Question-Guided Evidence Acquisition for Multimodal Visual Question Answering

Alin-Ionut Popa

Amazon Inc.
popaaln@amazon.com

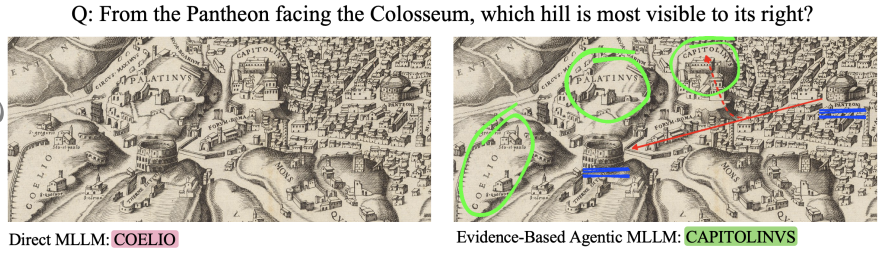

Fig. 1: From uniform perception to question-conditioned evidence recovery. Direct MLLM inference (*Left*) processes the document uniformly and answers from incomplete perception. Q-Guide (*Right*) directs perception toward the question—locating landmarks, reading targeted labels, and verifying spatial relationships—before committing to an answer.

Abstract. Multimodal LLMs can see a document, but they often can't read it reliably. Small text, tables, visual cues, and topological elements still trip them up under direct visual inference, even when the page is already sitting in the model's context. Most document-VQA systems treat perception as fixed: they encode the page once and answer from whatever the model extracted in that single fast pass. We think document VQA needs slower, more deliberate perception: instead of committing to one fixed encoding, the model should spend a little extra compute at inference time working out what to look at next, and only then answer. We build this into **Q-Guide**, a small agent that reads a question, works out what evidence it is still missing, and calls targeted tool(s) to recover it—reading text where text is needed, zooming in where detail is needed, or grounding a region where position matters. On DocVQA2026 and M109NC (a Manga109-based character-naming task), Q-Guide outperforms both direct prompting and recent multi-agent document systems (65.0% vs. 40.0% on DocVQA2026, 32.4% vs. 24.4% on M109NC), and the improvement holds across three Claude backbones (Opus 4.6, Sonnet 4.6, and Opus 4.5). We find that accuracy scales with the perception budget—most of the gain appears within two to three deliberate rounds—and that the gain comes from directing perception to the right place, not from complex control logic: adding planners, routers, or multiple collaborating agents does not help.

Keywords: Multimodal Reasoning · Slow Thinking · Test-Time Compute · Question-Conditioned Perception · Agentic VQA

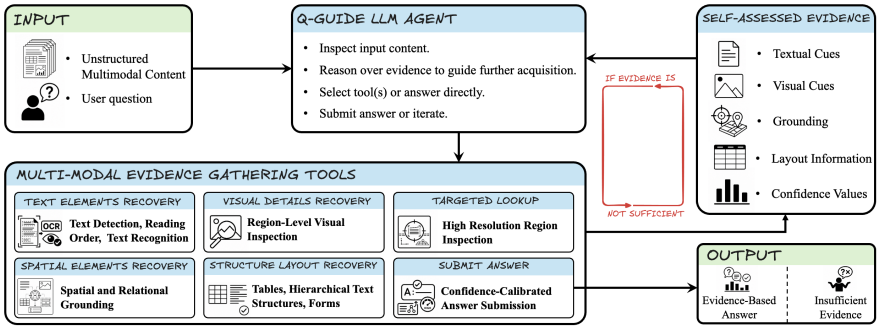

Fig. 2: Overview of Q-Guide. Given unstructured multimodal content and a user question, Q-Guide makes perception question-conditioned: the agent inspects the document, reasons about what evidence is still missing, selects perceptual recovery tools to fill the gap, and updates its evidence state. The loop continues until the recovered evidence supports an answer.

1 Introduction

Multimodal large language models have become very good at looking at documents, grounding regions, and reasoning across modalities [7, 11, 36, 49], building on steady progress in visual object representation and detection [23, 31]. But there is still a gap between seeing a document and actually reading it, and that gap grows as documents get visually denser. A model can have a page in its context window and still misread a dimension on an engineering drawing, miss which table cell is highlighted, or overlook a small label on a map (Figure 1). In our experiments this was the most common failure: the model could reason fine, but it did not reliably perceive the specific piece of evidence the question depended on.

One reason is that most document-VQA pipelines perceive the document in a single fast pass, the same way regardless of the question [7, 16, 26, 28]. The page is encoded once and the model answers from whatever it extracted—a System-1 style that commits before it has really looked. This works when the evidence is large and central, but visually rich documents are not like that [30, 33, 45, 51]: the evidence for one question might be a value printed in red, for the next a route on a map or a name in a speech bubble, each needing a different way of being read.

Our idea is to make perception slow and deliberate, and to condition it on the question [17, 30, 34, 44, 48]. Rather than committing after one pass, the model asks what this particular question requires it to perceive, spends test-time compute recovering exactly that, and repeats until it has enough evidence to answer—a System-2 loop over perception rather than over language alone. We build this into a compact agent we call **Q-Guide**.

Contributions. (i) We cast document VQA as test-time perceptual reasoning: a question-conditioned loop that spends extra compute on *what to perceive*

beats heavier reasoning orchestration (planning agents, routing, and multi-agent collaboration) on the same backbone [15,20,41,52], and accuracy scales with the perception budget until it saturates around two to three rounds. (ii) The same design transfers to very different visual domains, from document pages to manga, without any structural changes. (iii) The gains are not specific to one model: they hold across Claude Opus 4.6, Sonnet 4.6, and Opus 4.5.

We evaluate on two benchmarks that stress different things. DocVQA2026 [28] spans eight document categories with reasoning over up to 181 pages, and tests breadth across layouts, tables, and long documents. M109NC, a character-naming task we build from Manga109 [1, 6, 43], tests one hard axis instead: matching a character's identity across pages from Japanese dialogue and visual appearance. On both, Q-Guide beats direct prompting and recent agentic baselines [15, 34, 41]. Since these baselines were originally reported on different MLLM backbones, we re-implement each on our Claude backbones—keeping each method's own tools and control logic—so any difference reflects the evidence-acquisition strategy, not the underlying model. Ablations then confirm the gains come from directing perception toward question-relevant evidence; adding orchestration layers does not help and sometimes hurts [52].

2 Related Work

The gap we described in the introduction—models that see a page but do not reliably read the evidence a question needs—has been approached from two directions: model-side, training or adapting MLLMs to perceive text-rich documents better, including long-range sequence encoders for document structure [24, 40]; and system-side, keeping a general-purpose MLLM and giving it external perception it can call on. Q-Guide is system-side: the MLLM stays as-is but gains question-conditioned recovery tools that surface evidence it cannot reliably extract from pixels alone [1, 6, 8, 11, 16, 26, 28, 50].

