SEMANTIC FLASHLIGHT: PLAN-FIRST CONTEXT CONSTRUCTION WITH RESPONSE-ADAPTIVE EVIDENCE EXPOSURE FOR DEEP RESEARCH

Anonymous authors
Paper under double-blind review

ABSTRACT

Deep-research agents repeatedly search, inspect evidence, and refine their reasoning, causing their working context to grow with research depth. Existing context-management methods often decide what to retain only after the interaction history has accumulated, requiring a selector to reconstruct the information need behind earlier search decisions. We introduce **Semantic Flashlight**, a **plan-first and response-adaptive context construction** framework built around two complementary signals. First, during query formulation, the model already identifies an unresolved information gap and anticipates what evidence is needed next. Flashlight externalizes this information need as a **next-context plan**, which guides evidence selection for the following decision. Second, the model's response provides a signal about whether the current evidence is being effectively used. Across real research trajectories, higher-quality outcomes are associated with stronger alignment to the current context than to prior model responses. Flashlight therefore uses **response feedback** to adapt the breadth of evidence exposure, broadening raw evidence when response echo dominates source alignment. A dependency-aware evidence contract carries these two signals across planning, acquisition, review, and synthesis, while the Context Engine enforces provenance and bounded context construction. Across diverse deep-research benchmarks with two closed-source models, Flashlight improves report quality while reducing LLM token consumption by up to **45.6%** relative to Full-History, without requiring an auxiliary LLM context-selection call. Optional role-conditioned supervised fine-tuning and reinforcement learning with our Effective Information Acquisition (EIA) reward extend the workflow to an open-weight model. Together, these results suggest that context construction can be integrated into the research decision itself rather than treated as post-hoc transcript compression.

1 INTRODUCTION

Deep-research agents built on large language models (LLMs) repeatedly search, inspect evidence, and reason over retrieved sources to produce evidence-grounded reports (Du et al., 2026). As research proceeds, their working context accumulates source evidence together with model-generated plans, summaries, and prior responses. Retaining this history preserves continuity, but it also increases input cost and repeatedly exposes information that may no longer be relevant to the current decision. We call the native-agent baseline that replays this cumulative reasoning and tool transcript FULL-HISTORY. Existing methods address this growth through token-level compression, trajectory folding, or selective exposure of accumulated history (Jiang et al., 2023a; Ye et al., 2025; Liao et al., 2026). In selector-based systems, however, search planning and context selection are separated: the research model decides what to search for, while a later selector must infer from the accumulated history what information should remain visible. This requires reconstructing the information need that originally motivated the search.

We take a different view: the information needed for context construction is already formed when the research model decides what to search for. Query formulation requires the model to identify an unresolved gap and anticipate what evidence could resolve it. We externalize this latent information need as a *next-context plan*. For each active evidence requirement, the Controller emits a query

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.

together with a plan specifying the unresolved gap, the evidence needed next, and the decision that evidence is intended to support. After reviewing retrieved evidence, the plan is updated from the remaining gap and carried forward by the Context Engine to construct the next model input. This *plan-first* design couples evidence acquisition with subsequent context construction, avoiding a separate generative selector whose task is to recover the search intent after the fact.

A plan specifies what evidence should matter next, but it does not reveal whether the model actually uses that evidence once exposed. The resulting response provides a complementary control signal. Across 300 trajectories from an open-weight model and 300 from a closed-source model, we find that higher-quality outcomes are associated with a larger *context-over-echo margin*: responses align more strongly with the current working context than with the model's preceding response (Sec. 3). This observation motivates *response-adaptive evidence exposure*. In deployment, the Context Engine uses a source-grounded feedback signal to compare response alignment with displayed source evidence against similarity to earlier model responses. When response echo dominates source alignment, the engine broadens the raw evidence exposed within the direction specified by the current plan. The plan therefore controls *what* evidence should matter next, while response feedback controls *how broadly* that evidence should be exposed.

We instantiate these two signals in SEMANTIC FLASHLIGHT (FLASHLIGHT), an inference-time model–engine framework built around a dependency-aware evidence contract. A shared role-conditioned model acts as Designer, Controller, and Writer, while an external Context Engine validates state transitions, tracks source provenance, and constructs each budgeted context. The evidence contract carries unresolved requirements and source bindings across planning, acquisition, review, and synthesis, enabling the two context-control signals to persist throughout the research process. Across two closed-source models, FLASHLIGHT improves DRB RACE by 1.4–2.2 points while reducing LLM token consumption by 45.5–45.6% relative to native Full-History agents. In a controlled comparison within the same Designer–Controller–Writer pipeline, replacing full-state replay with Flashlight context construction improves RACE from 52.9 to 54.2 while reducing tokens from 118k to 72k, isolating the contribution of the context mechanism. For an open-weight model that does not initially execute the structured interface reliably, we study role-tagged supervised fine-tuning (SFT) and optional Group Relative Policy Optimization (GRPO) with our Effective Information Acquisition (EIA) reward separately from the core inference-time mechanism.

Contributions. **(i) Plan-first context construction.** We externalize the evidence need formed during query generation as a next-context plan and reuse it to guide the following context, avoiding post-hoc reconstruction of search intent. **(ii) Response-adaptive evidence exposure.** We identify a context-over-echo signal associated with research quality and use a source-grounded online variant to broaden plan-relevant evidence when responses become more self-referential than source-aligned. **(iii) Evidence-contract realization and controlled evaluation.** We implement both mechanisms in a provenance-aware model–engine framework and evaluate them through both end-to-end system comparisons and controlled within-pipeline comparisons, with interface SFT and EIA-GRPO studied as an optional open-weight extension.

2 RELATED WORK

Adaptive context management. Long-context reasoning is affected by evidence position and irrelevant context (Liu et al., 2024; Shi et al., 2023; Yoran et al., 2024). Existing methods manage accumulated context through compression, abstraction, memory, or selective exposure. LLMLingua, LongLLMLingua, and RECOMP compress prompts or retrieved evidence (Jiang et al., 2023a; 2024; Xu et al., 2024), while MemGPT and RAPTOR provide external or hierarchical memory (Packer et al., 2023; Sarthi et al., 2024). Trajectory-level methods such as Sculptor, AgentFold, ACM, ACE, ContextWeaver, Context as a Tool, and Self-GC organize or select prior agent states (Li et al., 2026a; Ye et al., 2025; Li et al., 2026c; Liao et al., 2026; Wu et al., 2026; Liu et al., 2026; Hao et al., 2026); active-inference work instead formulates context acquisition as a sequential decision problem (Dutta et al., 2026). These approaches primarily decide what to retain after information has entered history. Flashlight instead externalizes the evidence need when a search decision is formed and reuses it for subsequent context construction. Related feedback-driven methods, including active retrieval, Self-RAG, Corrective RAG, Self-Refine, and Reflexion, use retrieval or model feedback to improve generation (Jiang et al., 2023b; Asai et al., 2024; Yan et al., 2024; Madaan

et al., 2023; Shinn et al., 2023). Flashlight uses response behavior for a different purpose: adapting the breadth of plan-relevant source exposure.

Structured evidence workflows for deep research. Retrieval-augmented and tool-using language models connect reasoning with external evidence sources (Lewis et al., 2020; Yao et al., 2022; Nakano et al., 2021). Recent deep-research agents extend these workflows with structured search, planning, collaboration, and report synthesis (Li et al., 2026f; Lei et al., 2025; Li et al., 2026e; Jin et al., 2026; Li et al., 2026d; Shao et al., 2025; Team et al., 2025b; Alzubi et al., 2025). These systems organize research progress through outlines, checklists, search policies, or agent coordination, but they do not explicitly maintain whether individual evidence requirements have been resolved across stages. Flashlight introduces a dependency-aware evidence contract that persists requirements, dependencies, and source bindings across planning, acquisition, review, and synthesis, providing the state needed for plan-first context construction and response-adaptive evidence exposure.

3 Motivation: Two Signals for Context Construction

3.1 The search decision exposes the next evidence need

Context construction is usually treated as a post-hoc operation over accumulated history. Yet before retrieval occurs, the research model has already formed information that is directly useful for deciding what context should matter next. To formulate a search query, the model must identify what remains unresolved, anticipate what evidence could resolve that gap, and determine how the returned evidence would affect the next research decision. We refer to this decision-time information as the *next evidence need*.

This observation motivates a *next-context plan*: an explicit representation of the evidence need formed jointly with the search decision. The plan is scoped to an active *evidence requirement*, i.e., a claim, comparison, or fact that must eventually be supported by source evidence. In our realization, the plan records five semantic components: the active requirement, the unresolved gap, a provisional candidate claim, the desired evidence, and the intended next decision. Together, these fields characterize not merely what to retrieve, but what evidence should remain salient when the returned source is reviewed and the following decision is made.

The distinction between the query and the plan is therefore functional. The query specifies what to retrieve; the plan specifies why that retrieval is being performed and what evidence should inform the next decision. Because the plan is formed at the same time as the query, context construction can reuse the research model's current evidence need rather than asking a later selector to reconstruct it from accumulated history. After evidence review, the plan is updated from the remaining unresolved gap. The executable requirement graph, state transitions, and context compiler that realize this process are introduced in Sec. 4.

3.2 The Response Provides an Evidence-Use Signal

A next-context plan specifies what evidence should matter, but it does not reveal how strongly the model's subsequent response is shaped by that evidence. The response therefore provides a complementary signal about evidence utilization: it reflects the balance between alignment with the current working context and continuity with the model's prior reasoning.

We study this behavior offline using 600 observed deep-research trajectories drawn from two contrasting execution regimes: 300 trajectories from an open-weight Qwen3-8B Full-History agent and 300 trajectories from a closed-source GPT-5.6-Luna Flashlight agent. We use this cross-regime analysis as a behavioral diagnostic rather than as evidence for the effectiveness of Flashlight itself. By spanning both cumulative-history and bounded-context execution, as well as open-weight and closed-source models, the analysis tests whether the context-over-echo association is specific to a single model family or context-management policy.

For each eligible research turn, we compute the *context-over-echo margin*

$$m_t^{\mathrm{ctx}} = \mathrm{sim}(y_t, \mathcal{C}_t) - \mathrm{sim}(y_t, y_{t-1}), \tag{1}$$

where y_t is the current response, $\mathcal{C}_t$ is the working context presented at that turn, and y_{t-1} is the preceding model response. We average the margin over eligible turns within each trajectory and pair the resulting trajectory-level value with an independently judged RACE-style quality score normalized to $[0, 1]$. This diagnostic evaluator follows the RACE evaluation structure but is distinct from the official DRB RACE endpoint used in the main experiments.

Across the 300 Qwen3-8B Full-History and 300 GPT-5.6-Luna Flashlight trajectories, the trajectory-level mean margin is positively associated with report quality (Spearman's $\rho = 0.55$ and $\rho = 0.29$, respectively; Fig. 1). Additional sensitivity analyses and negative controls are reported in Sec. 5.2. These analyses do not establish causality, but they support response behavior as a nontrivial signal of whether newly exposed evidence is being incorporated.

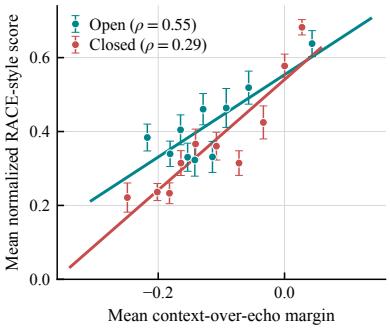


Figure 1: Context-over-echo margin versus normalized RACE-style score for 300 Qwen3-8B Full-History trajectories and 300 GPT-5.6-Luna Flashlight trajectories. Faint points show individual trajectories; opaque markers show equal-count bin means with standard errors, and lines are linear fits. Correlations reported in the text are computed over trajectory-level observations.

This observation motivates *response-adaptive evidence exposure*, but the offline diagnostic is not used directly as the deployment rule. The offline *context-over-echo margin* measures whether a response aligns more with its complete working context than with its immediately preceding response. For online control, Flashlight instead uses a source-grounded *source-over-echo margin*, comparing alignment with the displayed source evidence against similarity to the model's prior response history. When the calibrated source-over-echo margin is negative, the engine broadens raw evidence only within the direction specified by the current next-context plan. The two signals therefore serve complementary roles: the plan determines *what* evidence should matter next, while response feedback determines *how broadly* that evidence should be exposed. Sec. 4.3 formalizes the deployable trigger.

4 Semantic Flashlight

4.1 Framework and Evidence Contract

Flashlight realizes the two context-control signals introduced in Sec. 3 through a shared role-conditioned model and an external Context Engine. The model acts as *Designer*, *Controller*, or *Writer*: the Designer decomposes the request into evidence requirements, the Controller acquires and reviews source evidence, and the Writer synthesizes the final report from admitted evidence. The Context Engine maintains persistent state, validates model-proposed transitions, tracks source provenance, and constructs each budgeted model input. These roles communicate through a dependency-aware *evidence contract*. An evidence requirement is a claim, comparison, or fact that must be supported by source evidence; requirements form a directed acyclic graph (DAG), and an evidence binding links a reviewed source record and location to the requirement it supports. Together, requirement status, dependencies, and evidence bindings persist across planning, acquisition, review, and synthesis.

Let q denote the research request, $\mathcal{E}_t$ the source-evidence store, $\mathcal{S}_t$ the contract state, $\mathcal{N}_t$ the next-context plan, and f_{t-1} the response-feedback state. Under token budget B_t, the engine constructs

$$\mathcal{C}_t = \mathrm{Compile}(q, \mathcal{S}_t, \mathcal{E}_t, \mathcal{N}_t, f_{t-1}, B_t), \tag{2}$$

$$(y_t, a_t, \widehat{\mathcal{N}}_{t+1}) \sim p_\theta(\cdot \mid \mathcal{C}_t), \tag{3}$$

$$(\mathcal{S}_{t+1}, \mathcal{E}_{t+1}, \mathcal{N}_{t+1}, f_t) = \mathrm{Update}(\mathcal{S}_t, \mathcal{E}_t, y_t, a_t, \widehat{\mathcal{N}}_{t+1}, o_t), \tag{4}$$

where y_t is the model response, a_t its proposed action, $\widehat{\mathcal{N}}_{t+1}$ its proposed next-context plan, and o_t an optional tool observation. The engine validates actions, evidence references, plans, and state transitions before committing updates. Model-authored claims and summaries may remain in research state, but they are never treated as source evidence. Fig. 2 summarizes the workflow, and App. E gives the complete state and validation rules.

