

GATOR separates evidence retrieval from query-dependent granularity selection, producing a *mixed-granularity* context for answer generation (Figure 2). We first formulate the choice among representations of retrieved pages under a token budget, with at most one representation per page (Section 2.1). A sequential selector balances relevance, redundancy, and reading cost, with coverage and admission rules regulating evidence inclusion (Section 2.2). For standalone use, a lightweight scorer learns relevance and representation preferences from automatically derived supervision and on-policy distillation (Section 2.3). The same selector also accepts adapted host retrieval scores through a common candidate interface, enabling integration with existing memory systems without modifying their retrievers or rebuilding their search indices (Section 2.4).

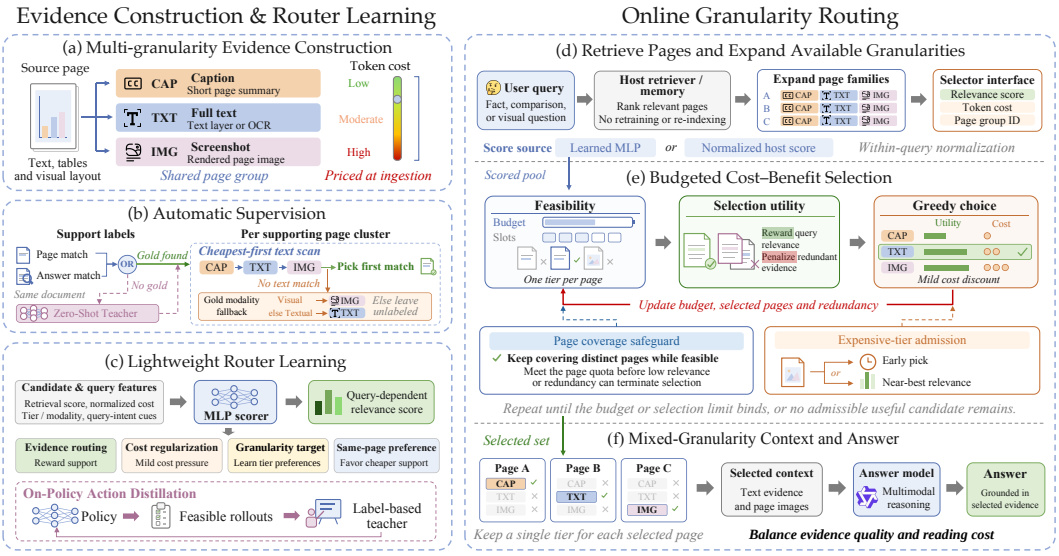


Figure 2: Overview of GATOR. **Offline (a–c):** construct page representations and train a lightweight scorer. **Online (d–f):** select evidence under a token budget and assemble the answer context.

2.1 Problem Formulation

Given a query q, we construct a candidate set $C = \{u_1, \ldots, u_n\}$ from retrieved sources and their available representations. Each source page may provide a *caption*, *full text*, or a rendered *page image*. Each candidate u_i has a relevance score $p_i \in [0, 1]$, a token-equivalent cost $c_i > 0$, a source-page identifier g_i, and a granularity label $\ell_i \in \mathcal{L} = \{\text{CAP}, \text{TXT}, \text{IMG}\}$. Scores are produced by the learned scorer in standalone use or adapted from host retrieval scores when mounted. Costs are assigned to individual representations; construction details appear in Appendix A.3.

Candidates sharing a source page form a group $C_g = \{u_i : g_i = g\}$, where $G = \{g_i : u_i \in C\}$ denotes the retrieved source pages. We seek a subset $S \subseteq C$ under a token budget B and a limit of M selected units, satisfying $\sum_{u_i \in S} c_i \leq B$ and $|S| \leq M$. To avoid selecting multiple representations of the same page, we impose a *family-mutex* constraint on each source group:

$$|S \cap C_g| \leq 1 \qquad \text{for every } g \in G. \tag{1}$$

Thus, each page contributes at most one available representation to the answer context. For a selected source, the desired granularity is the lowest-cost representation that retains the evidence needed for the query. Granularity selection therefore complements relevance ranking by considering both the evidence retained in each representation and its cost within the answer model's context.

2.2 Cost-Aware Granularity Selection

To choose among feasible representations, GATOR builds the answer context sequentially. Cost-aware scoring ranks candidates, page coverage regulates stopping, and expensive-tier admission screens costly additions. Starting from $S = \emptyset$, let $G(S) = \{g_u : u \in S\}$ denote the covered source pages and $b(S) = B - \sum_{v \in S} c_v$ the remaining budget. At each step with $|S| < M$, we consider candidates that fit the remaining budget and come from uncovered pages:

$$\mathcal{F}(S) = \{u \in C \setminus S : c_u \leq b(S),\ g_u \notin G(S)\}. \tag{2}$$

Each page's representations compete for one slot, allowing granularity to vary across pages.

Cost-aware scoring. We use a budgeted greedy policy to rank feasible candidates by a cost-adjusted relevance–redundancy utility. The utility before cost adjustment is

$$h(u \mid S) = \underbrace{\alpha p_u}_{\text{relevance}} - \underbrace{\gamma R(u, S)}_{\text{redundancy}}, \tag{3}$$

where $\alpha, \gamma \geq 0$ weight relevance and redundancy. With coverage vectors e_u, the overlap penalty for a candidate from an uncovered page is

$$R(u, S) = \frac{\rho}{|S|} \sum_{v \in S} \text{clip}(\cos(e_u, e_v), 0, 1), \qquad S \neq \emptyset. \tag{4}$$

The factor ρ moderates overlap penalties across distinct pages, since similar pages within a document can contain complementary evidence. We set $R(u, S) = 0$ when S is empty or coverage vectors are unavailable; individually missing vectors contribute zero overlap. Reading cost is incorporated through the budget-normalized quantity $\bar{c}_u = \max(c_u/B, \epsilon)$, yielding the cost-aware score:

$$s(u \mid S) = \begin{cases} h(u \mid S)/\bar{c}_u^{\beta}, & h(u \mid S) > 0, \\ h(u \mid S), & \text{otherwise.} \end{cases} \tag{5}$$

Here $\epsilon > 0$ stabilizes the denominator. The exponent $\beta \in [0, 1]$ interpolates between utility-only ranking ($\beta = 0$) and inverse-cost weighting of positive utility ($\beta = 1$). Intermediate values temper the preference for very cheap units while favoring lower cost at comparable utility. Nonpositive utilities remain unscaled for the stopping rule below.

Page coverage. The score above can become nonpositive before complementary pages are covered, especially under concentrated relevance or strong overlap penalties. We therefore target κ distinct pages before applying score-based stopping. After selecting the first unit $u^{(1)}$, we fix $p_{\text{ref}} = p_{u^{(1)}}$ as the reference relevance and define

$$\mathcal{F}_{\text{cov}}(S) = \{u \in \mathcal{F}(S) : |G(S)| < \kappa \ \text{ or } \ p_u \geq \delta p_{\text{ref}}\}, \tag{6}$$

with the relevance floor inactive before the first selection. Until κ pages are covered, we waive this floor and nonpositive-score stopping to admit complementary evidence with lower scores. Once the target is met, the floor applies and selection stops if the best eligible score is nonpositive. Budget, unit-limit, and admission constraints remain active, so coverage is pursued only while feasible.

Expensive-tier admission. Relaxing stopping for coverage can admit expensive evidence merely to add a page. We therefore require candidates with $c_u \geq \tau$ to satisfy

$$|S| \leq r \qquad \text{or} \qquad p_u \geq \sigma p_{\text{ref}}. \tag{7}$$

The first condition permits expensive evidence within the first $r + 1$ selections; later additions must retain relevance of at least σp_{ref}. The fixed within-query reference reduces dependence on absolute score calibration. Since τ thresholds token cost, the rule applies to long text as well as page images. Cost-aware scoring selects from $\mathcal{A}(S)$, the candidates in $\mathcal{F}_{\text{cov}}(S)$ that pass admission:

$$u^{\star} = \arg\max_{u \in \mathcal{A}(S)} s(u \mid S), \qquad S \leftarrow S \cup \{u^{\star}\}. \tag{8}$$

We update the remaining budget, page set, and overlap scores after each addition. Selection ends when M units have been selected, no eligible candidate remains, or the nonpositive-score stopping condition applies. The resulting mixed-granularity context satisfies the budget and family-mutex constraints by construction. Appendix B details the complete procedure.

2.3 OPD-Based Evidence Scorer

The selector uses scores to weigh source relevance and query-dependent representation preferences against reading cost. In standalone use, a compact MLP combines retrieval relevance, normalized cost, representation type and level, modality signals, and query-intent features. A bounded retrieval prior augments the logit for candidate u_i:

$$z_i = f_\theta(x(q, u_i)) + \omega \log_{10}(1 + 9r_i), \qquad p_i = \frac{\exp(z_i)}{\sum_{u_j \in C} \exp(z_j)}, \tag{9}$$

where $r_i \in [0, 1]$ is the retrieval score and $x(q, u_i)$ contains inference-time features. The weights p_i express relative selection preferences, not calibrated sufficiency probabilities.

To supervise these scores, we derive a binary relevance target y_i from annotated evidence pages and answer-string matches within the reference document. For each supporting page, m_i marks

the cheapest available representation whose text contains a reference answer. When no text match is found, evidence-modality annotations specify a page image for visual evidence or full text for textual evidence; unresolved pages receive no granularity target. If neither annotations nor answer matching identify a supporting candidate, Zeroshot selections can provide relevance pseudo-labels under the same granularity-labeling rules. These targets provide weak supervision for query-dependent representation preferences.