OCR-free and document-specialized MLLMs. OCR-free and document-specialized MLLMs improve the model-side representation of text-rich documents [7, 9, 16, 25, 26, 36, 39], building on a longer line of specialized text-spotting in document images [21], internalizing document perception through architecture, training data, or visual encoding. Q-Guide goes the other way, exposing question-conditioned recovery tools the model can invoke on demand [2, 35].

Tool-augmented and agentic VQA. There is growing interest in moving VQA from passive answer generation toward active evidence gathering [17,27,30, 48]. Several recent methods let vision-language models call external perception modules, break down visual queries, and iteratively collect information before committing to an answer [20, 38, 48]. In visually rich document understanding, closely related systems such as ARIAL [34] and AgenticOCR [44] treat document QA as an iterative process of locating and parsing evidence rather than relying on a fixed OCR transcript. Q-Guide shares this motivation but asks a more pointed question: which perceptual recovery actions actually help when conditioned on the question, and which ones just add overhead?

Agentic orchestration for document question answering. Recent document QA systems explore increasingly structured forms of orchestration, including planning agents, multi-agent collaboration, retrieval modules, answer grounding, and iterative self-correction [8, 12, 14, 15, 19, 29, 34, 41, 42, 46, 50, 53]. These systems show that decomposing document understanding into smaller actions can improve interpretability and robustness, especially for long or visually complex documents [18, 33, 45, 51]. The downside is that heavier orchestration brings its own failure modes: planner errors, noisy intermediate summaries, repeated reflection, and context that ends up distracting the reasoning model [52]. Q-Guide tests this trade-off head-on: rather than deeper orchestration, it uses a compact single-loop agent that directs perception toward question-relevant evidence and stops when the recovered modalities support an answer.

3 Q-Guide

The central idea is to make perception adaptive to the question. Rather than applying a fixed processing pipeline, the model iteratively identifies what evidence is missing and recovers it through targeted tools. Figure 2 gives an overview. The loop is short: the unstructured multimodal content and the user question come in, the Q-Guide agent inspects the document and picks question-conditioned recovery actions, the recovery tools extract the missing evidence, and the agent judges whether it now has enough to answer or needs another round.

3.1 Preliminaries and Formulation

Let $\mathcal{D} = \{p_1, \ldots, p_N\}$ be a document of N page images and q the user question. Q-Guide answers q—or returns `Unknown` when the evidence is insufficient—by iteratively gathering evidence and testing whether it suffices, an interaction defined by the tuple $(\mathcal{E}, \mathcal{A}, \mathcal{T}, \phi)$.

Evidence state. At turn t the agent holds an evidence state $e_t = (q, \mathcal{D}, O_t) \in \mathcal{E}$, where $O_t = (o_1, \ldots, o_{k_t})$ is the sequence of observations gathered so far, starting empty. Each observation $o_i = (a_i, r_i, y_i)$ records the tool a_i, the normalized region $r_i = (l, t, r, b) \in [0, 1]^4$ it read from, and the returned content y_i. Attaching the source region r_i to every observation ties each answer to the page location it was read from rather than to parametric priors.

Actions and transition. The action space $\mathcal{A} = \mathcal{A}_{\text{tool}} \cup \{\texttt{submit}\}$ holds the five recovery tools (Section 3.3) and the answer submission. A single policy π—the same LLM at every turn—proposes actions $\pi(e_t) \subseteq \mathcal{A}$ from the current evidence; the termination rule below arbitrates this proposal into the executed set $\mathbf{a}_t$, whose tool actions append their observations to the state:

$$e_{t+1} = \mathcal{T}(e_t, \mathbf{a}_t) = \Big(q,\ \mathcal{D},\ O_t \oplus \bigoplus_{a \in \mathbf{a}_t \cap \mathcal{A}_{\text{tool}}} \texttt{execute}(a)\Big), \tag{1}$$

where $\oplus$ concatenates the new observations onto O_t and `submit` adds none. Because the transition only ever *appends* to O_t and the agent places just $P_{\text{ctx}} \ll N$

pages in context (of the N total), the interaction cost scales with the number of rounds and the evidence gathered, not with the document length N—a property we quantify in Section 5.5.

Sufficiency and termination. Whether to gather more or answer is governed by the sufficiency predicate $\phi : \mathcal{E} \to \{0, 1\}$—the agent's judgment of whether the evidence suffices—which we make explicit. On submission the policy emits a self-reported confidence level $\sigma(e_t) \in \mathcal{L} = \{\textsc{low} \prec \textsc{med} \prec \textsc{high}\}$. The gate commits either when this confidence is at or above a threshold τ, or—for a less-confident submission—when a lightweight check finds the answer explicitly supported by the gathered evidence, $\textsc{ver}(e_t) = 1$:

$$\phi(e_t) \;=\; \mathbb{1}\big[\,\text{policy submits} \;\wedge\; \big(\sigma(e_t) \succeq \tau \;\vee\; \textsc{ver}(e_t)\big)\,\big], \qquad \tau = \textsc{med}\ \text{(default)}, \tag{2}$$

with $\mathbb{1}[\cdot]$ the indicator (1 iff its argument holds), $\succeq$ the order on $\mathcal{L}$, and $\textsc{ver}(e_t) = 1$ iff the candidate answer is explicitly supported by some observation $o_i \in O_t$. In words, high- and medium-confidence answers pass directly, while a less-confident answer commits only if verification backs it and otherwise abstains—a test applied by the policy itself, not a separate critic module. Each turn then resolves as:

$$\mathbf{a}_t = \begin{cases} \{\texttt{submit}(e_t)\} & \text{if } \phi(e_t) = 1, \\ \{\texttt{submit\_unknown}(e_t)\} & \text{if } \phi(e_t) = 0 \text{ and } \big(\text{policy submits or } t \geq T_{\max}\big), \\ \pi(e_t) & \text{otherwise (gather more).} \end{cases} \tag{3}$$

The second case is where verification does its work: a below-threshold submission that the evidence does not explicitly support ($\phi = 0$ with the policy nonetheless submitting) abstains to `Unknown` rather than emitting an unsupported answer; the same abstention applies once the budget $T_{\max}$ is spent. Because one model both selects actions and evaluates ϕ, Q-Guide needs no separate planner, router, or critic; it simply reasons over the accumulated evidence and decides what to perceive next.