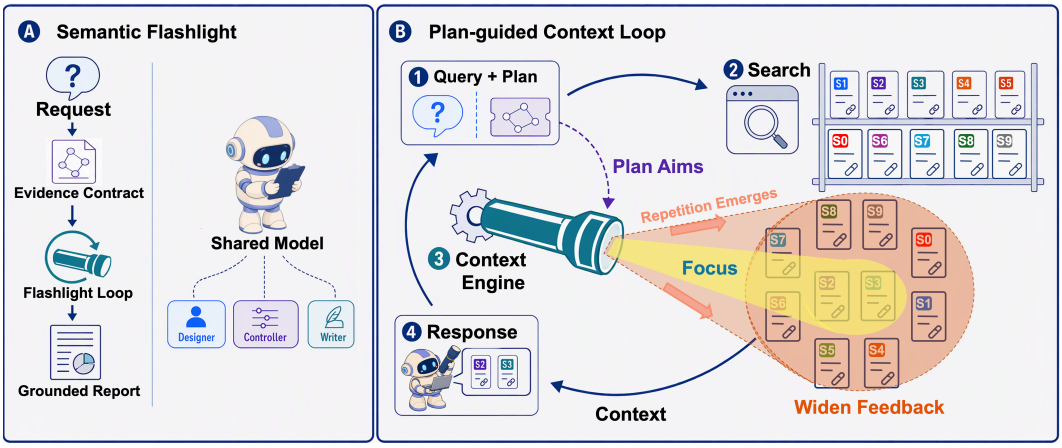


Figure 2: Overview of FLASHLIGHT. A shared role-conditioned model operates over an engine-validated evidence contract. The next-context plan determines which evidence is relevant to the next decision, while response feedback adaptively controls how broadly source evidence is exposed.

4.2 PLAN-FIRST CONTEXT CONSTRUCTION

Each Controller turn is either *acquisition* or *evidence review*. The Controller operates on one dependency-ready unresolved requirement at a time and produces a next-context plan containing the active requirement, unresolved gap, candidate claim, desired evidence, and intended next decision. During acquisition, it emits the search query and plan jointly: the query specifies what to retrieve, while the plan specifies why the retrieval is needed and what evidence should inform the next decision. Retrieved source records are stored as a pending evidence transaction. During review, the Controller may admit, reject, or mark evidence as conflicting, request a more precise source location, revise the remaining gap, propose requirement completion, and update the next-context plan. The engine validates these proposals before committing the transition.

To construct the next input, the engine scores eligible evidence against the semantic components of the next-context plan. Let n_j denote plan field j and $z(\cdot)$ the semantic encoder. For evidence record e,

$$s(e \mid \mathcal{N}_{t+1}) = \max_j \cos(z(e), z(n_j)) \,. \tag{5}$$

The first Controller input containing newly retrieved evidence includes all eligible records subject to the input budget. On later turns, the engine separates higher- and lower-relevance records using an Otsu boundary over the semantic scores (Otsu, 1979). For threshold τ, let $n_-(\tau)$ and $n_+(\tau)$ be the numbers of scores at or below and above the threshold, with corresponding means $\mu_-(\tau)$ and $\mu_+(\tau)$. The selected threshold is

$$\tau^\star = \arg\max_\tau n_-(\tau) n_+(\tau) \left(\mu_-(\tau) - \mu_+(\tau)\right)^2 . \tag{6}$$

Records above $\tau^\star$ form the primary raw-evidence set $\mathcal{R}_{t+1}$, while lower-scoring records may be represented as compact source-linked state $\mathcal{K}_{t+1}$ that preserves identifiers, locations, and reviewed relations without repeating full source text. Unselected records remain stored but are omitted from the current input, and pending evidence remains visible until review. The final context satisfies

$$\mathrm{Tok}(\mathcal{C}_{t+1}) \le B_{t+1}. \tag{7}$$

Thus, the engine reconstructs each input from persistent state and selected evidence rather than replaying the complete interaction history.

4.3 RESPONSE-ADAPTIVE EVIDENCE EXPOSURE

The next-context plan determines which evidence is relevant, but it does not indicate whether the model is actually using the exposed evidence. We therefore use the response as a second control

signal. Sec. 3 introduced an offline context-over-echo diagnostic based on the complete working context and preceding response. For online control, the engine instead compares the response with displayed source evidence and with the full prior response history. Let V_t denote the displayed source evidence and $\mathcal{H}_{<t}$ the set of earlier model responses:

$$g_t = \max_{e \in V_t} \cos(z(y_t), z(e)), \tag{8}$$

$$r_t = \max_{h \in \mathcal{H}_{<t}} \cos(z(y_t), z(h)), \tag{9}$$

$$m_t^{\mathrm{src}} = g_t - r_t. \tag{10}$$

Here, g_t measures source alignment and r_t measures response echo.

Because both terms are maxima and therefore depend on candidate-set cardinality, we standardize them using development trajectories with matching set sizes:

$$\widetilde{m}_t^{\mathrm{src}} = \frac{g_t - \mu_V(|V_t|)}{\sigma_V(|V_t|) + \epsilon_m} - \frac{r_t - \mu_H(|\mathcal{H}_{<t}|)}{\sigma_H(|\mathcal{H}_{<t}|) + \epsilon_m}. \tag{11}$$

The calibration functions and zero threshold are fixed before test evaluation. When $\widetilde{m}_t^{\mathrm{src}} < 0$, standardized response echo dominates source alignment, and the engine broadens evidence exposure within the direction specified by the current plan. It first performs the normal Otsu selection, removes the selected records, and applies a second Otsu split to the remaining plan-relevant candidates; records above the second boundary are added as raw evidence. Otherwise, the engine uses the normal single-stage selection.

The two signals therefore play complementary roles: the next-context plan determines *what* evidence should matter next, while response feedback determines *how broadly* that evidence should be exposed. The uncalibrated margin in Eq. 10 is retained as an interpretable diagnostic, while the calibrated trigger is evaluated separately in Sec. 5.2.

4.4 Evidence-Bounded Report Synthesis

The Writer is invoked only when no evidence transaction or source-location request remains pending and every required DAG node has been supported. Each completed requirement must contain at least one checked evidence binding and a completion certificate that states the answer claim, cites its admitted evidence, and records that no residual gap remains. The Writer receives the original request, compact contract state, and a source-linked evidence packet grouped by requirement; it does not receive the complete Controller trajectory. App. E gives the certificate and release-validation details.

Each citation handle in the generated report refers to a stored source identifier and location. Before release, the engine checks that cited identifiers exist, quoted text occurs in the referenced source, and the output satisfies the required format. These deterministic checks establish provenance and protocol validity but do not guarantee semantic entailment; we therefore evaluate claim–evidence support separately using an independent judge family and blinded human annotation (App. C.3).

4.5 Optional Open-Weight Adaptation

The core FLASHLIGHT mechanism is inference-time only and requires no parameter updates when the model reliably follows the structured interface. For open-weight models that do not initially produce valid Designer, Controller, and Writer outputs consistently, we first apply role-conditioned supervised fine-tuning using engine-validated trajectories. Each example pairs the exact turn input with a valid target output and contains only information available at that point in the trajectory.

We further optimize Controller acquisition with EIA-GRPO—GRPO using the EIA reward—in a replay environment constructed from saved Controller states and stored retrieval results. We operationalize effective information as newly validated required-fact coverage under a valid state transition. Accordingly, EIA rewards an acquisition only when it adds previously uncovered facts for an unresolved requirement; repeated evidence and invalid transitions receive no reward. It therefore trains the Controller to close outstanding information gaps rather than merely issue more or lexically different searches. App. A gives the exact reward definition, hidden coverage ledger, replay construction, GRPO objective, and optimization settings (Shao et al., 2024).

Table 1: End-to-end system results. The first block provides system-level references that use different model families, released components, or method-faithful adaptations; these rows are not used to attribute gains to Flashlight. In each subsequent model block, the backbone, task set, retrieval stack, maximum budgets, citation format, and evaluator are fixed, while each system retains its native orchestration and prompts. Efficiency is an unweighted macro-average over the five benchmarks, not a pooled per-task mean. Best results are bold and second-best results are underlined only within each fixed-backbone block. The Qwen3-8B block is not a direct method comparison with its raw Full-History row because SFT is required for interface execution. Metric definitions and provenance are detailed in App. B.

Method	DRB			DRB-II	RQA		SQA-CS2	DRGym		Efficiency		
	RACE ↑	FACT		Total (%) ↑	Cov. (%) ↑	Elo ↑	Score (%) ↑	KPR (%) ↑	KPC (%) ↓	Tokens (k) ↓	LLM calls ↓	Latency (s) ↓
		C.Acc. (%) ↑	E.Cit. (count) ↑									
System-level references: different model families or native system settings												
ACE	49.7	85.0	72.0	45.0	67.0	1030	70.0	58.0	1.20	92	22	520
ContextWeaver	48.9	83.0	68.0	44.0	66.0	1170	69.0	57.0	1.25	84	19	470
WebWeaver (GPT-4.1 adaptation)	50.6	93.4	200.8	49.0	74.0	1420	82.0	69.0	0.55	121	42	1120
RhinoInsight	50.9	89.0	88.0	48.0	72.0	1380	77.0	66.0	0.70	98	24	760
WebThinker-32B (report)	37.9	74.0	46.0	37.0	70.0	1080	46.7	52.0	1.10	82	18	640
Tongyi DeepResearch	40.6	78.0	52.0	41.0	64.0	1160	46.5	55.0	1.05	76	16	520
OpenDeepSearch	38.5	81.0	50.0	39.0	56.0	900	43.0	42.8	0.84	64	14	430
DR Tulu-8B (RL)	46.8	86.0	64.0	47.0	73.0	1040	88.3	61.0	0.75	48	12	280
Direct comparisons with fixed backbone, tasks, retrieval stack, budgets, and evaluator												
Main LLM: GPT-4.1												
FULL-HISTORY	49.8	82.0	62.0	46.0	68.5	1160	71.5	60.5	1.20	125	18	540
FLASHLIGHT (Ours)	**51.2**	**88.5**	**69.0**	**49.0**	**72.0**	**1290**	**75.0**	**64.0**	**0.80**	**68**	**14**	**410**
Main LLM: GPT-5.6-Luna												
FULL-HISTORY	52.0	84.0	70.0	50.0	73.0	1240	78.0	65.0	0.85	132	18	480
FLASHLIGHT (Ours)	**54.2**	**91.0**	**82.0**	**53.0**	**77.0**	**1380**	**82.0**	**69.0**	**0.55**	**72**	**14**	**365**
Main LLM: Qwen3-8B (Flashlight requires interface adaptation)												
FULL-HISTORY	40.5	70.0	44.0	36.0	56.0	760	40.4	41.0	1.35	108	18	620
FLASHLIGHT + SFT	46.0	83.0	61.0	45.0	67.0	1040	64.0	55.0	0.80	57	13	430
FLASHLIGHT + EIA-GRPO	**49.0**	**87.0**	**68.0**	**49.0**	**72.0**	**1160**	**72.0**	**61.0**	**0.65**	**54**	**12**	**395**

5 EXPERIMENTS

We evaluate FLASHLIGHT on five complementary deep-research benchmarks: DEEPRESEARCH BENCH (DRB) (Du et al., 2026), DEEPRESEARCH BENCH II (DRB-II) (Li et al., 2026b), ResearchQA (RQA) (Yifei et al., 2026), AstaBench ScholarQA-CS2 (SQA-CS2) (Bragg et al., 2026), and the Researchy Questions protocol of DeepResearchGym (DRGym) (Coelho et al., 2025). Together, they cover general-purpose research, expert-defined requirements, scholarly synthesis, source-linked reporting, and reproducible web retrieval. We use GPT-4.1 (OpenAI, 2025) and GPT-5.6-Luna as closed-source research models and Qwen3-8B (Team et al., 2025a) as the open-weight model. The first block of Tab. 1 reports external systems—including ACE, ContextWeaver, WebWeaver, RhinoInsight, DR Tulu, WebThinker, Tongyi DeepResearch, and OpenDeepSearch—as contextual system-level references rather than attribution baselines, because their released model families, tool stacks, or native orchestration differ. These systems are cited and discussed in Sec. 2. Exact task counts, metric definitions, baseline provenance, and implementation deviations are documented in App. B and App. B.3.

Our primary end-to-end comparison fixes the research backbone, task assignments, Bing v3 and Crawl4AI retrieval stack, maximum task and output budgets, citation format, and evaluator. Full-History retains its unified-agent prompt, orchestration, natural stopping behavior, and cumulative reasoning/tool transcript, whereas FLASHLIGHT uses the structured workflow described in Sec. 4; this comparison therefore measures the complete system difference. Mechanism attribution instead uses controlled comparisons within the Flashlight pipeline in Tab. 2, where roles, validators, tools, task limits, and evaluator are held fixed. DRB RACE is our primary metric for the quality of the final output report; the remaining official metrics measure citation, coverage, and contradiction. Direct controls are paired by task. We report paired 95% task-bootstrap intervals (Efron, 1992), apply Holm correction (Holm, 1979) to the primary mechanism contrasts, and additionally run three fresh retrieval and generation seeds on a stratified 100-task subset. Full statistical results, standardized effect sizes (Lakens, 2013), and win/tie/loss counts are reported in Tab. 16.

5.1 MAIN RESULTS

Table 2: End-to-end controls with GPT-5.6-Luna. All rows use the same tasks, retrieval stack, maximum budgets, and evaluator. Full-History retains its unified-agent orchestration; all other rows retain the Flashlight pipeline and change only the named component. The ACE-style and ContextWeaver-style rows replace context construction rather than reproduce the complete external systems. Efficiency metrics are per-task means; ranking is within each panel.

