Reflect and Retrieve: Multimodal Agentic Retrieval with Reflection and Query Agents for Document Search

Tryambak Gangopadhyay, Md Sirajus Salekin, Yiyue Qian,
Nkechinyere Agu, Shinan Zhang, Rinne Han
Amazon Web Services

Abstract

Retrieving relevant pages from visually rich documents, with evidence in charts, tables, and figures rather than body text, remains difficult: a single-pass query against a fixed index misses content in visual elements, and reordering the same candidates cannot recover a page never retrieved. We propose Reflect and Retrieve, an agentic retrieval approach centered on a Reflection Agent. After an initial page-level retrieval, the agent uses a multimodal large language model to visually reason over the retrieved page images, judge whether the evidence is sufficient, and, when gaps remain, an orchestrator issues targeted follow-up retrievals to recover the missing pages. Evaluating on two benchmarks, MMDocIR and ViDoRe V3, the Reflection Agent delivers the dominant agentic gain: it is the only configuration to beat single-step retrieval for every reasoning model on both, improving Recall@N from 0.628 to 0.663 on MMDocIR and 0.566 to 0.644 on ViDoRe V3. The improvement comes from recovering relevant pages absent in initial retrieval. Our approach indicates a clear design order: selecting an optimal embedding approach for the retrieval tool is the first step, after which the Reflection Agent contributes more value than a query decomposition agent, positioning visual reflection as the key component for document search.

1 Introduction

Real-world documents convey information through charts, tables, diagrams, and page layout as much as through text: financial reports present results in bar charts, research papers embed findings in figures, and contracts organize obligations in tables. Retrieval-Augmented Generation (RAG) systems that rely on extracted text (Lewis et al., 2020) lose this visual context. Page-level retrieval, which returns the most relevant pages for a query, is a natural granularity that preserves the full multimodal content of each page (Dong et al., 2025a; Faysse et al., 2025).

Recent work makes substantial progress here. ColPali (Faysse et al., 2025) introduces late-interaction retrieval directly from page images using a vision-language model (VLM), and VisRAG (Yu et al., 2025) reports 20 to 40% end-to-end gains over text-only RAG by embedding pages as images; other approaches embed VLM-generated page descriptions (Dong et al., 2025a; Ma et al., 2024). All treat retrieval as a *single-step* operation: embed the query, return the top-K, and stop. This one-shot paradigm caps performance: our best single-step configuration reaches only 0.628 Recall@N on MMDocIR and 0.566 on ViDoRe V3, with no mechanism to detect or recover from retrieval failures.

In parallel, the agentic RAG literature shows that iterative, self-correcting retrieval improves accuracy in text-only settings: Self-RAG (Asai et al., 2024) decides when to retrieve and self-critiques via reflection tokens, Corrective RAG (Yan et al., 2024) adds a retrieval evaluator that triggers corrective actions when confidence is low, Adaptive-RAG (Jeong et al., 2024) routes queries by complexity, and a recent survey (Singh et al., 2025) identifies reflection, planning, tool use, and multi-agent collaboration as core patterns. These methods operate exclusively on text and cannot inspect a retrieved chart, table, or diagram to judge its relevance, a fundamental limitation for multimodal documents.

We propose **REFLECT AND RETRIEVE**, an agentic retrieval approach for multimodal page-level document search with three components: (1) a *retrieval tool*, whose optimal embedding strategy and VLM we first identify; (2) a *Query Agent* that decomposes or refines queries before retrieval; and (3) a multimodal *Reflection Agent* that visually inspects the retrieved page images and decides whether to accept them or issue targeted follow-up

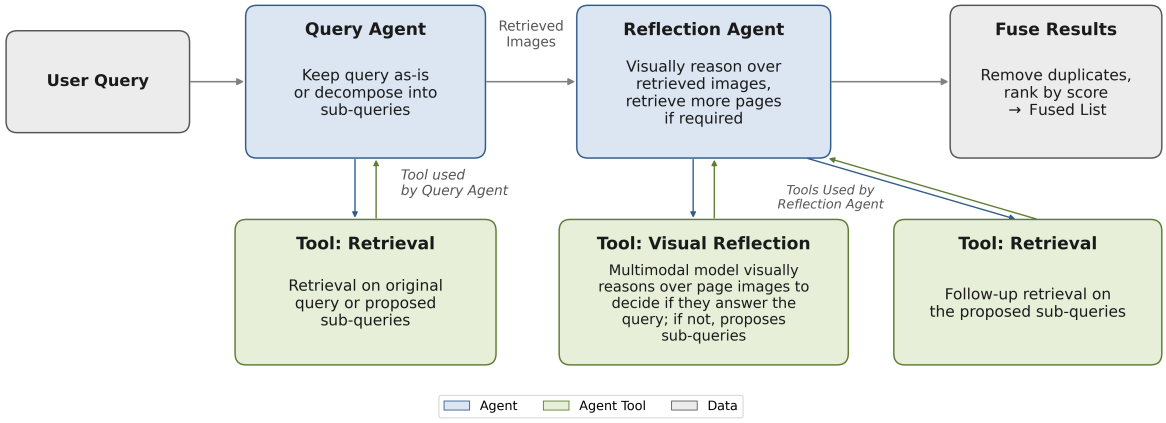


Figure 1: Reflect and Retrieve workflow centered on Reflection Agent. The Query Agent first keeps the user query as-is or decomposes it and retrieves an initial set of page images, which are sent to the Reflection Agent. The reflection agent calls a `visual reflection` tool, which uses a multimodal model to visually reason over the page images to decide whether they answer the query and, if not, proposes sub-queries; a `retrieval` tool then issues follow-up retrievals on those sub-queries. All retrieved items are fused by removing duplicates and ranking by score into the final fused list.

retrievals. The Reflection Agent is the key differentiator: by *seeing* the retrieved pages rather than only reading their text, it detects content gaps that text-only evaluation misses. Across enterprise document-search deployments in hospitality, security services, finance, and healthcare, first-pass retrieval repeatedly missed evidence inside charts, tables, and drawings that reranking cannot recover. Reflect and Retrieve emerged from those engagements: optimize the embedding strategy first, then add agentic retrieval with reflection and query agents. Our contributions are threefold: (1) a controlled comparison of five embedding configurations on two benchmarks to find optimal retrieval tool settings, establishing that VLM-generated descriptions outperform image-only embeddings by +16.7 to +19.7 Recall points; (2) the REFLECT AND RETRIEVE architecture, which introduces a multimodal Reflection Agent that delivers consistent gains (+3.4 to +7.9 points) on both benchmarks while query decomposition helps conditionally; and 3) a systematic study of agentic retrieval across five backbone LLMs on MMDocIR (1,658 queries) and ViDoRe V3 (1,487 queries) that yields practical guidance on when each agent helps, with a design distilled from production multimodal-search systems we built for enterprise customers.