3.2 Architecture

Q-Guide instantiates the loop of Section 3.1 with three parts. First, the **Q-Guide agent** (an MLLM) is the single policy π: it receives the unstructured multimodal content and the question, and at each turn inspects the evidence state e_t, reasons about what perceptual evidence is still missing, and selects one or more recovery tools to fill the gap. Multiple operators can be invoked in a single turn for complementary evidence. The instructions are intentionally minimal—short operator descriptions and answer-format guidance, with no explicit planning, category routing, or advisory hints. Second, the selected **question-conditioned recovery tools** extract information that is difficult to perceive from the raw image alone—text, visual detail, spatial grounding, layout structure, and targeted region answers—each addressing a perceptual gap conditioned on the current question. Every call returns a grounded observation

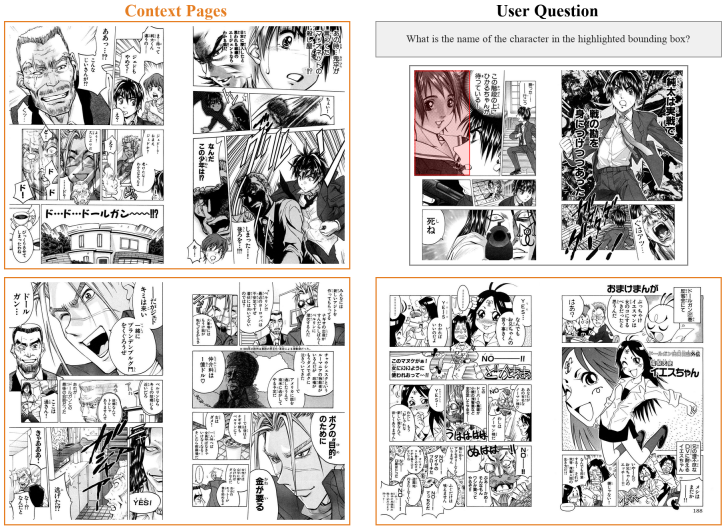

Fig. 3: M109NC task example. For visualization, we show 3 of the 6 context pages (orange border) that reference the query character; the query page highlights the target (red overlay). The task requires cross-page reasoning: reading Japanese dialogue, associating names with visual appearances, and matching the target's identity across scenes.

$o_i = (a_i, r_i, y_i)$ that the transition $\mathcal{T}$ appends to the evidence state. Third, Q-Guide performs **self-assessment**: the same policy that directs perception also evaluates the sufficiency predicate ϕ over the accumulated observations, looping back for more evidence while $\phi = 0$ and submitting once $\phi = 1$. Because the state grows one deliberate perception step at a time, this loop is where the model's test-time compute is spent.

Finally, the **output** follows Eq. 3: an evidence-based answer once $\phi(e_t) = 1$, or `Unknown` when the agent exhausts its budget or a low-confidence answer fails verification.

3.3 Question-Conditioned Recovery Tools

The model has direct access to the full document, yet seeing is not reading: small text, dense tables, colored cells, spatial arrangements, and stylized fonts all degrade under direct visual inference, so active recovery is needed even though the pixels are already in context. Each tool below targets one such gap, and the agent decides which to fill next based on what the question demands and what it has gathered so far.

Q-Guide exposes five evidence-recovery tools and one submission action, matching the tool groups in Figure 2. Each tool operates on a document page or on a region specified by normalized coordinates $(l, t, r, b) \in [0, 1]^4$. This allows the agent to move from a global view of the document to targeted local evidence.

Text elements recovery. This tool recovers text from a selected page or region. It supports text detection, reading-order recovery, and text recognition. It is useful for small text, scattered labels, dimensions, captions, form fields, and dense document regions.

Visual details recovery. This tool returns a high-resolution crop of a selected region. It does not perform OCR; instead, it gives the LLM a clearer view of the raw pixels. It is useful for color, shape, layout, relative position, chart appearance, and other details that may be lost in text extraction.

Targeted lookup. This tool asks a natural-language question over a selected region and returns a short answer with confidence. It is useful when the agent has already localized the relevant area and needs a precise value, such as a total, width, label, or table entry.

Spatial elements recovery. This tool provides grounding capabilities that anchor text and visual elements in the input coordinate space, recovering word-level positions, color distributions, and text along geometric corridors. It is useful for highlighted values, map-like layouts, or text arranged along a path.

Structure layout recovery. This tool recovers tables, forms, hierarchical text structure, and layout information. When needed, it also associates table cells with visual properties such as text color or background color. It is useful when the answer depends on both text and structure.

Submit answer. This action terminates the evidence acquisition loop. The agent submits a final answer, short reasoning, and confidence level. The output is either an evidence-based answer or `Unknown` when the evidence is insufficient.

Table 1: Tool importance on DocVQA2026. Leave-one-out over the five recovery tools (T: Text, V: Visual Inspection, Q: Targeted Query, S: Structure, P: Spatial), reporting accuracy and the drop when each is removed. Visual Inspection and Targeted Query are the most impactful; analyzed in Section 5.3.

Configuration	T	V	Q	S	P	Acc.	Drop
Q-Guide (full)	✓	✓	✓	✓	✓	65.0%	—
– Text	×	✓	✓	✓	✓	51.9%	↓ 13.1
– Visual	✓	×	✓	✓	✓	47.8%	↓ 17.2
– Query	✓	✓	×	✓	✓	48.8%	↓ 16.2
– Structure	✓	✓	✓	×	✓	57.8%	↓ 7.2
– Spatial	✓	✓	✓	✓	×	61.1%	↓ 3.9

Not all five tools contribute equally. A leave-one-out ablation (Table 1; analyzed in Section 5.3) removes each tool family in turn and measures the accuracy drop on DocVQA2026: Visual Inspection and Targeted Query are the most impactful, confirming that question-conditioned extraction—not the sheer number of tools—is what drives performance.

For long documents, we use a simple overlapping chunking strategy during evaluation. Each chunk j is processed by the same Q-Guide loop and returns an answer a_j with confidence σ_j; a lightweight adjudicator then combines the chunk-level answers by confidence-weighted majority,

$$\hat{a} \;=\; \arg\max_{v} \sum_{j:\, a_j = v} w(\sigma_j), \qquad w(\textsc{low}, \textsc{med}, \textsc{high}) = (1, 2, 3), \tag{4}$$

falling back to an LLM arbitration over the candidate set only when (4) is tied. This proved more reliable than page-level retrieval for DocVQA2026, where many answers depend on nearby consecutive pages or cross-page context (chunking 65.0% vs. page-level retrieval 50.0%; Section 5.5). A question-aware retriever that selects relevant pages under a holistic view of the document is a natural extension we leave to future work.