(a) Architecture and component ablations

Variant	DRB	SQA-CS2	Efficiency		
	RACE ↑	Score (%) ↑	Tokens (k) ↓	LLM calls ↓	Latency (s) ↓
FULL-HISTORY	52.0	78.0	132	18	480
Truncation	47.8	68.0	**64**	14	420
Summarization	49.2	72.0	78	16	515
Independent selector	51.5	77.5	88	19	560
ACE-style context compiler	52.4	78.8	86	19	545
ContextWeaver-style memory	52.6	79.0	80	15	430
Full-state replay	52.9	79.5	118	14	445
Flat requirements	50.8	74.0	70	14	350
Joint acquisition/review	51.0	73.5	67	**12**	**315**
FLASHLIGHT (Ours)	**54.2**	**82.0**	72	14	365

(b) Plan and feedback controls

Variant	DRB	SQA-CS2	Efficiency		
	RACE ↑	Score (%) ↑	Tokens (k) ↓	LLM calls ↓	Latency (s) ↓
Query only, no feedback	51.4	74.5	76	14	345
Equal-length dummy, no feedback	51.6	76.0	78	14	355
Task-shuffled plan, no feedback	51.2	74.2	77	14	354
Full plan, no feedback	52.6	79.0	**66**	14	**340**
Full plan + random broadening	52.8	79.4	72	14	365
Full plan + always broadening	53.7	81.5	96	14	520
Full plan + uncalibrated margin	54.0	81.8	74	14	370
FLASHLIGHT (Ours)	**54.2**	**82.0**	72	14	365

Tab. 1 shows that the complete FLASHLIGHT system improves quality while substantially reducing LLM token consumption across both closed-source backbones. Relative to native Full-History, DRB RACE increases from 49.8 to 51.2 with GPT-4.1 and from 52.0 to 54.2 with GPT-5.6-Luna, while token consumption decreases from 125k to 68k and from 132k to 72k, respectively—a 45.5–45.6% reduction. The gains are not confined to the primary endpoint: FLASHLIGHT also improves citation accuracy, effective citations, DRB-II, ResearchQA, SQA-CS2, and DR-Gym metrics within both fixed-backbone blocks. At the same time, GPT-5.6-Luna requires 14 rather than 18 LLM calls and reduces mean latency from 480 s to 365 s. These are complete-system comparisons and should therefore be interpreted jointly with the controlled within-pipeline results below.

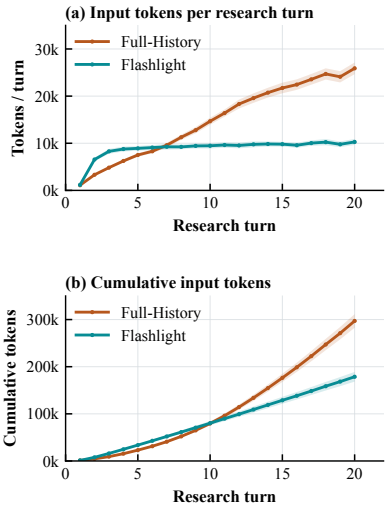


Figure 3: Input tokens by research depth for GPT-5.6-Luna trajectories reaching 20 turns. A research turn is one unified-agent call for Full-History or one Controller call for Flashlight. The curves exclude the one-time Designer and final Writer calls; Tab. 1 reports complete per-task totals.

Fig. 3 illustrates where the efficiency gain arises. Full-History repeatedly reintroduces an expanding transcript, causing the per-turn input to grow with research depth and the cumulative token cost to grow superlinearly. In contrast, FLASHLIGHT reconstructs each Controller input from persistent state and selected evidence, keeping per-turn context near 9–10k tokens even on trajectories reaching 20 research turns. As a result, cumulative input grows approximately linearly with research depth rather than through repeated replay of an expanding history. Equal-resource sweeps preserve the stronger quality–cost frontier (App. C.1), and the full cost decomposition is reported in Tab. 17. For Qwen3-8B, the raw checkpoint does not reliably execute the structured interface; after interface SFT, FLASHLIGHT reaches 46.0 RACE, and EIA-GRPO further raises it to 49.0 while reducing tokens, calls, and latency. We treat these adaptation results as an extension of the inference-time mechanism rather than as part of the closed-source comparison.

5.2 CONTROLLED MECHANISM ANALYSIS

The controlled within-pipeline comparisons in Tab. 2(a) isolate the contribution of context construction from that of the surrounding research workflow. Full-state replay retains the same Designer–Controller–Writer roles, validators, tools, task limits, and 14 LLM calls as FLASHLIGHT, but exposes all eligible accumulated state rather than selectively compiling context. Replacing full-state replay with Flashlight context construction improves RACE from 52.9 to 54.2 while reducing tokens from 118k to 72k, making this 1.3-point gain a cleaner estimate of the context mechanism than the native Full-History comparison. Simple truncation and summarization reduce token use but incur substantial quality loss, while a post-hoc independent LLM selector reaches 51.5 RACE and

requires 19 calls. ACE-style and ContextWeaver-style context controls, adapted within the same pipeline, also remain below FLASHLIGHT. These results indicate that the benefit does not come from bounded context alone, but from how the current evidence need is represented and reused.

We next isolate the plan signal itself. In addition to the end-to-end controls in Tab. 2(b), we evaluate plan variants on identical saved Controller states with a fixed candidate evidence pool, preventing later retrieval decisions from diverging. The complete next-context plan is compared against no-plan, requirement-only, stale-plan, task-shuffled, and equal-length controls. In this selection-only setting, the full plan improves evidence relevance and coverage by 7.8 and 8.5 percentage points, respectively, and the end-to-end controls preserve the same ordering: without response feedback, the full plan reaches 52.6 RACE, compared with 51.4 for query-only context, 51.6 for an equal-length dummy, and 51.2 for a task-shuffled plan. The equal-length and shuffled controls rule out prompt length and generic structured text as explanations, while the fixed-pool protocol shows that the gain appears already at evidence selection rather than only through downstream retrieval changes. Together, these results support the semantic content of the next-context plan as the operative signal.

The feedback controls test a different question: whether response behavior can identify *when* broader evidence exposure is useful, rather than merely benefiting from additional context. We compare the calibrated source-over-echo trigger with no broadening, the uncalibrated margin, always broadening, equal-activation random broadening, and a model-uncertainty trigger; triggered variants receive the same maximum additional-evidence and token budget. On held-out DRB and SQA-CS2 turns, a turn is labeled positive when broadening under this shared budget improves next-step evidence coverage. The calibrated trigger achieves 73.6% precision and 69.2% recall over 3,238 eligible turns and activates on 26.0% of them. It yields a +2.6-point gain on triggered cases and a +0.7-point pooled intent-to-treat gain, whereas equal-activation random broadening yields only +0.3 and +0.1 points, respectively. End-to-end, always broadening reaches 53.7 RACE but increases token use to 96k, while random broadening reaches 52.8 at approximately the same 72k-token cost as FLASHLIGHT; the calibrated trigger reaches 54.2 RACE with 72k tokens, and the uncalibrated variant reaches 54.0 with 74k. Replacing the added evidence with task-shuffled passages largely removes the gain, supporting the timing and evidence-specificity of the feedback rule rather than an explanation based on simply adding more context.

Finally, we examine implementation sensitivity and alternative explanations. Across two semantic encoders and four evidence-selection rules, end-to-end RACE varies by at most 0.6 points, indicating that the plan-first gain is not tied to the default encoder or Otsu thresholding rule (Tab. 9). For the response signal, the cross-regime association reported in Sec. 3 remains positive on a task-disjoint sensitivity split and in a regression controlling for answer length, number of searches, model family, benchmark, task difficulty, and research-turn count; shuffling evidence across tasks or response order removes the association. These analyses complement, rather than replace, the held-out broadening experiment above: the former tests whether the motivating behavioral signal is robust, whereas the latter tests whether the deployed trigger identifies useful broadening decisions. The default Context Engine adds only 2.8 CPU seconds per task. Full confidence intervals, negative controls, implementation variants, and protocol details are reported in App. C.2.

6 CONCLUSION

We introduced FLASHLIGHT, a plan-first and response-adaptive approach to context construction for deep research. Rather than reconstructing relevance from an accumulated transcript after the fact, Flashlight externalizes the evidence need when a search decision is formed and reuses it to construct the next context; the subsequent response then determines whether plan-relevant evidence should be exposed more broadly. A dependency-aware evidence contract carries these signals across acquisition, review, and synthesis. Across two closed-source backbones, the complete system improves DRB RACE by 1.4–2.2 points while using 45.5–45.6% fewer LLM tokens than native Full-History agents. More importantly, within the same Designer–Controller–Writer orchestration, replacing full-state replay with Flashlight context construction improves RACE by 1.3 points while reducing tokens from 118k to 72k. These results support a view of context construction as part of the research decision itself rather than as post-hoc transcript compression.

AI USE STATEMENT

In this work, we used generative AI tools to assist with checking parts of the system implementation and evaluation pipeline, refining experimental methods and evaluation procedures, and polishing the manuscript, figures, and tables. All AI-assisted content was reviewed and verified by the authors against the underlying code, experiments, results, and original literature. All final decisions regarding methodology, experimental design, interpretation of results, and scientific claims were made by the authors, who take full responsibility for the final content of this work.

REFERENCES

Mohammad Kalim Akram, Saba Sturua, Nastia Havriushenko, Quentin Herreros, Michael Günther, Maximilian Werk, and Han Xiao. jina-embeddings-v5-text: Task-targeted embedding distillation. *arXiv preprint arXiv:2602.15547*, 2026.

Salaheddin Alzubi, Creston Brooks, Purva Chiniya, Edoardo Contente, Chiara von Gerlach, Lucas Irwin, Yihan Jiang, Arda Kaz, Windsor Nguyen, Sewoong Oh, et al. Open Deep Search: Democratizing search with open-source reasoning agents. *arXiv preprint arXiv:2503.20201*, 2025.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avi Sil, and Hannaneh Hajishirzi. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In *International conference on learning representations*, volume 2024, pp. 9112–9141, 2024.

Jonathan Bragg, Mike D'Arcy, Nishant Balepur, Dan Bareket, Bhavana Dalvi Mishra, Sergey Feldman, Dany Haddad, Jena Hwang, Peter Jansen, Varsha Kishore, et al. AstaBench: Rigorous benchmarking of AI agents with a scientific research suite. In *International conference on learning representations*, volume 2026, pp. 110136–110223, 2026.

João Coelho, Jingjie Ning, Jingyuan He, Kangrui Mao, Abhijay Paladugu, Pranav Setlur, Jiahe Jin, Jamie Callan, João Magalhães, Bruno Martins, et al. DeepResearchGym: A free, transparent, and reproducible evaluation sandbox for deep research. *arXiv preprint arXiv:2505.19253*, 2025.

Mingxuan Du, Benfeng Xu, Chiwei Zhu, Licheng Zhang, Xiaorui Wang, and Zhendong Mao. DeepResearch Bench: A comprehensive benchmark for deep research agents. In *International Conference on Learning Representations*, volume 2026, pp. 42414–42448, 2026.

Sanchayan Dutta, Sai Niranjan Ramachandran, and Suvrit Sra. Active inference as context acquisition for AI agents. *arXiv preprint arXiv:2608.19202*, 2026.

Bradley Efron. Bootstrap methods: another look at the jackknife. In *Breakthroughs in statistics: Methodology and distribution*, pp. 569–593. Springer, 1992.

Xubin Hao, Hongjin Meng, Xin Yin, Jiawei Zhu, and Chenpeng Cao. Self-GC: Self-governing context for long-horizon LLM agents. *arXiv preprint arXiv:2607.00692*, 2026.

Sture Holm. A simple sequentially rejective multiple test procedure. *Scandinavian journal of statistics*, pp. 65–70, 1979.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. GPT-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. LLMLingua: Compressing prompts for accelerated inference of large language models. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pp. 13358–13376, 2023a.

Huiqiang Jiang, Qianhui Wu, Xufang Luo, Dongsheng Li, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. LongLLMLingua: Accelerating and enhancing LLMs in long context scenarios via prompt compression. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1658–1677, 2024.

Zhengbao Jiang, Frank F Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. Active retrieval augmented generation. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pp. 7969–7992, 2023b.

Jiarui Jin, Zexuan Yan, Shijian Wang, Wenxiang Jiao, and Yuan Lu. AgentDisCo: Towards disentanglement and collaboration in open-ended deep research agents. *arXiv preprint arXiv:2605.11732*, 2026.

Klaus Krippendorff. Computing Krippendorff's Alpha-reliability, 2011. Technical report.

Daniël Lakens. Calculating and reporting effect sizes to facilitate cumulative science: a practical primer for t-tests and anovas. *Frontiers in psychology*, 4:863, 2013.

Yu Lei, Shuzheng Si, Wei Wang, Yifei Wu, Gang Chen, Fanchao Qi, and Maosong Sun. RhinoInsight: Improving deep research through control mechanisms for model behavior and context. *arXiv preprint arXiv:2511.18743*, 2025.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive NLP tasks. *Advances in neural information processing systems*, 33: 9459–9474, 2020.

Mo Li, LH Xu, Qitai Tan, Long Ma, Hongyong Song, Ting Cao, and Yunxin Liu. Sculptor: Empowering LLMs with cognitive agency via active context management. In *International Conference on Learning Representations*, volume 2026, pp. 153411–153440, 2026a.

Ruizhe Li, Mingxuan Du, Benfeng Xu, Chiwei Zhu, Xiaorui Wang, and Zhendong Mao. DeepResearch Bench II: Diagnosing deep research agents via rubrics from expert reports. *arXiv preprint arXiv:2601.08536*, 2026b.

Xiaochuan Li, Ryan Ming, Meng Chu, Shuai Shao, Rong Jin, and Chenyan Xiong. ACM: Agentic context management for long horizon tasks. *arXiv preprint arXiv:2607.23809*, 2026c.

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. *Advances in Neural Information Processing Systems*, 38:120091–120131, 2026d.

Yishan Li, Wentong Chen, Yukun Yan, Mingwei Li, Sen Mei, Xiaorong Wang, Kunpeng Liu, Xin Cong, Shuo Wang, Zhong Zhang, et al. AgentCPM-Report: Interleaving drafting and deepening for open-ended deep research. *arXiv preprint arXiv:2602.06540*, 2026e.

Zijian Li, Xin Guan, Bo Zhang, Shen Huang, Houquan Zhou, Shaopeng Lai, Ming Yan, Yong Jiang, Pengjun Xie, Fei Huang, et al. WebWeaver: Structuring web-scale evidence with dynamic outlines for open-ended deep research. In *International Conference on Learning Representations*, volume 2026, pp. 4173–4217, 2026f.

