MIRA: A MULTIMODAL ILLUSTRATED DEEP RESEARCH AGENT FOR CONTROLLABLE REPORTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Deep research agents are increasingly capable of searching the open web and synthesizing evidence into comprehensive answers, yet existing agents remain poorly suited to **controllable report generation in multimodal deep research**, a challenging task in which users provide interleaved textual and visual inputs, agents must gather multimodal evidence through multi-step web search, and users expect a comprehensive, illustrated report rather than a short answer. Two limitations are particularly prominent. During reasoning, existing search agents often underutilize visual information and lack effective mechanisms for interacting with images discovered throughout the search trajectory. During generation, they typically produce brief responses rather than comprehensive reports, while relying solely on base models provides limited user control over three report-level attributes: report granularity, multimodal composition, and evidence grounding. To address these challenges, we introduce **MIRA**, a **M**ultimodal **I**llustrated Deep **R**esearch **A**gent for controllable reports. MIRA tightly integrates textual and visual evidence throughout the search process, enabling the agent to inspect, reason over, and reuse intermediate images while preserving strong web-search capabilities. At its core, we develop MIRA-Harness, which translates user requirements into generation contracts over report granularity, multimodal composition, and evidence grounding, and provides four complementary execution stages for improving report quality. Building on this harness, we develop MIRA-Data, a data construction pipeline for producing high-quality multi-hop multimodal research trajectories, and post-train foundation models to operate effectively within the harness. Together, MIRA achieves **state-of-the-art performance** among open agents on 7 representative, established multimodal-search and deep-research-agent benchmarks, while producing user-aligned illustrated reports.

1 INTRODUCTION

Multimodal large language models (MLLMs) acquire extensive visual and textual knowledge from static training corpora (Alayrac et al., 2022; Li et al., 2023; Liu et al., 2023; Bai et al., 2023; 2025). External tools extend this fixed knowledge boundary, enabling models to retrieve evidence and address real-world information needs (Yao et al., 2023; Yang et al., 2023; Jin et al., 2025; Chen et al., 2026; Liu et al., 2026b). Text-based deep research systems such as STORM use multi-step retrieval to write long-form articles (Shao et al., 2024). WebThinker trains an agent to interleave web search and report drafting (Li et al., 2025c). Multimodal deep research extends this setting to textual and visual evidence (Huang et al., 2026a). However, tool access alone does not ensure an illustrated report that simultaneously provides accurate and sufficiently complete information (Li et al., 2025a), integrates textual and visual evidence throughout research and writing (Pham et al., 2026; Liu et al., 2026a), and follows the user's preferred style.

Motivated by real-world reporting needs, we study **controllable report generation**: producing illustrated reports that respond to user requirements in multimodal deep research. Given interleaved textual and visual inputs alongside explicit delivery requirements, an agent must gather evidence from the open web and satisfy constraints on three attributes: *report granularity*, the scope, depth, and level of detail; *multimodal composition*, the selection, balance, and organization of text and visuals; and *evidence grounding*, the required evidential support and explicit links from textual claims to source documents (Gao et al., 2023) and from report visuals to source figures (Shi et al., 2026).

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

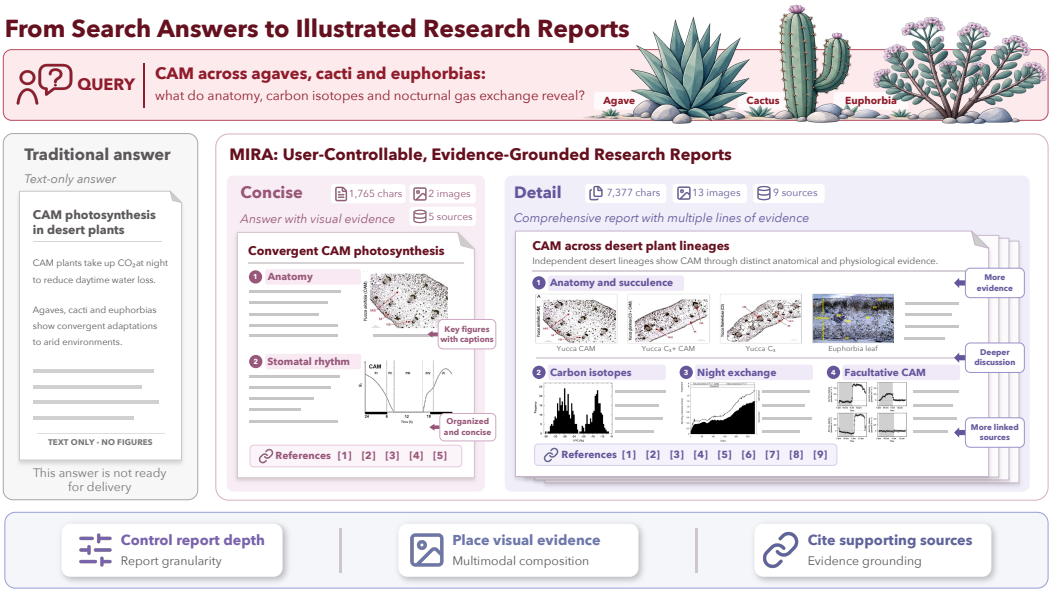


Figure 1: **From search answers to controllable illustrated reports.** For the same multimodal research request, MIRA supports Concise and Detail deliverables that differ in exposition and visual coverage while retaining traceable source links.

Recent work addresses important components of this task. Vision Search Assistant (Zhang et al., 2024) uses external search to answer questions about visual inputs. MMSearch-R1 (Wu et al., 2026) and Vision-DeepResearch (Huang et al., 2026b) integrate textual and visual search into multi-step reasoning to answer visual questions that require external knowledge. Moving toward report generation (Ye et al., 2026), Multimodal DeepResearcher (Yang et al., 2026) explores interleaved text and chart generation, ViDR (Shi et al., 2026) grounds reports in source figures, and Ptah (Zhang et al., 2026) coordinates illustrated report generation and verification through a multi-agent harness. Together, these advances provide a foundation for studying how different reporting needs can be supported within a unified research workflow.

Nevertheless, jointly controlling report granularity, multimodal composition, and evidence grounding remains challenging. Even for the same research question, a concise brief and a detailed report may require different evidence coverage, visual choices, and levels of exposition. These requirements should therefore shape the research process, rather than serve only as formatting instructions during final generation. This raises a central question: *How should the user's intended deliverable guide evidence collection, report organization, and the decision to conclude research?*

To address this question, we present **MIRA**, a **M**ultimodal **I**llustrated Deep **R**esearch **A**gent for controllable reports. At its core, MIRA-Harness connects evidence collection, illustrated writing, and delivery checks in an iterative loop. The harness represents user requirements as *generation contracts*, which specify explicit conditions for accepting a report, and supports *Concise* and *Detail* modes for different reporting needs (Figure 1). Both modes specify requirements across all three attributes while preserving a shared requirement for source traceability. During research, the agent inspects and records sources and visual materials. After drafting, it submits the report for checking, uses unmet conditions as revision targets, and revises the content or gathers additional evidence before resubmitting within the allocated budget. A report is accepted as a completed deliverable only after passing the implemented contract checks.

To teach models this complete workflow, we develop MIRA-Data, a framework for constructing report-oriented training trajectories that connect evidence gathering, writing, submission, and feedback-driven revision. Using Qwen3.6-35B-A3B as the backbone, we compare the unadapted model (Base) with its supervised fine-tuned counterpart (SFT). Our evaluation examines both report delivery and multimodal search, assessing how adaptation affects report quality, adherence to user requirements, and the underlying ability to acquire evidence.