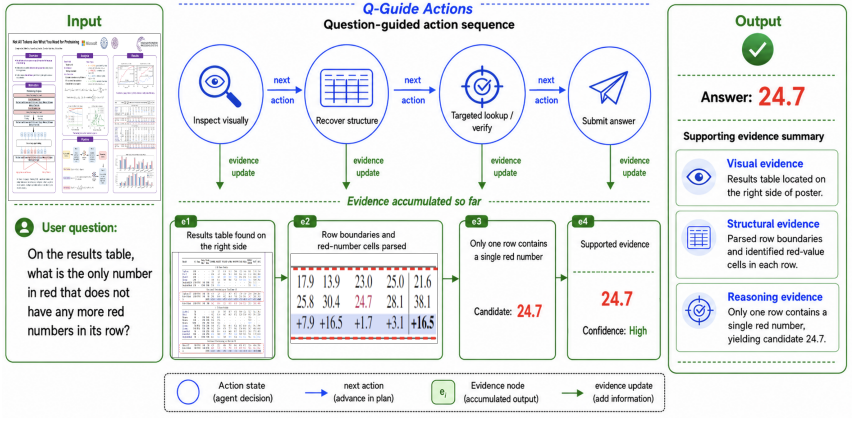

Fig. 4: Q-Guide in action. A science-poster example where the answer depends on both table structure and color. The agent locates the relevant region, recovers the table and red cell values, and submits the answer only after the evidence supports it. Each evidence node e_i shows the observation o_i added at that step (Section 3.1); together they form the accumulated evidence state.

4 Implementation Details

This section grounds the three parts from Section 3.2 in concrete components. The Q-Guide agent is a single MLLM (the *LLM backbone* below); the question-conditioned recovery tools are backed by off-the-shelf perception services (*tool backends*); and the self-assessment loop, including the low-confidence verification check, is realized as an explicit state machine (*agent execution*).

LLM backbone. We use Claude Opus 4.6 [5] as the multimodal reasoning backbone. The model receives unstructured multimodal content and the user question, decides which tools to call, and produces the final answer. We use deterministic decoding with temperature 0. Page images are resized to preserve readability while fitting within the context budget.

Tool backends. Structured text extraction, layout recovery, table/form parsing, and targeted region queries are implemented with Textract [2]. Spatial text detection uses a coordinate-aware text detector at word polygonal representation level [35]. For visual text grounding, i.e. anchoring detected text elements

to their spatial coordinates in the document, we use Qwen3.5-2B [37], which provides bounding-box-level localization of textual elements. Visual inspection returns image crops from the source document. Color sampling and corridor text reading are implemented with lightweight image-processing routines. Region crops include padding and are upscaled before OCR or query-based extraction.

Agent execution. The agent is implemented as a state machine using LangGraph [22], with nodes for initialization, agent reasoning, tool execution, answer extraction, low-confidence verification, and forced termination; this cleanly separates decision logic from tool execution and supports reproducible multi-turn interactions. The evidence-gathering budget is $T_{\max} = 3$ turns, and multiple parallel tool calls in a single turn count as one turn. For long documents, we combine outputs from overlapping chunks via the confidence-weighted adjudication of Eq. 4 (majority vote, with LLM arbitration on ties).

		Document Category								
Method	**Type**	**Bus. Report**	**Comics**	**Eng. Draw.**	**Infogr.**	**Maps**	**Sci. Paper**	**Sci. Poster**	**Slides**	**Overall**
Direct prompting baselines (single LLM call, no tools)										
Visual Prompt	Direct	20.0%	40.0%	20.0%	20.0%	20.0%	30.0%	40.0%	30.0%	27.5%
OCR Prompt	Direct	30.0%	20.0%	10.0%	40.0%	50.0%	20.0%	50.0%	40.0%	32.5%
Visual+OCR Prompt	Direct	40.0%	20.0%	20.0%	50.0%	30.0%	50.0%	50.0%	50.0%	38.8%
Agentic baselines (multi-step, tool use)										
ARIAL [34]	Agentic	20.0%	40.0%	40.0%	30.0%	10.0%	30.0%	30.0%	60.0%	32.5%
DocAgent [41]	Agentic	30.0%	40.0%	40.0%	30.0%	30.0%	40.0%	50.0%	60.0%	40.0%
MDocAgent [15]	Agentic	30.0%	40.0%	30.0%	40.0%	40.0%	50.0%	60.0%	30.0%	40.0%
Q-Guide	**Agentic**	**60.0%**	**60.0%**	**90.0%**	**60.0%**	**50.0%**	**50.0%**	**90.0%**	**60.0%**	**65.0%**

Table 2: Results on the DocVQA2026 validation split (80 questions, 8 categories). Q-Guide achieves the best overall accuracy, with the largest gains on engineering drawings and science posters, where precise local evidence is critical. Even at this sample size, Q-Guide's Wilson [47] 95% confidence interval on the overall column ($[54.1, 74.5]$) sits clearly above those of the strongest baselines (Visual+OCR $[28.8, 49.7]$, DocAgent/MDocAgent $[30.0, 51.0]$); per-category cells ($n = 10$) are noted for trends only, and the gaps are robust under paired [10, 32] and bootstrap [13] tests.

5 Experiments

5.1 Datasets and Task Formulation

We evaluate Q-Guide on two deliberately complementary settings that exercise different failure modes of the same framework: DocVQA2026 stresses *breadth* across document types and layouts, while M109NC stresses *depth on one hard axis*—naming a character, which forces cross-page identity matching from low-resource Japanese text and visual appearance. DocVQA2026 [28] contains 80 questions across 8 document categories: business reports, comics, engineering drawings, infographics, maps, science papers, science posters, and slides. Documents range from 1 to 181 pages, and many questions require correlating entities across regions, matching labels to visual elements, reading table cells, and

combining evidence from multiple pages. Although compact, DocVQA2026 is deliberately adversarial rather than easy: the eight categories mix dense tables, engineering dimensions, colored cells, and map topology, stressing a breadth of perception that far larger single-domain benchmarks do not probe. Its difficulty is evident in the numbers—the strongest direct-prompting baseline reaches only 38.8%, and every prior agentic system we test stays at or below 40.0%—so headroom here reflects genuine perceptual challenge, not annotation noise.

The second setting is a character naming benchmark that we construct from Manga109 [1, 6], inspired by the multi-task comic understanding setup of CoMiX [43]. We call this task M109NC, for Manga109 Naming Characters. Given a set of context pages containing character names and a query page with a highlighted character (Figure 3), the system must produce the character's canonical name. This requires active evidence acquisition beyond standard OCR: the model must read Japanese text from context pages, associate names with character appearances, and match the highlighted character on a different page. We use Manga109-v2026 annotations, which provide character body and face boxes linked to a book-level character roster.

For M109NC, we sample questions from 10 Manga109 books selected for diversity and difficulty, yielding 504 questions in total (roughly 50 per book). Q-Guide is evaluated zero-shot, so no training on the remaining books is involved. For each book, we select a compact set of context pages via a greedy set-cover over character name mentions, then draw query samples from the non-context pages. We deliberately withhold the full book: the goal is to test whether the method can recover a character's identity from a small evidence set where names and visual references are present but not trivially matched.