Ning Liao, Zihao Long, Xiaoxing Wang, Xue Yang, Yaoming Wang, Ziyuan Zhuang, Xunliang Cai, Rongxiang Weng, and Junchi Yan. ACE: Pluggable adaptive context elasticizer across agents. *arXiv preprint arXiv:2606.31564*, 2026.

Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *Transactions of the association for computational linguistics*, 12:157–173, 2024.

Shukai Liu, Bo Jiang, Jian Yang, Yizhi Li, Jinyang Guo, Xianglong Liu, and Bryan Dai. Context as a tool: Context management for long-horizon SWE-agents. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 20604–20617, 2026.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. Self-Refine: Iterative refinement with self-feedback. *Advances in neural information processing systems*, 36:46534–46594, 2023.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, et al. WebGPT: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332*, 2021.

OpenAI. Introducing GPT-4.1 in the API. `https://openai.com/index/gpt-4-1/`, 2025. Accessed: 2026-09-26.

Nobuyuki Otsu. A threshold selection method from gray-level histograms. *IEEE transactions on systems, man, and cybernetics*, 9(1):62–66, 1979.

Arnold Overwijk, Chenyan Xiong, and Jamie Callan. ClueWeb22: 10 billion web documents with rich information. In *Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval*, pp. 3360–3362, 2022.

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

Qwen Team, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

Parth Sarthi, Salman Abdullah, Aditi Tuli, Shubh Khanna, Anna Goldie, and Christopher Manning. RAPTOR: Recursive abstractive processing for tree-organized retrieval. In *International Conference on Learning Representations*, volume 2024, pp. 32628–32649, 2024.

Rulin Shao, Akari Asai, Shannon Zejiang Shen, Hamish Ivison, Varsha Kishore, Jingming Zhuo, Xinran Zhao, Molly Park, Samuel G Finlayson, David Sontag, et al. DR Tulu: Reinforcement learning with evolving rubrics for deep research. *arXiv preprint arXiv:2511.19399*, 2025.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed H Chi, Nathanael Schärli, and Denny Zhou. Large language models can be easily distracted by irrelevant context. In *International conference on machine learning*, pp. 31210–31227. PMLR, 2023.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *Advances in neural information processing systems*, 36:8634–8652, 2023.

Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, Aidan McLaughlin, Aiden Low, AJ Ostrow, Akhila Ananthram, et al. OpenAI GPT-5 system card. *arXiv preprint arXiv:2601.03267*, 2025.

Qwen Team et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 6(7):13, 2025a.

Tongyi DeepResearch Team, Baixuan Li, Bo Zhang, Dingchu Zhang, Fei Huang, Guangyu Li, Guoxin Chen, Huifeng Yin, Jialong Wu, Jingren Zhou, et al. Tongyi DeepResearch technical report. *arXiv preprint arXiv:2510.24701*, 2025b.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533*, 2022.

Yating Wu, Yuhao Zhang, Sayan Ghosh, Sourya Basu, Anoop Deoras, Jun Huan, and Gaurav Gupta. ContextWeaver: Selective and dependency-structured memory construction for LLM agents. *arXiv preprint arXiv:2604.23069*, 2026.

Fangyuan Xu, Weijia Shi, and Eunsol Choi. RECOMP: Improving retrieval-augmented LMs with context compression and selective augmentation. In *International Conference on Learning Representations*, volume 2024, pp. 43478–43502, 2024.

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

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2022.

Rui Ye, Zhongwang Zhang, Kuan Li, Huifeng Yin, Zhengwei Tao, Yida Zhao, Liangcai Su, Liwen Zhang, Zile Qiao, Xinyu Wang, et al. AgentFold: Long-horizon web agents with proactive context management. *arXiv preprint arXiv:2510.24699*, 2025.

Li S Yifei, Allen Chang, Chaitanya Malaviya, and Mark Yatskar. ResearchQA: Evaluating scholarly question answering at scale across 75 fields with survey-mined questions and rubrics. *Transactions of the Association for Computational Linguistics*, 14:1344–1368, 2026.

Ori Yoran, Tomer Wolfson, Ori Ram, and Jonathan Berant. Making retrieval-augmented language models robust to irrelevant context. In *International Conference on Learning Representations*, volume 2024, pp. 29862–29883, 2024.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging LLM-as-a-judge with MT-Bench and Chatbot Arena. *Advances in neural information processing systems*, 36:46595–46623, 2023.

A EIA REWARD AND GRPO TRAINING DETAILS

EIA reward and fixed evidence worlds for GRPO. Each training world contains a request, an evidence contract state, a set of candidate acquisition actions, stored tool results, and labels that connect each result to the required facts it contains. Every candidate has a stable action ID and an acquisition descriptor containing its requirement and query proposal. The policy receives the request, current contract state, IDs, and descriptors, but no stored result or required-fact label. It emits an ID and query, after which the simulator resolves the stored result by exact ID and runs the same Controller review and engine validation used at inference time. Unknown or unsupported IDs receive no result and $v_i = 0$; there is no semantic nearest-neighbor fallback. This construction gives all sampled actions a stable evidence environment. Let $\mathcal{U}_t$ denote the unresolved requirements before an action and ℓ_t a reward-only coverage ledger hidden from the policy. For candidate action a_i, the Effective Information Acquisition (EIA) reward is

$$R_i^{\mathrm{EIA}} = v_i \sum_{r \in \mathcal{U}_t} \Big[c_r(\ell_{t+1}^{(i)}) - c_r(\ell_t) \Big]_+ , \tag{12}$$

where $c_r(\ell)$ counts the distinct required facts covered for requirement r and $[x]_+ = \max(x, 0)$. We operationalize effective information narrowly as newly validated required-fact coverage under a valid transition: an acquisition earns reward for previously missing facts, whereas repeated facts and invalid transitions earn no additional reward. EIA does not directly optimize token cost, latency, source quality, or lexical novelty. The reward evaluator updates ℓ_t only after runtime validation, and the ledger is never serialized into $\mathcal{C}_t$. We publish action-to-result maps, report the out-of-support action rate, and test the promoted policy with live search to measure replay-to-deployment transfer.

For one Controller input $\mathcal{C}_t$, GRPO samples a group of G candidate actions. Their normalized advantages are

$$\widehat{A}_i = \frac{R_i^{\mathrm{EIA}} - \overline{R}}{\mathrm{std}(R_1^{\mathrm{EIA}}, \ldots, R_G^{\mathrm{EIA}}) + \epsilon_A}, \tag{13}$$

where $\overline{R}$ is the group mean reward and $\epsilon_A > 0$ is a numerical stability constant. For token j in action a_i, define $\rho_{i,j}(\theta) = p_\theta(a_{i,j} \mid \mathcal{C}_t, a_{i,<j}) / p_{\theta_{\mathrm{old}}}(a_{i,j} \mid \mathcal{C}_t, a_{i,<j})$. We minimize

$$\mathcal{L}_{\mathrm{GRPO}}(\theta) = -\mathbb{E}\Bigg[\frac{1}{G} \sum_{i=1}^{G} \frac{1}{|a_i|} \sum_{j=1}^{|a_i|} \min\Big(\rho_{i,j}(\theta)\widehat{A}_i, \; \mathrm{clip}(\rho_{i,j}(\theta), 1-\epsilon, 1+\epsilon)\widehat{A}_i \Big) - \beta D_{\mathrm{KL}}(\pi_\theta \,\|\, \pi_{\mathrm{ref}}) \Bigg], \tag{14}$$

where ϵ is the clipping radius, β weights the KL penalty, and π_{ref} is the frozen reference policy.

Optimization settings. We initialize GRPO from the SFT checkpoint and optimize Controller acquisition turns with group size 8, clipping radius 0.2, KL coefficient 0.01, learning rate 5×10^{-6}, and 1,000 updates. The reference policy is the frozen SFT checkpoint. We report end-to-end benchmark results for SFT and SFT plus GRPO in Tab. 1.

Role-tagged SFT data. We curate 6,940 high-quality role examples from 1,000 training questions. Each example contains the exact model input constructed by the engine, one role tag, and a validated target output. The three tags are `Designer`, `Controller`, and `Writer`. They train one shared model rather than three separate models. Designer examples cover requirement decomposition and DAG construction. Controller examples cover query and next-context plan generation, evidence review, and state updates. Writer examples cover synthesis from the reviewed evidence packet. We split questions and evidence worlds before constructing examples so that held-out sources do not enter training inputs. The verified mixture contains 3,764 Designer, 2,000 Controller, and 1,176 Writer examples after quality filtering and train/validation splitting.

B EXPERIMENTAL DETAILS

All methods use the task sets listed in Tab. 3. We record the exact model version, retrieval date, decoding configuration, prompts, tool limits, final-report settings, random seed, and evaluator for

every run. The Full-History and Flashlight system comparison fixes the research model, task assignments, search and browsing backend, maximum tool and output budgets, citation format, and evaluator. Full-History retains its unified-agent prompt, cumulative transcript, and natural stopping behavior, whereas Flashlight uses the Designer–Controller–Writer workflow and Context Engine. This comparison therefore measures the complete system difference, not the isolated effect of the context compiler. Component attribution instead uses Flashlight-pipeline controls that share roles and validators and change one mechanism at a time. We use the official set sizes (100/132/776/100/1,000) for DRB, DRB-II, ResearchQA, ScholarQA-CS2, and DeepResearchGym, respectively. The system-level evaluation uses one run per task and reports 95% task-bootstrap confidence intervals; training and focused mechanism studies use three seeds.

B.1 Benchmark metric structure

Table 3: Official metric structure of the evaluation benchmarks. Tab. 1 reports the primary comparison metrics; the remaining dimensions provide the detailed diagnostic breakdown.

Benchmark	Main-table metrics	Additional official diagnostics
DeepResearch Bench	RACE overall, FACT citation accuracy, FACT effective citations	RACE comprehensiveness, depth, instruction following, and readability
DeepResearch Bench II	Total rubric score	Information recall, analysis, and presentation
ResearchQA	Rubric coverage and leaderboard score	Coverage by research domain and rubric type
ScholarQA-CS2	Official aggregate score	Answer coverage, answer relevance, citation precision, and citation recall
DeepResearchGym	Key-point recall and contradiction	Citation precision, citation recall, clarity, and insightfulness

B.2 Metric definitions and units

Tab. 4 defines the units used throughout the paper. The main distinctions are that FACT effective citations is a count, ResearchQA's leaderboard value is an Elo score, and DeepResearchGym's KPR and KPC metrics are percentages.

Table 4: Metric definitions and units used in our evaluation. Metrics with different units or evaluator semantics are reported separately.

Benchmark	Metric and unit	Interpretation
DeepResearch Bench	RACE overall	Overall report quality; higher is better.
DeepResearch Bench	FACT C.Acc. (%) / E.Cit. (count)	Citation accuracy is a percentage. Effective citations is a count and therefore depends on report length, citation style, and statement–URL deduplication.
DeepResearch Bench II	Total satisfied-rubric score (%)	Fraction of the benchmark rubric satisfied by the report.
ResearchQA	Coverage (%) / leaderboard score (Elo)	Coverage is a per-report percentage. Elo is fitted from pairwise comparisons within a shared tournament pool and is not a percentage.
ScholarQA-CS2	Official four-component average (%)	Average of answer coverage, answer relevance, citation precision, and citation recall.
DeepResearchGym	KPR / KPC (%)	Key-point recall is higher-is-better. Key-point contradiction is lower-is-better.

We preserve the official scorer's precision in released result files while rounding paper tables for readability. Missing official metrics remain blank rather than being inferred from a similarly named field.

Table 5: Baseline implementation and protocol audit. "Adapted" means that we implemented the paper's operative mechanism inside the common research harness; it does not mean that the original production system was reproduced bit for bit. Native and common-protocol results are reported separately.

Method	Artifact status	Native paper setting	Our benchmark implementation and sources of deviation
ACE	Adapted; no runnable official repository identified in the reviewed paper version	GPT-4.1 as the agent and elasticizer, GPT-4o (Hurst et al., 2024) for abstraction/vision, Google Serper search, Jina Reader, and at most 50 actions	We implement raw/abstract state typing and elastic compression in the common harness with GPT-4.1, Bing v3, Crawl4AI, common task IDs, and official scorers. Adapter fidelity, retrieval breadth, and index date remain deviation sources.
Context-Weaver	Adapted to deep research	Originally evaluated as a dependency-graph memory for SWE-Agent on SWE-Bench across GPT-5 (Singh et al., 2025) and other proprietary backbones; it is not a released drop-in deep-research agent	We port node extraction, dependency edges, ancestry retrieval, and the five-step recent window to research traces, using GPT-4.1 with common Bing/Crawl4AI and Writer settings. This tests the mechanism, not the original coding-agent product.
WebWeaver	Author workflow/components where runnable	The released workflow uses a proprietary planner/writer backbone, auxiliary URL selection, dynamic outlines, hierarchical drafting, and much larger evidence/output budgets	Our adaptation uses GPT-4.1 for planning and writing while preserving the released workflow and normalizing task IDs, retrieval date, Bing/Crawl4AI, and scorer inputs. Reduced saved-page and output budgets can lower FACT citation counts.
RhinoInsight	Style-adapted; complete production stack is not released	A proprietary backbone with a verifiable checklist, iterative outline, evidence audit, critic ranking, and production search/crawler services	We implement the checklist and evidence-audit interfaces with GPT-4.1 in the common harness and label the result as adapted because the critic details and production retrieval service cannot be reproduced exactly.
WebThinker	Public code	QwQ-32B main model, Qwen2.5 assistant models (Qwen Team et al., 2024), Bing Web Search (US-English, top 10), Crawl4AI, deep page navigation, iterative drafting, and an 81,920-token generation ceiling	We run the compatible public report-mode workflow with pinned Bing locale, page parser, task budget, and scorer. Assistant-model choice and navigation behavior remain explicit deviation sources.
Tongyi DeepResearch	Public model, code, and reproduction scripts	Qwen3-30B-A3B with Search, Visit, Python, Google Scholar, and File Parser; Google-based retrieval and up to 128 tool calls	We retain the released checkpoint and agent loop and also evaluate a common Bing/Crawl4AI condition with the shared call ceiling. The removal of Scholar/File tools and native summarization is disclosed.
OpenDeep-Search	Open agent evaluated in DeepResearchGym	Default agent loop with its native commercial retrieval integration; DeepResearchGym also reports a controlled ClueWeb22-B (Overwijk et al., 2022) condition	We preserve the agent loop and evaluate native-retrieval and common Bing/Crawl4AI variants. Corpus coverage, ranking, and URL-to-citation conversion remain score-shift sources.
DR Tulu-8B	Public checkpoint, data, code, and evaluation stack	Qwen3-8B with Google Serper, Jina browsing, Semantic Scholar full text, top-10 retrieval, and at most 10 tool calls	We run the released checkpoint and canonicalize only transport formatting before official scoring. The separate common-Bing condition is labeled as an adaptation; Tab. 1 uses our pinned evaluator.