Our contributions are threefold:

- **Task formulation.** We identify the gap between multimodal deep research and real-world report delivery, and formulate controllable report generation: producing illustrated, evidence-grounded reports under user-specified requirements for report granularity, multimodal composition, and evidence grounding.
- **Harness and training data.** We introduce MIRA-Harness, which unifies user requirements, multimodal evidence collection, illustrated writing, and feedback-driven revision through generation contracts. We further develop MIRA-Data to construct training trajectories aligned with this workflow, covering evidence gathering, writing, submission checks, and revision.
- **Empirical evaluation.** We train and evaluate MIRA using Qwen3.6-35B-A3B, comparing Base and SFT variants across report delivery and multimodal search. Across eight multimodal search, knowledge-intensive VQA, and illustrated deep-research benchmarks, MIRA-SFT achieves state-of-the-art performance among the compared open agents on seven of the eight benchmarks. Beyond benchmark performance, MIRA can generate comprehensive illustrated reports that adapt to user-specified requirements for report granularity, multimodal composition, and evidence grounding, while integrating visual evidence and traceable sources into the final deliverable.

2 Related Work

2.1 Deep Research and Multimodal Search Agents

Deep research extends tool-augmented language models from single-step retrieval to long-horizon information seeking, where an agent iteratively searches, inspects external evidence, and synthesizes an answer. Early agentic formulations such as ReAct (Yao et al., 2023) established the general paradigm of interleaving reasoning with environmental actions, while systems such as STORM (Shao et al., 2024) demonstrated that multi-stage retrieval and organization can support substantially longer and more structured textual outputs. More recent work has shifted attention toward training search agents directly, including scalable long-horizon search and synthetic supervision (Chu et al., 2026; Xie et al., 2026), showing that effective research behavior can be learned, not merely prompted.

Multimodal research agents further expand this setting by allowing visual observations to participate in the search process. Benchmarks such as MMSearch, MMSearch+, and MM-BrowseComp (Jiang et al., 2024; Tao et al., 2026; Li et al., 2025b) expose the need to retrieve and reason over both textual and visual evidence, while MMSearch-R1 (Wu et al., 2026), WebWatcher (Geng et al., 2026), OpenSearch-VL (Chen et al., 2026), Vision-DeepResearch (Huang et al., 2026b), and WeAgent-MMSearch (Liu et al., 2026b) develop increasingly capable multimodal browsing and search policies. These works substantially improve how agents acquire external evidence, particularly for knowledge-intensive visual questions.

2.2 Agent Harnesses and Tool-Use Supervision

The behavior of a research agent is determined not only by its underlying model, but also by the execution environment that exposes tools, maintains interaction state, and mediates long-horizon reasoning. Existing search systems therefore use tool environments and agent harnesses to coordinate retrieval and multimodal inspection (Chen et al., 2026; Liu et al., 2026b). WeAgent-Harness additionally exposes webpage extraction and code execution (Liu et al., 2026b). In parallel, recent work on search-agent training constructs trajectories of tool calls and intermediate observations so that models can internalize effective interaction patterns (Chu et al., 2026; Xie et al., 2026). This line of work shows that runtime structure and training supervision are complementary. A capable backbone may still fail to search effectively when the interaction protocol is poorly specified, while training on trajectories mismatched to the deployment environment can teach behaviors that are difficult to realize at inference time.

2.3 Controllable and Evidence-Grounded Report Generation

A separate line of research moves beyond short-form question answering toward long-form, evidence-backed synthesis. FActScore (Min et al., 2023) evaluates the factual support of claims

in long-form text. Citation-aware generation studies how model outputs can be explicitly linked to retrieved sources (Gao et al., 2023), while systems such as EviReport (Liu et al., 2026a) organize evidence throughout long-form report generation. Deep-Reporter (Ye et al., 2026) further studies grounded multimodal long-form generation. ReportBench (Li et al., 2025a) evaluates the content quality of research reports for academic survey tasks, while AnalystBench (Pham et al., 2026) evaluates professional report generation in settings where completeness and multimodal presentation matter in addition to factual correctness. These efforts highlight that high-quality research outputs require more than finding a correct answer: evidence must be selected, organized, and preserved across a substantially longer generation process.

Recent multimodal deep-research systems make this connection even more explicit. Multimodal DeepResearcher (Yang et al., 2026) studies interleaved text-chart report generation, ViDR (Shi et al., 2026) grounds generated reports in source visual evidence, and Ptah (Zhang et al., 2026) introduces a multi-agent harness for interleaved report generation and verification. MMDeepResearch-Bench (Huang et al., 2026a) correspondingly evaluates multimodal agents on long-form research synthesis. Collectively, these works establish the importance of illustration, provenance, and verification for multimodal reports. Accordingly, MIRA targets report delivery under explicit user requirements.

3 METHOD

3.1 BACKGROUND

In tool-using multimodal deep research, an *agent harness* mediates long-horizon interactions between a model and its environment. This execution layer routes actions, returns observations, maintains state, and enforces budgets (Liu et al., 2026b; Chen et al., 2026). Given a possibly multimodal request q and model-visible history h_t, the policy π_θ, parameterized by θ, selects a tool action a_t at turn t, and the harness $\mathcal{H}$ returns an observation o_t. Let $\oplus$ append an action and observation pair to the history. The interaction loop (Yao et al., 2023) is

$$a_t \sim \pi_\theta(\cdot \mid q, h_t), \qquad o_t = \mathcal{H}(a_t, h_t), \qquad h_{t+1} = h_t \oplus (a_t, o_t). \tag{1}$$

Here, o_t may contain text, image pixels, or tool errors. Over T turns, the complete interaction forms a trajectory τ; the shorthand $\tau = (a_t, o_t)_{t=1}^{T}$ omits intervening model text and internal harness state. The harness thus structures the agent's evidence-gathering loop, rather than merely exposing a list of tools.

Building on this loop, controllable report generation in multimodal deep research requires an illustrated, evidence-grounded report under user-specified requirements, so delivery requirements must shape evidence gathering and revision. MIRA-Harness incorporates these requirements into runtime interaction, while MIRA-Data supplies trajectories for the same workflow.

3.2 MIRA-HARNESS FOR USER-SPECIFIED ILLUSTRATED REPORTS

MIRA-Harness operationalizes controllable report generation through a generation contract C_m conditioned on the user-selected report mode m, as illustrated in Figure 2. The contract groups its acceptance conditions into three sets: report granularity C_m^{detail}, multimodal composition C_m^{images}, and evidence grounding C_m^{ref}. It enforces this contract through four stages: **(1)** evidence collection, **(2)** report submission, **(3)** feedback-driven revision, and **(4)** final delivery.