Joint training objective. Using these targets, we combine relevance and granularity supervision with cost regularization and on-policy distillation:

$$\mathcal{L} = \lambda_{\text{route}}\mathcal{L}_{\text{route}} + \lambda_{\text{cost}}\mathcal{L}_{\text{cost}} + \lambda_{\text{level}}\mathcal{L}_{\text{level}} + \lambda_{\text{margin}}\mathcal{L}_{\text{margin}} + \lambda_{\text{opd}}\mathcal{L}_{\text{opd}}. \tag{10}$$

The routing loss maximizes probability mass on supporting candidates, while the cost loss penalizes expected normalized reading cost. The level loss allocates probability to weak granularity targets across supporting pages; the margin loss favors each such target over more expensive candidates from the same supporting page. We assign a lower weight to cost than to routing; loss definitions and weights appear in Appendix B.3.

On-policy distillation. The preceding losses operate on complete candidate pools, whereas successive selections remove alternatives and consume budget. We therefore train on partial selections visited by the current student policy. At state S_t, we renormalize the candidate logits over the budget-feasible, uncovered-page set $\mathcal{F}(S_t)$ from Equation 2: $\pi_\theta(\cdot \mid S_t) = \text{softmax}_{\mathcal{F}(S_t)}(z/T_s)$. The teacher weights relevance, granularity preference, retrieval score, and normalized cost $\tilde{c}_i$ by a_y, a_m, a_r, a_c:

$$z_i^{\text{T}} = a_y y_i + a_m m_i + a_r r_i - a_c \tilde{c}_i, \qquad \pi_{\text{T}}(\cdot \mid S_t) = \text{softmax}_{\mathcal{F}(S_t)}(z^{\text{T}}/T_{\text{T}}), \tag{11}$$

Here T_s, T_{T} denote the student and teacher policy temperatures. We minimize reverse KL between the two policies on states S_t visited by sequentially sampling actions from the current student policy:

$$\mathcal{L}_{\text{opd}} = \mathbb{E}_{S_t \sim d_{\pi_\theta}}\left[D_{\text{KL}}(\pi_\theta(\cdot \mid S_t) \,\|\, \pi_{\text{T}}(\cdot \mid S_t))\right], \tag{12}$$

where d_{π_θ} is the rollout state distribution. We compute KL over all feasible candidates without differentiating through sampled actions. Rollouts enforce budget and family-mutex masks; coverage and expensive-tier admission apply only at inference. The teacher is computed directly from supervision labels and candidate metadata, so optimization requires no LLM calls.

2.4 Integration with Memory Systems

Mounted deployment reuses the selector with host retrieval scores, bypassing the standalone learned scorer. The adapter supplies each representation's score, token-equivalent cost, source group, identifier, and payload. Shared source metadata defines groups; records without it form separate groups. Stored representations and costs are reused. Plain-text records may also provide a truncated caption proxy; images require accessible payloads.

Score adaptation. The adapter first converts host scores to a convention in which larger values indicate greater relevance, including reversing distance-based scores. For the resulting candidate scores s_i, we apply within-query min–max normalization followed by a softmax:

$$\hat{s}_i = \frac{s_i - s_{\min}}{s_{\max} - s_{\min}}, \qquad p_i = \frac{\exp(4\hat{s}_i)}{\sum_{u_j \in C} \exp(4\hat{s}_j)}. \tag{13}$$

Equal scores yield a uniform distribution. Otherwise, candidate order is preserved for the selection rules in Section 2.2; representation preferences derive from host scores and adapter-defined priors.

Selection and handoff. The adapter passes these scores, costs, and group identifiers to the selector. The minimal interface requires no host embeddings, so $R(u, S) = 0$ in this configuration; budget, family-mutex, coverage, and admission rules still apply. Selected identifiers and payloads are returned for assembly into the answer context. Integration adds a selection step after native retrieval, without changing the host's index, embedding model, or retrieval procedure. In our paired evaluation, both arms use the same retrieved results and answer model, isolating the effect of evidence selection. Appendix A.7 details the adapters and mounting configuration.

Table 1: Standalone routing on UniDoc and MMDocRAG under a shared five-unit budget. **Bold** and underlined mark the best and second best results, respectively; Oracle is an upper bound and is excluded from the ranking. GATOR is highlighted.

Router	UniDoc				MMDocRAG			
	Cost ↓	F1 ↑	Judge ↑	F1/1k ↑	Cost ↓	F1 ↑	Judge ↑	F1/1k ↑
Upper bound								
Oracle	1948	0.366	0.609	0.188	1543	0.343	0.496	0.222
Rule	494	0.018	0.027	0.036	2046	0.005	0.012	0.002
Zeroshot	**177**	0.006	0.013	0.034	**643**	0.003	0.021	0.005
BM25	3845	0.274	0.371	0.071	1237	0.133	0.243	0.108
Vanilla RAG	4160	0.194	0.330	0.047	2377	0.121	0.234	0.051
Chain-of-Thought	4160	0.176	0.353	0.042	2377	0.172	0.213	0.072
Self-Ask	4160	0.230	0.291	0.055	2377	0.123	0.261	0.052
GATOR	2440	**0.296**	**0.429**	**0.121**	965	**0.187**	**0.287**	**0.194**

3 EXPERIMENTS

Our conduct evaluations addressing the following three research questions concerning method effectiveness, robustness, and indivisual component contributions:

- **RQ1:** How does GATOR balance answer quality and evidence cost in standalone use and when mounted on existing memory systems? ***(Main Results**; Section 3.2)*
- **RQ2:** Does the contributions of GATOR remain persistent across answer models, training corpora, and image-cost assumptions? ***(Case Studies**; Section 3.3)*
- **RQ3:** How do individual components contribute to the gains of GATOR? ***(Ablations**; Section 3.4)*

3.1 EXPERIMENTAL SETUP

Datasets. We evaluate GATOR on two large multimodal datasets. **UniDoc** (Peng et al., 2025) spans eight domains giving 815 documents and 1,600 QA. Its pages are scans with no text layer, so the middle rung of the ladder exists only after OCR. **MMDocRAG** (Dong et al., 2025) contributes 1,082 documents and 4,055 QA and offers all three granularities for most pages. Detailed per-corpus dataset statistics are shown in Appendix A.1.

Models and Baselines. We adopt **Qwen3-VL-8B-Instruct** as the primary backbone model, with **Qwen3-VL-Flash** and **GPT-5.6-Luna** evaluated as alternative backbones in Section 3.3. Text and visual retrieval use `all-MiniLM-L6-v2` and `CLIP ViT-B/32`, respectively, combined through reciprocal rank fusion. We compare GATOR against baselines from three categories: *granularity policies* (**Zeroshot**, **Rule**), *prompting strategies* (**Chain-of-Thought**, **Self-Ask**), and *retrieval methods* (**Vanilla RAG**, **BM25**). We assess portability by mounting GATOR on **Mem0** (Chhikara et al., 2025) and **A-Mem** (Xu et al., 2025) while retaining their native retrievers. Detailed model, baseline, and memory-system configurations are provided in Appendix A.

Metrics. We evaluate answer quality using token-level F1 and LLM-judge accuracy, and report average token-equivalent evidence cost. **F1 per 1,000 tokens** measures answer quality per unit of reading cost. Mounting experiments additionally report cost savings, retained unit counts, and relative changes in average cost per unit. Metric definitions and cost accounting are provided in Appendix A.8.

3.2 MAIN RESULTS (RQ1)

To address RQ1, we evaluate GATOR as a standalone router (Table 1) and as a component mounted on Mem0 and A-Mem (Table 2). The standalone comparison evaluates the learned scorer and selector jointly, while mounting assesses selection under host scores. We examine the overall quality–cost balance in both settings.

Table 2: GATOR mounted on two memory systems. Changes are relative to Host. Δ Cost/unit estimates the change (%) in average cost per retained unit. F1 and F1/1k colors indicate lower, higher, and unchanged; darker shading denotes larger absolute changes within each metric.

	Cost			Units kept		F1		F1/1k		Δ Cost/unit
k	Host	+GATOR	Saving	Host	+GATOR	Host	+GATOR	Host	+GATOR	(%)
Mem0 on MMDocRAG										
5	411	182	55.7%	5.0	2.3	0.153	0.142	0.372	0.780	−3.7
10	821	283	65.5%	10.0	3.7	0.170	0.170	0.207	0.601	−6.8
15	1232	343	72.2%	15.0	4.4	0.193	0.188	0.157	0.548	−5.1
20	1880	538	71.4%	20.0	6.5	0.195	0.203	0.104	0.377	−11.9
Mem0 on UniDoc										
5	2394	579	75.8%	5.0	2.0	0.170	0.200	0.071	0.345	−39.5
10	4789	682	85.8%	10.0	2.3	0.157	0.199	0.033	0.292	−38.1
15	6958	820	88.2%	15.0	2.4	0.240	0.251	0.034	0.306	−26.3
20	8533	980	88.5%	20.0	3.8	0.230	0.252	0.027	0.257	−39.6
A-Mem on MMDocRAG										
5	583	458	21.4%	5.0	3.9	0.148	0.145	0.254	0.317	+0.7
10	1283	567	55.8%	10.0	5.4	0.147	0.142	0.115	0.250	−18.2
15	1722	664	61.4%	15.0	5.3	0.152	0.152	0.088	0.229	+9.1
20	2404	661	72.5%	20.0	5.9	0.144	0.155	0.060	0.234	−6.8
A-Mem on UniDoc										
5	2238	1259	43.7%	5.0	4.5	0.147	0.142	0.066	0.113	−37.5
10	4036	1428	64.6%	10.0	5.3	0.185	0.183	0.046	0.128	−33.2
15	5830	1565	73.2%	15.0	5.1	0.177	0.180	0.030	0.115	−21.0
20	7871	1693	78.5%	20.0	5.7	0.187	0.192	0.024	0.113	−24.5

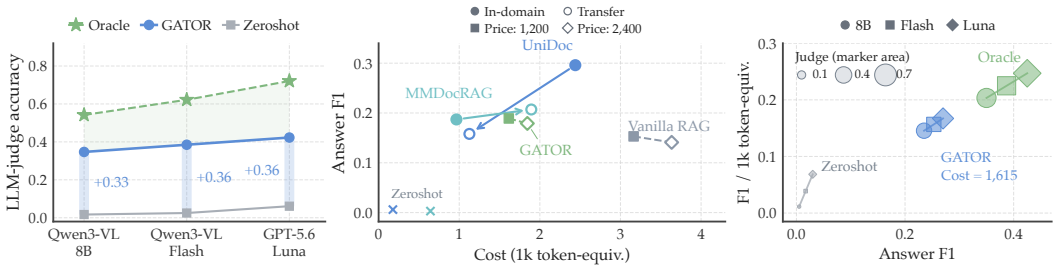
❶ ***Granularity-aware routing improves both quality and efficiency.*** We compare standalone routers with the same questions, answer model, and five-unit limit. Vanilla RAG fixes representation preferences, BM25 changes page ranking, and prompting baselines reuse the Vanilla RAG evidence. GATOR achieves the highest F1, judge accuracy, and F1/1k among non-oracle methods on both benchmarks. Relative to Vanilla RAG, it reduces evidence cost by 41.3% on UniDoc and 59.4% on MMDocRAG, while increasing F1 from 0.194 to 0.296 and from 0.121 to 0.187, respectively. These improvements yield 2.60× and 3.81× the F1/1k of Vanilla RAG. It also surpasses the strongest prompting baseline on each corpus at lower evidence cost, showing that learned evidence selection improves the quality–cost balance beyond modifying the answering strategy alone.