ResearchQA tournament protocol. Coverage is computed independently for every report and is the directly reproducible ResearchQA metric. For the leaderboard score, all compared systems enter one frozen tournament pool with a predeclared pairing schedule, judge prompt, tie rule, random seed, and Elo-fitting implementation. We publish the pairwise comparison matrix and fitting code. Scores from different pools are not compared as though they shared an absolute scale.

B.3 Baseline provenance, adaptation, and retrieval normalization

Exact reproduction is not equally possible for every comparison system. Tab. 5 distinguishes an author release from a method-faithful adapter and records the native retrieval stack before any normalization. A paper number is cited as a published/native result. A number produced by our infrastructure is labeled as a local reproduction. We never merge these two provenance classes into a single row or treat a port to Bing as an exact reproduction of a Google-, Serper-, proprietary-, or corpus-backed system. Their common-protocol measurements appear in the first block of Tab. 1; the fixed-backbone blocks below them keep the research model, task set, retrieval stack, maximum budgets, citation format, and evaluator unchanged.

Common Bing adapter. For an adapted run, only the retrieval and transport boundary is normalized: the baseline retains its own planning, memory, stopping, and writing logic. We map a native search call to Bing v3 with a pinned market, language, safe-search setting, top-k, date, and retry policy; map page visits to Crawl4AI; preserve query order and the method's native stopping rule; and store the raw search response, fetched page, timestamp, and parser status. If a native tool has no faithful counterpart, such as Google Scholar, Python execution, a proprietary critic, or a file parser, we either retain that tool in a native-protocol run or mark the common-protocol cell unsupported. We do not silently replace it with extra Bing searches.

Two-view reporting. Whenever feasible, each externally proposed system receives two rows in the released result sheet: an author/native view and a common-protocol view. The native view answers whether we can recover the paper's behavior; the common view controls Bing/Crawl4AI, task IDs, evaluator, search ceiling, and report format. Only the common view is used for direct paired comparisons with Flashlight. Closed-source model version, provider region, safety filters, unavailable URLs, parser failures, retries, and fallback behavior are logged per task.

B.4 Diagnostic and efficiency metrics

The runtime scores each acquisition candidate with the same fixed semantic encoder used for context construction. Query relevance (QRel) averages the selected candidate's semantic relevance to the active requirement and its remaining gap. Query diversity (QDiv) averages two novelty terms. The first compares the selected query and expected answer change with earlier executed queries. The second compares them with the other candidates proposed in the same turn. Cosine similarities are mapped from $[-1, 1]$ to $[0, 1]$, and QRel and QDiv are reported as percentages. Evidence coverage (ECov) is the percentage of required Designer nodes that finish with an engine-checked evidence binding and completion certificate. In Tabs. 1 and 2, "Tokens" is total input plus output tokens from the Designer, Controller, and Writer. LLM calls count all calls to these roles and any selector call used by a baseline. Tool calls count search and browsing operations. End-to-end latency is measured from the initial request to the accepted Writer report, including model generation, search, browsing, parsing, retries, and queueing. Each efficiency entry in the main and ablation tables is a per-task benchmark-macro mean: we first average all assigned tasks within each of the five datasets and then take the unweighted mean of those five dataset means under intent-to-treat accounting. Tokens, calls, and latency are therefore not summed across datasets, and the larger ResearchQA and DeepResearchGym sets do not dominate the aggregate.

B.5 Controlled ablation protocol

The Flashlight-pipeline ablations use the same tasks, Designer, Controller, review schedule, model, tools, Writer, stopping rule, validators, and task limits. The "full-state replay" control retains this pipeline but exposes every eligible accumulated state and source record up to the common input limit, without plan-guided filtering or response-triggered broadening. This control, rather than the unified-agent Full-History system, isolates context compilation. The Full-History row is retained as an end-to-end system reference under the same backbone, task set, retrieval stack, maximum budgets, and evaluator. They retain live retrieval because the requirement DAG and Controller schedule can change later queries. We repeat the stratified mechanism subset with 3 seeds and report uncertainty. Selection-only comparisons use saved Controller states and the same candidate evidence pool. These comparisons isolate next-context planning, semantic selection, and response feedback from retrieval variation. The Full-plan/no-feedback uses one Otsu selection. Query-only and query-plus-active- requirement controls separate query wording from the remaining plan fields. Always broadening applies the second selection on every eligible turn. Flashlight applies it only when the cardinality-calibrated source-over-echo margin is negative. The uncalibrated margin is reported separately. LLM-call counts include every executed Designer, Controller, Writer, and baseline-selector call. Relative to independent selection, Flashlight removes the additional selector call.

The independent selector uses the same GPT-5.6-Luna checkpoint at temperature zero and runs once before each evidence-bearing Controller input. It receives the active requirement, current query, accumulated Controller history, and the same candidate record identifiers, titles, and passages available to Flashlight, but no evaluator labels. It returns an ordered list of record IDs under the same exposure

budget. This adds 5.0 model calls, 15.6k input tokens, and 0.4k output tokens per task on average, explaining the increase from 14 to 19 calls and from 72k to 88k total tokens in Tab. 2.

The ACE-style context-compiler control retains the same Designer, Controller, Writer, evidence contract, tools, validators, stopping rule, and task limits as Flashlight. It replaces the next-context-plan selector and response trigger with raw and abstract memory views followed by a separate LLM exposure decision before each evidence-bearing Controller call. This isolates a post-hoc generative selection strategy inside the common pipeline; it is not a reproduction of the complete ACE system or its native orchestration. The ContextWeaver-style dependency-memory control likewise retains the common pipeline but replaces plan-guided compilation with dependency nodes over executed research steps. Each input retrieves relevant ancestor nodes together with a five-step recent window, without next-context-plan scoring or response-triggered broadening. It tests dependency-structured memory in the deep-research pipeline rather than reproducing the original coding-agent system. All model calls, including auxiliary selection or abstraction calls, are counted in the reported totals.

C Additional Validation and Robustness Results

The main text establishes the central result with two tables: Tab. 1 reports end-to-end quality and cost, while Tab. 2 isolates the core mechanism through Full-state replay, plan controls, and feedback controls. This appendix provides the supporting validation needed to interpret those results. It tests resource limits, contract and citation quality, interface compliance, reward attribution, failure accounting, evaluator dependence, retrieval perturbations, leakage, and model-family sensitivity. These analyses rule out alternative explanations; they are not additional co-equal components of the core Flashlight mechanism. Tab. 6 summarizes which controlled comparison addresses each principal alternative explanation.

Table 6: Controls for the principal empirical conclusions. All comparisons are paired by task and use intent-to-treat accounting.

Conclusion	Alternative explanation	Controlled comparison	Result
Quality–efficiency	Gains arise from a larger token, call, or latency budget	Equal-token, equal-call, and equal-latency comparisons plus a budget sweep of Full-History, truncation, summarization, independent selection, and Flashlight	Flashlight retains the stronger quality–cost frontier under identical resource caps.
Next-context plan	The benefit comes from extra prompt text or a stronger template	Full plan, requirement-only, stale, task-shuffled, equal-length dummy, and oracle plans with identical candidate pools and serialized lengths	The complete plan improves evidence relevance and coverage beyond content controls with the same serialized length.
Response feedback	Any additional evidence or heuristic trigger would help	No feedback, uncalibrated and cardinality-calibrated margins, random broadening with the same activation rate, always broadening, and an uncertainty trigger under an equal added-evidence budget	Cardinality-calibrated feedback improves diversity with lower token and latency cost than always broadening.
DAG and review schedule	Improvements are caused by more calls or more verbose state	Dependency DAG versus flat and edge-shuffled requirements; separate versus joint review at native resources and at identical token, call, and latency limits	The DAG improves evidence coverage, and separate review improves citation accuracy under identical resource limits.
Generalization	Results depend on one model, benchmark, context window, or search backend	Five benchmark strata, two context windows, a frozen corpus, a second search backend, and multiple model families	The quality–efficiency direction persists across models, benchmarks, and retrieval perturbations.
SFT and GRPO	Gains come from data scale, formatting, invalid-action filtering, or leakage rather than EIA	SFT data-scale and role-mixture ablations; from one SFT checkpoint compare no RL, EIA, terminal-quality, validity-only, ungated-coverage, shuffled-label, shuffled action–result, and zero-reward controls	SFT establishes interface compliance; EIA-GRPO then adds 3.0 RACE points from the same checkpoint.
Reliability and judging	Failed tasks are dropped or results depend on one LLM judge	Intent-to-treat accounting, schema and failure audits, an independent judge, and a blinded human subset	Failure-inclusive and independently rescored estimates preserve the primary effect direction.

C.1 Budget-Controlled Baselines and Pareto Results

Flashlight retains its quality advantage under identical test-time resource limits, and its operating points trace a stronger quality–cost frontier than standard context strategies. Each backbone is evaluated at four predeclared budgets: 25%, 50%, 75%, and 100% of the median Full-History token budget. We additionally enforce identical maximum LLM calls and identical end-to-end latency cutoffs. A run that reaches a cutoff must emit its current best report; it is not restarted with a larger budget. This produces three complementary comparisons: equal token budget, equal model-call budget, and equal latency budget. Tab. 7 reports the shared token-budget curve and its quality–log-cost area. The equal-call and equal-latency comparisons are reported below the table. The native

setting is retained because it measures the intended system, but it cannot by itself establish efficiency. Baseline prompts, stopping rules, retry policies, and tool limits are archived, and the table uses only common-protocol reproductions.

Table 7: DRB RACE under GPT-5.6-Luna token budgets, expressed as fractions of the 132k Full-History reference. AUC is the normalized RACE–log-token area over 25–100%. Best values are bold and second-best values are underlined.

Method	25% (33k)	50% (66k)	75% (99k)	100% (132k)	AUC ↑
FULL-HISTORY	43.8	47.2	50.3	<u>52.0</u>	47.6
Truncation	44.6	46.9	47.6	47.8	46.6
Summarization	45.1	47.8	48.8	49.2	47.5
Independent selector	<u>46.0</u>	<u>49.0</u>	<u>50.7</u>	51.5	<u>48.9</u>
Flashlight	**48.3**	**52.5**	**53.8**	**54.2**	**52.0**

Flashlight leads at every shared budget and improves normalized quality–log-cost AUC by 3.1 points over the strongest alternative selector. The advantage is already 2.3 RACE points at the smallest budget and grows to 3.5 points at the 50% budget, where context allocation is most consequential. The same direction holds under the other two resource definitions. At a shared cap of 72k total tokens, Full-History and Flashlight score 48.9 and 54.2 RACE. At a shared cap of 14 model calls, they score 50.1 and 54.2. At a shared latency cap of 365 seconds, they score 49.7 and 54.2. The corresponding paired advantages are 5.3, 4.1, and 4.5 RACE points. Thus the native operating point is not advantaged by receiving a larger token, call, or latency budget.

C.2 MECHANISM ATTRIBUTION BEYOND REMOVAL ABLATIONS

Motivating correlation audit. Fig. 1 analyzes 600 observed trajectories and documents the positive association between the context-over-echo margin and report quality. The 300 open-weight and 300 closed-source model observations shown in Fig. 1 yield $\rho = 0.55$ and $\rho = 0.29$, respectively. A task-disjoint sensitivity split that was not used to choose the trigger, encoder, or threshold retains correlations of $\rho = 0.51$ with a 95% paired-bootstrap interval of $[0.42, 0.59]$ and $\rho = 0.27$ with an interval of $[0.16, 0.37]$. A task-level regression controlling for answer length, number of searches, model family, benchmark, task difficulty, and research-turn count retains a positive standardized coefficient of 0.19 ($p < .001$). The association disappears after shuffling evidence across tasks ($\rho = 0.02$) or response order within trajectories ($\rho = -0.01$). Cardinality calibration is fitted on development data and the zero trigger is frozen before end-to-end evaluation. Tab. 10 reports precision and recall for identifying turns whose next-step evidence broadening improves held-out quality.

Next-context plan controls. We test whether the complete plan improves evidence relevance and coverage beyond controls that only add prompt content. We compare it with five controls: no plan, requirement identifier only, the previous turn's stale plan, a plan shuffled across tasks within the same benchmark, and an oracle plan written from the gold rubric but without gold evidence. Equal-length controls receive inert text so that a longer prompt is not mistaken for a better mechanism. Selection-only evaluation uses the same saved Controller state and candidate evidence pool for all variants. End-to-end evaluation then tests whether plan changes alter later retrieval. Tab. 8 reports evidence-selection Jaccard overlap together with QRel, ECov, RACE, tokens, and latency.

Table 8: Held-out plan controls with a fixed candidate pool and no response feedback. Jaccard is overlap with evidence selected by the complete plan. The oracle plan is an upper-bound diagnostic and is not used by Flashlight. Best non-oracle values are bold and second-best values are underlined.

Plan input	Jaccard	QRel (%) ↑	ECov (%) ↑	DRB RACE ↑	Tokens (k) ↓	Latency (s) ↓
Query only	0.52	64.8	66.9	51.4	<u>76</u>	<u>345</u>
Requirement identifier only	0.69	<u>68.9</u>	<u>71.6</u>	<u>51.9</u>	74	352
Stale previous-turn plan	0.57	65.7	67.8	50.9	76	358
Task-shuffled plan	0.44	62.9	65.8	51.2	77	354
Equal-length dummy	0.46	64.1	66.7	51.6	78	355
Complete plan	1.00	**72.6**	**75.4**	**52.6**	**66**	**340**
Oracle rubric plan	0.82	76.9	79.8	55.1	69	348

The complete plan improves QRel by 7.8 pp and ECov by 8.5 pp over query-only selection while reducing both tokens and latency. Shuffling or replacing its content removes the gain, so prompt length alone does not explain the result. The 2.5-point RACE gap to the oracle plan quantifies remaining headroom in plan quality.