2 Related Work

Multimodal Document Retrieval: Traditional retrieval discards visual signals by operating on extracted text (Lewis et al., 2020). Recent methods embed pages directly as images: ColPali (Faysse et al., 2025) adapts ColBERT (Khattab and Zaharia, 2020) late-interaction to page images, DSE (Ma et al., 2024) uses single-vector screenshot embeddings, VLM2Vec (Jiang et al., 2025) trains VLMs as multimodal encoders, and VisRAG (Yu et al., 2025) reports 20 to 40% gains over text-only RAG. Alternatively, embedding VLM-generated page descriptions yields strong results (Dong et al., 2025a), with extensions to cross-modal and multi-page settings (Li et al., 2025; Cho et al., 2025; Dong et al., 2025b). All treat retrieval as single-step with no self-correction mechanism.

Agentic and Iterative Retrieval: A growing body of work makes retrieval adaptive and self-correcting. Self-RAG (Asai et al., 2024) uses reflection tokens to decide when to retrieve and to assess relevance; Corrective RAG (Yan et al., 2024) adds a retrieval evaluator that triggers corrective actions when confidence is low; Adaptive-RAG (Jeong et al., 2024) routes queries by complexity; and Auto-RAG (Yu et al., 2024) plans multi-turn retrieval autonomously. On the query side, HyDE (Gao et al., 2023) retrieves with a hypothetical answer document, and RAPTOR (Sarthi et al., 2024) builds recursive passage trees. A recent survey (Singh et al., 2025) distils four agentic patterns: reflection, planning, tool use, and multi-agent collaboration. All of these operate in the text domain and cannot inspect charts, tables, or diagrams. REFLECT AND RETRIEVE bridges this

gap with a multimodal Reflection Agent that judges relevance from the page image itself, for example verifying that a retrieved financial page actually contains the queried chart rather than merely sharing keywords, a capability no prior agentic retrieval system provides.

3 Method: Reflect and Retrieve

REFLECT AND RETRIEVE places agents on top of a shared retrieval tool backed by a dense vector index. We describe the tool, then the two agents and their composition.

3.1 Retrieval Tool

The retrieval tool performs a single-step nearest neighbour search over a dense vector index. Its central design variable is the modality of the indexed page representation, governed by two dimensions: (i) the VLM used to generate a per-page textual description, and (ii) the embedding strategy applied to each page.

Each page is described by a VLM in up to 1,600 tokens using a structured prompt (Appendix A.2). We compare **Claude Sonnet 4.5** and **Qwen3-VL 235B**, both receiving the same prompt. We derive three variants using Amazon Nova Multimodal Embeddings (1,024-d): **Image-Only** from the raw page image; **Desc.-Only** from the VLM-generated description; and **Image+Desc.** from both jointly.

When the retrieval tool is called multiple times (because the query was decomposed or because the Reflection Agent issued follow-up retrievals), the returned lists are *fused*: concatenated, sorted by descending cosine similarity score, and deduplicated by page identity, yielding a single ranked list of up to N unique pages. Each retrieval call returns its top $k = 10$ pages; we use this cutoff in every scenario.

3.2 Reflection Agent

The Reflection Agent is the core of our approach (Figure 1). It operates downstream of an initial retrieval and performs *multimodal* reasoning over the retrieved page images (the standalone reflection workflow is shown in Appendix Figure 4). It comprises two components: an **orchestrator** agent that coordinates the workflow, and a `visual_reflection` tool, backed by a multimodal LLM, that inspects page images and returns a retrieval adequacy verdict.

Initial retrieval first produces a ranked list of k page images. The orchestrator passes these images together with the user query to the `visual_reflection` tool, which visually reasons over the rendered pages and returns either an *adequate* verdict, indicating the set sufficiently covers the query and triggering immediate output, or a *not adequate* verdict accompanied by up to two refined sub-queries that target the identified gaps. In the latter case the orchestrator issues follow-up retrieval calls on the suggested sub-queries, and all items (initial retrieval plus follow-ups) are fused and ranked by score.

By reasoning over rendered page images along with the user query, the reflection agent detects visual gaps, for example a chart referenced in the query but absent from the retrieved pages, and formulates targeted follow-up queries to recover the missing pages. To bound latency and cost, the agent is limited to at most two `visual_reflection` calls and two follow-up retrieval calls per question. We decouple the orchestrator (fixed to Claude Haiku 4.5 for efficiency) from the `visual_reflection` model (swept across four multimodal LLMs of varying capacity). We provide the orchestrator and `visual_reflection` prompts in Appendix.

3.3 Query Agent

The Query Agent is a text-only LLM that uses the retrieval tool. For each user query it chooses one of two actions: (A) *pass-through*, issuing the query unchanged as a single retrieval call, or (B) *decomposition*, splitting a multi-faceted query into at most two sub-queries and issuing one retrieval call each. With $m \leq 2$ (sub-)queries the fused list holds up to $N = m \times k$ items before deduplication. The agent only retrieves: it never answers from its own knowledge, rephrases beyond decomposition, or summarizes the returned pages. We evaluate it by sweeping the backbone LLM across five models of varying capacity; The query agent prompt appear in Appendix.

3.4 Query and Reflection Agent

In the most agentic configuration, the two agents are composed sequentially: the Query Agent first rewrites and potentially decomposes the query to produce a seed retrieval set, and the Reflection Agent then reviews and augments that set through visual inspection (Figure 1). This configuration tests whether query decomposition provides gains

complementary to the reflection loop.

4 Experimental Setup

4.1 Datasets

We evaluate on two page-level visual document retrieval benchmarks; each question references one or more ground-truth (GT) page images, and a retrieval is scored by how well its returned page set covers those GT pages. **MMDocIR** (Dong et al., 2025a) provides 1,658 questions over 313 source documents spanning 10 professional domains (academic papers, financial reports, news, laws, guidebooks, and others). 81% of questions have exactly one GT page (mean 1.27), making it primarily a "find the single right page" task. **ViDoRe V3** (Loison et al., 2026) is a page-level benchmark with human-verified annotations spanning 5 enterprise domains (finance, HR, industrial, pharmaceuticals, computer science). It ships each query in six languages; we use the English subset of 1,487 queries over 96 source documents and 12,968 pages. Its GT density is roughly four times that of MMDocIR (mean 5.09 pages per question; only 25% of questions are single page), making it a multi-page retrieval challenge. Per-domain question and page counts for both benchmarks are given in Tables 5 and 6.

4.2 Evaluation Metrics

All scenarios are scored on the fused, deduplicated ranked list of page images, formed by concatenating the per-query retrieval blocks, sorting by descending k-NN cosine similarity, and keeping the first occurrence of each page. For single step retrieval no fusion is needed, so the list has length k; agentic scenarios add pages, so the realized list length N can exceed k. We report six standard information retrieval metrics computed against this list: *Recall@N* (primary), *Precision@N*, *F1@N*, *NDCG@N*, *MAP*, and *MRR*, all at k=10 in the main body. We use k=10 because ViDoRe V3 averages 5.09 ground-truth pages per question; MMDocIR (mean 1.27 ground-truth pages) is unaffected by this choice, and a common k keeps the two benchmarks directly comparable. Unless stated otherwise, a reported value is the *micro* (per-question) average, computed per question then averaged with equal weight over all questions. Because the fused list length grows for agentic scenarios, we report the average fused N alongside *Recall@N*.