❷ ***Efficiency gains extend to unchanged memory systems.*** Table 2 compares each host with and without GATOR using *identical retrieval results*. We vary retrieval depth over $k \in \{5, 10, 15, 20\}$ on both corpora, retaining the answer model and changing only how retrieved evidence is selected. Across sixteen configurations, GATOR reduces token-equivalent evidence cost by 21.4–88.5% and achieves 1.25–9.54× the hosts' F1/1k, with F1 changes of −0.011 to +0.042. The Δ Cost/unit metric separates changes in average unit cost from changes in retained count: 14 of 16 configurations reduce cost per unit by 3.7–39.6%. On UniDoc at $k = 20$, for example, Mem0 retains 3.8 units on average with a 39.6% lower mean unit cost; total cost falls from 8,533 to 980, while F1 rises from 0.230 to 0.252. Thus, savings combine fewer retained units with less costly evidence, whose average cost reflects both page composition and representation choice. This supports reducing the reading burden while preserving useful evidence across memory architectures. Appendix C.1 further analyzes how retrieval depth affects total savings, retained unit counts, and average cost per unit across both corpora.

Takeaway: GATOR improves standalone quality and efficiency; mounted routing reduces evidence count and usually cost per unit, with limited F1 changes.

3.3 Case Studies (RQ2)

To address RQ2, we examine whether routing gains persist across answer models, training corpora, and image-cost assumptions. These studies separate the quality of the reader, the transferability of the router, and the efficiency of the selected evidence under changing deployment conditions.



(a) Answer quality (b) Corpus and price shifts (c) Backbone efficiency

Figure 3: Routing across backbones, corpora, and image costs. (a) Judge accuracy. (b) F1 versus cost under corpus and image-price shifts; arrows denote transfer and dashed lines connect prices. (c) Backbone F1 and F1/1k, with marker area proportional to Judge. Connected points share a router and evidence cost; comparisons use study-specific controls. Full results appear in Tables 4 and 5.

Routing gains persist across answer models. We hold ingestion, retrieval, and selected evidence fixed while varying the answer model across Qwen3-VL-8B, Qwen3-VL-Flash, and GPT-5.6-Luna. Each router supplies identical contexts across models. We compare GATOR with the training-free Zeroshot policy and an Oracle supplied with annotated evidence, using the same questions and judge protocol. This design tests whether the advantage of learned selection depends on a particular reader. Figure 3(a) shows that GATOR achieves judge accuracy 0.347, 0.385, and 0.423, respectively, exceeding Zeroshot by 0.330–0.362. The ordering of Oracle, GATOR, and Zeroshot holds across backbones, indicating that stronger readers retain the quality advantage of selected evidence. Oracle also improves across models, showing that reader capability and evidence quality remain distinct sources of variation in the final answer.

Corpus and price shifts affect the quality–cost balance differently. For corpus transfer, we train the router on one corpus and evaluate it on the other without target-corpus adaptation, comparing it with an in-domain router and Zeroshot on the same target questions. A separate price-sensitivity experiment doubles the page-image cost parameter from 1,200 to 2,400 token-equivalents. We rebuild the stores and rerun evaluation for both GATOR and Vanilla RAG at each price, keeping the remaining configuration fixed. This tests whether the observed savings depend on a particular assigned image cost. Figure 3(b) combines both studies while retaining their respective controls. Transfer to MMDocRAG raises F1 from 0.187 to 0.207 but lowers F1/1k as cost rises from 965 to 1,894; transfer to UniDoc lowers F1 from 0.296 to 0.158 while increasing F1/1k as cost falls from 2,440 to 1,126. Both transferred routers outperform Zeroshot, although the direction of transfer determines whether the policy spends more for additional evidence or sacrifices quality for lower expenditure. Under both image prices, GATOR achieves higher F1 than Vanilla RAG, with 48.9–49.2% lower cost and 2.42–2.50× its F1/1k. Thus, corpus transfer changes evidence allocation and its trade-offs, whereas the efficiency advantage persists under higher image prices.

Stronger backbones improve returns from fixed evidence. We further analyze the same backbone runs using F1 and F1/1k to measure how effectively each reader uses its supplied context. For each router, evidence cost is fixed across backbones, and F1/1k divides mean F1 by mean cost and scales it by 1,000. Any efficiency change therefore reflects answer quality under the same evidence allocation. Table 4 and Figure 3(c) show that GATOR increases F1 from 0.235 to 0.270 and F1/1k from 0.146 to 0.167 across backbones, with evidence cost fixed at 1,615. Even Qwen3-VL-8B with GATOR exceeds GPT-5.6-Luna with Zeroshot in F1 (0.235 versus 0.030), suggesting that reader capacity alone cannot compensate for insufficient evidence. Backbone improvements and evidence selection therefore provide complementary gains. Oracle F1 nevertheless ranges from 0.350 to 0.425, and its margin over GATOR widens from 0.115 to 0.155. Stronger readers improve quality for a fixed routed context while also benefiting more from annotated evidence, suggesting that further gains depend on both reader capability and the context it receives.

Takeaway: Routing complements stronger readers and remains effective under corpus and price shifts, with transfer-dependent quality–cost trade-offs.

3.4 Ablation Studies (RQ3)

To examine the roles of representation availability and the OPD objective, we restrict the available granularities or remove OPD on UniDoc, using the full model in Table 1 as the reference. Granularity removal tests inexpensive evidence availability; OPD removal tests supervision on partial selections. In Figure 4, removing captions lowers F1 from 0.296 to 0.088 and raises evidence cost from 2,440 to 3,642; additionally removing full text yields F1 0.062 at cost 5,925. Relative to the w/o-OPD variant, the full model improves F1 by 0.194 while reducing evidence cost by 31.3%. Table 6 reports complete results, including answer F1, judge accuracy, evidence cost, and F1/1k for all variants.

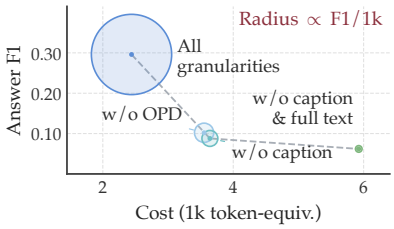


Figure 4: UniDoc ablations (Table 6). Radius ∝ F1/1k.

Takeaway: The full configuration achieves the best quality–cost balance among the tested ablations, supporting the inclusion of low-cost representations and the OPD objective.

4 Related Work

Multimodal Agent Memory System. Memory streams and hierarchical context management allow agents to retain and revisit information across interactions (Park et al., 2023; Packer et al., 2023). Subsequent designs organize memories as consolidated facts (Chhikara et al., 2025), interlinked notes (Xu et al., 2025), or entity-relation graphs (Gutiérrez et al., 2024; Guo et al., 2025c), reflecting the diversity of memory representations and operations reviewed in recent surveys (Zhang et al., 2025; Du et al., 2025). These mechanisms govern what knowledge is retained, how it is updated, and which evidence remains accessible to subsequent queries across tasks and an agent's evolving interaction history. For multimodal evidence, visual document retrieval preserves information from page layouts, tables, and figures through image-based representations (Faysse et al., 2025; Ma et al., 2024; Yu et al., 2025; Cho et al., 2024), while multimodal graph frameworks connect textual and visual evidence for retrieval across modalities (Guo et al., 2025b). *Our work complements these systems with budgeted granularity selection that balances reading cost and evidence sufficiency.*

Policy Learning and On-Policy Distillation. Learning sequential decisions requires accounting for how earlier actions shape subsequent states. DAgger addresses this distribution shift in imitation learning by collecting expert supervision on states visited by the learner, aligning supervision with the learner's state distribution (Ross et al., 2011). For autoregressive models, ImitKD (Lin et al., 2020) and Generalized Knowledge Distillation (GKD) (Agarwal et al., 2024) use teacher feedback on student-generated sequences to mitigate training–inference mismatch as prediction errors alter the contexts encountered during generation. MiniLLM uses reverse KL divergence to discourage students from overestimating low-probability teacher outputs, concentrating probability mass on responses favored by the teacher (Gu et al., 2024). DistiLLM combines skew KL objectives with adaptive off-policy reuse of student-generated outputs (Ko et al., 2024), while DistiLLM-2 proposes contrastive objectives that treat teacher and student responses differently (Ko et al., 2025).