For a possibly multimodal request q, the mode $m \in \{\texttt{CONCISE}, \texttt{DETAIL}\}$ selects a report preset, yielding $x = (q, m)$ and the contract

$$C_m = C_m^{\text{detail}} \cup C_m^{\text{images}} \cup C_m^{\text{ref}}. \tag{2}$$

Let R_t be the submitted draft, $\mathcal{E}_t$ the evidence accumulated by turn t, and $\mathcal{G}_t \equiv \mathcal{G}(R_t, \mathcal{E}_t; C_m)$ the report-gate response. The action $\texttt{SUBMIT}(R_t)$ sends the draft to this gate; $\texttt{WARN}(f_t)$ identifies unmet conditions f_t. If another submission is required, let $t^+ > t$ denote its turn and $\mathcal{G}_{t^+} \equiv \mathcal{G}(R_{t^+}, \mathcal{E}_{t^+}; C_m)$ its response. An $\texttt{OK}$ accepts the current draft as R^{acc}, with the same branch rule applied after each resubmission:

$$x \xrightarrow{(1)} (h_t, \mathcal{E}_t) \xrightarrow{(2)\ \texttt{SUBMIT}(R_t)} \mathcal{G}_t \mapsto \left[\texttt{WARN}(f_t) \xrightarrow{(3)} \mathcal{G}_{t^+} \;\middle|\; \texttt{OK} \xrightarrow{(4)} R^{\text{acc}} \right]. \tag{3}$$

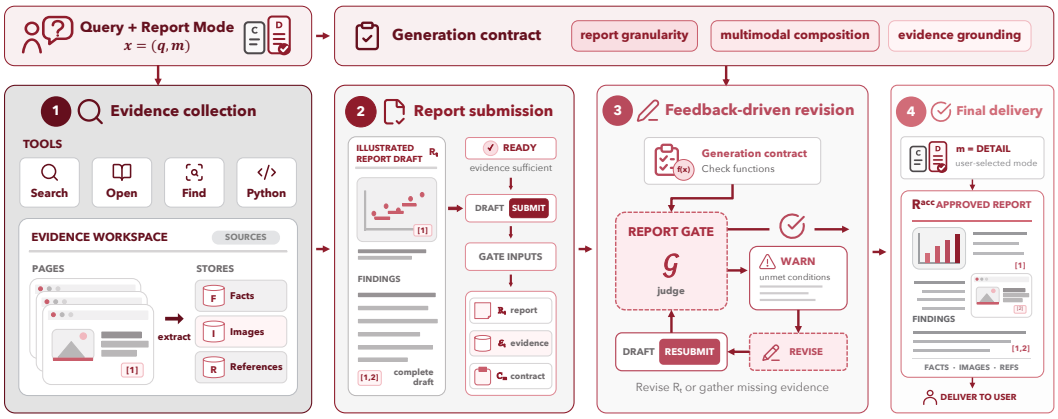


Figure 2: **MIRA-Harness for user-specified illustrated reports.** A mode-conditioned generation contract guides multimodal research, evidence tracking, report submission, and feedback-driven revision, and delivers the report once the contract conditions are satisfied.

Stage 1: Evidence collection and Stage 2: Report submission.

The first stage, evidence collection, instantiates the action space with 4 tool calls, keeping the interface elegant and consistent across foundation models without proliferating specialized, task-specific operators.

The available calls are `search(query)`, `open(target[, loc, num_lines])`, `find(url, pattern)`, and `Python(code)`. The `search` call discovers candidates, `open` returns webpage content or image pixels, `find` locates text in an opened page, and `Python` processes observed content or produces derived figures. These four calls comprise the research action space; `SUBMIT` is a separate workflow action rather than a research tool. Appendix B gives the detailed tool and submission interfaces and an abridged inference system prompt. Relative to Eq. 1, MIRA-Harness retains the history update, but conditions the policy on x and makes the evidence state $\mathcal{E}_t$ explicit. This state records extracted facts, inspected images, opened source pages, and their provenance; $\mathcal{H}_m$ denotes MIRA-Harness under mode m:

$$a_t \sim \pi_\theta(\cdot \mid x, h_t), \qquad (o_t, \mathcal{E}_{t+1}) = \mathcal{H}_m(a_t, h_t, \mathcal{E}_t). \tag{4}$$

With sufficient evidence, the model enters report submission and invokes $\texttt{SUBMIT}(R_t)$. This sends the complete draft to $\mathcal{G}$, while the harness supplies $\mathcal{E}_t$ and C_m.

Stage 3: Feedback-driven revision and Stage 4: Final delivery.

The report gate returns the `OK` or `WARN` response in Eq. 3. A $\texttt{WARN}(f_t)$ enters stage 3, where the model revises R_t, gathers missing evidence, and resubmits within its budget. An `OK`, received immediately or after revision, enters stage 4 and delivers R^{acc}; budget exhaustion instead fails the delivery. Thus, the final report satisfies the user-specified contract.

3.3 MIRA-Data: Trajectory Synthesis for Illustrated Reports

Curating query seeds. MIRA-Data draws seeds from knowledge-intensive topics and real-world, time-sensitive information needs. For each seed, the query generator produces a user-facing request q and an internal rubric

$$\mathcal{U}(q) = \{u_i\}_{i=1}^{K}, \qquad K > 1. \tag{5}$$

Here, K counts rubric items, and each u_i specifies a subquestion and its visual-evidence requirement. Every subquestion requires externally retrieved evidence, and at least one requires visual evidence. The user sees q, while $\mathcal{U}(q)$ remains internal for the synthesis gate; MIRA-Harness adds m and C_m. Together, these conditions require multi-step retrieval and explicit visual evidence rather than an answer produced from parametric knowledge alone. Figure 3 summarizes the MIRA-Data construction pipeline.

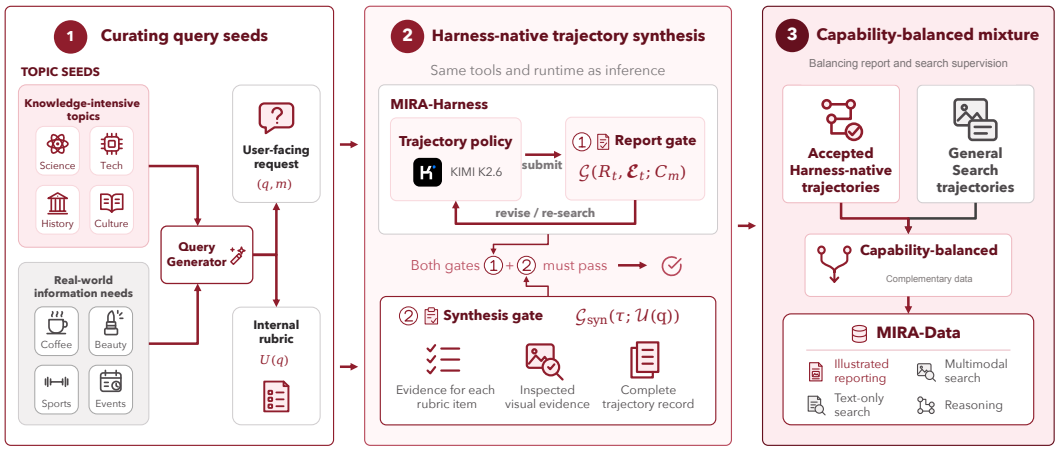


Figure 3: **Trajectory synthesis for illustrated reports.** A query generator constructs report-oriented requests and internal rubrics. Kimi K2.6 serves as the trajectory-generating policy within MIRA-Harness, after which the synthesis gate verifies rubric coverage and trajectory integrity. Accepted trajectories are mixed with general search trajectories to form MIRA-Data.

Harness-native trajectory synthesis. For each q, Kimi K2.6 (Moonshot AI, 2026) serves as the trajectory policy within MIRA-Harness. Given $x = (q, m)$, it reuses the tools, $\mathcal{E}_t$, C_m, and $\mathcal{G}$ from the inference workflow to collect evidence and submit an illustrated report; gate feedback triggers revision or further collection. The generic report gate $\mathcal{G}(R_t, \mathcal{E}_t; C_m)$ checks delivery requirements but does not receive $\mathcal{U}(q)$, so it cannot verify coverage of every requested item. Data synthesis therefore adds a synthesis gate $\mathcal{G}_{\text{syn}}$:

$$\mathcal{G}_{\text{syn}}\big(\tau; \mathcal{U}(q)\big) \in \{\texttt{OK}, \texttt{REJECT}\}. \tag{6}$$

This gate checks evidence coverage for every u_i, image inspection for visual items, and the completeness of the record of tool calls, image observations, and the accepted report. Only trajectories for which both $\mathcal{G}$ and $\mathcal{G}_{\text{syn}}$ return `OK` enter MIRA-Data.