Performance varies across volumes (Figure 5), tracking how visually distinctive and unambiguously named the characters are: Q-Guide is strongest on YoumaKourin and EienNoWith (unique hairstyles and clothing) and all methods struggle on Akuhamu, where many hamster characters share the same body shape. For both benchmarks, we use ANLS-based matching as the primary evaluation criterion, supplemented by strict matching for numeric and date answers and order-invariant matching for list answers on DocVQA2026. Unless otherwise stated, all methods use Claude Opus 4.6 as a fixed reasoning-capable backbone LLM so that the comparison isolates the evidence-acquisition strategy from differences in the underlying model. We verify that Q-Guide's gains generalize across backbones; results with Claude Sonnet 4.6 [4] and Claude Opus 4.5 [3] are reported in Section 5.4.

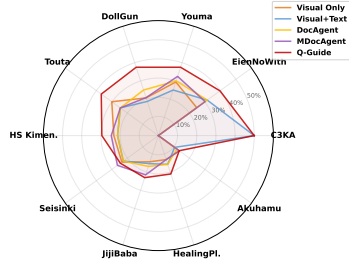

Fig. 5: Per-book accuracy on M109NC. Q-Guide (red) covers the largest area, indicating consistent gains across diverse manga styles.

5.2 Main Results

Table 2 compares Q-Guide with direct prompting baselines (a single LLM call over page images, OCR text, or both) and recent agentic document-understanding frameworks on the DocVQA2026 validation split.

On DocVQA2026, Q-Guide reaches 65.0%, improving over the strongest direct baseline (Visual+OCR Prompt) by 26.2 points and over DocAgent and MDocAgent by 25.0 points—so giving the model both images and OCR text is not enough. These agentic baselines are recent multi-step frameworks: DocAgent [41] decomposes questions with a planning agent, while MDocAgent [15] uses multi-agent reader/synthesizer collaboration. The largest gains appear on engineering drawings and science posters (90.0%), which hinge on precise local evidence—dimensions, table cells, colored values, chart entries—where targeted lookup, visual inspection, and structure recovery beat global document processing. Q-Guide also improves on maps (50.0%), though these remain challenging because many require spatial or topological reasoning that is only partially captured.

Figure 4 illustrates the process on a science-poster example: the agent locates the relevant table, recovers its structure and color, then verifies the answer before submitting. On M109NC, Q-Guide also improves over direct and agentic baselines (Table 3). With Claude-based OCR it reaches 32.4%, improving over Visual+OCR Prompt (23.1%) by 9.3 points and over DocAgent (24.4%) by 8.0 points; with ground-truth OCR it reaches 53.7%, a 28.8-point gain over Visual+OCR Prompt with GT-OCR (24.9%). Q-Guide's tools are thus highly effective when text is reliable, while imperfect OCR limits its potential. Notably, Q-Guide with imperfect Claude-OCR (32.4%) already surpasses Visual+OCR Prompt with *perfect* GT-OCR (24.9%), suggesting that iterative tool use and cross-page visual matching add value beyond better text extraction alone. One pattern holds across both benchmarks: visual input alone consistently trails structured multi-modal evidence (visual + text)—MLLMs reason better when evidence arrives already separated into complementary modalities than when they must decompose a scene internally. On M109NC this is especially striking, since the OCR is itself produced by Claude: the same model reads better when its own text extraction is surfaced as an explicit tool than end-to-end from pixels.

Table 3: Character naming on M109NC ($n = 504$). Q-Guide benefits strongly from higher-quality text: ground-truth OCR lifts its accuracy to 53.7%. The two Wilson 95% intervals—$[28.4, 36.5]$ under Claude-OCR and $[49.4, 58.1]$ under GT-OCR—do not overlap, so this text-quality effect is well separated from sampling noise.

	Text Source	
Method	**Claude-OCR**	**GT-OCR**
Direct prompting baselines		
Visual Prompt	21.8%	21.8%
Visual+OCR Prompt	23.1%	24.9%
Agentic approaches		
DocAgent [41]	24.4%	29.3%
MDocAgent [15]	25.6%	27.8%
Q-Guide	**32.4**%	**53.7**%

5.3 Ablation Studies

Our central claim is that closing the perception gap comes from pointing perception at the right place, not from heavier reasoning control. We ablate two axes this implies: orchestration strategy (does adding planning, routing, or extra stages help?) and tool composition (which recovery modalities surface the missing evidence?).

Configuration	Strategy	Acc.	Δ
Direct prompting baselines			
Visual Prompt	Direct Answer	27.5%	—
OCR Prompt	Direct Answer	32.5%	+5.0
Visual+OCR Prompt	Direct Answer	38.8%	+11.3
Two-stage agentic: research for the answer			
Extract-Synthesize	Investigate → Answer	44.3%	+16.8
Complex reasoning agentic			
3-Stage Research	Plan → Gather → Answer	35.7%	+8.2
Advisory Routing	Category-aware routing	44.3%	+16.8
Tool-focused agentic			
Basic agentic	Inspect ↻ OCR loop	50.0%	+22.5
Q-Guide	**5 recovery tools**	**65.0%**	**+37.5**

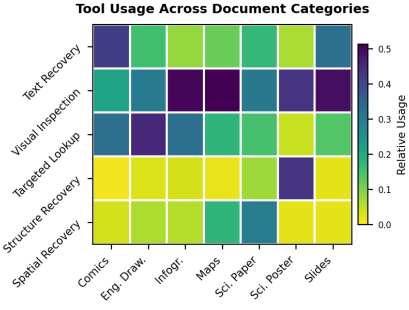

Fig. 6: Orchestration and tool-use analysis. *Left*: Q-Guide performs best with a compact iterative tool-use loop, while heavier planning and routing do not improve accuracy. *Right*: tool usage changes naturally across document categories, showing that the agent adapts evidence acquisition without explicit category routing.

Orchestration strategy. Figure 6 (*Left*) compares different ways of organizing the reasoning process. Direct prompting baselines range from 27.5% to 38.8%, with Visual+OCR Prompt performing best among them. A simple two-stage agentic setup, Extract-Synthesize, reaches 44.3%, showing that separating investigation from answering already helps.

Yet more structure does not always help. The 3-stage research pipeline reaches only 35.7%—below the Visual+OCR baseline—and Advisory Routing (44.3%) merely matches Extract-Synthesize: the bottleneck is not the absence of a plan. Extra planning produces brittle intermediate decisions, and category routing forces a fixed strategy even when the question needs different evidence. The strongest gains come from a simple iterative loop: Basic agentic reasoning reaches 50.0% and Q-Guide 65.0%, a 37.5-point improvement over Visual Prompt—the payoff is in perceptual recovery tools, not orchestration complexity. What that extra compute buys is visible in how accuracy grows with the perception budget (Fig. 8, right): from 40.0% at $T_{\max} = 1$ to 55.0% at $T_{\max} = 2$ and $\sim 65\%$ at $T_{\max} \geq 3$, after which it plateaus—most of the gain is already captured within two to three deliberate rounds.