4.3 Models and Implementation

All indexes use Amazon Nova Multimodal Embeddings at 1,024 dimensions over OpenSearch Serverless with an HNSW index (cosine similarity). Per-page descriptions are generated by one of two VLMs, Claude Sonnet 4.5 and Qwen3-VL 235B. For the agentic scenarios we sweep five backbone LLMs for the Query Agent: Claude Haiku 4.5, Claude Sonnet 4.6, Claude Opus 4.8, Qwen3-VL 235B, and Qwen3-Next 80B. For the Reflection Agent the orchestrator is fixed to Claude Haiku 4.5 and the `visual_reflection` model is swept across four multimodal models: Haiku 4.5, Sonnet 4.6, Opus 4.8, and Qwen3-VL 235B. We fix the per-call retrieval cutoff at k=10 and cap the Query Agent at two sub-queries and the Reflection Agent at two `visual_reflection` calls and two follow-up retrievals.

5 Results

5.1 Retrieval Tool Optimization (Single-Step Retrieval)

We first evaluate the five embedding configurations in single-step retrieval mode, where each query is embedded once and the top-k nearest neighbours are returned directly. This identifies the optimal retrieval tool that agentic workflows will use.

Table 1 presents the single-step retrieval results. The most striking finding is the dominance of text based embeddings: Desc.-Only under Claude Sonnet 4.5 outperforms Image-Only by 16.7 Recall points on MMDocIR (0.461 to 0.628) and 19.7 points on ViDoRe V3 (0.369 to 0.566), the largest single gain in the entire pipeline. Fusing image and text signals does not help: Image+Desc. trails Desc.-Only by 3.0 (MMDocIR) and 4.7 (ViDoRe) Recall points, suggesting that averaging the two vectors at 1,024 dimensions dilutes the text signal rather than complementing it. VLM quality also proves consequential: Claude Sonnet 4.5 descriptions yield 3.8 (MMDocIR) and 5.0 (ViDoRe) Recall points over Qwen3-VL 235B under identical conditions, a gap attributable solely to description quality, since Image-Only does not depend on the VLM. The three embedding strategies are illustrated in Appendix (Figure 2), and the full per-metric results are repeated in Appendix (Table 7). Based on these results, we fix the Desc.-Only (Claude Sonnet 4.5) index as the retrieval tool for all subsequent agentic experiments.

VLM	Strategy	R@N	NDCG	MAP	MRR
MMDocIR (N=1,658)					
N/A	Image-Only	0.461	0.342	0.297	0.317
Claude Sonnet 4.5	Desc.-Only	**0.628**	**0.505**	**0.455**	**0.486**
	Image+Desc.	0.598	0.478	0.429	0.456
Qwen3-VL 235B	Desc.-Only	0.591	0.470	0.423	0.449
	Image+Desc.	0.595	0.469	0.420	0.447
ViDoRe V3 (N=1,487)					
N/A	Image-Only	0.369	0.321	0.231	0.417
Claude Sonnet 4.5	Desc.-Only	**0.566**	**0.511**	**0.399**	**0.624**
	Image+Desc.	0.519	0.462	0.352	0.572
Qwen3-VL 235B	Desc.-Only	0.516	0.463	0.349	0.588
	Image+Desc.	0.511	0.450	0.338	0.562

Table 1: Single-step retrieval *Recall@N*. **Bold** = best per benchmark. Desc.-Only with Claude Sonnet 4.5 is optimal embedding approach and used in the retrieval tool for agentic retrieval experiments.

visual_reflection	R@N	NDCG	MAP	MRR	Fused N	Cost/q ($)	Time/q (s)
MMDocIR (baseline R@N = 0.628)							
Sonnet 4.6	**0.663**	**0.521**	**0.467**	**0.495**	13.25	0.076	30.3
Opus 4.8	0.659	0.519	0.465	0.492	12.90	0.093	33.5
Qwen3-VL 235B	0.653	0.508	0.452	0.481	13.43	0.044	27.7
Haiku 4.5	0.650	0.505	0.450	0.478	13.34	0.048	24.1
ViDoRe V3 (baseline R@N = 0.566)							
Opus 4.8	**0.644**	0.496	0.364	0.571	18.73	0.080	37.7
Qwen3-VL 235B	0.632	0.490	0.361	0.568	18.12	0.039	34.1
Sonnet 4.6	0.608	**0.503**	**0.379**	**0.596**	15.74	0.055	36.0
Haiku 4.5	0.601	0.498	0.376	0.587	14.69	0.036	22.0

Table 2: Reflection Agent *Recall@N*, with rank-aware metrics (NDCG@N, MAP, MRR) and per-query cost and latency. The orchestrator is fixed to Haiku 4.5; the `visual_reflection` model is swept. Fused N is the average deduplicated list length after follow-up retrievals. **Bold** = best per benchmark (Recall@N determines the leading configuration; the best value in each rank-aware column is also bolded). Per-domain breakdowns are in Appendix Tables 13 and 14.

5.2 Reflection Agent

Having fixed Desc.-Only (Claude Sonnet 4.5) as the optimal embedding strategy (Section 5.1), we evaluate the Reflection Agent on top of it. The orchestrator is fixed to Claude Haiku 4.5 and the `visual_reflection` model is swept across four multimodal models; all configurations share the same index and k=10, so the only variable is the visual reasoning model.

Table 2 reports the results. Recall@N is our primary selection metric; the leading `visual_reflection` model per benchmark is therefore Sonnet 4.6 on MMDocIR and Opus 4.8 on ViDoRe V3. We report NDCG@N, MAP, and MRR alongside for completeness. On ViDoRe V3 these rank-aware metrics favor Sonnet 4.6, but we select Opus 4.8 on the basis of the primary Recall@N metric. On **both** benchmarks, *every* `visual_reflection` model beats the single-step baseline. On MMDocIR the gains range from +0.022 (Haiku 4.5) to +0.034 (Sonnet 4.6); on ViDoRe V3 they range from +0.035 (Haiku 4.5) to +0.079 (Opus 4.8). This is the only configuration that reliably exceeds the non-agentic baseline for every reasoning model on both benchmarks.

The improvement comes from *recovering relevant pages the first pass misses*. When the visual evaluator judges the seed set inadequate, it issues targeted follow-up retrievals that surface ground-truth pages absent from the original top-10, expanding the fused list to 12.90–13.43 pages on average on MMDocIR and 14.69-18.73 on ViDoRe V3. The gain is not merely an effect of the longer list: for example, single-step retrieval at k=13 reaches only 0.648 on MMDocIR, still below the reflection agent's 0.663 at a comparable

fused length of 13 pages. The gain scales with how effectively each evaluator drives this recovery: the two strongest models on ViDoRe (Opus 4.8 and Qwen3-VL 235B) surface the most relevant pages (fused N near 18 to 19). The visual modality is essential: inspecting page images lets the agent detect missing charts, tables, and figures that text-only query rewriting cannot reason about.

The advantage of reflection is not confined to a small retrieval budget. At a larger budget of k=30, where single-step already recovers substantially more pages, adding the Reflection Agent on the same retrieval tool still improves Recall on both benchmarks (0.7095 to 0.7424 on MMDocIR and 0.7075 to 0.7756 on ViDoRe V3; Appendix Table 15), confirming the gain is not an artifact of the small-k regime.