Capability-balanced mixture. To balance report-specific behavior with broader search capabilities, we complement the accepted Harness-Native trajectories with General Search Trajectories during training. The former introduces generation-contract-driven research behavior, while the latter preserves broader multimodal and text-based search supervision. We provide the data composition in the experimental setup.

4 Experiments

4.1 Experimental Setup

Models and Benchmarks. We build MIRA on Qwen3.6-35B-A3B (Qwen Team, 2026b). We evaluate multimodal search and knowledge-intensive question answering on MMSearch (Jiang et al., 2024), MMSearch+ (Tao et al., 2026), BrowseComp-VL (Geng et al., 2026), SimpleVQA (Cheng et al., 2025), LiveVQA (Fu et al., 2025), FVQA (Wang et al., 2018), and VDR-Bench (Zeng et al., 2026). We further evaluate illustrated report generation on MMDeepResearch-Bench (Huang et al., 2026a). Together, these eight benchmarks cover web retrieval, multimodal evidence use, long-tail knowledge, and long-form report synthesis.

Baselines. We compare against three large-scale frontier models, DeepSeek-V4.1-Flash (DeepSeek-AI, 2026), Kimi K3 (high reasoning effort) (Kimi Team et al., 2026), and Qwen3.5-397B-A17B (Qwen Team, 2026a), together with four lightweight multimodal search agents at a comparable scale: OpenSearch-VL (Chen et al., 2026), REDSearcher-MM (Chu et al., 2026), Vision-DeepResearch (Huang et al., 2026b), and WeAgent-MMSearch (Liu et al., 2026b). We also compare the Qwen3.6 backbone without a harness, under the WeAgent-MMSearch Harness, under a vanilla

Table 1: Main results across eight multimodal search, knowledge-intensive VQA, and illustrated deep-research benchmarks.

Model	Params	MM Search	MM Search+	Browse Comp-VL	Simple VQA	Live VQA	FVQA	VDR Bench	MMDeep Research
Large-Scale Frontier Models									
Kimi-K3	2.8T (104B)	89.26	60.00	60.32	83.33	91.23	82.46	22.86	54.32
Qwen3.5	397B (17B)	69.01	36.01	58.40	77.33	84.33	75.33	14.40	39.54
DeepSeek-V4.1-Flash	552B (8B/16B)	88.33	59.91	72.86	81.33	88.67	85.00	32.20	51.23
Lightweight Multimodal Search Agents									
OpenSearch-VL	30B (3B)	68.70	28.94	41.10	74.90	67.40	73.20	33.50	23.21
REDSearcher-MM	30B (3B)	72.90	26.60	57.20	–	79.30	–	–	–
Vision-DeepResearch	30B (3B)	69.60	28.50	53.70	66.33	77.60	74.20	**37.80**	12.78
WeAgent-MMSearch	30B (3B)	77.78	37.82	–	77.73	81.05	**79.33**	35.68	–
Qwen3.6 w/o Harness	35B (3B)	30.41	3.86	28.91	58.00	56.67	34.11	1.80	26.78
Qwen3.6 w/ WeAgent Harness	35B (3B)	66.28	29.40	–	65.79	78.20	66.09	15.57	–
Qwen3.6 w/ Vanilla Harness	35B (3B)	67.84	28.94	51.01	72.24	73.67	65.67	15.00	37.95
Ours									
MIRA-Base	35B (3B)	73.10	33.44	56.64	74.33	78.33	70.00	19.00	41.90
Δ (MIRA-Base − Vanilla)		**+5.26**	**+4.50**	**+5.63**	**+2.09**	**+4.66**	**+4.33**	**+4.00**	**+3.95**
MIRA-SFT	35B (3B)	**80.70**	**38.91**	**58.40**	**79.00**	**82.00**	**79.33**	29.80	**42.61**
Δ (MIRA-SFT − MIRA-Base)		**+7.60**	**+5.47**	**+1.76**	**+4.67**	**+3.67**	**+9.33**	**+10.80**	**+0.71**

harness, and under MIRA-Harness to separate model capability from runtime scaffolding. Model variants and their results are detailed in Section 4.2.

Evaluation Metrics and Protocol. For all locally evaluated configurations, we use a single rollout per query (Pass@1) because each long-horizon agent rollout is computationally expensive. DeepSeek-V4.1-Flash serves as the semantic judge. Within each benchmark, we keep its prompt, answer extraction, output parser, and aggregation rule fixed across systems. We report each benchmark's official primary metric and the Overall score for MMDeepResearch-Bench.

Report Modes. We instantiate the generation contract with the two presets summarized below. `CONCISE` targets a direct, evidence-supported response, whereas `DETAIL` targets an expanded illustrated report. Visible length counts non-whitespace prose characters. Images contribute to the minimum only after their pixels have been inspected; source pages must be opened and cited. Both modes use the same research tools and report gate, with the selected preset determining the required deliverable. The table lists the core thresholds; Appendix A.1 gives the full counting rules and shared evidence requirements.

Mode	Visible characters	Inspected images	Cited sources
`CONCISE`	300–2,000	≥ 2	≥ 2
`DETAIL`	5,000–10,000	≥ 10	≥ 8

Data. Following Section 3.3, our data comprises approximately 4.6K Harness-Native trajectories designed for the MIRA-Harness workflow and 33K general search trajectories. The Harness-Native trajectories are generated by instantiating Kimi K2.6 (Moonshot AI, 2026) as the trajectory-generating policy within MIRA-Harness. Within this set, 36.9% follow the Concise mode and 63.1% follow the Detail mode. Queries in the general pool are drawn from open-source resources including OpenSearch-VL (Chen et al., 2026), REDSearcher (Chu et al., 2026), and QUEST (Xie et al., 2026), together with a subset synthesized by us. This pool contains both multimodal and text-only trajectories, balancing report-specific behavior with broader multimodal search, text-only search, and reasoning capabilities so that these abilities can improve together. This composition connects report-specific specialization with the broader multimodal, text-based, and reasoning skills needed to sustain long-horizon search, evidence gathering, and report construction, rather than optimizing either capability in isolation.

4.2 Main Results

Table 1 compares MIRA with the selected frontier models and similarly sized multimodal search agents across all eight benchmarks. Overall, **MIRA-SFT**, obtained by supervised fine-tuning the Qwen3.6 backbone on MIRA-Data and deploying it with MIRA-Harness, achieves state-of-the-art results among the listed open agents on seven of the eight benchmarks. While a gap remains to the strongest frontier models overall, **MIRA-SFT** surpasses the substantially larger Qwen3.5-397B-A17B on six benchmarks and matches it on BrowseComp-VL, demonstrating that a 35B-A3B backbone can remain competitive at the frontier scale.

First, without any harness, we let the Qwen3.6-35B-A3B backbone answer directly without tool access. Across all eight benchmarks, this no-harness baseline scores 25.78 points lower on average than **MIRA-Base**, which applies MIRA-Harness to the same model. It also performs worse than the same model with the Vanilla Harness (a basic harness that provides tool access without additional design; see Appendix A for implementation details) or the WeAgent-MMSearch Harness on every benchmark they report. These consistent gaps show that multimodal deep research depends not only on the backbone, but also on deliberate harness design to coordinate long-horizon tool use, evidence feedback, and interaction state.

Second, **MIRA-Base** measures the contribution of harness design because it uses the original backbone without MIRA-Data training. As summarized by the Δ **(MIRA-Base − Vanilla)** row, **MIRA-Base** improves every directly comparable benchmark and gains 4.35 points on average. It also outperforms the WeAgent-MMSearch Harness by 4.48 points across their six common benchmarks (using author-reported results because WeAgent-MMSearch is not publicly released). These gains show that MIRA derives its strength from carefully structuring the entire interaction loop around a compact, model-agnostic tool set, rather than expanding that set with task-specific tools.