Encoder and threshold sensitivity. We test whether plan-guided selection is stable across encoder and threshold choices. We compare Jina Embeddings v5 Text Nano (Akram et al., 2026) with E5 (Wang et al., 2022) and repeat selection-only experiments with cosine thresholding, fixed top-k, percentile thresholding, and Otsu selection. Encoder and threshold hyperparameters are chosen only on development tasks. Tab. 9 reports the complete encoder–rule comparison.

Table 9: Encoder and selection-rule sensitivity. Rank agreement is Spearman agreement with the default Jina v5 Nano Otsu ordering. Paraphrase stability is evidence-selection Jaccard after meaning-preserving plan rewrites. CPU duration is the per-task Context Engine total.

Encoder	Rule	Rank agreement ↑	Paraphrase Jaccard ↑	DRB RACE ↑	Engine CPU (s) ↓	No-pass turns (%) ↓
Jina v5 Nano	Otsu (default)	1.00	0.82	**54.2**	2.8	4.8
E5	Otsu	0.88	0.80	<u>54.0</u>	3.1	5.2
Jina v5 Nano	Fixed top-k	<u>0.93</u>	**0.84**	53.9	**2.5**	**0.0**
E5	Fixed top-k	0.89	0.82	53.8	2.9	**0.0**
Jina v5 Nano	Percentile	0.90	<u>0.83</u>	<u>54.0</u>	<u>2.7</u>	**0.0**
E5	Percentile	0.89	0.81	53.9	3.0	**0.0**
Jina v5 Nano	Cosine threshold	0.84	0.78	53.7	3.0	1.7
E5	Cosine threshold	0.82	0.76	53.6	3.4	2.1

Every encoder–rule combination remains within 0.6 RACE points of the default. The default Jina–Otsu rule leaves 4.8% of turns without a positive passing candidate, for which the engine follows the declared fallback policy. Fixed top-k removes that case but slightly lowers quality by admitting weak evidence.

Response-feedback controls. We test whether the alignment trigger gives a better diversity–cost trade-off than always or indiscriminately broadening the context. It is compared with no broadening, the uncalibrated source-over-echo margin trigger, always broadening, random broadening with the same empirical activation rate, and a model-uncertainty trigger. Each triggered method receives the same maximum number of additional evidence records and tokens. We report the activation rate, the fraction of activations followed by a new source domain, the change in response–source alignment, and the score delta separately for triggered and non-triggered tasks. A negative control with task-shuffled evidence tests whether gains come from additional relevant evidence rather than simply additional text.

Table 10: Held-out response-feedback diagnostics. A turn is positive when broadening under the same added-evidence budget improves next-step evidence coverage. Precision and recall use this turn-level label; score deltas compare paired broaden and no-broaden continuations. Triggered Δ conditions on the trigger's selected cases, whereas overall Δ is the intent-to-treat effect across all eligible cases. Within the pooled comparison, best values are bold and second-best values are underlined; activation rate is descriptive and is not ranked.

Trigger	Evaluation slice	Eligible turns	Precision (%) ↑	Recall (%) ↑	Activation (%)	New domain (%) ↑	Triggered Δ (points) ↑	Overall Δ (points) ↑
Cardinality-calibrated margin	DRB	1,846	72.4	68.1	24.6	63.8	+2.3	+0.6
Cardinality-calibrated margin	SQA-CS2	1,392	75.1	70.6	27.9	68.4	+3.1	+0.9
Cardinality-calibrated margin	Pooled	3,238	**73.6**	<u>69.2</u>	26.0	**65.9**	**+2.6**	**+0.7**
Uncalibrated margin	Pooled	3,238	<u>65.4</u>	**73.8**	31.8	<u>59.7</u>	<u>+1.9</u>	<u>+0.6</u>
Model uncertainty	Pooled	3,238	61.7	62.5	26.4	56.8	+1.4	+0.4
Equal-activation random	Pooled	3,238	49.2	48.6	26.0	44.9	+0.3	+0.1

The cardinality-calibrated trigger trades a modest reduction in recall relative to the uncalibrated margin trigger for higher precision, more new-domain evidence, and a larger activation-conditional score gain. Its pooled activation rate is 26.0%, so most turns retain the narrow primary selection. Random broadening with the same activation rate yields only a 0.3-point conditional gain, supporting the timing and evidence-specificity of the feedback rule rather than an additional-context explanation. On triggered turns, calibrated feedback raises response–source alignment by 0.09 and task score by 2.6 points; on non-triggered tasks the corresponding score change is 0.2 points. Replacing the added evidence with task-shuffled passages reduces the conditional gain to 0.1 points. The pooled intent-to-treat effect is +0.7 points for calibrated feedback and +0.1 points for random broadening with the same activation rate, so the conclusion does not rely only on comparing differently selected triggered subsets.

Requirement graph and review schedule. We test whether the dependency DAG improves coverage and whether a separate review turn improves citation accuracy when resources are held constant. The dependency DAG is compared with a flat requirement list, randomly shuffled edges that preserve node count, and an oracle dependency order derived from the evaluation rubric. The separate acquisition/review schedule is compared with a joint turn both at native resources and with identical call and output-token limits by allowing the joint system an additional reasoning pass. These controls separate the effect of dependency structure from prompt length and the effect of review separation from extra inference. We record illegal transitions, unsupported completion certificates, repeated queries, and latency to first validated evidence binding. Tab. 11 reports the controlled DAG and review-schedule comparisons.

Table 11: Requirement-graph and review-schedule controls. Native rows retain each variant's natural computation. Equal-resource rows fix tokens, calls, and latency to isolate review separation. Best and second-best quality values are marked within each compute regime.

Variant	Compute	DRB RACE ↑	ECov (%) ↑	C.Acc. (%) ↑	Tokens (k) ↓	Calls ↓	Latency (s) ↓
Flat requirements	Native	50.8	69.0	86.8	70	14	350
Edge-shuffled requirements	Native	51.3	71.2	87.5	71	14	358
Full DAG	Native	<u>54.2</u>	<u>82.0</u>	<u>91.0</u>	72	14	365
Oracle dependency order	Native	**55.0**	**85.2**	**92.0**	74	14	372
Joint acquisition/review	Native	51.0	68.5	86.5	67	12	315
Joint acquisition/review	Equal resources	<u>52.1</u>	<u>74.6</u>	<u>88.1</u>	72	14	365
Separate review	Equal resources	**54.0**	**80.9**	**90.6**	72	14	365

The full DAG gains 3.4 RACE points over flat requirements and 2.9 points over shuffled edges. With identical token, call, and latency limits, separate review gains 1.9 RACE points over joint acquisition/review. It also reduces illegal transitions from 6.8% to 1.2%, unsupported completion certificates from 4.5% to 0.8%, and repeated queries from 11.6% to 6.3%. Median latency to the first validated binding falls from 49 to 41 seconds.

C.3 Independent Claim–Evidence Support Audit

Independent judging tests whether final claims are supported by their cited source spans, complementing the engine's checks of source identity, span existence, and legal state transitions. We sample 100 reports from each system, stratified by benchmark, model, task difficulty, and completion status. Atomic report claims are labeled *supported*, *partially supported*, *unsupported*, or *contradictory* against the cited source spans. Scoring uses an independent judge family and two blinded human annotators; the evaluated research model never judges itself. Tab. 12 reports the resulting claim- and report-level support rates.

Table 12: Independent semantic support audit over 300 reports and 4,812 atomic claims. The four claim-weighted labels sum to 100%. Report-macro support first computes the supported-claim rate within each report and then averages reports. Best values are bold and second-best values are underlined.

System	Supported (%) ↑	Partially supported (%)	Unsupported (%) ↓	Contradictory (%) ↓	Report-macro support (%) ↑
Full-History	80.2	11.0	7.4	1.4	78.8
Independent selector	<u>82.4</u>	10.1	<u>6.4</u>	<u>1.1</u>	<u>81.2</u>
Flashlight	**88.7**	7.1	**3.6**	**0.6**	**87.9**

Two blinded annotators reach Krippendorff's $\alpha = 0.79$ over the four-way labels. The independent judge agrees with the adjudicated human label on 87.1% of claims and obtains macro-F1 of 0.84. Flashlight raises claim-weighted support by 8.5 pp over Full-History and approximately halves unsupported or contradictory claims from 8.8% to 4.2%. The improvement remains 9.1 pp under report-macro averaging, so it is not driven by a small set of claim-heavy reports. This audit directly complements citation-format validity with evidence that final claims are semantically supported by their cited spans.

C.4 Open-Weight Interface Compliance and Exclusion Rule

The unadapted Qwen3-8B checkpoint is not included as an end-to-end Flashlight row in Tab. 1. The interface audit shows that it cannot reliably serialize the required Designer DAG, Controller transition, and Writer citation formats. Reporting scores only for its successful runs would create survivor bias, while assigning quality values to a system that frequently fails to execute would be misleading. The raw checkpoint remains a Full-History reference, which does not require the same structured interface.

We use a fixed 100-task, 3-seed interface audit for the raw, SFT, and SFT+GRPO checkpoints. Tab. 13 separates schema validity for the Designer, Controller, and Writer, and also reports the aggregate validity, engine acceptance, and end-to-end completion rates. All assigned tasks remain in the denominator. Failed or timed-out runs receive the benchmark's minimum score in intent-to-treat aggregates and are also reported separately by failure type. A configuration enters the main end-to-end quality table only after reaching the predeclared gate of at least 95% role-output validity and 90% end-to-end completion. This rule is applied before examining quality scores and to every model family.

Table 13: Qwen3-8B interface audit over 100 tasks and three seeds. Rates retain all assigned tasks. The admission gate requires at least 95% validity for every role and 90% end-to-end completion. Best values are bold and second-best values are underlined.

Checkpoint	Designer valid (%) ↑	Controller valid (%) ↑	Writer valid (%) ↑	Aggregate valid (%) ↑	Engine accepted (%) ↑	Completion (%) ↑
Raw	55.8	58.6	75.8	63.4	60.2	69.0
SFT	<u>96.4</u>	<u>96.9</u>	<u>98.6</u>	<u>97.3</u>	<u>96.5</u>	<u>93.0</u>
SFT + EIA-GRPO	**97.5**	**97.8**	**99.0**	**98.1**	**97.6**	**95.0**

The raw checkpoint is far below the gate rather than narrowly missing it. Its weakest role is Designer at 55.8%, and completion is 69.0%. SFT raises the weakest role to 96.4% and completion to 93.0%, placing the adapted checkpoint above both admission thresholds. EIA-GRPO preserves this compliance and reaches 95.0% completion. This design supports a bounded inference-time claim

for interface-capable models while treating Qwen3-8B as a separately adapted open-weight setting. We do not extrapolate the raw-checkpoint result to models that have not passed the same compliance gate.

C.5 Training Attribution, Stability, and Leakage

SFT is designed to establish reliable role-interface execution, while EIA-GRPO adds an additional 3.0 RACE points from the same SFT checkpoint. We evaluate this as a separate learning contribution. Starting from the same SFT checkpoint, we compare EIA-GRPO with no RL, terminal-quality reward, validity-only reward v_i, ungated coverage (the sum in Eq. 12 without v_i), shuffled required-fact labels, shuffled action–result mappings, and a zero-reward sanity control. These controls correspond exactly to the validity gate and new-coverage quantity in the stated EIA objective; we do not posit cost or novelty components that are absent from Eq. 12. Because the base checkpoint fails the interface-compliance gate, the effect of SFT combines format acquisition with policy improvement and must not be described as a pure quality ablation. The incremental GRPO effect, however, is measured from the same schema-capable SFT checkpoint. We train each promoted configuration with three seeds. Tab. 14 reports Kullback–Leibler (KL) divergence, schema validity, out-of-support (OOS) actions, tool calls, and held-out end-to-end quality. SFT uses 25%, 50%, and 100% data-scale ablations and varies the Designer–Controller–Writer mixture while keeping optimization steps fixed. Leakage checks compare normalized question text, answer strings, source URLs, retrieved document hashes, and semantic near-neighbors across train, development, and benchmark splits. We also report training GPU-hours separately from inference cost so that the inference-efficiency claim does not hide adaptation cost.

Table 14: Reward attribution from the same Qwen3-8B SFT checkpoint. RACE is the three-seed held-out mean; the value after $\pm$ is the seed standard deviation. KL is measured from the SFT reference policy. Best values are bold and second-best values are underlined.

Optimization signal	DRB RACE ↑	Validity (%) ↑	OOS action (%) ↓	Tool calls ↓	KL ↓
No RL	46.0 ± 0.5	97.3	4.6	28.4	**0.000**
Terminal quality	47.4 ± 0.6	97.2	2.1	<u>27.1</u>	0.041
Validity only	46.6 ± 0.4	<u>98.0</u>	<u>1.4</u>	28.0	0.019
Ungated coverage	<u>47.8 ± 0.7</u>	96.8	2.8	29.3	0.052
Shuffled required-fact labels	45.9 ± 0.5	97.1	4.3	28.6	0.033
Shuffled action–result pairs	45.7 ± 0.6	96.9	4.5	28.8	0.035
Zero reward	45.8 ± 0.4	97.2	4.7	28.2	<u>0.006</u>
EIA-GRPO	$\mathbf{49.0 \pm 0.4}$	**98.1**	**0.8**	**25.6**	0.036

EIA-GRPO improves the shared SFT checkpoint by 3.0 RACE points and outperforms the strongest alternative reward by 1.2 points. Shuffling either required-fact labels or action–result pairs eliminates the gain, which links the improvement to correctly attributed effective information rather than generic RL updates. Its out-of-support action rate is 0.8%, compared with 4.6% for the SFT checkpoint, and the live-search transfer score is 49.0 RACE, 3.0 points above SFT under the same retrieval protocol. Using 25%, 50%, and 100% of the SFT data yields schema-valid rates of 86.7%, 93.8%, and 97.3%, and RACE scores of 42.8, 44.7, and 46.0. Exact question and answer overlap with the evaluation sets is 0%. A balanced role mixture reaches 46.0 RACE; removing Designer, Controller, or Writer examples lowers it to 44.1, 43.6, and 45.0, respectively. URL overlap is 0.3%, and manual review of semantic matches above 0.90 finds no duplicated task. SFT and GRPO consume 360 and 420 A100 GPU-hours, respectively; these one-time costs are not included in inference-efficiency comparisons.