Table 2 reports the per-query cost and latency of each configuration: the Reflection Agent adds one to two multimodal calls, costing $0.036 to $0.093 and 22 to 38 seconds per query. The best evaluator per benchmark is not always the most expensive one: on MMDocIR, Sonnet 4.6 gives the top Recall at $0.076/query, below Opus 4.8 ($0.093), and the cheaper Qwen3-VL and Haiku evaluators retain most of the gain at roughly half the cost while also being faster. On ViDoRe V3, Qwen3-VL 235B achieves the second-best Recall (0.632, close to Opus 4.8's 0.644) while retaining most of the benefit at roughly half the cost ($0.039 vs $0.080). The recall-cost-latency trade-off, not recall alone, can guide the choice of visual reasoning model in deployment.

5.3 Query and Reflection Agent

We next study whether adding a Query Agent in front of the reflection loop yields gains complementary to reflection. On its own, the Query Agent helps only conditionally: on ViDoRe V3, where questions average 5.09 ground-truth pages, Claude Opus 4.8 is the one backbone to beat single-step (0.585, +0.020) by decomposing multi-faceted queries into genuinely separable facets; on MMDocIR, where 81% of questions have a single ground-truth page, no backbone beats the baseline and recall falls as decomposition rises. Full Query-Agent-only results for both benchmarks are reported in Appendix (Tables 11 and 12). We therefore place the Query Agent in front of the reflection loop with a dataset-appropriate backbone. On ViDoRe V3 we use Opus 4.8, its only backbone to beat single-step. On MMDocIR the choice is more deliberate:

Configuration	R@N	Fused N
MMDocIR		
Reflection (vr Sonnet 4.6)	0.663	13.25
Query (QA Sonnet 4.6) + Reflection (vr Sonnet 4.6)	0.653	14.43
ViDoRe V3		
Reflection (vr Opus 4.8)	0.644	18.73
Query (QA Opus 4.8) + Reflection (vr Opus 4.8)	**0.651**	20.51

Table 3: Query and Reflection Agent *Recall@N*, compared against the best reflection-alone configuration. QA denotes the Query Agent backbone. The Query Agent and the `visual_reflection` model are both set to the dataset-matched best (Sonnet 4.6 on MMDocIR, Opus 4.8 on ViDoRe V3); the orchestrator is Haiku 4.5. **Bold** = best per benchmark.

the models that match the single-step baseline (the Qwen backbones) do so precisely because they almost never decompose (Appendix Table 10), leaving the query as-is, so pairing them with reflection would simply reproduce reflection-alone and tell us nothing. We therefore use Sonnet 4.6, the Claude backbone that decomposes most often (22.6% of queries), which provides a genuine test of whether feeding the reflection loop a heavily decomposed seed set helps on this single-page benchmark.

Table 3 shows that composing the two agents does not improve substantially on reflection agent alone. On MMDocIR, the Query Agent feeds a more decomposed seed set into the reflection loop but reaches 0.653, just below reflection-alone (0.663): the extra decomposition does not help when most questions have a single relevant page. On ViDoRe V3, the composed pipeline reaches 0.651, marginally above reflection-alone (0.644, +0.007) and above the Query Agent alone (0.585), since the dataset rewards both decomposition and visual recovery. In both cases the reflection loop accounts for nearly all of the agentic gain, and the Query Agent contributes at most a small increment. The practical implication is that the Reflection Agent is the component to invest in; a query decomposition layer is optional and helps only when the queries have multi-faceted information needs.

Table 4 traces the cumulative gains. Two design choices compound. First, optimizing the embedding turns a weak image-only index into a strong retrieval tool (+0.167 on MMDocIR, +0.197 on ViDoRe V3). Second, on top of this optimized tool the Reflection Agent recovers pages the first

Stage	Best config.	R@N
MMDocIR		
Single-step (Image-Only)	N/A	0.461
Single-step (Desc.-Only)	Claude Sonnet 4.5	0.628
Query Agent	Qwen3-VL/Next	0.628
Reflection Agent	**vr Sonnet 4.6**	**0.663**
Query + Reflection	QA Sonnet 4.6, vr Sonnet 4.6	0.653
ViDoRe V3		
Single-step (Image-Only)	N/A	0.369
Single-step (Desc.-Only)	Claude Sonnet 4.5	0.566
Query Agent	Opus 4.8	0.585
Reflection Agent	vr Opus 4.8	0.644
Query + Reflection	**QA Opus 4.8, vr Opus 4.8**	**0.651**

Table 4: Progression from the weakest baseline to the best configuration on each benchmark, reporting *Recall@N*. For Query + Reflection, QA denotes the Query Agent backbone (the `visual_reflection` model is the dataset-best, as in Table 3). Optimizing the embedding (the Desc.-Only index lifts Recall by +0.167 on MMDocIR and +0.197 on ViDoRe V3 over Image-Only); on top of this optimized retrieval tool, the Reflection Agent adds a further, consistent gain on both benchmarks, while the additional Query Agent helps conditionally.

pass still misses, and improves Recall for every reasoning model on both benchmarks; the additional Query Agent contributes a smaller, conditional gain. The full REFLECT AND RETRIEVE pipeline therefore pairs the Reflection Agent with an optimized retrieval tool, and a Query Agent in front.

6 Conclusion

We present REFLECT AND RETRIEVE, an agentic retrieval approach for multimodal document search built around a multimodal Reflection Agent that visually reasons over retrieved page images and recovers content the first pass misses. Through systematic evaluation on two page-level benchmarks (MMDocIR and ViDoRe V3), we identify three actionable findings for practitioners deploying multimodal retrieval systems. First, *the retrieval tool must be optimized first*: converting document pages to VLM-generated text descriptions before embedding yields +16.7 to +19.7 Recall points over image-only embeddings, and higher-quality VLMs produce better descriptions that directly improve retrieval. Second, the *Reflection Agent* on top of the optimized retrieval tool, improves Recall (by +0.034 on MMDocIR and +0.079 on ViDoRe V3) and is the only configuration to beat single-step retrieval for every reasoning model on both benchmarks, by visually inspecting retrieved pages and issuing targeted follow-up retrievals for missing content. Single-step retrieval has no way to recognize or repair an incomplete result set, whereas the Reflection Agent consistently recovers relevant pages it leaves behind. Third, *the Query Agent is a complementary addition*: query decomposition helps when the corpus has multi-page ground truths requiring sub-queries, and adds a smaller increment on top of the reflection loop. Together, REFLECT AND RETRIEVE uses a Reflection Agent with an optimized retrieval tool, with an optional Query Agent in front, and we recommend adopting them in that order.

Limitations

Our evaluation uses English-language documents for a controlled comparison. The pipeline itself is language-agnostic: the query and reflection agents and the visual reflection tool are not English-specific, and the same setup applies by indexing documents in other languages; we leave multilingual evaluation to future work. We have not explored alternative fusion strategies (for example, reciprocal rank fusion or learned re-ranking) or late-interaction retrieval models that may further improve performance of the retrieval tool. All experiments use a single embedding model (Amazon Nova Multimodal Embeddings at 1,024 dimensions); results may differ with other embedding providers or higher-dimensional representations.