5 Conclusion

We presented GATOR, a cost-aware router that separates evidence retrieval from query-dependent granularity selection in multimodal agent memory. Its budgeted selector balances relevance and redundancy, retains at most one representation per page, and enforces coverage across distinct pages. A lightweight scorer learns relevance and representation preferences from weak supervision, while on-policy distillation aligns its sequential decisions with a cost-aware teacher without LLM calls during training. Through an interface of relevance scores, token costs, and group identifiers, GATOR integrates with existing memory systems without host retraining or re-indexing. Experiments on UniDoc and MMDocRAG demonstrate improved standalone answer quality and efficiency, with mounting on Mem0 and A-Mem reducing token-equivalent evidence cost by up to 88.5%. Future work includes adaptive granularities and deployment-specific cost models.

AI USE STATEMENT

Generative AI tools were used to assist with drafting and polishing the manuscript, including improving clarity, grammar, and organization. The authors are responsible for the final text, claims, and research artifacts.

ETHICS STATEMENT

We evaluate GATOR on existing multimodal document QA benchmarks and credit their original sources. Human annotations are used to assess LLM-judge reliability, as described in Appendix E. Use and redistribution of the benchmark data remain subject to the terms specified by their providers.

REPRODUCIBILITY STATEMENT

Appendices A and B document the datasets, evaluation metrics, model configurations, and training procedure. Supplementary results and ablations are provided in Appendices C and D; judge validation and prompts appear in Appendices E and F. Code and data are linked in the abstract.

REFERENCES

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://arxiv.org/abs/2306.13649.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In *International Conference on Learning Representations (ICLR)*, 2024. arXiv:2310.11511.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-VL technical report. *arXiv preprint arXiv:2511.21631*, 2025. URL https://arxiv.org/abs/2511.21631.

Prateek Chhikara, Dev Khant, Saket Aryan, Taranjeet Singh, and Deshraj Yadav. Mem0: Building production-ready AI agents with scalable long-term memory. In *European Conference on Artificial Intelligence (ECAI)*, 2025. doi: 10.3233/FAIA251160. arXiv:2504.19413.

Jaemin Cho, Debanjan Mahata, Ozan Irsoy, Yujie He, and Mohit Bansal. M3DocRAG: Multimodal retrieval is what you need for multi-page multi-document understanding. *arXiv preprint arXiv:2411.04952*, 2024.

Kuicai Dong, Yujing Chang, Shijie Huang, Yasheng Wang, Ruiming Tang, and Yong Liu. Benchmarking retrieval-augmented multimodal generation for document question answering. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-0619. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/1a93178950e92fd2e7b7448f7d68fd7d-Abstract-Datasets_and_Benchmarks_Track.html.

Yiming Du, Wenyu Huang, Danna Zheng, Zhaowei Wang, Sebastien Montella, Mirella Lapata, Kam-Fai Wong, and Jeff Z. Pan. Rethinking memory in LLM based agents: Representations, operations, and emerging topics. *arXiv preprint arXiv:2505.00675*, 2025.

Manuel Faysse, Hugues Sibille, Tony Wu, Bilel Omrani, Gautier Viaud, Céline Hudelot, and Pierre Colombo. ColPali: Efficient document retrieval with vision language models. In *International Conference on Learning Representations (ICLR)*, 2025. arXiv:2407.01449.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. MiniLLM: Knowledge distillation of large language models. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/8ac015d409635f196f9e3e9dcfb9a94e-Abstract-Conference.html.

Shuyu Guo, Shuo Zhang, and Zhaochun Ren. Enhancing RAG efficiency with adaptive context compression. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 24061–24076. Association for Computational Linguistics, 2025a. doi: 10.18653/v1/2025.findings-emnlp.1307. URL https://aclanthology.org/2025.findings-emnlp.1307/.

Zirui Guo, Xubin Ren, Lingrui Xu, Jiahao Zhang, and Chao Huang. RAG-Anything: All-in-one RAG framework. *arXiv preprint arXiv:2510.12323*, 2025b.

Zirui Guo, Lianghao Xia, Yanhua Yu, Tu Ao, and Chao Huang. LightRAG: Simple and fast retrieval-augmented generation. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 10746–10761. Association for Computational Linguistics, 2025c. doi: 10.18653/v1/2025.findings-emnlp.568. URL https://aclanthology.org/2025.findings-emnlp.568/.

Bernal Jiménez Gutiérrez, Yiheng Shu, Yu Gu, Michihiro Yasunaga, and Yu Su. HippoRAG: Neurobiologically inspired long-term memory for large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. arXiv:2405.14831.

Zhengjun Huang, Zhoujin Tian, Qintian Guo, Fangyuan Zhang, Yingli Zhou, Di Jiang, Zeying Xie, and Xiaofang Zhou. LiCoMemory: Lightweight and cognitive agentic memory for efficient long-term reasoning. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 36842–36858, 2026. doi: 10.18653/v1/2026.findings-acl.1835. URL https://aclanthology.org/2026.findings-acl.1835/.

Soyeong Jeong, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, and Jong C. Park. Adaptive-RAG: Learning to adapt retrieval-augmented large language models through question complexity. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*, pp. 7036–7050, 2024. arXiv:2403.14403.

Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. LLMLingua: Compressing prompts for accelerated inference of large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2023. arXiv:2310.05736.

Jongwoo Ko, Sungnyun Kim, Tianyi Chen, and Se-Young Yun. DistiLLM: Towards streamlined distillation for large language models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pp. 24872–24895. PMLR, 2024. URL https://proceedings.mlr.press/v235/ko24c.html.

Jongwoo Ko, Tianyi Chen, Sungnyun Kim, Tianyu Ding, Luming Liang, Ilya Zharkov, and Se-Young Yun. DistiLLM-2: A contrastive approach boosts the distillation of LLMs. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267, pp. 31044–31062. PMLR, 2025. URL https://proceedings.mlr.press/v267/ko25a.html.

Alexander Lin, Jeremy Wohlwend, Howard Chen, and Tao Lei. Autoregressive knowledge distillation through imitation learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 6121–6133. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.emnlp-main.494. URL https://aclanthology.org/2020.emnlp-main.494/.

Xueguang Ma, Sheng-Chieh Lin, Minghan Li, Wenhu Chen, and Jimmy Lin. Unifying multimodal retrieval via document screenshot embedding. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2024. arXiv:2406.11251.

Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G. Patil, Ion Stoica, and Joseph E. Gonzalez. MemGPT: Towards LLMs as operating systems. *arXiv preprint arXiv:2310.08560*, 2023.

Zhuoshi Pan, Qianhui Wu, Huiqiang Jiang, Menglin Xia, Xufang Luo, Jue Zhang, Qingwei Lin, Victor Rühle, Yuqing Yang, Chin-Yew Lin, H. Vicky Zhao, Lili Qiu, and Dongmei Zhang. LLMLingua-2: Data distillation for efficient and faithful task-agnostic prompt compression. In *Findings of the Association for Computational Linguistics: ACL 2024*, 2024. arXiv:2403.12968.

Joon Sung Park, Joseph C. O'Brien, Carrie J. Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology (UIST)*, 2023. doi: 10.1145/3586183.3606763.

Xiangyu Peng, Can Qin, Zeyuan Chen, Ran Xu, Caiming Xiong, and Chien-Sheng Wu. UNIDOC-BENCH: A unified benchmark for document-centric multimodal RAG. *arXiv preprint arXiv:2510.03663*, 2025. URL https://arxiv.org/abs/2510.03663.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A. Smith, and Mike Lewis. Measuring and narrowing the compositionality gap in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, 2023. arXiv:2210.03350.

Stephane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15, pp. 627–635. PMLR, 2011. URL https://proceedings.mlr.press/v15/ross11a.html.

Parth Sarthi, Salman Abdullah, Aditi Tuli, Shubh Khanna, Anna Goldie, and Christopher D. Manning. RAPTOR: Recursive abstractive processing for tree-organized retrieval. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://openreview.net/forum?id=GN921JHCRw.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, 2022. URL https://proceedings.neurips.cc/paper/2022/hash/9d5609613524ecf4f15af0f7b31abca4-Abstract-Conference.html.

Fangyuan Xu, Weijia Shi, and Eunsol Choi. RECOMP: Improving retrieval-augmented LMs with context compression and selective augmentation. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/bda88ed2892f5e61c9a9bf215c566913-Abstract-Conference.html. arXiv:2310.04408.

Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. A-Mem: Agentic memory for LLM agents. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. arXiv:2502.12110.

Wei Yang, Jingjing Fu, Rui Wang, Jinyu Wang, Lei Song, and Jiang Bian. OMGM: Orchestrate multiple granularities and modalities for efficient multimodal retrieval. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 24545–24563. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.acl-long.1198. URL https://aclanthology.org/2025.acl-long.1198/.

Woongyeong Yeo, Kangsan Kim, Soyeong Jeong, Jinheon Baek, and Sung Ju Hwang. UniversalRAG: Retrieval-augmented generation over corpora of diverse modalities and granularities. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3843–3871. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.acl-long.177. URL https://aclanthology.org/2026.acl-long.177/.

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 *International Conference on Learning Representations (ICLR)*, 2025. arXiv:2410.10594.

Zeyu Zhang, Quanyu Dai, Xiaohe Bo, Chen Ma, Rui Li, Xu Chen, Jieming Zhu, Zhenhua Dong, and Ji-Rong Wen. A survey on the memory mechanism of large language model-based agents. *ACM Transactions on Information Systems*, 43(6):155:1–155:47, 2025. doi: 10.1145/3748302.

Zijie Zhong, Hanwen Liu, Xiaoya Cui, Xiaofan Zhang, and Zengchang Qin. Mix-of-Granularity: Optimize the chunking granularity for retrieval-augmented generation. In *Proceedings of the 31st International Conference on Computational Linguistics*, pp. 5756–5774. Association for Computational Linguistics, 2025. URL https://aclanthology.org/2025.coling-main.384/.