Tool composition. Figure 6 (*Right*) shows the agent is not a fixed pipeline: it reaches for text recovery on text-heavy documents, visual inspection when local detail matters, and structure or spatial recovery for specialized cases—the question directs perception toward the relevant modality gap. Removing tools

Method	DocVQA2026			M109NC		
	Opus 4.6	Sonnet 4.6	Opus 4.5	Opus 4.6	Sonnet 4.6	Opus 4.5
Direct prompting baselines						
Visual Prompt	27.5%	30.0%	27.1%	21.8%	21.8%	21.8%
Visual+OCR Prompt	38.8%	31.4%	37.1%	23.1%	22.9%	22.9%
Agentic baselines						
DocAgent	40.0%	34.3%	37.1%	24.4%	24.2%	22.5%
MDocAgent	40.0%	31.4%	31.4%	25.6%	23.8%	25.1%
Q-Guide	**65.0%**	**62.8%**	**64.2%**	**32.4%**	**32.2%**	**31.5%**

Table 4: Accuracy across MLLM backbones. Q-Guide consistently achieves the best accuracy regardless of the underlying model. On M109NC, direct prompting baselines plateau around 22% irrespective of backbone, suggesting the bottleneck is the reasoning architecture rather than perceptual capability.

one at a time (Table 1) shows how much each one carries: dropping Visual Inspection or Targeted Query costs the most (−17.2 and −16.2 points), Text Recovery somewhat less (−13.1), and Structure and Spatial least (−7.2 and −3.9). Tellingly, losing just those top two together (−33.4 points) wipes out more than Q-Guide's entire lead over the best baseline—so it is the question-driven *targeting* of evidence, not the length of the toolset, that does the work.

On M109NC, the +21.3-point Claude-OCR-to-GT-OCR gap (32.4%→53.7%) shows the payoff is largest when recovered text is reliable, yet Q-Guide still beats all baselines under imperfect OCR. Broken down by context-page count (Figure 7), Q-Guide leads at every context size and scales with available evidence, exceeding 50% with 12 or more naming pages, whereas the baselines stay nearly flat.

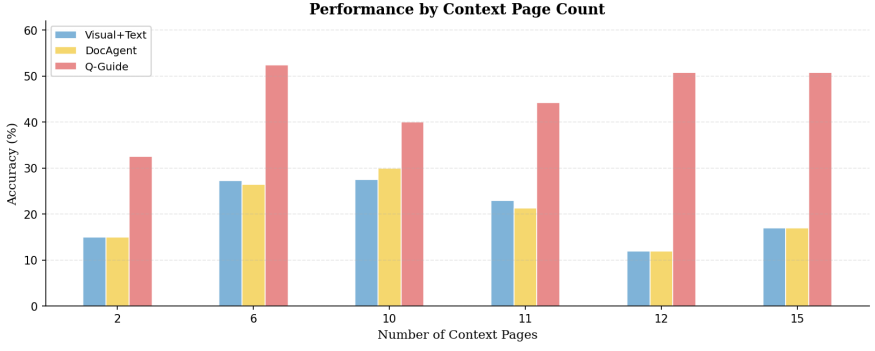

Fig. 7: M109NC accuracy by context page count. Q-Guide leads at every context size and exploits additional naming pages that single-call baselines cannot, exceeding 50% with 12 or more naming pages while the baselines stay nearly flat.

5.4 LLM Backbone Versatility

To check that Q-Guide's gains generalize beyond one backbone, we re-run all methods with Claude Sonnet 4.6 [4] and Claude Opus 4.5 [3]; Table 4 reports the results. Q-Guide improves over every baseline regardless of the model, so the gains come from the evidence-acquisition strategy rather than a specific model's capability. Stronger backbones amplify its DocVQA2026 advantage, but weaker ones still recover substantially with its tools. On M109NC the direct baselines are flat across models (∼22%) while Q-Guide holds ∼32%, so character naming benefits more from structured evidence acquisition than from raw perceptual capability. MDocAgent, by contrast, degrades on DocVQA2026 with weaker backbones (40.0%→31.4%): its multi-agent orchestration is fragile under reduced capability, while Q-Guide's simpler loop stays robust.

5.5 Limitations

Q-Guide's gains come with limitations that also mark where the method can improve: it costs more compute than single-pass prompting, it still struggles with spatial and topological content, and it remains bounded by the quality of the text it recovers.

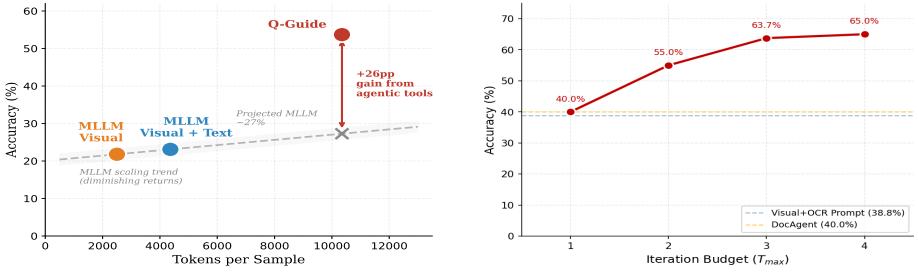

Fig. 8: Cost and budget analysis. *Left:* accuracy vs. token cost on M109NC; Q-Guide breaks above the single-call MLLM ceiling (dashed, +26pp) by spending tokens on active investigation rather than context stuffing. *Right:* accuracy vs. iteration budget ($T_{\max}$) on DocVQA2026; most gain is captured within 2–3 rounds, with diminishing returns beyond $T_{\max} = 3$.

Compute cost. Token use follows the simple relation Tokens $\approx \alpha P_{\text{ctx}} + \beta \sum_i |y_i| + \gamma T$—one term per cost source: page images in context, recovered tool outputs, and reasoning turns—so cost scales with the rounds and evidence gathered, not with document length. Table 5 bears this out on M109NC: Q-Guide's overhead is overwhelmingly input-side (9,668 of 10,351 tokens), i.e. surfacing recovered evidence rather than longer generation, and it spends ∼2.5× the tokens of the strongest agentic baseline. Since this cost grows with T while accuracy saturates by $T \approx 3$ (Fig. 8), a small budget captures most of the benefit, but the method is materially more expensive than a single pass.

Spatial and topological reasoning. Q-Guide's clearest accuracy weakness is on map questions, which often hinge on a value that depends on a route, adjacency, or orientation the model must follow visually. Its tools recover local text and detail but do not represent the map as a navigable structure, so a dedicated map-to-graph tool is the clearest next step. Relatedly, long-document handling relies on a simple fixed-window chunking with cross-chunk adjudication; this already beats page-level retrieval (65.0% vs. 50.0%, as many answers span consecutive pages), but a question-aware retriever remains future work.