Ethical Considerations

All experiments use publicly available benchmarks (MMDocIR, ViDoRe V3) with no personally identifiable information. The quantitative results reported here use offline public benchmarks (MMDocIR, ViDoRe V3) without live user traffic; the architecture itself is derived from production multimodal-search deployments, and the offline protocol is chosen for reproducibility on non-proprietary data. We acknowledge that VLM-generated descriptions may introduce biases present in the underlying language models; however, since descriptions are used solely as embedding inputs (not surfaced to end users), the risk of downstream harm is limited in this context.

References

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

Jaemin Cho, Debanjan Yoon, Minhyuk Ji, Seonghyeon Lee, Sungdong Yoon, and Minjoon Seo. 2025. M3docvqa: Multi-modal multi-page multi-document understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*.

Kuicai Dong, Yujing Chang, Derrick Goh Xin Deik, Dexun Li, Ruiming Tang, and Yong Liu. 2025a. MMDocIR: Benchmarking multimodal retrieval for long documents. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 30971–31005, Suzhou, China. Association for Computational Linguistics.

Kuicai Dong, Yujing Chang, Shijie Huang, Yasheng Wang, Ruiming Tang, and Yong Liu. 2025b. Benchmarking retrieval-augmented multimodal generation for document question answering. In *Advances in Neural Information Processing Systems (NeurIPS)*.

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

Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. 2023. Precise zero-shot dense retrieval without relevance labels. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL)*.

Soyeong Jeong, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, and Jong C. Park. 2024. 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)*.

Ziyan Jiang, Rui Meng, Xinyi Yang, Semih Yavuz, Yingbo Zhou, and Wenhu Chen. 2025. Vlm2vec: Training vision-language models for massive multimodal embedding tasks. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 39–48.

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. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Zirui Li, Siwei Wu, Yizhi Li, Xingyu Wang, Yi Zhou, and Chenghua Lin. 2025. Docmmir: A framework for document multi-modal information retrieval. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 13117–13130.

António Loison, Quentin Macé, Antoine Edy, Victor Xing, Tom Balough, Gabriel Moreira, Bo Liu, Manuel Faysse, Céline Hudelot, and Gautier Viaud. 2026. Vidore v3: A comprehensive evaluation of retrieval augmented generation in complex real-world scenarios. *arXiv preprint arXiv:2601.08620*.

Xueguang Ma, Sheng-Chieh Lin, Minghan Li, Wenhu Chen, and Jimmy Lin. 2024. Unifying multimodal retrieval via document screenshot embedding. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 6492–6505.

Parth Sarthi, Salman Abdullah, Aditi Tuli, Shubh Khanna, Anna Goldie, and Christopher D. Manning. 2024. Raptor: Recursive abstractive processing for tree-organized retrieval. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Aditi Singh, Abul Ehtesham, Saket Kumar, Tala Talaei Khoei, and Athanasios V. Vasilakos. 2025. Agentic retrieval-augmented generation: A survey on agentic rag. *arXiv preprint arXiv:2501.09136*.

Shi-Qi Yan, Jia-Chen Gu, Yun Zhu, and Zhen-Hua Ling. 2024. Corrective retrieval augmented generation. *arXiv preprint arXiv:2401.15884*.

Shi Yu, Chaoyue Tang, Bokai Xu, Junbo Cui, Junhao Ran, Yukun Yan, Zhenghao Liu, Shuo Wang, Xu Han, Zhiyuan Liu, and Maosong Sun. 2025. Visrag: Vision-based retrieval-augmented generation on multi-modality documents. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Tian Yu, Shaolei Zhang, and Yang Feng. 2024. Auto-rag: Autonomous retrieval-augmented generation for large language models. *arXiv preprint arXiv:2411.19443*.

A Appendix

This appendix provides dataset statistics (§A.1), the embedding strategies and VLM prompt (§A.2), full single-step retrieval metrics and per-domain results (§A.3), and the agentic workflows, prompts, and per-domain results for the Query and Reflection agents (§A.4–A.5).

A.1 Dataset Statistics

Tables 5 and 6 report the per-domain question and page counts for MMDocIR and the ViDoRe V3 English subset. MMDocIR is dominated by Academic Paper, Financial Report, and Research Report (about 56% of questions) and ViDoRe V3 is more evenly balanced across its five enterprise domains.

Domain	**#Q**	**%**
Academic Paper	389	23.5
Financial Report	344	20.7
Research Report / Intro	194	11.7
News	137	8.3
Laws	132	8.0
Guidebook	115	6.9
Government	111	6.7
Tutorial/Workshop	104	6.3
Brochure	76	4.6
Admin./Industry	56	3.4
Total	**1,658**	100

Table 5: Distribution of evaluation questions across document domains in the MMDocIR benchmark (N=1,658).

Domain	**#Q**	**#Pages**
Pharmaceuticals (FDA)	362	2,313
HR (EU Reports)	318	1,110
Finance (U.S. Banks)	309	2,942
Industrial (USAF)	283	5,244
Computer Science	215	1,360
Total	**1,487**	**12,969**

Table 6: Distribution of evaluation queries and corpus pages across domains in the ViDoRe V3 benchmark (English-only subset, N=1,487).

A.2 Embedding Strategies and VLM Prompt

Figure 2 illustrates the three page-embedding strategies compared in Section 3.1. The structured prompt used to generate page descriptions for the Desc.-Only and Image+Desc. strategies is given below; it is held constant across both VLMs.

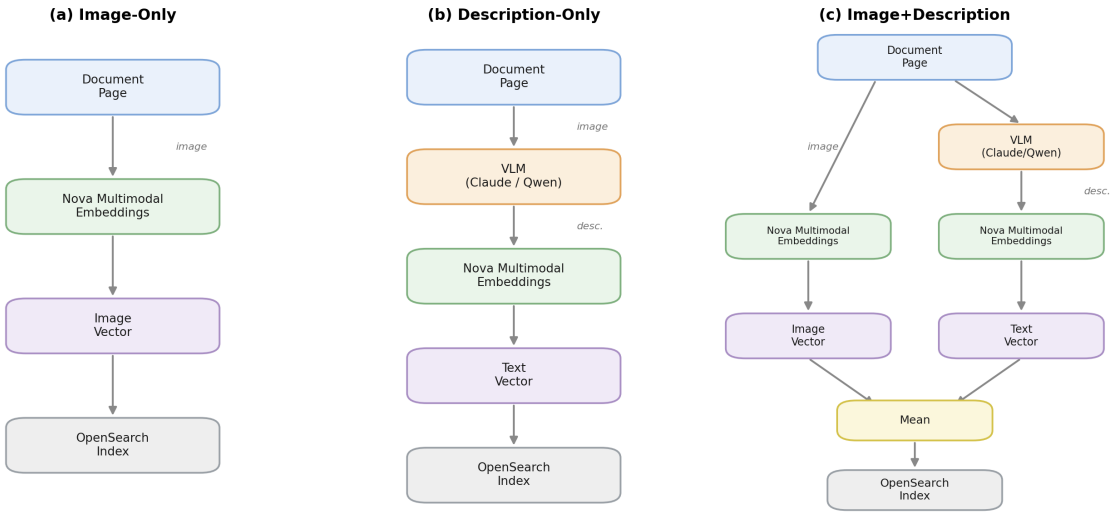


Figure 2: Three embedding strategies for page-level document indexing. (a) Image-Only embeds the raw page image directly. (b) Desc.-Only first generates a textual description with a VLM, then embeds the text. (c) Image+Desc. embeds the image and the description separately through Amazon Nova Multimodal Embeddings, then averages the two vectors.