Yifan Zhu, Yu Mi, Yue Lu, Yanchu Guan, and Zhixuan Chu. LFRAG: Layout-oriented fine-grained retrieval-augmented generation on multimodal document understanding. *arXiv preprint arXiv:2605.22829*, 2026. URL https://arxiv.org/abs/2605.22829.

Yilong Zuo, Xunkai Li, Jing Yuan, Qiangqiang Dai, Hongchao Qin, and Ronghua Li. MAGE-RAG: Multigranular adaptive graph evidence for agentic multimodal RAG in long-document QA. *arXiv preprint arXiv:2606.15906*, 2026. URL https://arxiv.org/abs/2606.15906.

APPENDIX

GATOR: Granularity-Aware Token-Optimized Routing for Efficient Multimodal Retrieval

TABLE OF CONTENTS

A Datasets and Experimental Setup . . . 15
A.1 Datasets Statistics . . . 15
A.2 Hardware and Software . . . 15
A.3 Ingestion and Cost Accounting . . . 15
A.4 Answer Models . . . 15
A.5 Retrieval Configuration . . . 16
A.6 Baseline Details . . . 16
A.7 Memory Systems . . . 17
A.8 Evaluation Metrics . . . 17
B Additional Method Details . . . 18
B.1 Selection Procedure . . . 18
B.2 Selector Hyperparameters . . . 18
B.3 Router Training . . . 18
C Supplementary Main Results and Robustness . . . 20
C.1 Mounting Across Retrieval Depths . . . 20
C.2 Answer-Model Robustness . . . 20
C.3 Cross-Corpus Transfer . . . 21
D Ablations and Diagnostic Studies . . . 21
D.1 Component Ablations–Full Results . . . 21
D.2 Representation and Budget Analysis . . . 22
D.3 Selector Sensitivity . . . 22
D.4 Scorer Capacity . . . 22
D.5 Optimization Dynamics . . . 23
E Reliability of the LLM Judge . . . 23
F Prompt Templates . . . 24
G Extended Related Work . . . 26

A Datasets and Experimental Setup

A.1 Datasets Statistics

UniDoc provides document page images; extracted text is supplied by OCR. MMDocRAG provides quote-level textual and visual evidence, which the loader maps to source groups. Both corpora support multiple representations where the corresponding text or image is available.

Table 3: Corpus inventory and QA splits recorded for the main experiments.

	UniDoc	MMDocRAG
Documents	815	1,082
Train QA	1,223	3,420
Test QA	377	635
Caption units	7,198	55,672
Full-text units	7,147	55,672
Page-image units	7,198	19,597
Total units	21,543	130,941

Splits. Train and test partitions are grouped by document identifier. All questions and representations associated with a document remain in the same partition. Within each reported comparison, methods use the same evaluation questions; study-specific populations are stated with their results.

A.2 Hardware and Software

All experiments run on NVIDIA A100 80GBs PCIe under Linux 5.15 with Python 3.13.9 and PyTorch 2.9.1 (CUDA 12.8). The answer model is served locally through vLLM behind an OpenAI-compatible API, so token costs are the quantity of interest rather than monetary prices. The single exception is GPT-5.6-Luna in the answer-model ablation, which is reached through a hosted endpoint; its token counts are recorded the same way. Appendix A.4 describes the three models.

A.3 Ingestion and Cost Accounting

PDF pages are rendered at 150 DPI with a cap of 40 pages per document. Text is extracted from an available PDF text layer or obtained through Tesseract OCR. The default caption is a heuristic summary based on the first sentence or heading, capped at 240 characters. Representations share a source identifier and carry a cost assigned during ingestion:

$$c_{\text{text}} = |\text{chars}|/4, \qquad c_{\text{image}} = b_{\text{image}}\, v, \tag{14}$$

where the default page-image base budget is $b_{\text{image}} = 1200$ and the visual multiplier is $v = 1.2$, yielding 1440 token-equivalents per stored page image. Selected image groups can additionally include companion content, whose cost is included in the evidence total. This distinction separates the image-price parameter from the cost of a complete selected context. The generic host adapter uses its own text/image estimator when stored cost metadata is unavailable (Appendix A.7).

A.4 Answer Models

The answer-model comparison uses three vision-capable models, with Qwen3-VL-8B-Instruct also serving as the default LLM judge. Additional judges are specified in Appendix E. None of these models is fine-tuned or called during router training.

Qwen3-VL-8B-Instruct (Bai et al., 2025). The default answer model for every experiment unless stated otherwise. It is the 8B instruction-tuned member of the Qwen3-VL family, an open-weights vision-language series with native resolution handling and interleaved image–text inputs, which is what the mixed-granularity context requires: a single prompt may carry page screenshots alongside caption and full-text units.

Qwen3-VL-Flash. A smaller, latency-oriented member of the same family, used in the answer-model ablation (Appendix C.2). Holding the family fixed while changing capacity separates the effect of model scale from the effect of model lineage.

GPT-5.6-Luna. A proprietary multimodal model accessed through a hosted OpenAI-compatible endpoint, included in the answer-model ablation as the strongest available reader. Its role is to raise the Oracle ceiling: if the routing advantage were an artefact of a weak reader, it would vanish under a stronger one. Because it is a hosted model, its internal size and architecture are not public, and we treat it purely as a black-box reader.

A.5 Retrieval Configuration

Text retrieval uses `all-MiniLM-L6-v2` (384-d) over 60 nearest neighbours; visual retrieval uses `CLIP ViT-B/32` (512-d) over 40. The two rankings are fused by reciprocal rank ($k = 60$) and normalized to $[0, 1]$. Family expansion then adds the sibling granularities of every retrieved page, and one-hop graph expansion adds neighbours at a decayed score.

A.6 Baseline Details

We compare retrieval, prompting, and routing baselines under a shared evaluation protocol to assess the overall quality–cost balance.

Vanilla RAG. This baseline, denoted `Full` in the evaluation code, ranks pages by their highest candidate retrieval score and applies a fixed representation preference: full text, page image, then caption. For each page, it selects the first available representation that fits the remaining budget, stopping after five units. The representation preference is independent of the query.

BM25. Okapi BM25 re-ranks the same candidate pages using lexical matching, with $k_1 = 1.5$ and $b = 0.75$. Text associated with each page is concatenated for scoring, and term statistics are computed over the candidate pool. The representation preference and budget handling remain identical to Vanilla RAG, isolating the effect of page ranking.

Chain-of-Thought. Following chain-of-thought prompting (Wei et al., 2022), this baseline asks the answer model to reason step by step before producing its final answer. It uses the same evidence as Vanilla RAG, without demonstrations or additional retrieval. Only the extracted final answer is evaluated.

Self-Ask. Adapted from Press et al. (2023), this baseline prompts the model to decompose a question into follow-up questions and intermediate answers before returning a final answer. Decomposition is performed in a single generation over the Vanilla RAG context, without external search. Both prompting baselines receive three times the direct-answer output allowance to accommodate reasoning; the evidence budget is unchanged.

Zeroshot. This training-free granularity policy scores candidate i as $s_i = b_i + \pi_i - 0.5\tilde{c}_i$, where b_i is its retrieval score and $\tilde{c}_i$ is its cost normalized by 6,000. The tier prior π_i is 0.15 for captions, -0.05 for full text, and zero for page images. Scores are softmax-normalized and passed to the same constrained selector as `GATOR`, testing whether a fixed preference for cheaper representations suffices.

Rule. A hand-weighted linear scorer combines retrieval relevance, modality bias, representation-type prior, and normalized cost, with respective weights 0.4, 0.3, 0.2, and -0.1. Type priors are 0.6 for captions, 0.5 for full text, and 0.7 for page images. Softmax uses temperature 1.0, followed by the shared selector. This baseline tests manually specified feature weights without supervised training or distillation.

Oracle. Oracle bypasses retrieval using annotated evidence pages and prioritizes the annotated modality: images for exclusively visual evidence and full text for textual or mixed evidence. It

respects the same evidence budget and unit limit and provides a reference for answer quality with gold evidence. Because it requires test-set annotations, it is excluded from the baseline ranking.

A.7 Memory Systems

Mem0. Mem0 (Chhikara et al., 2025) supports persistent memory through extraction, consolidation, and retrieval of salient information. In our experimental configuration, records are inserted with `infer=False`, preserving their text without LLM-based fact extraction. Document identifiers, page identifiers, granularity labels, and costs are retained as metadata. Retrieval uses the native Mem0 search interface with `all-MiniLM-L6-v2` embeddings and a Qdrant vector store; the adapter passes the returned relevance scores to the router.

A-Mem. A-Mem (Xu et al., 2025) organizes memories as interlinked notes enriched with contextual descriptions, keywords, and tags. New notes are analyzed by an LLM to establish links and update related memories. We retain its native note insertion and search operations, using `all-MiniLM-L6-v2` embeddings. Linked-note expansion can return more items than the requested retrieval depth. The adapter converts retrieval distances to relevance scores, with smaller distances receiving higher scores, before applying the selector.

Mounted scoring. The adapters reuse host relevance scores, with lightweight representation priors for derived views where needed. These inputs support cost-aware selection through the shared interface; the standalone learned scorer is not applied in the mounting experiments.

A.8 Evaluation Metrics

Token-level F1. For each question, we compare the predicted answer with the reference after lowercasing, removing English articles and numeric thousands separators, and normalizing punctuation and whitespace. Tokenization preserves decimal numbers and hyphenated terms; Latin text is tokenized into words and CJK text into characters. Let T_q and R_q be the predicted and reference token multisets, and o_q their overlap count, including repeated occurrences. Precision, recall, and F1 are

$$p_q = \frac{o_q}{|T_q|}, \qquad r_q = \frac{o_q}{|R_q|}, \qquad F_q = \frac{2p_q r_q}{p_q + r_q}. \tag{15}$$

Zero overlap receives zero F1. When multiple reference aliases are available, we retain the highest F1 across them. The reported score is the macro-average $\overline{F} = N^{-1}\sum_{q=1}^{N} F_q$ over evaluation questions, so each question contributes equally regardless of answer length.