OCR dependence. Q-Guide is only as good as the text it can read. Accuracy on M109NC rises almost linearly with text quality—from 21.8% with no text (image only), to 32.4% with Claude's own OCR, to 53.7% with ground-truth text—so if we place these on a quality scale from 0 to 1, the trend follows $\text{acc} \approx 21.8\% + 31.9x$. On that scale Claude-OCR lands at only $x \approx 0.33$: it realizes about a third of the gain that perfect text would, which tells us how much of the remaining error is bad OCR, and how much headroom better text extraction would unlock.

Table 5: Measured per-question token cost on M109NC (mean over 50 questions, Claude Opus 4.6). Q-Guide's overhead is input-dominated (93%).

Method	In	Out	Total
Visual Only	2,336	166	2,502
Visual+Text	3,941	422	4,364
DocAgent	3,452	603	4,055
Q-Guide	**9,668**	**683**	**10,351**

Cross-page matching. On M109NC the agent resolves character identity by combining visual-appearance similarity with the textual name mentions gathered across context pages, evaluating both natively from tool-surfaced evidence rather than through a separate face-embedding pipeline. Its residual errors are dominated by perceptual visual-matching failures rather than language-level coreference: single-call methods hit a ceiling regardless of context size, while Q-Guide breaks through by spending tokens on active tool use rather than passive input expansion (Fig. 8).

6 Conclusion

Q-Guide is a compact agent that spends test-time compute deciding what to perceive, directing recovery tools toward the evidence a question needs. It outperforms both single-pass (System-1) prompting and heavier multi-agent orchestration across three backbones and two visually distinct domains, with accuracy scaling with the perception budget—most of the gain arriving within 2–3 rounds. For document VQA, slow thinking is thus most useful applied to perception itself, not only to language-level reasoning: across cost, the confidence gate, and residual failures alike, the lever for further progress is richer *perception*, not deeper control logic. That the same loop carried unchanged from documents to manga character naming suggests a general recipe for evidence-limited multimodal QA; the clearest next steps are a structure-aware tool for maps, a question-aware retriever for long documents, and a port to open-weight backbones.

References

1. Aizawa, K., Fujimoto, A., Otsubo, A., Ogawa, T., Matsui, Y., Tsubota, K., Ikuta, H.: Building a manga dataset "manga109" with annotations for multimedia applications. IEEE Multimedia **27**(2), 72–80 (2020)
2. Amazon Web Services: Amazon textract documentation. `https://docs.aws.amazon.com/textract/` (2026), developer documentation. Accessed: 2026-06-02
3. Anthropic: Claude opus 4.5 model card. `https://www.anthropic.com/research/claude-opus-4-5-system-card` (2025), accessed: 2026-06-01
4. Anthropic: Claude sonnet 4.6 model card. `https://www.anthropic.com/research/claude-sonnet-4-6-system-card` (2025), accessed: 2026-06-01
5. Anthropic: Introducing claude opus 4.6. `https://www.anthropic.com/news/claude-opus-4-6` (2026), model release announcement. Accessed: 2026-06-02
6. Baek, J., Miyai, A., Onohara, S., Ikuta, H., Aizawa, K.: Manga109-v2026: Revisiting Manga109 Annotations for Modern Manga Understanding. In: ICML Workshop (2026)
7. Bai, S., Chen, K., Liu, X., Wang, J., Ge, W., Song, S., Dang, K., Wang, P., Wang, S., Tang, J., et al.: Qwen2.5-vl technical report. arXiv preprint arXiv:2502.13923 (2025)
8. Cho, J., Mahata, D., Irsoy, O., He, Y., Bansal, M.: M3docrag: Multi-modal retrieval is what you need for multi-page multi-document understanding. arXiv preprint arXiv:2411.04952 (2024)
9. Cui, C., Sun, T., Liang, S., Gao, T., Zhang, Z., Liu, J., Wang, X., Zhou, C., Liu, H., Lin, M., Zhang, Y., Zhang, Y., Zheng, H., Zhang, J., Zhang, J., Liu, Y., Yu, D., Ma, Y.: Paddleocr-vl: Boosting multilingual document parsing via a 0.9b ultra-compact vision-language model. arXiv preprint arXiv:2510.14528 (2025)
10. Dietterich, T.G.: Approximate statistical tests for comparing supervised classification learning algorithms. Neural Computation **10**(7), 1895–1923 (1998)
11. Ding, Y., et al.: A survey on mllm-based visually rich document understanding. arXiv preprint arXiv:2507.09861 (2026)
12. Dong, K., Chang, Y., Huang, S., Wang, Y., Tang, R., Liu, Y.: Benchmarking retrieval-augmented multimodal generation for document question answering. arXiv preprint arXiv:2505.16470 (2025)
13. Efron, B., Tibshirani, R.J.: An Introduction to the Bootstrap. No. 57 in Monographs on Statistics and Applied Probability, Chapman & Hall/CRC, New York (1993)
14. Faysse, M., Sibille, H., Wu, T., Omrani, B., Viaud, G., Hudelot, C., Colombo, P.: Colpali: Efficient document retrieval with vision language models. arXiv preprint arXiv:2407.01449 (2024)
15. Han, S., et al.: Mdocagent: A multi-modal multi-agent framework for document understanding. arXiv preprint arXiv:2503.13964 (2025)
16. Hu, A., Xu, H., Zhang, L., Ye, J., Yan, M., Zhang, J., Jin, Q., Huang, F., Zhou, J.: mplug-docowl2: High-resolution compressing for ocr-free multi-page document understanding. arXiv preprint arXiv:2409.03420 (2024)
17. Hu, Y., Shi, W., Fu, X., Roth, D., Ostendorf, M., Zettlemoyer, L., Smith, N.A., Krishna, R.: Visual sketchpad: Sketching as a visual chain of thought for multimodal language models. In: Advances in Neural Information Processing Systems (2024)
18. Indrehus, K., et al.: Towards self-explainable document visual question answering. arXiv preprint arXiv:2605.06058 (2026)