VLM description generation prompt. For the Desc.-Only and Image+Desc. strategies, each page is described by a VLM using a structured, domain-agnostic prompt held constant across both VLMs, so that performance differences are attributable to the VLM rather than the task specification. Full prompt is reproduced as follow.

[System] You are a professional document analyst working for a research institution's digital archive. Your role is to create objective, factual descriptions of document page images for building a searchable document index. This work supports document research and retrieval systems.

You provide neutral, comprehensive descriptions without interpretation or judgment. Your descriptions focus solely on observable content to enable text-based search and retrieval of document pages.

[User] Analyze this document page image and provide a comprehensive, factual description.

DESCRIPTION STRUCTURE:

1. Document Overview:
- Document type (research report, tutorial, presentation slide, financial statement, legal contract, technical manual, etc.)
- Page number if visible
- Overall layout and design elements (single/multi-column, sidebars, text boxes)

2. Textual Content:
- All headings, titles, and section headers (transcribe verbatim)
- Key textual information including body text, captions, and labels
- Quotes, statistics, or highlighted information
- Footnotes, references, citations, or legal disclaimers if present

3. Visual Elements:
- Charts, graphs, tables, and diagrams with detailed descriptions
- Chart types (bar, line, pie, scatter, etc.) and represented data
- Technical diagrams, flowcharts, or process illustrations
- Colors, legends, axes labels, and data ranges
- Images, photographs, maps, or illustrations

```
  - Logos, icons, or graphical elements
4. Tables (if present):
  - Structure: number of rows and columns, headers
  - What each column represents
  - Key numerical values, trends, or patterns
  - Totals, subtotals, or summary rows
  - Cell formatting, highlights, or special notations
5. Data and Insights:
  - Specific numerical values, percentages, or trends
  - Comparisons, correlations, or key findings presented
  - Financial figures or legal references if applicable
GUIDELINES:
  - Maintain objectivity; describe observable content only
  - Use precise language for colors and positions (top-left, center, bottom-right)
  - Include verbatim text for titles and headings
  - For tables: describe structure AND key data comprehensively
  - For charts: specify axis representations and notable data points
  - Ensure description enables understanding without viewing the image
  - Pay special attention to multi-modal elements (text + images + tables)
OUTPUT: Provide a cohesive 500-word description integrating all elements while following the
structure above.
```

A.3 Single-Step Retrieval: Full Metrics and Per-Domain Results

Table 7 reports all six metrics for the five single-step configurations (the main text shows Recall@N only). Tables 8 and 9 then give the per-domain Recall@N breakdown with macro and micro averages.

Benchmark	VLM	Strategy	R@N	P@N	F1@N	NDCG@N	MAP	MRR
MMDocIR (N=1,658)	N/A	Image-Only	0.461	0.057	0.099	0.342	0.297	0.317
	Claude Sonnet 4.5	Desc.-Only	**0.628**	**0.075**	**0.132**	**0.505**	**0.455**	**0.486**
		Image+Desc.	0.598	0.073	0.127	0.478	0.429	0.456
	Qwen3-VL 235B	Desc.-Only	0.591	0.070	0.123	0.470	0.423	0.449
		Image+Desc.	0.595	0.072	0.126	0.469	0.420	0.447
ViDoRe V3 (N=1,487)	N/A	Image-Only	0.369	0.127	0.162	0.321	0.231	0.417
	Claude Sonnet 4.5	Desc.-Only	**0.566**	**0.193**	**0.249**	**0.511**	**0.399**	**0.624**
		Image+Desc.	0.519	0.177	0.227	0.462	0.352	0.572
	Qwen3-VL 235B	Desc.-Only	0.516	0.177	0.227	0.463	0.349	0.588
		Image+Desc.	0.511	0.174	0.223	0.450	0.338	0.562

Table 7: Single-step retrieval on both benchmarks (micro average). **Bold** = best per metric within each benchmark. Desc.-Only with Claude Sonnet 4.5 is optimal across all metrics and serves as the fixed embedding foundation for the agentic experiments (§5.2).

Tables 8 and 9 give the per-domain single-step Recall@N at k=10 for all five embedding configurations, with micro (equal question weight) averages. The Desc.-Only (Claude Sonnet 4.5) strategy dominates in nearly every domain on both benchmarks.

A.4 Query Agent: Workflow, Prompt, and Results

Figure 3 shows the Query-Agent-only workflow, followed by key prompt elements and the overall and per-domain results (Tables 10, 11, 12).

Domain	#Q	Img	Claude Sonnet 4.5		Qwen3-VL 235B	
			Desc	I+D	Desc	I+D
Academic Paper	389	0.428	**0.660**	0.633	0.614	0.646
Admin./Industry	56	0.304	0.327	0.384	0.271	**0.390**
Brochure	76	0.484	0.572	**0.595**	0.507	0.543
Financial Report	344	0.346	**0.595**	0.526	0.554	0.527
Government	111	0.405	**0.613**	0.568	0.577	0.523
Guidebook	115	0.617	0.713	**0.731**	0.627	0.707
Laws	132	0.636	**0.727**	0.712	0.697	0.720
News	137	0.095	**0.380**	0.277	**0.380**	0.256
Research Report	194	0.746	0.773	0.764	0.765	**0.774**
Tutorial/Workshop	104	0.653	**0.679**	0.672	0.652	0.677
Macro avg.	N/A	0.471	**0.604**	0.586	0.564	0.576
Micro avg.	1,658	0.461	**0.628**	0.598	0.591	0.595

Table 8: Single-Step Retrieval. MMDocIR per-domain Recall@N. Img = Image-Only, Desc = Desc.-Only, I+D = Image+Desc.. **Bold** = best of the four description-based columns per row.

Domain	#Q	Img	Claude Sonnet 4.5		Qwen3-VL 235B	
			Desc	I+D	Desc	I+D
Finance	309	0.252	**0.577**	0.500	0.550	0.495
HR	318	0.336	**0.563**	0.508	0.482	0.477
Industrial	283	0.230	0.381	0.345	**0.389**	0.377
Pharmaceuticals	362	0.517	**0.630**	0.604	0.551	0.571
Computer Science	215	0.521	**0.689**	0.648	0.626	0.657
Macro avg.	N/A	0.371	**0.568**	0.521	0.520	0.515
Micro avg.	1,487	0.369	**0.566**	0.519	0.516	0.511

Table 9: Single-Step Retrieval. ViDoRe V3 per-domain Recall@N. Img = Image-Only, Desc = Desc.-Only, I+D = Image+Desc.. **Bold** = best of the four description-based columns per row.