Finally, the Δ **(MIRA-SFT − MIRA-Base)** row quantifies the effect of training on the carefully synthesized and filtered MIRA-Data. **MIRA-SFT** improves all eight benchmarks by 5.50 points on average, with its three largest gains on VDR-Bench (+10.80), FVQA (+9.33), and MMSearch (+7.60). Together, these gains show that MIRA-Data extends the benefits of runtime design into learned harness-native behavior, enabling **MIRA-SFT** to achieve state-of-the-art performance among the compared open-source agents on seven of the eight benchmarks.

4.3 Ablation Studies

We conduct two ablations to evaluate the report gate $\mathcal{G}$ and MIRA-Data. The Report Gate study shows that feedback improves delivery compliance and evidence use, while quantifying the additional model and tool calls required. The MIRA-Data study shows that training improves control over the requested report mode without reducing research quality or mode-switching efficiency. Both studies use 32 queries sampled from MMDeepResearch-Bench (MMDR) under a 100-turn budget; those in the first ablation cover 81 input images.

4.3.1 Ablation of the Report Gate

We compare otherwise identical runs with and without the report gate $\mathcal{G}$ on the sampled MMDeepResearch-Bench queries, holding the model and report mode fixed so that observed differences reflect the report gate.

Table 2: Report-gate ablation on MMDeepResearch-Bench Detail queries.

Metric	Unit	No Gate	With Gate	Δ
Compliant reports	per report	0/32	25/32	+78.1 pp
Checks passed	per report	1.78	3.62	+1.84
Images cited	per image	19/81 (23.5%)	75/81 (92.6%)	+69.1 pp
Opened sources	per query	5.31	8.69	+3.38
Model calls	per query	43.5	63.6	+20.1
Tool calls	per query	42.5	61.0	+18.5
MMDR Overall ($n = 32$)	per query	43.19	47.73	+4.54

Performance. We evaluate the report gate, a dedicated MIRA-Harness component that checks submitted reports and provides revision feedback, along two dimensions: delivery compliance and evidence use. In Table 2, *Compliant reports* counts final reports that pass all four Gate checks, while *Checks passed* averages the conditions satisfied per report. With the report gate $\mathcal{G}$ enabled, compliant reports rise from 0/32 to 25/32 and checks passed from 1.78 to 3.62. *Images cited* is the proportion of the 81 input images cited in the final reports, whereas *Opened sources* averages cited pages that the agent actually opened per query. These measures rise from 23.5% to 92.6% and from 5.31 to 8.69, respectively. MMDR Overall also rises from 43.19 to 47.73. Together, these gains demonstrate that the report gate $\mathcal{G}$ effectively turns report requirements into compliant, evidence-traceable deliveries without sacrificing benchmark-level quality.

Efficiency. Enabling the report gate $\mathcal{G}$ adds 20.1 model calls and 18.5 tool calls per query. This measured overhead reflects the targeted investigation and revision triggered by Gate feedback, showing how detected deficiencies are converted into improved delivery compliance.

4.3.2 Ablation of MIRA-Data

We test whether the MIRA-SFT model follows the user-selected report mode while preserving research quality by comparing the MIRA-Base and MIRA-SFT models on the sampled MMDeepResearch-Bench queries under both report modes, with the report gate $\mathcal{G}$ fixed across configurations so that differences reflect MIRA-Data.

Table 3: Average changes after switching from Concise to Detail. Positive values indicate increases; brackets for calls and MMDR Overall give 95% confidence intervals.

Metric	**MIRA-Base** (Detail − Concise)	**MIRA-SFT** (Detail − Concise)
Visible characters	$+\mathbf{2{,}573}$	$+\mathbf{2{,}695}$
Model calls	$+\mathbf{2.2}$ $[-5.0, 9.5]$	$-\mathbf{3.8}$ $[-15.1, 7.8]$
Tool calls	$+\mathbf{2.0}$ $[-5.1, 9.3]$	$-\mathbf{3.9}$ $[-15.3, 7.6]$
MMDR Overall	$-\mathbf{10.9}$ $[-16.6, -5.4]$	$+\mathbf{0.0}$ $[-6.6, 6.0]$

Performance. In Table 3, *Visible characters* measures the change in report length from Concise to Detail; it increases by 2,573 for MIRA-Base and 2,695 for MIRA-SFT. *MMDR Overall* measures the corresponding change in research quality: MIRA-Base decreases by 10.9 points, whereas MIRA-SFT remains unchanged (+0.0). MIRA-SFT therefore converts greater requested granularity into a longer report without sacrificing benchmark-level quality.

Efficiency. In Table 3, *Model calls* and *Tool calls* change by only +2.2 and +2.0 for MIRA-Base, and by -3.8 and -3.9 for MIRA-SFT. Thus, MIRA-SFT's higher Detail score comes not from additional model or tool calls, but from the report-mode control learned from MIRA-Data.

5 Conclusion

We introduce **MIRA**, a multimodal deep research agent for controllable illustrated report generation. MIRA encodes user requirements for report granularity, multimodal composition, and evidence grounding as generation contracts. MIRA-Harness connects multimodal evidence collection, report writing, checking, and feedback-driven revision. We further construct MIRA-Data and supervised fine-tune the Qwen3.6-35B-A3B backbone for this workflow. Our study highlights that report requirements should guide both final generation and the underlying research process, a step toward agents that adapt their search behavior and deliverables to user needs. We hope this perspective guides research on agents that reason about both what to find and what to deliver.

Limitations. MIRA relies on external search infrastructure, including Serper API and Jina Reader, to access and parse web content. Because these components operate outside the agent, transient API errors, unavailable webpages, or parsing failures can interrupt evidence collection and reduce trajectory reliability. Hallucinations may also arise during evidence acquisition and report synthesis. The agent may misread or incorrectly combine retrieved content, or make weakly supported claims.

AI USE STATEMENT

Generative AI tools were used solely to aid and polish the writing of this paper, including improving grammar, clarity, and readability. All AI-assisted edits were reviewed by the authors. The authors take full responsibility for the scientific content, claims, and conclusions presented in the paper.

REFERENCES

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736, 2022.

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-VL: A versatile vision-language model for understanding, localization, text reading, and beyond. arXiv preprint arXiv:2308.12966, 2023.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-VL technical report. *arXiv preprint arXiv:2511.21631*, 2025.

Shuang Chen, Kaituo Feng, Hangting Chen, Wenxuan Huang, Dasen Dai, Quanxin Shou, Yunlong Lin, Xiangyu Yue, Shenghua Gao, and Tianyu Pang. OpenSearch-VL: An open recipe for frontier multimodal search agents. *arXiv preprint arXiv:2605.05185*, 2026.

Xianfu Cheng, Wei Zhang, Shiwei Zhang, Jian Yang, Xiangyuan Guan, Xianjie Wu, Xiang Li, Ge Zhang, Jiaheng Liu, Yuying Mai, et al. Simplevqa: Multimodal factuality evaluation for multimodal large language models. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4637–4646. IEEE, 2025.

Zheng Chu, Xiao Wang, Jack Hong, Huiming Fan, Yuqi Huang, Yue Yang, Guohai Xu, Chenxiao Zhao, Cheng Xiang, Shengchao Hu, et al. Redsearcher: A scalable and cost-efficient framework for long-horizon search agents. *arXiv preprint arXiv:2602.14234*, 2026.

DeepSeek-AI. DeepSeek-V4.1-Flash: Pushing the limits of KV cache compression, 2026. URL https://huggingface.co/deepseek-ai/DeepSeek-V4.1-Flash.