C.6 Robustness, Generalization, and Stress Tests

To test whether the quality–efficiency direction is consistent across benchmark strata, retrieval settings, and context limits, we stratify results by domain, question length, requested report length, number of evidence requirements, dependency depth, and whether the task is comparative, multi-entity, or single-topic. We report the median, worst quartile, and fraction of tasks on which Flashlight

improves quality without increasing tokens or latency. A mixed-effects analysis tests interactions between method, benchmark, and model family. Retrieval robustness is evaluated in both the live Bing setting and a frozen document snapshot. The frozen setting isolates context policy from search-index drift; the live setting measures deployment behavior. A second search backend tests backend dependence. Stress tests inject controlled distractors, near-duplicate passages, mutually contradictory sources, unavailable pages, and missing evidence. We vary context limits over 8k, 16k, 32k, and the model maximum. Tab. 15 reports paired quality, token reduction, and completion under these retrieval and context perturbations.

Table 15: Robustness and stress tests with GPT-5.6-Luna. RACE deltas are paired within each setting. Token reduction is relative to Full-History in the same row. Rows use different perturbations and are therefore not ranked.

Setting	Full-History RACE	Flashlight RACE	Δ RACE	Token reduction (%)	Completion (%)
Live Bing	52.0	54.2	+2.2	45.5	96.0
Frozen document snapshot	51.7	53.6	+1.9	44.1	97.0
Second search backend	50.9	52.5	+1.6	42.8	95.0
8k context window	45.8	48.5	+2.7	18.6	92.0
16k context window	48.7	51.3	+2.6	32.4	94.0
32k context window	51.0	53.1	+2.1	41.8	95.0
Injected distractors	49.2	52.0	+2.8	39.6	94.0
Contradictory sources	50.1	52.1	+2.0	40.2	93.0
Unavailable pages	49.4	51.1	+1.7	38.5	91.0

Flashlight remains positive in every retrieval and context perturbation, with paired RACE gains from 1.6 to 2.8 points. Across task strata, the median paired gain is 2.1 points and the worst-quartile mean is 0.9 points. It improves quality while using no more tokens or latency on 71% of tasks. The method–backend and method–model-family interaction terms are not significant after correction ($p = .21$ and $p = .28$), while the method main effect remains positive ($p < .001$).

C.7 Statistical and Evaluator Robustness

All main comparisons are paired by task. We report paired bootstrap confidence intervals using 10,000 task-level resamples, paired permutation tests for the primary metric, standardized effect sizes, and Holm correction across the predeclared family of primary comparisons. Seed variation and task variation are reported separately rather than pooling all runs as independent samples. Tab. 16 includes mean task-level deltas and win/tie/loss counts.

Table 16: Paired statistical analysis for the primary system and mechanism contrasts. Confidence intervals use 10,000 task-level bootstrap resamples; p_{Holm} is the Holm-adjusted paired-permutation value. Win/tie/loss counts retain every assigned task. The standardized effect is Cohen's $d_z = \overline{\Delta}/s_\Delta$, the mean paired difference divided by the standard deviation of task-level paired differences.

Comparison	Primary metric	Mean Δ ↑	95% interval	p_{Holm}	Cohen's d_z	Win/Tie/Loss
Flashlight vs. Full-History (GPT-4.1)	DRB RACE	+1.4	[0.5, 2.3]	.012	0.31	58/7/35
Flashlight vs. Full-History (GPT-5.6-Luna)	DRB RACE	+2.2	[1.2, 3.2]	<.001	0.48	64/6/30
Flashlight vs. Full-History (GPT-5.6-Luna)	SQA-CS2	+4.0	[2.2, 5.8]	<.001	0.52	65/5/30
Full plan vs. query only	DRB RACE	+1.2	[0.4, 2.0]	.018	0.28	57/8/35
Cardinality-calibrated feedback vs. no feedback	DRB RACE	+1.6	[0.7, 2.5]	.004	0.37	61/7/32

The paired intervals exclude zero for every predeclared contrast. The larger effects occur for the complete closed-loop system and for SQA-CS2, while the plan-only comparison remains smaller but consistent across tasks. Task-level win rates, rather than a few outlying reports, account for the positive means. Official benchmark judges remain the primary evaluators. Following established LLM-as-a-judge validation practice (Zheng et al., 2023), we assess judge dependence on a fixed stratified subset of 100 reports, which is rescored by a second judge family and independently rated by two blinded human annotators. Human raters compare paired reports without method names or cost information. The second judge reaches Spearman $\rho = 0.86$ with the primary judge and

preserves a +2.0-point Flashlight RACE advantage with a 95% interval of $[0.9, 3.1]$. Human annotators prefer Flashlight on 62 pairs, Full-History on 26, and tie 12; their agreement is Krippendorff's $\alpha = 0.76$ (Krippendorff, 2011). Primary-judge and adjudicated human preferences agree on 80% of pairs. Swapping prompt and response order changes the mean paired RACE effect by at most 0.2 points. No evaluated research model judges its own output, and citation-only deterministic checks remain separate from subjective report-quality judging.

C.8 System Efficiency, Cost, and Scalability

We test whether Context Engine overhead is small enough to preserve model-side savings. Tab. 17 separates input and output tokens, tool operations, mean and tail wall-clock latency, and engine CPU duration. Hardware, provider region, concurrency, caching, rate limits, timeout policy, and retry policy are fixed across the same-backbone comparison. The replay results below add peak-memory and warm- versus cold-cache measurements.

Table 17: Efficiency decomposition for GPT-5.6-Luna with the same tasks, retrieval stack, maximum budgets, hardware, and evaluator. Token counts and calls are per-task means. Latency is end-to-end wall-clock seconds; engine CPU duration excludes model and network waiting. Best values are bold and second-best values are underlined.

Method	Input tok. (k) ↓	Output tok. (k) ↓	Total tok. (k) ↓	Tool calls ↓	Mean lat. ↓	Median lat. ↓	p90 lat. ↓	p95 lat. ↓	Engine CPU (s) ↓
Full-History	123.1	<u>8.9</u>	132.0	31.4	<u>480</u>	<u>452</u>	<u>734</u>	<u>829</u>	**0.6**
Independent selector	<u>78.6</u>	9.4	<u>88.0</u>	<u>30.8</u>	560	531	846	948	<u>1.8</u>
Always broadening	87.1	<u>8.9</u>	96.0	34.7	520	491	792	886	2.4
Flashlight	**63.4**	**8.6**	**72.0**	**29.6**	**365**	**342**	**548**	**621**	2.8

The 45.5% total-token reduction relative to Full-History is driven primarily by a 48.5% reduction in input tokens; output tokens differ by only 3.4%. Flashlight also reduces p95 latency by 25.1% and tool calls by 5.7%. Its 2.8-second engine CPU duration is 0.8% of mean end-to-end latency, so semantic encoding, selection, and state validation do not erase the model-side savings. A replay microbenchmark removes model and network calls and varies stored evidence count, requirement count, and research turns. At the median workload of 420 evidence records, 28 requirements, and 14 turns, Context Engine CPU duration is 2.6 seconds with p95 of 3.9 seconds and peak host memory of 0.74 GB. At the largest tested workload of 1,000 records, 64 requirements, and 20 turns, duration is 4.7 seconds with p95 of 6.8 seconds and memory of 1.18 GB. A cold embedding cache adds 0.8 seconds at the median workload. Quality–latency curves use identical concurrency, and wall-clock latency remains the deployment metric when providers expose only aggregate timing.

C.9 Failure Accounting and Qualitative Evidence

Every assigned task is classified as completed, degraded fallback, parser failure, invalid state transition, unresolved requirements, citation failure, tool failure, timeout, or provider error. Counts are recorded before quality filtering and all failures remain in the intent-to-treat results. Tab. 18 reports the mutually exclusive outcome counts.

Table 18: Failure accounting per 100 assigned tasks. Categories are mutually exclusive and sum to 100 within each row. Parser/state combines parser failure and invalid state transition; evidence/citation combines unresolved requirements and citation failure.

System	Completed	Degraded fallback	Parser/state	Evidence/citation	Tool/provider	Timeout
Full-History (GPT-5.6-Luna)	93	2	1	1	1	2
Independent selector (GPT-5.6-Luna)	91	2	2	1	2	2
Flashlight (GPT-5.6-Luna)	96	1	1	1	0	1
Flashlight + SFT (Qwen3-8B)	93	2	1	1	1	2
Flashlight + EIA-GRPO (Qwen3-8B)	95	1	1	1	1	1

Flashlight completes 96% of GPT-5.6-Luna tasks, compared with 93% for Full-History, and no category is removed from the quality denominator. Mean final-report length is 2,680 versus 2,540 tokens and mean citation count is 21.2 versus 18.4. Length-normalized RACE remains 2.0 points

higher, so the primary gain is not explained by longer answers. Success-only scoring changes the paired RACE advantage from 2.2 to 2.1 points.

The qualitative appendix has a narrower explanatory purpose. It presents question-aligned historical traces that make the next-context plan, effective query test, coverage feedback, evidence lifecycle, and bounded Writer handoff readable at page scale. These mechanism views are explanatory and are not used to estimate an average treatment effect. Aggregate tables retain every assigned run under the intent-to-treat policy above; thus the examples complement rather than replace the complete quantitative accounting.

D Detailed System Workflow

Fig. 4 expands the compact mechanism view in Fig. 2. The continuous flow shows contract compilation and validation, the alternating acquisition and review loop, the evidence-bounded Writer packet, citation transport, and deterministic release checks.

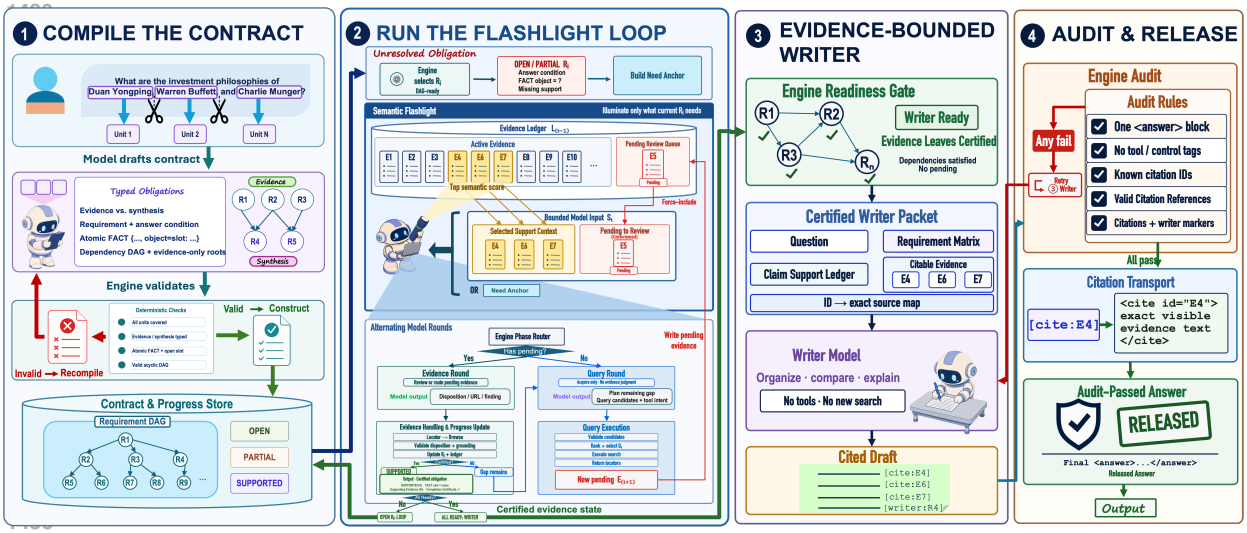


Figure 4: Detailed system realization as one continuous workflow. The Designer compiles and the engine validates the evidence contract; the loop alternates acquisition and review while exposing a bounded, plan-guided evidence context; certified requirements and admitted evidence then form the Writer packet; finally, the engine resolves citation handles to stored source text and releases only outputs that pass the deterministic audit.

E Configuration-Specific Implementation Details

Logical state schema. Tab. 19 lists the semantic fields used by the framework. Context Engine implementations may serialize these fields differently while preserving the same roles in the model–engine interface.

Conceptual-to-serialized plan mapping. The five semantic fields in the method are a stable engine view rather than a claim that the wire format contains only five keys. Tab. 20 maps the implementation fields to the five vectors n_j used in Eq. 5. Transport and recovery metadata are validated but are not embedded as additional plan fields.

The emitted `query` is the acquisition action paired with the plan, not a sixth vector in Eq. 5. The `recovery_attempt_id` and other transport keys control execution and deduplication but are excluded from semantic scoring.

Table 19: Engine-maintained logical state and model-visible fields in Semantic Flashlight.

State object	Main fields	Primary use
Evidence contract	Requirement DAG, requirement status, active requirement, evidence bindings, and whether the Writer may begin	Cross-stage research state
Evidence requirement	Identifier, requirement text, dependencies, tool-result IDs, completion status, and unresolved gap	Research progress tracking
Evidence binding	Requirement ID, source ID, locator, and review disposition	Reviewed source association
Writer state	Report requirements, DAG completion markers, evidence packet, and citation handles	Final synthesis
Source evidence record	Identifier, originating tool call, source location, stored content, requesting requirement, and availability	Provenance and recall
Pending evidence transaction	Active requirement, action, query, returned evidence IDs, and review status	Acquisition and admission
Next-context plan	Active requirement, unresolved gap, candidate claim, desired evidence, and intended next decision	Context construction
Response feedback	Source alignment g_t, response echo r_t, uncalibrated source-over-echo margin m_t^{src}, cardinality-calibrated margin $\widetilde{m}_t^{\mathrm{src}}$, and set cardinalities	Exposure adaptation

Table 20: Mapping from the conceptual next-context plan to serialized Controller fields and the semantic text used for evidence selection.