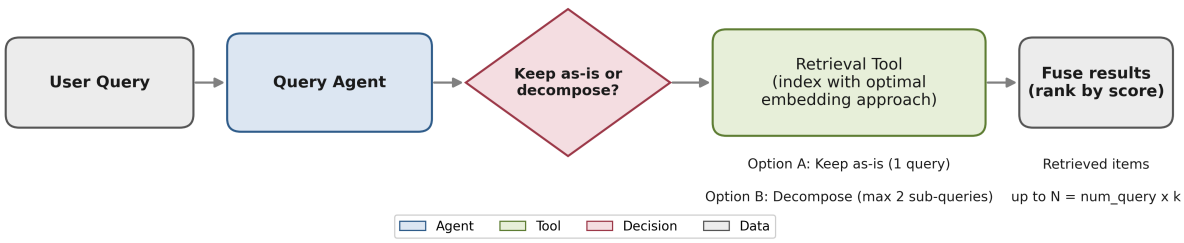


Figure 3: Query Agent Only workflow. The user query is rewritten and optionally decomposed into at most two sub-queries, each sent to the retrieval tool; the returned lists are fused and ranked by score.

Query Agent prompt. The full system prompt for the Query Agent is reproduced as follow.

Role and Purpose. You are a Query Agent. Your ONLY purpose is to call the retrieval tool over a document index and return the raw results. You do not answer the user's question. You do not summarize. You do not synthesize. You retrieve.

Hard rules (never violate).

1. You MUST call the retrieval tool at least once on EVERY user query.
2. You MUST NOT answer the user's question from your own knowledge, even if you know the answer or the question seems trivial. Always defer to retrieval.
3. You MUST NOT skip the tool call because the question looks unanswerable, ambiguous, or out of scope. Call retrieval anyway and return whatever it returns.
4. The user's query is treated as a search query, not a question to be answered.

Two retrieval options.

(A) DEFAULT: pass the user's query VERBATIM as a single retrieval query.
(B) DECOMPOSE: split the user's query into AT MOST 2 sub-queries, when it has clearly independent retrievable facets. There is no third option; you may NOT rephrase, paraphrase, fix typos, resolve pronouns, reword, condense, expand, or translate the query. If not decomposing, pass it byte-for-byte unchanged.

Workflow.

Step 1, Decide verbatim or decompose. If decomposing, produce at most two complete natural-language sub-queries that together cover the original intent, and do not include the verbatim query as one of them.
Step 2, Retrieve. Call the retrieval tool once per planned query (one call for (A), at most two for (B)). Do not summarize, modify, or filter the results.
Step 3, Return. Present the complete retrieved results exactly as returned by the tool, as a JSON array with one {query, results} element per retrieval call.

Constraints. Always call retrieval at least once and at most twice; never answer from your own knowledge; never fabricate results (if retrieval is empty, return the empty envelope); do not modify, reorder, or summarize retrieved items.

Overall results. Table 10 reports the overall Query-Agent-only Recall@N on both benchmarks. On MMDocIR no backbone beats the single-step baseline, and Recall falls as the decomposition rate rises; on ViDoRe V3 only Claude Opus 4.8 beats the baseline (+0.020).

Model	R@N	Decomp.	Fused N
MMDocIR (baseline R@N = 0.628)			
Qwen3-VL 235B	0.628	0.12%	10.00
Qwen3-Next 80B	0.628	0.06%	10.00
Claude Opus 4.8	0.620	9.83%	10.47
Claude Sonnet 4.6	0.610	22.62%	11.10
Claude Haiku 4.5	0.610	3.38%	10.20
ViDoRe V3 (baseline R@N = 0.566)			
Claude Opus 4.8	**0.585**	41.36%	12.28
Qwen3-Next 80B	0.567	3.09%	10.22
Qwen3-VL 235B	0.566	1.28%	10.06
Claude Sonnet 4.6	0.562	54.40%	12.88
Claude Haiku 4.5	0.547	14.93%	10.98

Table 10: Query-Agent-only Recall@N on both benchmarks, on the Desc.-Only (Claude Sonnet 4.5) index. Decomp. is the fraction of questions decomposed into two or more sub-queries; Fused N is the average deduplicated list length. **Bold** = best per benchmark.

Domain	#Q	Opus 4.8	Sonnet 4.6	Haiku 4.5	Qwen3-VL	Qwen3-Next
Academic Paper	389	0.637	0.616	0.596	**0.660**	**0.660**
Admin./Industry	56	**0.330**	0.304	0.309	0.327	0.327
Brochure	76	0.572	**0.576**	0.564	0.572	0.572
Financial Report	344	0.588	0.588	**0.599**	0.595	0.592
Government	111	0.595	0.559	0.595	**0.613**	**0.613**
Guidebook	115	0.707	**0.725**	0.710	0.708	0.713
Laws	132	0.720	0.720	0.720	**0.727**	**0.727**
News	137	**0.380**	0.372	**0.380**	**0.380**	**0.380**
Research Report	194	**0.775**	0.774	0.756	0.770	0.773
Tutorial/Workshop	104	0.679	0.645	**0.694**	0.679	0.679
Macro avg.	N/A	0.598	0.588	0.592	0.603	**0.604**
Micro avg.	1,658	0.620	0.610	0.610	**0.628**	**0.628**

Table 11: MMDocIR per-domain Recall@N by query-agent only workflow. All configurations use Description-Only (Sonnet 4.5) as the embedding index. Single-step baseline: micro avg. 0.628. No query-agent model surpasses the baseline on aggregate; models that rarely decompose (Qwen variants, $<0.2\%$ decomposition rate) most closely reproduce baseline performance.

Domain	#Q	Opus 4.8	Sonnet 4.6	Haiku 4.5	Qwen3-VL	Qwen3-Next
Finance (U.S. Banks)	309	**0.603**	0.562	0.555	0.577	0.578
HR (EU Reports)	318	**0.584**	0.570	0.547	0.563	0.566
Industrial (USAF)	283	**0.393**	0.368	0.356	0.376	0.376
Pharmaceuticals (FDA)	362	**0.650**	0.621	0.608	0.633	0.633
Computer Science	215	0.705	**0.706**	0.686	0.691	0.691
Macro avg.	N/A	**0.587**	0.565	0.550	0.568	0.569
Micro avg.	1,487	**0.585**	0.562	0.547	0.566	0.567

Table 12: ViDoRe V3 per-domain Recall@N by query-agent only workflow. All configurations use Description-Only (Sonnet 4.5) as the embedding index. Single-step baseline: micro avg. 0.566. Claude Opus 4.8 is the only model to beat single-step (+0.020), winning 4 of 5 domains.

A.5 Reflection Agent: Workflow, Prompts, and Per-Domain Results

Figure 4 shows the standalone Reflection-Agent workflow. We then give the orchestrator and `visual_reflection` prompts, and the per-domain Recall@N for each evaluator model (Tables 13, 14).

Reflection Agent (orchestrator) prompt. The full system prompt for the Reflection Agent orchestrator is reproduced below.

Role and Purpose. You are the Reflection Agent. You orchestrate a reflect-and-retrieve loop on top of an initial set of retrieved document pages. Your goal is to ensure the final result set is sufficient to answer the user's query. Do not answer the user's query yourself, and do not summarize, reorder, or fabricate documents; your only job is to decide whether to retrieve more and to merge the results.