LLM-judge accuracy. The default judge is Qwen3-VL-8B-Instruct. It receives the question, reference answer, and candidate answer, with retrieved evidence withheld, and assesses substantive correctness. The rubric accepts equivalent wording, reordered lists, and equivalent date formats, while rejecting contradictions, incorrect numbers, and missing required content. Judge accuracy is the fraction of valid judgments whose binary `correct` field is true; the auxiliary graded score is not the reported accuracy. Failed calls and unparseable judgments are excluded from this denominator. The prompt appears in Figure 13, and Appendix E reports agreement with human annotations and alternative judges.

Token-equivalent evidence cost. For selected evidence set S_q, the per-question reading cost is $C_q = \sum_{i \in S_q} c_i$, including any mandatory companion content. Text costs are estimated at one token per four characters; visual evidence uses the fixed modality prices assigned during ingestion. These stored costs provide a common accounting basis across answer models and are not model-specific tokenizer counts or monetary charges. We report $\overline{C} = N^{-1}\sum_{q=1}^{N} C_q$. This metric covers selected evidence only: question and instruction tokens, generated reasoning and answers, judge calls, and offline ingestion or training are excluded. API-reported generation and judging usage is recorded separately. Appendix A.3 describes the representation-level cost accounting.

Efficiency and relative savings. The efficiency metric uses the ratio of corpus-level means,

$$\mathrm{F1/1k} = 1000\,\frac{\overline{F}}{\overline{C}}, \qquad \mathrm{Saving}(\%) = 100\left(1 - \frac{\overline{C}_{\mathrm{GATOR}}}{\overline{C}_{\mathrm{Host}}}\right). \tag{16}$$

F1/1k is not the mean of per-question F1-to-cost ratios. Higher values indicate greater answer quality per unit of evidence cost; we report F1 and cost alongside it to make the underlying trade-off explicit. Savings compare the mounted system with its host over the same questions. For mean retained count $\overline{U}$, average cost per unit is $\bar{c} = \overline{C}/\overline{U}$. Table 2 reports its relative change, $\Delta\text{Cost/unit}(\%) = 100(\bar{c}_{\text{GATOR}}/\bar{c}_{\text{Host}} - 1)$. The cost ratio decomposes as $\overline{C}_{\text{GATOR}}/\overline{C}_{\text{Host}} = (\overline{U}_{\text{GATOR}}/\overline{U}_{\text{Host}})(\bar{c}_{\text{GATOR}}/\bar{c}_{\text{Host}})$, distinguishing count reduction from changes in average unit cost. The latter reflects both selected pages and their representations; values calculated from rounded table means are approximate.

B Additional Method Details

B.1 Selection Procedure

We detail the greedy policy in Section 2.2.

Overlap computation. When coverage vectors are available and $S \neq \emptyset$, the redundancy term is

$$R(u,S) = \frac{\eta(u,S)}{|S|} \sum_{v \in S} \text{clip}(\cos(e_u, e_v), 0, 1)\,, \qquad \eta(u,S) = \begin{cases} 0.25, & g_u \notin G(S), \\ 1, & \text{otherwise}, \end{cases} \tag{17}$$

where $G(S) = \{g_v : v \in S\}$. The reduced penalty for an uncovered page limits the suppression of distinct evidence from semantically similar sources. Under the family-mutex constraint, every eligible candidate has an uncovered source page. We set $R(u,S) = 0$ when S is empty or no coverage vectors are provided; individually missing vectors contribute zero overlap. Positive utilities are discounted by $\bar{c}_u^{\beta}$ with $\bar{c}_u = \max(c_u/B, 10^{-3})$, while nonpositive utilities retain their original values, as specified in Equation 5.

Selection procedure. The core selector proceeds as follows:

1. Initialize $S = \emptyset$, spent cost $C(S) = 0$, and an unset reference score p_{ref}.
2. While $|S| < M$, form the eligible set from unselected candidates whose source pages are absent from $G(S)$, whose costs fit $B - C(S)$, and which satisfy Equation 7 when $c_u \geq \tau$.
3. If $|G(S)| \geq \kappa$, also remove candidates with $p_u < \delta p_{\text{ref}}$. Stop if the eligible set is empty.
4. Select the candidate with the largest score in Equation 5. Stop if this score is nonpositive and the page quota has already been reached; otherwise, add the candidate to S.
5. Update $C(S)$ and $G(S)$. On the first addition, set p_{ref} to the selected candidate's relevance score and retain that value for all subsequent steps.

Every addition respects the token budget, unit limit, and source-page mutex. The page quota relaxes early stopping but does not override these restrictions or the expensive-tier admission rule. Consequently, the returned set can cover fewer than κ pages when no further candidate is eligible.

B.2 Selector Hyperparameters

$\alpha = 1.0$, $\gamma = 0.35$, $\beta = 0.20$, budget $B = 6000$, page quota $\kappa = 4$, expensive threshold $\tau = 800$, rank limit $r = 1$, admission share $\sigma = 0.90$. Appendix D.3 reports the sweeps over κ and σ.

B.3 Router Training

Features and scorer. The scorer is a $28 \to 64 \to 32 \to 1$ MLP. Its 18 candidate features comprise retrieval relevance, modality-bias score, normalized cost, six node-type indicators, four hierarchy-level indicators, three active-modality indicators, modality confidence, and normalized node count. Ten query features encode character and word counts, yes/no form, and lexical cues for counting, temporal reasoning, location or enumeration, visual evidence, comparison, explanation, and multiple requested facts. All features are available without reference answers at inference. Costs are normalized as $\tilde{c}_i = c_i/C_0$ with $C_0 = 6000$. The bounded retrieval prior in Equation 9 uses $\omega = 1.5$ during both training and inference.

Label construction. A candidate receives $y_i = 1$ when its source page is annotated as evidence or its text matches an answer alias within the reference document. Matching uses normalized, lowercased strings with punctuation removed. Under the configured combined labeling mode, queries without such candidates use Zeroshot selections as relevance pseudo-labels. Queries that still have no positive candidate are omitted. Within each positively labeled source group, representations are ordered by cost and the first answer-containing one receives $m_i = 1$. Without a text match, the labeler uses an available image for visual evidence, followed by full text when textual evidence is annotated. If neither rule applies, the group has no granularity target. These heuristics provide weak representation preferences for joint training, without requiring answer-model inference or certifying evidence sufficiency.

Loss definitions. For a query with candidate probabilities p_i, the routing and cost losses are

$$\mathcal{L}_{\text{route}}(q) = -\log \sum_i y_i p_i, \qquad \mathcal{L}_{\text{cost}}(q) = \sum_i p_i \tilde{c}_i. \tag{18}$$

Let $n_q^{\min} = \sum_i m_i$ and $\mathcal{P}_q = \{(i,j) : m_i = y_j = 1,\ g_i = g_j,\ c_i < c_j\}$. The granularity losses are

$$\mathcal{L}_{\text{level}}(q) = -\frac{1}{n_q^{\min}} \sum_i m_i \log p_i, \tag{19}$$

$$\mathcal{L}_{\text{margin}}(q) = \frac{1}{|\mathcal{P}_q|} \sum_{(i,j)\in\mathcal{P}_q} [\mu - z_i + z_j]_+, \tag{20}$$

where $[x]_+ = \max(x, 0)$. Each loss is averaged over its applicable queries: the level term excludes queries without granularity targets, and the margin term excludes queries without eligible pairs. An empty set of applicable queries contributes zero. Loss weights are $\lambda_{\text{route}} = 1.0$, $\lambda_{\text{cost}} = 0.15$, $\lambda_{\text{level}} = 0.8$, $\lambda_{\text{margin}} = 0.1$, and $\lambda_{\text{opd}} = 0.35$.

Distillation and optimization. The teacher coefficients in Equation 11 are $(a_y, a_m, a_r, a_c) = (2.0, 1.5, 0.5, 0.75)$, with student temperature $T_s = 1.0$ and teacher temperature $T_{\text{T}} = 0.5$. Each rollout takes at most five actions, removing same-source alternatives and candidates that exceed the remaining budget after each action. Candidate logits remain fixed during the rollout; masking and renormalization determine the policy at each state. Training samples actions from the detached student distribution, whereas validation uses its highest-probability action. Reverse KL is computed over all feasible candidates and averaged over active query–state pairs. Optimization uses Adam with learning rate 10^{-3}, batches of 32 queries, a 15% validation split, and early stopping with patience 30 under a 400-epoch ceiling. Teacher construction and optimization require no LLM calls.

C SUPPLEMENTARY MAIN RESULTS AND ROBUSTNESS

The standalone and mounting summaries below use values in Tables 1 and 2.

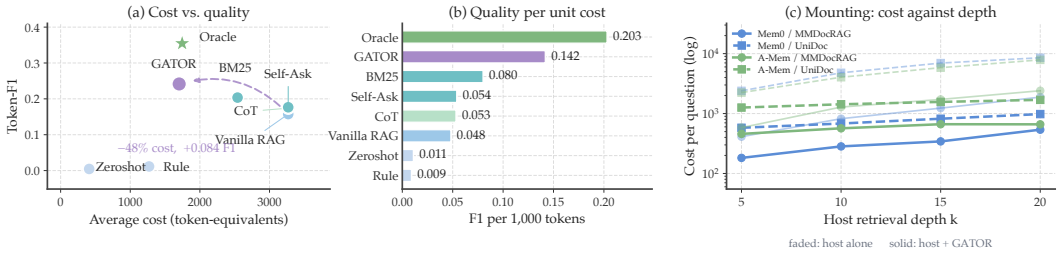


Figure 5: Supplementary views of the main results. (a) Cost versus F1 and (b) F1 per 1,000 tokens, using equal-weight means across the two corpora in Table 1. (c) Host and mounted evidence costs from Table 2; lighter curves denote the hosts.