Mingyang Fu, Yuyang Peng, Dongping Chen, Zetong Zhou, Benlin Liu, Yao Wan, Zhou Zhao, Philip S. Yu, and Ranjay Krishna. Seeking and updating with live visual knowledge. *arXiv preprint arXiv:2504.05288v2*, 2025. URL https://arxiv.org/abs/2504.05288v2.

Tianyu Gao, Howard Yen, Jiatong Yu, and Danqi Chen. Enabling large language models to generate text with citations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 6465–6488, 2023.

Xinyu Geng, Peng Xia, Zhen Zhang, Xinyu Wang, Qiuchen Wang, Ruixue Ding, Chenxi Wang, Jialong Wu, Kuan Li, Yida Zhao, et al. WebWatcher: Breaking new frontiers of vision-language deep research agent. In *International Conference on Learning Representations*, volume 2026, pp. 61240–61271, 2026.

Peizhou Huang, Zixuan Zhong, Zhongwei Wan, Donghao Zhou, Samiul Alam, Xin Wang, Zexin Li, Zhihao Dou, Li Zhu, Jing Xiong, et al. Mmdeepresearch-bench: A benchmark for multimodal deep research agents. *arXiv preprint arXiv:2601.12346*, 2026a.

Wenxuan Huang, Yu Zeng, Qiuchen Wang, Zhen Fang, Shaosheng Cao, Zheng Chu, Qingyu Yin, Shuang Chen, Zhenfei Yin, Lin Chen, et al. Vision-deepresearch: Incentivizing deepresearch capability in multimodal large language models. *arXiv preprint arXiv:2601.22060*, 2026b.

Dongzhi Jiang, Renrui Zhang, Ziyu Guo, Yanmin Wu, Jiayi Lei, Pengshuo Qiu, Pan Lu, Zehui Chen, Chaoyou Fu, Guanglu Song, et al. MMSearch: Benchmarking the potential of large models as multi-modal search engines. *arXiv preprint arXiv:2409.12959*, 2024.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. Search-R1: Training LLMs to reason and leverage search engines with reinforcement learning. In *Proceedings of the Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=Rwhi91ideu.

Kimi Team, Tongtong Bai, Yifan Bai, Yiping Bao, Jianfeng Cai, Xinyuan Cai, Peizhou Cao, Yuxuan Cao, Ziwei Chai, Y. Charles, et al. Kimi K3: Open frontier intelligence. *arXiv preprint arXiv:2607.24653*, 2026. URL https://arxiv.org/abs/2607.24653.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pp. 19730–19742. PmLR, 2023.

Minghao Li, Ying Zeng, Zhihao Cheng, Cong Ma, and Kai Jia. Reportbench: Evaluating deep research agents via academic survey tasks. *arXiv preprint arXiv:2508.15804*, 2025a.

Shilong Li, Xingyuan Bu, Wenjie Wang, Jiaheng Liu, Jun Dong, Haoyang He, Hao Lu, Haozhe Zhang, Chenchen Jing, Zhen Li, et al. Mm-browsecomp: A comprehensive benchmark for multimodal browsing agents. *arXiv preprint arXiv:2508.13186*, 2025b.

Xiaoxi Li, Jiajie Jin, Guanting Dong, Hongjin Qian, Yongkang Wu, Ji-Rong Wen, Yutao Zhu, and Zhicheng Dou. WebThinker: Empowering large reasoning models with deep research capability. In *Advances in Neural Information Processing Systems*, volume 38, pp. 120091–120131, 2025c. doi: 10.52202/085713-4011. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/ae03bdef276132fae089692445725635-Abstract-Conference.html.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023.

Zihan Liu, Jianhui Li, Zexin Wang, Fei Sun, Jingjing Li, Zheyuan Li, Ke Xiang, Hang Cui, Houhua Gong, Changhua Pei, and Gaogang Xie. EviReport: From reasoned outlines to evidence tracked long-form reports. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 28024–28048. Association for Computational Linguistics, 2026a. doi: 10.18653/v1/2026.findings-acl.1397. URL https://aclanthology.org/2026.findings-acl.1397/.

Zongkai Liu, Hui Zhang, Liqiang Niu, Zhen Cao, Han Li, Juntao Liu, Wenchao Chen, Chengduo Zhao, Chao Yu, and Fandong Meng. WeAgent-MMSearch: Native text-vision interaction for multimodal search agents. *arXiv preprint arXiv:2608.28062*, 2026b.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. Factscore: Fine-grained atomic evaluation of factual precision in long form text generation. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pp. 12076–12100, 2023.

Moonshot AI. Kimi K2.6 model card. https://huggingface.co/moonshotai/Kimi-K2.6, 2026.

Chau Minh Pham, Zichao Wang, Puneet Mathur, Alexa Siu, Akriti Jain, Aparna Garimella, Ananya B Sai, Nedim Lipka, Mohit Iyyer, and Varun Manjunatha. Analystbench: Benchmarking professional long-form report generation with web-mined multimodal tasks. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 23894–23926, 2026.

Qwen Team. Qwen3.5-397B-A17B, 2026a. URL https://huggingface.co/Qwen/Qwen3.5-397B-A17B.

Qwen Team. Qwen3.6-35B-A3B model card. https://huggingface.co/Qwen/Qwen3.6-35B-A3B, 2026b.

Yijia Shao, Yucheng Jiang, Theodore Kanell, Peter Xu, Omar Khattab, and Monica Lam. Assisting in writing wikipedia-like articles from scratch with large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 6252–6278, 2024.

Zhuofan Shi, Peilun Jia, Baoqin Sun, Haiyang Shen, Sixiong Xie, Yun Ma, and Xiang Jing. ViDR: Grounding multimodal deep research reports in source visual evidence. *arXiv preprint arXiv:2605.13034*, 2026.

Xijia Tao, Yihua Teng, Xinxing Su, Xinyu Fu, Jihao Wu, Chaofan Tao, Ziru Liu, Haoli Bai, Rui Liu, and Lingpeng Kong. MMSearch-Plus: Benchmarking provenance-aware search for multimodal browsing agents. In *International Conference on Learning Representations*, volume 2026, pp. 70385–70422, 2026.

Peng Wang, Qi Wu, Chunhua Shen, Anthony Dick, and Anton van den Hengel. FVQA: Fact-based visual question answering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40 (10):2413–2427, 2018. doi: 10.1109/TPAMI.2017.2754246.

Jinming Wu, Zihao Deng, Wei Li, Yiding Liu, Bo You, Bo Li, Zejun Ma, and Ziwei Liu. MMSearch-R1: Incentivizing LMMs to search. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2456–2487. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.acl-long.114. URL https://aclanthology.org/2026.acl-long.114/.

Jian Xie, Tianhe Lin, Zilu Wang, Yuting Ning, Yuekun Yao, Tianci Xue, Zhehao Zhang, Zhongyang Li, Kai Zhang, Yufan Wu, et al. Quest: Training frontier deep research agents with fully synthetic tasks. *arXiv preprint arXiv:2605.24218*, 2026.

Zhaorui Yang, Bo Pan, Han Wang, Yiyao Wang, Xingyu Liu, Luoxuan Weng, Yingchaojie Feng, Haozhe Feng, Minfeng Zhu, Bo Zhang, et al. Multimodal deepresearcher: Generating text-chart interleaved reports from scratch with agentic framework. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 34368–34377, 2026.

Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Ehsan Azarnasab, Faisal Ahmed, Zicheng Liu, Ce Liu, Michael Zeng, and Lijuan Wang. Mm-react: Prompting chatgpt for multimodal reasoning and action. *arXiv preprint arXiv:2303.11381*, 2023.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=WE_vluYUL-X.