Tools.

visual_reflection(query, retrieved_results): sends the current retrieved page images plus the user's query to a visual evaluator; returns a verdict (ADEQUATE or NOT ADEQUATE), reasoning, and up to two suggested sub-queries when gaps exist.

retrieval(query): kNN search over the document index for a single query string, returning the top matching pages.

Workflow.

Step 1, Reflect: call visual_reflection on the current retrieved set with the user's query.

Step 2, Decide: if the verdict is ADEQUATE, stop and return the current set unchanged; if NOT

ADEQUATE, proceed.
Step 3, Retrieve gaps: for each suggested sub-query (up to two), call retrieval once (at most two calls).
Step 4, Reflect again (optional): you may call visual_reflection once more on the augmented set to verify the gaps are closed.
Step 5, Return: return the union of the original and newly retrieved pages, exactly as retrieved.

Budgets and constraints. At most 2 visual_reflection calls and 2 retrieval calls per run. Never call retrieval without a NOT ADEQUATE verdict; never fabricate sub-queries (use those returned by visual_reflection); never modify, reorder, merge, or summarize retrieved pages; never answer the query yourself.

Visual Reflection tool prompt. The full system prompt for the `visual_reflection` tool is reproduced below.

Role and Purpose. You are a Visual Reflection Evaluator. You receive a user's query and a set of retrieved document pages (as images, possibly with text snippets). Your job is to look at the page images and judge whether they are sufficient to answer the query, and if not, identify exactly what is missing. You do not retrieve documents, rewrite queries, or answer the user's question; you only evaluate adequacy.

Evaluation criteria. Relevance (do the pages address the core topic?), completeness (do they cover every key aspect and sub-question?), specificity (specific enough, not too general?), consistency (coherent, non-contradictory?), visual evidence (do charts, tables, or figures support the query?), and sufficiency (enough detail for a well-supported answer?).

Decision. Output ADEQUATE if the pages suffice, otherwise NOT ADEQUATE.

Output format.
VERDICT: ADEQUATE | NOT ADEQUATE
REASONING: query intent (one sentence), coverage (aspects covered), gaps (what is missing, vague, or insufficient).
If NOT ADEQUATE, also SUGGESTED SUB-QUERIES: up to two, each targeting a distinct gap, rephrased with synonyms or decomposed, not repeating the original query.

Constraints. Never fabricate page content; never answer the query yourself; never modify, reorder, or summarize the retrieved pages; always ground reasoning in what is visible on the pages.

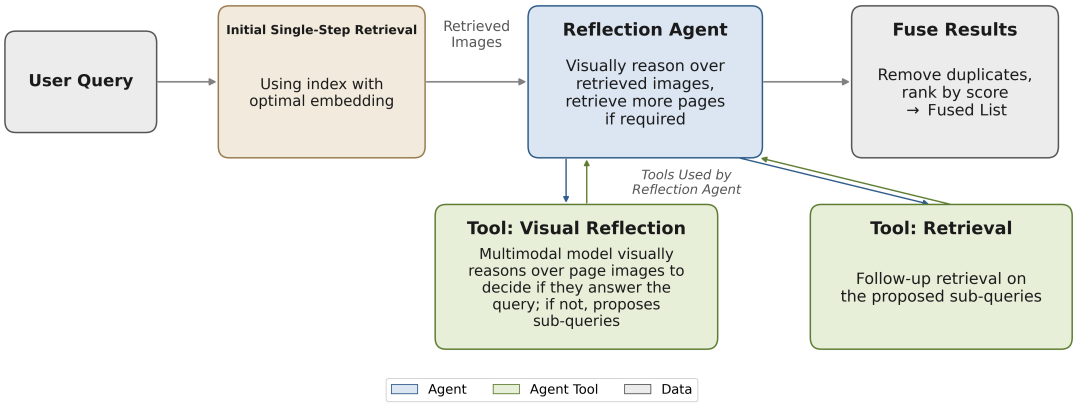


Figure 4: Reflection Agent Only workflow. An initial single-step retrieval, using the index with the optimal embedding, returns page images that are sent to the Reflection Agent. The agent calls a `visual reflection` tool, which uses a multimodal model to visually reason over the page images to decide whether they answer the query and, if not, proposes sub-queries; a `retrieval` tool then issues follow-up retrievals on those sub-queries. All retrieved items are fused by removing duplicates and ranking by score into the final fused list.

Domain	#Q	Sonnet 4.6	Opus 4.8	Qwen3-VL	Haiku 4.5
Academic Paper	389	**0.690**	0.678	0.671	0.673
Admin./Industry	56	**0.414**	0.399	0.381	0.381
Brochure	76	0.623	0.602	**0.632**	0.599
Financial Report	344	0.612	0.617	0.615	**0.621**
Government	111	**0.631**	**0.631**	0.613	0.622
Guidebook	115	**0.788**	0.777	0.762	0.762
Laws	132	0.727	**0.735**	**0.735**	0.727
News	137	**0.445**	**0.445**	0.423	0.416
Research Report	194	**0.796**	0.795	0.792	0.795
Tutorial/Workshop	104	0.739	**0.740**	0.726	0.694
Macro avg.	N/A	**0.646**	0.642	0.635	0.629
Micro avg.	1,658	**0.663**	0.659	0.653	0.650

Table 13: MMDocIR per-domain Recall@N by Reflection Agent Only workflow. Orchestrator fixed to Claude Haiku 4.5. Single-step baseline: micro avg. 0.628. Every evaluator beats single-step (+0.022 to +0.034); Sonnet 4.6 leads on 6 of 10 domains and achieves the highest micro average.

Domain	#Q	Opus 4.8	Qwen3-VL	Sonnet 4.6	Haiku 4.5
Finance	309	**0.659**	0.651	0.621	0.615
HR	318	**0.636**	0.628	0.620	0.610
Industrial	283	**0.441**	0.437	0.416	0.401
Pharmaceuticals	362	**0.719**	0.693	0.665	0.662
Computer Science	215	**0.779**	0.767	0.725	0.724
Macro avg.	N/A	**0.647**	0.635	0.610	0.603
Micro avg.	1,487	**0.644**	0.632	0.608	0.601

Table 14: ViDoRe V3 per-domain Recall@N by Reflection Agent Only workflow. Orchestrator fixed to Claude Haiku 4.5. Single-step baseline: micro avg. 0.566. Every evaluator beats single-step; Opus 4.8 leads on every domain and achieves the highest micro average.

Dataset	Config	Single-step	Reflection	Δ
MMDocIR	vr = Sonnet 4.6	0.7095	**0.7424**	+0.0329
ViDoRe V3	vr = Opus 4.8	0.7075	**0.7756**	+0.0681

Table 15: Reflection Agent Only vs. single-step at a larger retrieval budget (k=30), Recall@N. Orchestrator fixed to Claude Haiku 4.5; the `visual_reflection` model is the best per dataset. The advantage of reflection over single-step *persists at higher* k: even when single-step already retrieves 30 pages, adding the Reflection Agent on the same retrieval tool improves Recall on both benchmarks (+0.0329 on MMDocIR, +0.0681 on ViDoRe V3), confirming the gain is not an artifact of the small-k regime.