C.1 MOUNTING ACROSS RETRIEVAL DEPTHS

Figure 6 plots the cost savings in Table 2 over the shared retrieval-depth grid $k \in \{5, 10, 15, 20\}$. Increasing k from 5 to 20 raises savings on UniDoc from 75.8% to 88.5% for Mem0 and from 43.7% to 78.5% for A-Mem. On MMDocRAG, A-Mem savings increase from 21.4% to 72.5%, while Mem0 savings rise from 55.7% to 71.4%, peaking at 72.2% when $k = 15$. Deeper retrieval therefore generally increases the savings from routed evidence selection, although the relationship is not uniformly monotonic.

The trend reflects the different growth rates of host and routed evidence costs. The host forwards all retrieved items, while GATOR selects a subset whose cost grows more slowly over the full sweep. The Mem0/MMDocRAG exception at $k = 20$ illustrates the dependence on the retained evidence: relative to $k = 15$, the average number of selected units increases from 4.4 to 6.5 and routed cost rises from 343 to 538, reducing the saving despite a larger candidate pool. Average cost per retained unit nevertheless falls relative to the host at both depths (5.1% and 11.9%). The two A-Mem/MMDocRAG settings at $k = 5$ and $k = 15$ illustrate the opposite case: unit cost rises by 0.7% and 9.1%, but fewer retained units still reduce total cost by 21.4% and 61.4%. Thus, retrieval depth expands selection opportunities, while realized savings depend jointly on the number and average cost of retained units.

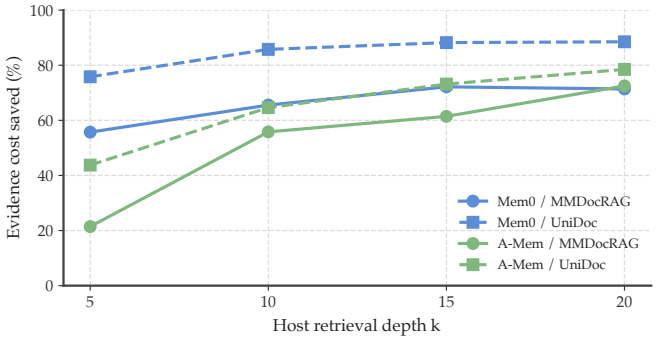


Figure 6: Evidence cost savings across the sixteen mounting configurations in Table 2. Savings generally increase with retrieval depth; Mem0 on MMDocRAG peaks at $k = 15$.

C.2 ANSWER-MODEL ROBUSTNESS

To assess sensitivity to the answer model, we evaluate Qwen3-VL-8B, Qwen3-VL-Flash, and GPT-5.6-Luna on UniDoc and MMDocRAG. Ingestion, retrieval, and routing remain fixed, so each router

supplies identical evidence across backbones. Table 4 reports the complete results, and Figure 3(a) shows judge accuracy; panel (c) jointly shows F1, token efficiency, and judge accuracy. Judge accuracy uses valid verdicts under the protocol in Appendix A.8.

GATOR achieves F1 0.235, 0.253, and 0.270 across the three backbones, exceeding Zeroshot by 0.230–0.240. Judge accuracy likewise improves by 0.330–0.362. With evidence cost fixed at 1,615, F1/1k rises from 0.146 to 0.167 across backbones, indicating that stronger readers extract more answer quality from the same selected evidence. Zeroshot consumes fewer tokens but has lower F1/1k in every setting. Oracle remains ahead in F1 (0.350–0.425), leaving scope to improve evidence selection. These results support the quality–cost advantage of learned routing across the evaluated answer models.

Table 4: Answer-model comparison on UniDoc and MMDocRAG. Each router reads identical selected evidence across the three backbones. Oracle uses gold evidence.

Answer model	Router	Cost ↓	F1 ↑	Judge ↑	F1/1k ↑
Qwen3-VL-8B	Oracle	1721	0.350	0.542	0.203
	Zeroshot	438	0.005	0.017	0.011
	GATOR	1615	0.235	0.347	0.146
Qwen3-VL-Flash	Oracle	1721	0.387	0.622	0.225
	Zeroshot	438	0.017	0.025	0.039
	GATOR	1615	0.253	0.385	0.157
GPT-5.6-Luna	Oracle	1721	0.425	0.721	0.247
	Zeroshot	438	0.030	0.061	0.068
	GATOR	1615	0.270	0.423	0.167

C.3 Cross-Corpus Transfer

Figure 3(b) visualizes the cost–F1 trade-offs in Table 5. Each trained router is evaluated on both corpora without target-corpus adaptation; Zeroshot is evaluated on the same splits. The table additionally reports F1 per 1,000 tokens and gold-page selection coverage.

Table 5: Cross-dataset transfer of GATOR. Each trained router is evaluated on both corpora; Zeroshot provides a training-free reference. Gold page is the fraction of questions for which a gold evidence page is selected.

Train	Test	F1 ↑	F1/1k ↑	Cost ↓	Gold page ↑
MMDocRAG	MMDocRAG	0.187	0.194	965	0.447
	UniDoc	0.158	0.140	1126	0.389
UniDoc	MMDocRAG	0.207	0.109	1894	0.532
	UniDoc	0.296	0.121	2440	0.784
Zeroshot	MMDocRAG	0.003	0.005	643	—
	UniDoc	0.006	0.034	177	—

D Ablations and Diagnostic Studies

We report component removals, representation-level summaries, parameter sensitivity, scorer capacity, and optimization dynamics.

D.1 Component Ablations–Full Results

These ablations assess representation availability and the auxiliary OPD objective within the complete routing pipeline. Since page selection and representation choice share an evidence budget, we rerun selection after removing granularities to capture the resulting changes in evidence allocation. The w/o-OPD variant evaluates the contribution of the distillation objective as a whole, including teacher supervision on learner-visited feasible sets.

Table 6: Granularity and OPD ablations on UniDoc. Cost denotes token-equivalent evidence cost; F1/1k measures answer quality per 1,000 token-equivalents.

Variant	Cost ↓	F1 ↑	Judge ↑	F1/1k ↑
All available granularities	2440	0.296	0.429	0.121
w/o caption	3642	0.088	0.127	0.024
w/o caption and full text	5925	0.062	0.095	0.010
w/o OPD	3553	0.102	0.133	0.029

D.2 Representation and Budget Analysis

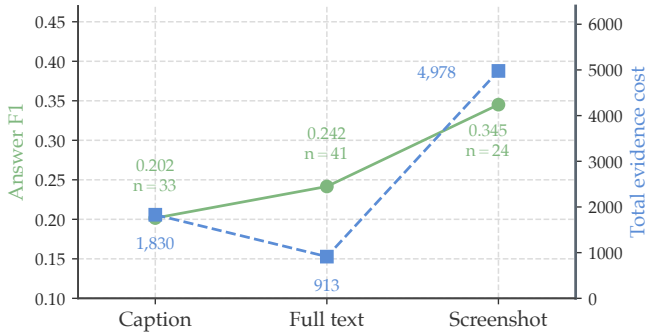


Figure 7: Answer F1 and total evidence cost, grouped by the representation selected on a gold page. Each point summarizes its corresponding question–run observations.

Quality and cost by selected granularity. Figure 7 pools question–run observations from the three granularity arms, retaining those with a selected gold page. Full-text selections have higher observed F1/1k than caption or screenshot selections. Cost denotes the entire selected context, which explains why the caption group can have higher average cost than the full-text group. The comparison therefore describes selection outcomes without isolating the intrinsic quality or unit cost of a representation.

D.3 Selector Sensitivity

Figure 8 examines the page quota κ and expensive-tier admission threshold σ. Larger page quotas generally increase gold-page coverage at a cost to token efficiency, while admission thresholds alter the balance between coverage and expenditure. The response is not uniformly monotonic. Each panel summarizes sensitivity within its respective experimental configuration.

D.4 Scorer Capacity

Table 7 compares scoring architectures while retaining the same features, training objectives, selector, and evaluation protocol. Results are aggregated over seeds 42, 43, and 44 on the document-question subset. A 481-parameter MLP achieves F1 0.180, compared with 0.068 for the linear scorer. Increasing capacity to 14,081 parameters yields F1 0.183 and improves F1/1k from 0.084 to 0.111 through lower evidence cost. Thus, nonlinear scoring substantially improves answer quality in this setting; additional capacity primarily improves efficiency, with modest F1 gains.

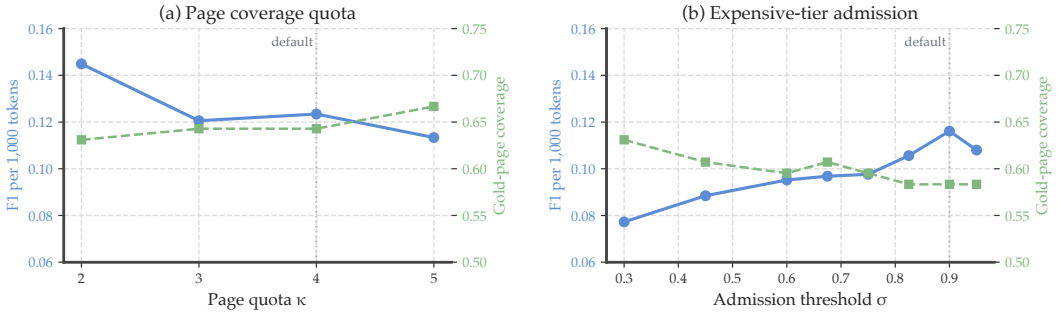


Figure 8: Sensitivity to (a) the page quota κ and (b) the expensive-tier admission threshold σ. Blue curves show F1 per 1,000 tokens; green curves show gold-page coverage. Dotted lines mark the default settings.

Table 7: Scorer capacity on UniDoc and MMDocRAG. Only the network architecture changes; results are mean $\pm$ standard deviation over three seeds.