19. Jain, C., Wu, Y., Zeng, Y., Liu, J., Dai, S., Shao, Z., Wu, Q., Wang, H.: Simpledoc: Multi-modal document understanding with dual-cue page retrieval and iterative refinement. arXiv preprint arXiv:2506.14035 (2025)
20. Jiang, B., Zhuang, Z., Shivakumar, S.S., Roth, D., Taylor, C.J.: Multi-agent vqa: Exploring multi-agent foundation models in zero-shot visual question answering. arXiv preprint arXiv:2403.14783 (2024)
21. Krubiński, M., Matcovici, S., Grigore, D., Voinea, D., Popa, A.I.: Watermark text pattern spotting in document images (2024), https://arxiv.org/abs/2401.05167
22. LangChain: Langgraph: Build resilient language agents as graphs (2024), https://github.com/langchain-ai/langgraph
23. Leotescu, G., Popa, A.I., Grigore, D.N.N., Voinea, D., Perona, P.: Self-supervised incremental learning of object representations from arbitrary image sets. In: Proceedings of the Winter Conference on Applications of Computer Vision (WACV). pp. 8133–8143 (February 2025)
24. Leotescu, G., Voinea, D., Popa, A.I.: Bidirectional long-range parser for sequential data understanding (2024), https://arxiv.org/abs/2404.05210
25. Liao, W., Wang, J., Li, H., Wang, C., Huang, J., Jin, L.: Doclayllm: An efficient and effective multi-modal extension of large language models for text-rich document understanding. arXiv preprint arXiv:2408.15045 (2024)
26. Liu, Y., Yang, B., Liu, Q., Li, Z., Ma, Z., Zhang, S., Bai, X.: Textmonkey: An ocr-free large multimodal model for understanding document. arXiv preprint arXiv:2403.04473 (2024)
27. Liu, Z., Dong, Y., Rao, Y., Zhou, J., Lu, J.: Chain-of-spot: Interactive reasoning improves large vision-language models. arXiv preprint arXiv:2403.12966 (2024)
28. Llabrés, A., Serra Ortega, M., Ockier, T., Singh, A., Georgakilas, C., Barsky, A., Valveny, E., Karatzas, D.: DocVQA2026: ICDAR2026 competition on multimodal reasoning over documents in multiple domains. https://www.docvqa.org/challenges/2026 (2026), cVC-UAB. Accessed: 2026-05-25
29. López, E., Llabrés, A., Valveny, E.: Enhancing document vqa models via retrieval-augmented generation. arXiv preprint arXiv:2508.18984 (2025)
30. Luan, B., Feng, H., Chen, H., Wang, Y., Zhou, W., Li, H.: Textcot: Zoom in for enhanced multimodal text-rich image understanding. arXiv preprint arXiv:2404.09797 (2024)
31. Matcovici, S., Voinea, D., Popa, A.I.: $k-$ NN embeded space conditioning for enhanced few-shot object detection. In: 2023 IEEE/CVF Winter Conference on Applications of Computer Vision Workshops (WACVW). pp. 401–410 (2023). https://doi.org/10.1109/WACVW58289.2023.00044
32. McNemar, Q.: Note on the sampling error of the difference between correlated proportions or percentages. Psychometrika **12**(2), 153–157 (1947)
33. Mo, Y., Shao, Z., Ye, K., Mao, X., Zhang, B., Xing, H., Ye, P., Huang, G., Chen, K., Huan, Z., Yan, Z., Zhou, S.: Doc-cob: Enhancing multi-modal document understanding with visual chain-of-boxes reasoning. arXiv preprint arXiv:2505.18603 (2025)
34. Mohammadshirazi, A., et al.: Arial: An agentic framework for document vqa with precise answer localization. arXiv preprint arXiv:2511.18192 (2025)
35. Paruchuri, V., Datalab: Surya: Ocr, layout analysis, reading order, and table recognition. https://github.com/datalab-to/surya (2024), software package. Accessed: 2026-06-02
36. Qin, Y., Wei, B., Ge, J., Kallidromitis, K., Fu, S., Darrell, T., Wang, X.: Chain-of-visual-thought: Teaching vlms to see and think better with continuous visual tokens. arXiv preprint arXiv:2511.19418 (2025)

37. Qwen Team: Qwen3.5: Towards native multimodal agents (February 2026), https://qwen.ai/blog?id=qwen3.5
38. Reddy, N.D., Snyder, D., Kiragu, L., Mohin, M., Bin Amin, S., Pillai, S.: Orion: A unified visual agent for multimodal perception, advanced visual reasoning and execution. arXiv preprint arXiv:2511.14210 (2025)
39. Rodriguez, J.A., Jian, X., Panigrahi, S.S., Vazquez, D., Rajeswar, S., et al.: Bigdocs: An open dataset for training multimodal models on document and code tasks. In: Proceedings of ICLR (2025)
40. Sandu, I.C., Voinea, D., Popa, A.I.: Large sequence representation learning via multi-stage latent transformers. In: COLING. International Committee on Computational Linguistics, Gyeongju, Republic of Korea (Oct 2022)
41. Sun, L., et al.: Docagent: An agentic framework for multi-modal long-context document understanding. In: Proceedings of EMNLP (2025)
42. Tanaka, R., Iki, T., Hasegawa, T., Nishida, K., Saito, K., Suzuki, J.: Vdocrag: Retrieval-augmented generation over visually-rich documents. arXiv preprint arXiv:2504.09795 (2025)
43. Vivoli, E., Bertini, M., Karatzas, D.: Comix: A comprehensive benchmark for multi-task comic understanding. arXiv preprint arXiv:2407.03550 (2024)
44. Wang, Z., Ma, D., Zhong, H., Li, J., Zhang, W., Wang, B., He, C.: Agenticocr: Parsing only what you need for efficient retrieval-augmented generation. arXiv preprint arXiv:2602.24134 (2026)
45. Wang, Z., Guan, T., Fu, P., et al.: Marten: Visual question answering with mask generation for multi-modal document understanding. In: Proceedings of CVPR (2025)
46. Wen, S., Zhang, Z., Bian, X., Zhu, H., He, L., Shama, L., Ergu, D., Cai, Y.: Ocr-agent: Agentic ocr with capability and memory reflection. arXiv preprint arXiv:2602.21053 (2026)
47. Wilson, E.B.: Probable inference, the law of succession, and statistical inference. Journal of the American Statistical Association **22**(158), 209–212 (1927)
48. Yin, S., et al.: Toolvqa: A dataset for multi-step reasoning vqa with external tools. In: Proceedings of ICCV (2025)
49. Yin, S., Fu, C., Zhao, S., Li, K., Sun, X., Xu, T., Chen, E.: A survey on multimodal large language models. National Science Review (2024), also available as arXiv:2306.13549
50. Yu, S., Tang, C., Xu, B., Cui, J., Ran, J., Yan, Y., Liu, Z., Wang, S., Han, X., Liu, Z., Sun, M.: Visrag: Vision-based retrieval-augmented generation on multi-modality documents. In: Proceedings of ICLR (2025)
51. Yu, W., Chen, W., Qi, G., Li, W., Li, Y., Sha, L., Xia, D., Huang, J.: Bbox docvqa: A large scale bounding box grounded dataset for enhancing reasoning in document visual question answer. arXiv preprint arXiv:2511.15090 (2025)
52. Zhang, S., Yin, M., Zhang, J., Liu, J., Han, Z., Zhang, J., Li, B., Wang, C., Wang, H., Chen, Y., Wu, Q.: Which agent causes task failures and when? on automated failure attribution of LLM multi-agent systems. In: Forty-second International Conference on Machine Learning (2025), https://openreview.net/forum?id=GazlTYxZss
53. Zhang, T., Popa, A.I., Xu, Y., Song, R., Dimitriadis, D.: Pivot: Bridging planning and execution in llm agents via trajectory refinement (2026), https://arxiv.org/abs/2605.11225