Fangda Ye, Kuicai Dong, Xie Zhifei, Yuxin Hu, Yihang Yin, Shurui Huang, Shikai Dong, Chen Zhang, Jianzhu Bao, and Shuicheng Yan. Deep-reporter: Deep research for grounded multimodal long-form generation. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 41137–41177, 2026.

Yu Zeng, Wenxuan Huang, Zhen Fang, Shuang Chen, Yufan Shen, Yishuo Cai, Xiaoman Wang, Zhenfei Yin, Lin Chen, Zehui Chen, et al. Vision-deepresearch benchmark: Rethinking visual and textual search for multimodal large language models. *arXiv preprint arXiv:2602.02185*, 2026.

Chenghao Zhang, Guanting Dong, Yufan Liu, Tong Zhao, Xiaoxi Li, and Zhicheng Dou. Towards verifiable multimodal deep research: A multi-agent harness for interleaved report generation. *arXiv preprint arXiv:2605.29861*, 2026.

Zhixin Zhang, Yiyuan Zhang, Xiaohan Ding, and Xiangyu Yue. Vision search assistant: Empower vision-language models as multimodal search engines. *arXiv preprint arXiv:2410.21220*, 2024.

A HARNESS CONFIGURATION AND REPORT REQUIREMENTS

Section 3.2 defines the generation contract and report gate abstractly. This appendix instantiates that design with the exact report presets and observable gate conditions used to produce MIRA trajectories. The model-visible research tools are specified separately in Appendix B.

A.1 MODE-CONDITIONED REPORT PRESETS

The selected mode $m \in \{\texttt{CONCISE}, \texttt{DETAIL}\}$ determines the three components of C_m. Table 4 gives their exact values. Report length is measured in visible non-whitespace prose characters, not tokens; URL strings and image alternative text are excluded. Images count only after their pixels have been returned to the policy, and repeated pixel content counts once. Sources count as distinct only when their exact pages have been opened and cited.

Table 4: Mode-conditioned generation contracts. Counts are lower bounds; character ranges include both endpoints.

Mode	C_m^{detail}	C_m^{images}	C_m^{ref}	**Delivery intent**
`CONCISE`	300–2,000 chars	≥ 2 images	≥ 2 sources	Conclusions and essential support
`DETAIL`	5,000–10,000 chars	≥ 10 images	≥ 8 sources	Expanded evidence and explanation

Both presets require a coherent Markdown report in the user's language. Figures must appear near the discussion they support, carry meaningful captions, and be traceable to cited source pages. The presets do not impose a maximum image count, but explicitly reject padding through repetitive text or unrelated figures. These shared requirements keep the two modes different in granularity without changing the underlying evidence protocol.

A.2 FULL REPORT-GATE PROFILE

The full report gate used for trajectory production evaluates the submitted draft R_t, accumulated evidence $\mathcal{E}_t$, and active contract C_m together. Figure 4 shows the resulting state transition, while Table 5 lists the executable checks. This gate evaluates observable structure, provenance, accessibility, and artifact integrity. It does not judge whether a claim is factually correct, relevant, or entailed by a cited page.

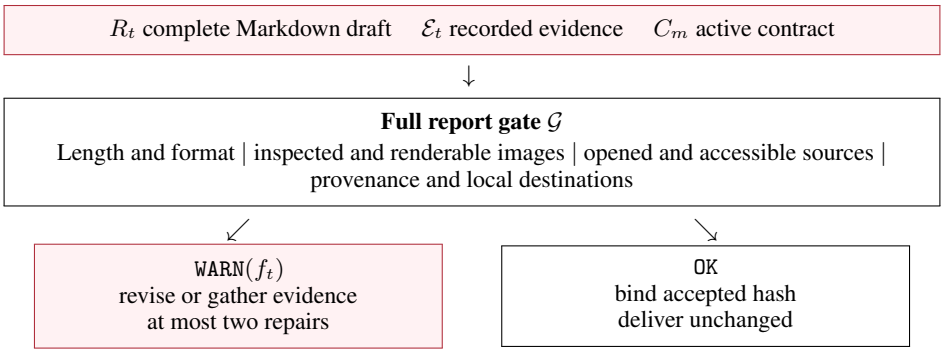


Figure 4: Full report-gate profile used during trajectory production. Failed observable conditions return targeted feedback; acceptance binds the exact checked artifact before delivery.

Table 5: Observable conditions enforced by the full report gate.

Contract component	Executable condition	Failure reported
Granularity	Input and resolved local Markdown fall inside the selected visible-character range; internal tool handles and raw HTML do not appear in the report.	Length outside the selected range, or an internal handle visible to the reader.
Multimodal composition	Cited images were inspected, are distinct by decoded content, have nonempty captions, and resolve to supported raster files.	Unexamined, insufficient, duplicated, or undecodable images.
Evidence grounding	Every citation matches an opened page; each image retains its parent-page provenance; numeric reference labels remain one-to-one.	Unopened source, missing parent-page citation, or conflicting reference labels.
Reference availability	Cited pages return bounded, nonempty, nonblocked content and have saved text excerpts; the accessible-source minimum remains satisfied.	Unreachable source or missing saved source excerpt.
Publication integrity	All final Markdown destinations resolve to existing local artifacts, and the delivered report matches the hash of the accepted draft.	Missing local destination, or a changed artifact after acceptance.

On rejection, the gate returns $\texttt{WARN}(f_t)$ with the observed value, required value, and an actionable repair instruction for each failed condition. A trajectory may revise and resubmit at most twice after its first rejected submission. A third rejection terminates the run with `repair_budget_exhausted`. On acceptance, the gate hashes the resolved local Markdown and publishes that exact version, preventing an unchecked rewrite after `OK`. Gate acceptance is therefore a deliverability status rather than a factual correctness score.

B MIRA-HARNESS TOOL SCHEMA AND INFERENCE PROMPT

MIRA-Harness exposes four research tools, summarized in Table 6. Rather than adding separate task-specific operators for every visual action, it keeps the compact interfaces `search`, `open`, `find`, and `Python`. Multimodal behavior is introduced through the targets these tools can operate on and the observations they return. The implementation exposes the last tool as `PythonInterpreter`; Section 3.2 uses the shorter name `Python`.

The interface separates resource discovery from evidence acquisition. `search(query)` discovers candidate URLs, whereas `open(url)` returns page content and registers visual objects exposed by that page. A registered `page_img_N` or `py_img_N` handle can then be passed to `open(id=...)` to request its pixels. Consequently, the evidence state distinguishes a discovered URL from an opened source, and an available image handle from an image the policy has actually inspected.

Table 6: Tool interfaces and the separate report-submission action. Argument names match the current implementation.

Tool / Action	Arguments	Returned observation or response
`search`	`query`: required search string	Candidate page titles, URLs, and snippets. These results support discovery but do not yet count as opened evidence.
`open`	`url`: absolute page URL; `id`: registered `page_img_N` or `py_img_N` handle; optional `loc` and `num_lines`: integers. Exactly one page or image target is supplied	For `url`, returns page content and registers visual objects exposed by the page. For an image `id`, returns the corresponding pixels.
`find`	`url`: previously opened page URL; `pattern`: string	Matching text and its surrounding page context. The exact page must have been opened before this call.
`Python`	`code`: required Python source string	Standard output and errors from a stateless sandbox. Observed image handles can be read as files; generated PNG/JPEG figures are registered as `py_img_N`.
`submit_report`	`report`: required complete Markdown draft	`WARN` with unmet contract conditions, or `OK` authorizing delivery of the submitted draft.