Scorer	Params	Cost ↓	F1 ↑	Judge ↑	F1/1k ↑
Linear	29	1924 ± 91	0.068 ± 0.001	0.088 ± 0.007	0.035 ± 0.002
MLP (1 hidden)	481	2138 ± 61	0.180 ± 0.012	0.278 ± 0.019	0.084 ± 0.005
MLP (2 hidden)	3969	1716 ± 89	0.179 ± 0.010	0.301 ± 0.028	0.105 ± 0.011
MLP (3 hidden)	14081	1650 ± 101	0.183 ± 0.002	0.316 ± 0.010	0.111 ± 0.006

D.5 Optimization Dynamics

Figure 9 summarizes training and validation losses over three seeds under the optimization settings in Appendix B.3. Curves show the mean and range over the common first 90 epochs, allowing comparison over the same training interval. Both losses decrease as optimization proceeds, with smaller changes in later epochs. The dashed marker denotes the median validation-selected checkpoint at epoch 87. These curves summarize scorer optimization across runs; downstream answer quality is evaluated separately.

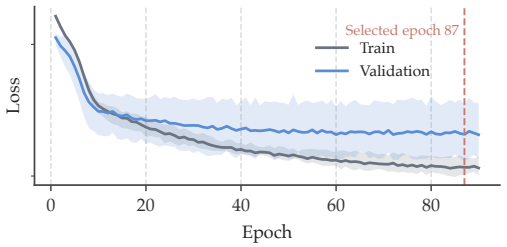


Figure 9: Training and validation losses over three seeds.

E Reliability of the LLM Judge

We assess judge reliability against human labels on a balanced sample from the evaluation runs. For each selected granularity—caption, full text, and screenshot—we sample 50 human-labeled correct answers and 50 incorrect answers, yielding 300 items. Each judge receives the question, reference answer, and candidate answer, with retrieved evidence withheld. We evaluate Qwen3-VL-8B, GPT-5.6-Luna, and Gemini-3.7-Flash on the same items using the shared correctness rubric. Accuracy is computed against human labels for each item, counting both false positives and false negatives as errors; predicted class counts alone can conceal offsetting errors.

Table 8 reports accuracies of 98%, 100%, and 97% for the default Qwen3-VL-8B judge across the three granularities. All three judges achieve at least 96% accuracy in every group, indicating high agreement with human labels across the evaluated representations and judge models.

Table 8: Judge accuracy against human labels on 50 correct and 50 incorrect answers per granularity. Count denotes predicted correct/incorrect answers; Acc. measures item-level agreement with human labels.

Judge	Caption		Full text		Screenshot	
	Count	Acc. (%)	Count	Acc. (%)	Count	Acc. (%)
Human annotation	50 / 50	100	50 / 50	100	50 / 50	100
GPT-5.6-Luna	50 / 50	100	52 / 48	98	49 / 51	99
Gemini-3.7-Flash	53 / 47	97	52 / 48	96	52 / 48	98
Qwen3-VL-8B (ours)	48 / 52	98	50 / 50	100	51 / 49	97

F Prompt Templates

We provide the four templates used for direct answering, chain-of-thought, Self-Ask, and LLM judging. Context and question fields are populated at inference; attached images accompany the selected evidence where applicable.

Answering. Figure 10 is the prompt every router arm shares. Two of its instructions exist to keep the cost comparison honest. The completeness clause ("include every requested list item, comparison point, date, number and unit") prevents a model from scoring well on token-F1 by emitting a short hedge, which would otherwise reward exactly the arms that supply the least evidence. The `UNKNOWN` escape gives a model an explicit way to decline when the evidence is insufficient, rather than forcing it to guess; this is what makes the model's abstention behavior interpretable, and it is how a cost-greedy baseline reveals itself as cheap-but-useless rather than merely inaccurate.

Answer Generation (all routers)

Answer the question using ONLY the provided context and attached images. Return a concise but COMPLETE final answer without a preamble. Include every requested list item, comparison point, date, number and unit. Perform any arithmetic explicitly requested by the question before giving the result. Do not shorten a multi-part answer to only one item. For yes/no questions, answer yes or no and add only the fact needed to disambiguate. If the evidence is genuinely insufficient, answer `UNKNOWN`.

Context: `<selected evidence>`
Question: `<question>`
Answer:

Figure 10: Answer prompt, identical across every routing arm. Only the contents of `<selected evidence>` differ, which is what makes the comparison a test of selection rather than of prompting.

Chain-of-thought. Figure 11 is the prompting baseline of Table 1. It receives identical evidence and differs only in asking for reasoning before the answer. Two details keep it comparable. The answer is delimited by a marker so that it is scored on the same target as every other arm rather than on its reasoning trace; and the token ceiling is raised threefold, because reasoning is emitted before the answer and the standard ceiling would truncate the answer away. The reported evidence cost is unchanged because both arms receive the same context; additional reasoning tokens are recorded separately. This comparison isolates the effect of prompting on answer quality.

Self-Ask. Figure 12 gives the second prompting baseline. It uses the Vanilla RAG evidence and three times the direct-answer output allowance. Only the final answer following the designated marker is scored.

Judging. Figure 13 is the judge of Appendix E. Its accept and reject lists are written to target the specific failure modes of token-F1 in this setting: equivalent wording and reordered lists are accepted because penalising them would measure phrasing rather than correctness, while missing list items and unsupported `UNKNOWN` answers are rejected because accepting them would let an arm

Chain-of-Thought (prompting baseline)

Answer the question using ONLY the provided context and attached images. Think step by step, then give the final answer on a new line beginning with 'Answer:'. The final answer must be concise but COMPLETE: include every requested list item, comparison point, date, number and unit. If the evidence is genuinely insufficient, answer UNKNOWN.

Context: <selected evidence>
Question: <question>
Reasoning:

Figure 11: Chain-of-thought prompt. Only the text after the Answer: marker is scored; generated reasoning tokens are recorded separately from the reported evidence cost.

Self-Ask (prompting baseline)

Answer the question using ONLY the provided context and attached images. First decide whether follow-up questions are needed. If so, write each as 'Follow up: ...' with its 'Intermediate answer: ...'. Then write the final answer on a new line beginning with 'So the final answer is:'. The final answer must be concise but COMPLETE: include every requested list item, comparison point, date, number and unit. If the evidence is genuinely insufficient, answer UNKNOWN.

Context: <selected evidence>
Question: <question>

Figure 12: Self-Ask prompt. The final answer is extracted after the So the final answer is: marker.

score well by answering less. The output is constrained to a single JSON object so that a parse failure is detectable rather than silently scored; a failure is recorded as a judge error and excluded.

LLM Judge

You are an answer evaluator. Determine whether the candidate answer is substantively correct for the question given the reference answer. Accept equivalent wording, reordered lists, harmless extra units, and normalized date formats. Reject contradictions, wrong numbers, missing required list items, and unsupported UNKNOWN answers. Output exactly one JSON object and nothing else using this schema: {"correct": true, "score": 0.0, "reason": "brief reason"}. The score must be between 0 and 1.

Question: <question>
Reference answer: <gold answer>
Candidate answer: <model answer>

Figure 13: Judge prompt. The accept/reject lists encode which token-F1 failure modes the judge is meant to correct for.

G Extended Related Work

Granularity-aware retrieval encompasses several decisions: selecting the extent of a text span, choosing a modality-specific corpus, localizing a document region, or determining how much detail to expose to a reader. We distinguish these decisions from selecting among differently priced representations of an already retrieved source page.

Granularity selection in text-based RAG. RAPTOR (Sarthi et al., 2024) recursively clusters and summarizes text chunks into a retrieval tree, allowing evidence to be retrieved at different levels of abstraction; its collapsed-tree strategy also supports a token budget. Mix-of-Granularity (MoG) (Zhong et al., 2025) constructs multiple chunking levels and learns a query-dependent router that combines retrieval signals across these levels to select an appropriate textual extent. ACC-RAG (Guo et al., 2025a) addresses inference efficiency through hierarchical context compression: a trained compressor produces representations at multiple compression levels, and an adaptive selector progressively supplies compressed embeddings until the decoder has sufficient information. Its selection policy uses decoder hidden states to assess whether further context is needed. *These methods establish the value of adaptive granularity, while* `GATOR` *targets a different deployment interface: caption, text, and image representations remain directly consumable by an existing multimodal reader. It allocates an explicit evidence budget across source pages without rebuilding a retrieval hierarchy or accessing decoder hidden states. Adapting these text-based methods to our setting would require multimodal representations and, depending on the method, changes to the index or reader interface beyond the post-retrieval selection policy.*

Granularity selection in multimodal RAG. UniversalRAG (Yeo et al., 2026) routes queries to appropriate modality–granularity combinations across heterogeneous corpora before performing specialized retrieval. OMGM (Yang et al., 2025) uses a coarse-to-fine pipeline for knowledge-based visual question answering: an initial entity search is followed by multimodal reranking and fine-grained section selection. These approaches determine where to search and which evidence to retrieve. LFRAG (Zhu et al., 2026) instead localizes evidence within documents, combining layout-based block construction with semantic–layout representations and block-level retrieval to reduce the context passed to generation. MAGE-RAG (Zuo et al., 2026) constructs a graph of pages and document elements, then uses an iterative controller to activate, open, search for, or prune evidence. Its budgets constrain page entries, activation actions, controller iterations, and the final set of opened nodes. LFRAG and MAGE-RAG thus explicitly pursue efficient evidence use, through finer retrieval units and adaptive graph exploration, respectively. `GATOR` *offers a complementary cost-control interface: it selects among available representations after retrieval, jointly accounting for reading cost, redundancy, and page coverage without additional LLM calls for routing. This allows deployment on existing memory systems while preserving their retrievers. We therefore do not treat the above complete pipelines as drop-in controls in our fixed-host mounting experiments: they also change corpus routing, indexing, retrieval, or evidence expansion. Comparing their overall efficiency would require accounting for these stages and any controller calls alongside reader cost; our evaluation focuses on the cost and quality of the selected answer context.*