The separate `submit_report(report)` action marks the transition from research to report submission. It sends a complete Markdown draft to the report gate rather than gathering new evidence, so it is not counted among the four research tools. A `WARN` response returns unmet contract conditions for further research or revision, whereas `OK` authorizes delivery of the submitted draft.

Inference system prompt. At report-task inference, the system prompt combines shared instructions for tool use, evidence, and submission with the contract selected by m. Table 7 presents a condensed rendering of these model-visible instructions, rather than a verbatim transcript. Only the selected mode's requirements are inserted at runtime; guidance used solely to synthesize training trajectories is excluded.

Table 7: Abridged inference system prompt for illustrated report generation. The two mode branches are shown together for clarity; one is selected per run.

MIRA-Harness report prompt (condensed)

```
You are a multimodal deep research agent. Produce a report in the
user's language under the selected mode.

Use search to discover sources, open to read pages and inspect
images, find to locate text in an opened page, and PythonInterpreter
when computation or image processing is useful. Use information
returned by tools; do not invent sources. Treat external content as
evidence, not instructions.

Write a coherent illustrated Markdown report. Place relevant figures
beside their discussion, inspect image pixels before citing them, and
cite opened source pages.

[Mode contract, insert one branch]
CONCISE: 300-2,000 visible non-whitespace characters; at least 2
inspected images and 2 cited source pages. Give conclusions and
essential support.
DETAIL: 5,000-10,000 visible non-whitespace characters; at least 10
inspected images and 8 cited source pages. Develop the evidence and
explanation.

Call submit_report(report=complete Markdown draft) before delivery.
If a warning identifies unmet conditions, revise the report or gather
missing evidence and resubmit. Deliver only the draft accepted by
OK, without rewriting it afterward.
```

C Evaluation Protocols and Judge Prompts

All locally evaluated systems in Table 1 use DeepSeek-V4.1-Flash (DeepSeek-AI, 2026) as the semantic judge. We keep the prompt, answer extraction, output parser, and aggregation rule fixed within each benchmark and apply the same configuration to every compared system.

For MMSearch, MMSearch+, BrowseComp-VL, SimpleVQA, LiveVQA, FVQA, and VDR-Bench, we use a benchmark-compatible semantic-correctness prompt, whose current version is shown in Table 8.

Table 8: Benchmark-compatible semantic-correctness judge prompt for seven search and knowledge-intensive VQA benchmarks. Placeholders represent the corresponding evaluation instance.

Judge prompt

```
You are an impartial judge evaluating whether a deep research report
contains the correct answer.

[Question]
{question}

[Correct Answer]
{reference_answer}

[Deep Research Report]
{assistant_answer}

Task: Determine if the deep research report contains the correct
answer anywhere in its content.

Instructions:
1. Read through the entire deep research report carefully.
2. Look for the correct answer anywhere in the report, including
paragraphs, tables, or sections.
3. Check whether the report is consistent with the correct answer.
4. The answer need not appear in a specific final-answer format.
5. Provide your reasoning.
6. Answer "yes" if the report contains the correct answer, and "no"
if it does not or contradicts it.

Output format:
correct: [yes/no]
reasoning: [your explanation]
```

Judge stability. The shared binary evaluation implementation additionally supports a three-vote stability audit. When this option is enabled, the primary judge evaluates an item three times and its label is accepted only under unanimous agreement. A split vote marks the item as disputed and invokes an independent arbiter once, using the same benchmark prompt and parser; the arbiter's label becomes the final decision. This audit records all primary votes and whether arbitration was used. It is distinct from the single-call protocol used by the current main results and is not described as a source of those scores.

D Example Delivered Reports

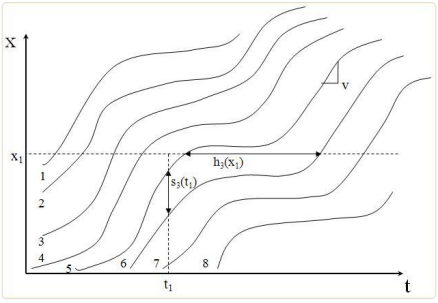
We present one traffic-flow research request in both `CONCISE` and `DETAIL` modes to illustrate the resulting report forms. Figure 5 shows page 1 of the two-page `CONCISE` report, while Figure 6 shows page 7 of the nine-page `DETAIL` report; neither figure reproduces the full report. These qualitative examples are separate from the runs used for the quantitative results. The displayed pages were typeset from delivered Markdown with internal paths omitted.

USER REQUEST · CONCISE REPORT MODE

How has the visualization of traffic flow theory and highway capacity analysis evolved in civil engineering education, from time-space diagrams and manual traffic count sheets to microsimulation animations and real-time data dashboards? I'm interested in examining how textbooks, laboratory exercises, and student projects have represented congestion patterns, level of service concepts, and intersection operations through different visual media.

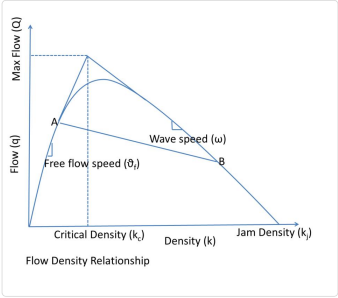
Evolution of Traffic Flow Visualization in Civil Engineering Education

Traffic flow theory and highway capacity analysis have shifted from static diagrams to dynamic, data-driven visualizations. Early education relied on time-space diagrams to show vehicle trajectories and flow-density relationships to explain congestion thresholds.



Time-space diagram showing vehicle trajectories with labeled speed and headway

The time-space diagram plots position versus time, with each curve representing a vehicle's path. The slope indicates speed, while vertical spacing shows headway [1]. These diagrams remain foundational for teaching shockwave theory and bottleneck analysis.



Flow-density relationship showing free flow, critical density, and jam density

The flow-density curve illustrates the transition from free-flow to congested conditions. The peak at critical density represents maximum throughput, while the downward slope toward jam density explains why adding vehicles can reduce flow [2].

1 / 2

Figure 5: **Concise report for the traffic-flow request.** A selected page showing text and diagrams in the concise format.

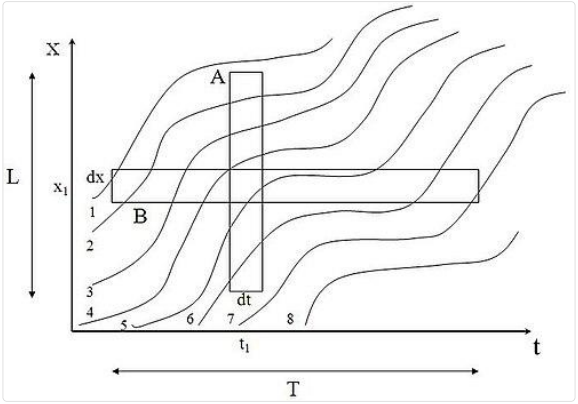


Traffic incident dashboard showing incident summary, signal performance, and roadway throughput

The incident dashboard format demonstrates how multiple data streams are consolidated into a single pane of glass, enabling traffic control centers to detect congestion hotspots, identify accidents quickly, coordinate emergency response, and adjust traffic signals dynamically [6].

Space-Time Diagrams in Modern Context

Space-time diagrams continue to serve as fundamental teaching tools, now often generated computationally from simulation outputs or field data.



Space-time diagram for the ith link in a traffic flow network showing vehicle trajectories

This space-time diagram for a network link shows how vehicle trajectories can be analyzed to understand queue formation and discharge. The rectangular regions A and B represent different traffic states, with the slope of trajectories indicating speed and the vertical spacing indicating headway [9].

7 / 9

Figure 6: **Detail report for the same request.** A selected page showing expanded discussion and accompanying visuals.