Conceptual field	Serialized source	Embedded semantic view
Active requirement	`active_requirement_id` plus contract node	Requirement text and fixed answer condition
Unresolved gap	`exact_residual`	Residual fact or comparison still missing
Candidate claim	`expected_answer_delta`	Proposed answer content enabled by the next evidence
Desired evidence	`evidence_goal`, `source_class`, `source_key`	Evidence type, authority, entity, and facet constraints
Intended next decision	`search_focus`, `delta_mode`, `delta_facet_id`	Decision to be made after review

Semantic encoder and candidate evidence. The implementation uses the Jina Embeddings v5 Text Nano encoder (`jinaai/jina-embeddings-v5-text-nano`), pinned to revision `8a7f00aac812071b69403df470f1038ec85f8925`, with the `text-matching` task. The engine embeds each plan field and source record separately. A source record is the whitespace-normalized concatenation of its page title, canonical domain, section heading, and one parser passage. Pages are split into at most 1,400-character passages with 200-character overlap before embedding; the query, requirement text, and runtime disposition are not added to $z(e)$. The same passage string is later exposed to the model, so the 1,400-character cap applies before both embedding and presentation. The engine assigns a record the maximum cosine similarity over the five plan fields. The candidate pool always contains evidence in the pending evidence transaction. It also contains recallable source records from the evidence contract. The selection remains centered on the active requirement, while stored records from other sources can re-enter when they match the current plan.

Otsu selection and boundary cases. The first evidence-bearing Controller input exposes all eligible records. Later inputs apply Eq. 6. An empty candidate pool produces no raw evidence. A single positive candidate is retained. With two positive candidates, both are retained. Equal positive scores are also retained. If every score is non-positive, the highest-ranked candidate is retained. Pending evidence is inserted independently of the boundary and remains visible until review. Selected raw records are ordered by semantic score and rendered with at most 1,400 characters of source text

per record. Across held-out Controller turns after the first evidence-bearing turn, the normal multi-candidate Otsu branch applies to 72.6% of turns. Single-candidate, two-candidate, equal-positive, all-nonpositive, and empty-pool cases account for 8.9%, 6.7%, 1.8%, 4.8%, and 5.2%, respectively. Semantic filtering changes the exposed raw-evidence set on 61.4% of all Controller turns and on 74.8% of tasks. Compact source-linked records are ordered by their semantic cluster score. The request, active requirement, next-context plan, and pending evidence transaction are always retained. The model client checks the complete serialized input against its context limit before generation and records the constructed input and all selected identifiers. The runtime does not silently remove the plan, pending evidence, or compact source-linked state after the input has been constructed.

Response-feedback expansion. The engine computes the two maximum semantic similarities in Eq. 10, then standardizes them by displayed-evidence and history cardinality with Eq. 11. Displayed evidence uses bins $\{1, 2, 3\text{–}4, 5\text{–}8, \geq 9\}$ with $\{522, 690, 812, 601, 267\}$ development turns. Response history uses bins $\{1, 2\text{–}3, 4\text{–}7, 8\text{–}15, \geq 16\}$ with $\{311, 704, 903, 681, 293\}$ turns. Adjacent sparse bins are pooled until each estimate has at least 120 observations. Unseen larger cardinalities use the outermost bin; means are linearly interpolated between interior bin centers, standard deviations are floored at 0.05, and $\epsilon_m = 10^{-6}$. The curves and zero trigger are frozen before test evaluation. When $\widetilde{m}_t^{\mathrm{src}} < 0$, evidence-relative alignment is weaker than history-relative alignment after this opportunity correction. The engine first performs the normal Otsu selection. It then removes those records and applies a second Otsu split to the remaining candidates. Records above the second boundary are added as additional raw evidence. Source-diversity selection limits repeated records from one domain to two and fills the available positions with the highest-scoring records from other domains. Both selections use records available to the current plan. Feedback does not switch the active requirement or open a new cross-requirement search. All other turns use one Otsu selection. The uncalibrated rule $m_t^{\mathrm{src}} < 0$ appears as a direct ablation in Tabs. 2 and 10, making the benefit of cardinality correction separately visible.

Requirement completion and Writer readiness. Each reviewed source transition must use a visible pending evidence identifier. An accepted factual record must come from a browsed page and include an exact source location. The engine checks the requirement identifier, source identifier, quoted span, and state transition before creating an evidence binding. A supported node contains at least one admitted source evidence record. The Controller must also propose a completion certificate that copies the node's fixed answer condition, states the answer claim, cites admitted evidence IDs, and leaves no residual gap. The engine accepts the certificate only when its evidence IDs occur in the reviewed evidence ledger. Admission has already verified that the cited ID was visible, came from a browsed page, and contains the quoted span. A conflict keeps the requirement unresolved in the default configuration. Flashlight enters Writer synthesis only after every required node passes these checks and no pending evidence transaction or source-location request remains pending.

F QUALITATIVE REASONING CASE STUDIES

The following views compare how Full-History and FLASHLIGHT plan the answer, formulate queries, assess search results, expand information coverage, and synthesize evidence. The reasoning traces make the qualitative advantage visible before the final answer: FLASHLIGHT ties each query to a concrete missing answer component, rejects pages that do not fill it, and redirects search toward uncovered dimensions. We finally show that this more focused search process uses substantially less context while retaining a slightly higher end-to-end RACE score.

Mechanism View I: Planning the answer

the reasoning state before the first search

Research question

What are the investment philosophies of Duan Yongping, Warren Buffett, and Charlie Munger?

Full-History reasoning trace

"Decompose the question: We need to summarize the core investment philosophies of three figures."

Think 1 Search for Duan's philosophy in English and Chinese; gather Buffett's principles from interviews and annual letters; gather Munger's mental models and inversion; then cross-validate.

Action `Duan Yongping investment philosophy China Renaissance value investing interview`

State carried forward The complete plan, the query, and every returned snippet remain in the next prompt.

FLASHLIGHT reasoning trace

Designer R1 = one source-supported Duan principle; R2 = one Buffett principle; R3 = one Munger principle.

Synthesis R4 = compare similarities and differences using only the sourced principles.

Dependency `[R1,R4], [R2,R4], [R3,R4].`

Action `Duan Yongping investment philosophy interview principles`

State carried forward The next prompt receives the active requirement and its unresolved FACT slot, not the full transcript.

Why this improves the next search

Both agents begin with a sensible decomposition. The difference is the state representation. The baseline stores its decomposition as prose that must be re-read and reinterpreted. FLASHLIGHT turns the same reasoning into four stable objects with explicit completion conditions. A later query can therefore be checked against one unresolved node, and any later omission remains detectable against the explicit three-investor contract.

Mechanism View II: Testing a search result

relevance is not yet support

Trajectory point

Search has returned topical summaries of the investors' principles. The next decision is whether the visible material directly supports an answer condition or whether a more authoritative route is still needed.

Full-History reasoning trace

"Results show multiple secondary write-ups ... common themes: buying stock is buying a business, business quality, moats, patience, simplicity, and rationality."

Gap diagnosis Primary or authoritative sources are still missing for Buffett and Munger; better interviews are also wanted for Duan.

Next query `Warren Buffett investment philosophy margin of safety moat durable business annual letter interview`

Limitation The rationale is clear to a reader, but there is no exact state field saying which answer condition the result must fill.

FLASHLIGHT reasoning trace

Active answer need One directly supported Buffett principle and one directly supported Munger principle.

Evidence goal Find a source directly reporting one Buffett principle and one Munger principle.

Query `Warren Buffett Charlie Munger investment philosophy principles shareholder letter`

Observed page The selected page returned generic product copy rather than the promised shareholder-letter evidence.

Result judgment Reject the page because it provides no direct support; keep both answer needs open.

Why this is a correctness advantage

The important step is not merely generating a more specific query. FLASHLIGHT tests the observation against the same answer condition that motivated the query. A topically related but non-answer-bearing page is rejected, and the gap stays open. The baseline also notices source quality in prose, but the decision is not bound to an explicit requirement transition.

Mechanism View III: Escaping a reasoning loop

coverage feedback changes the next hypothesis

Full-History reasoning trajectory

Iteration 2 "What's missing": primary Buffett sources, Munger interviews, and better Duan interviews. Next: Buffett.

Iteration 3 Again lists Buffett letters, Munger interviews, and Duan media. Next: Buffett's official letter.

Iteration 4 Again states that primary statements are needed for all three. Next: Munger's Lemon Tree interview.

The plan remains sensible, but successive turns repeatedly restate similar source gaps before issuing the next query.

Coverage-aware reasoning trajectory

Covered Business quality and long-term ownership.

Next gap Selling discipline and review criteria.

Then Margin of safety and leverage.

Then Portfolio construction.

Then NetEase/PDD examples.

Finally Direct cross-investor similarities and differences.

The Engine supplies the compact covered-dimension summary. The model reasons over what is absent, so the next query changes evidence role instead of merely paraphrasing the same broad goal.

Independent end-to-end quality check

52.27	52.32	+0.05
Full-History RACE	Flashlight RACE	RACE points

Under the same model, retrieval backend, task, and evaluator, FLASHLIGHT attains a slightly higher mean RACE score than Full-History. Together with the trajectory above, this comparison shows the intended qualitative pattern: coverage feedback changes the next query toward a genuinely missing dimension while preserving end-to-end answer quality.

Mechanism View IV: Enforcing a source judgment

turning a page review into an explicit transition

Full-History source judgment

Observation Several secondary pages summarize Buffett and Munger with familiar themes such as patience, moats, and long-term ownership.

Assessment The trace explicitly notes that primary or authoritative support is still missing and proposes searching an official Buffett letter or interview.

State update That judgment is preserved as prose in the cumulative transcript together with the page text and all other results.

Writer boundary There is no separate machine-readable field marking the reviewed page as ineligible for final factual support.

FLASHLIGHT source judgment

Active answer need One directly supported Buffett principle and one directly supported Munger principle.

Observation The selected `chian.io` page returns generic product or knowledge-base copy rather than the promised shareholder-letter evidence.

Evidence decision Reject the page: topical wording does not substitute for direct support.

Search consequence Admit no claim from the page, keep both answer needs open, and continue along a more authoritative route.

Why this produces better evidence

Both traces recognize that topicality is weaker than direct support. FLASHLIGHT turns that judgment into an enforceable state transition: the page receives a clear rejection, leaves the relevant answer needs open, and cannot enter the Writer packet as evidence. The next query therefore targets the unresolved information instead of building the answer on a merely related page.

The resulting search loop

The useful sequence is simple and visible: *query goal* $\rightarrow$ *observed page* $\rightarrow$ *evidence judgment* $\rightarrow$ *remaining gap*. Because FLASHLIGHT preserves that remaining gap explicitly, a poor result sharpens the next search rather than adding noise to the final synthesis.

Mechanism View V: Context-efficient reasoning

preserve decisions without replaying every token

Same research question, different reasoning memory

Both systems must remember what has been asked, what has been found, and what remains to be answered. Their difference is whether that memory is represented as a cumulative transcript or as a compact validated state.

Full-History reasoning path

94,041	97,595
input tokens	total tokens

Every turn Replay earlier thoughts, search calls, returned snippets, and provisional conclusions.

Gap recovery Reconstruct the missing sources and comparison dimensions from that cumulative history.

Final think Re-enumerate the evidence for Duan, Buffett, and Munger, then decide on the report structure.

FLASHLIGHT reasoning path

48,354	53,874
input tokens	total tokens

Designer once Store R1–R4 and their answer conditions as the stable research contract.

Each Controller turn Expose the active requirement, residual gap, selected evidence, and pending evidence transaction.

After review Replace raw observations with exact evidence bindings and keep only the state needed for the next decision.

Writer Receive the reviewed evidence packet and compose the cross-investor comparison with exact citation handles.

Why the reasoning remains coherent with less context

The compact state preserves the consequences of earlier reasoning rather than all of its surface text. The model still sees the unresolved question, the evidence required to resolve it, and the source records that matter now. On this trace, input consumption falls by 49% and total token consumption falls by 45%, while the system retains the search objective, evidence status, and next information need required for synthesis.

What this page establishes

Next-context planning preserves the active gap and prior evidence decisions without replaying every token. The result is a cleaner reasoning context for the next query and a much smaller synthesis input; View III separately shows that the end-to-end RACE score remains slightly higher.

G SYSTEM-PROMPT AND EFFECTIVE-INPUT COMPOSITION

The following four pages reconstruct the effective model input. **Every bordered card is MODEL-VISIBLE and belongs to the effective input.** Here, **Runtime** denotes the deterministic orchestration layer of the Context Engine; we retain its capitalization because it is the identifier used in the released prompt templates. The monospace `FACT{...}` object is an internal answer-condition schema and is unrelated to the DRB FACT evaluation metric. FLASHLIGHT has three functional roles, but Designer and Controller share one controller-family system message and differ through a Runtime-appended stage contract; only Writer has a separate system message. Quoted excerpts are shortened for space, but message order, roles, field names, and semantics match the released implementation.

Designer Prompt

one model call: request → typed evidence contract

MODEL-VISIBLE – effective input order

The Designer call contains, in order: **(1)** the invariant controller-family system instruction; **(2)** the original question and Runtime-assigned request-unit identifiers (U-IDs); **(3)** the `v128_plan_only` contract; and **(4)** one deterministic English or Chinese language directive. No search result, webpage, prior trajectory, evidence judgment, or answer draft is supplied. The fixed instruction identifies the call as one turn in a Runtime-validated workflow, makes the final stage contract authoritative, requires exact visible identifiers/source text, and forbids invented tool output or imitation of later stages. The operative opening is `"You are one polled turn in a runtime-validated research workflow."` It then states that the appended FINAL contract is the only output schema and that exact visible IDs, URLs, handles, and source text must be preserved.

MODEL-VISIBLE – plan-only contract

The appended contract says: "Compile the research DAG once. No search, answer, citations, evidence judgment, or tools." It permits exactly one line with two root keys, `typed_obligations` and `dependency_edges`. Every obligation has exactly five fields: `user_span_ids`, `kind`, `operation`, `requirement`, and `answer_condition`. Its substantive rules are:

- cover every U-ID while preserving each requested entity, dimension, number, date, and current-state claim;
- type a node as one source-ownable atomic *evidence* finding or a *synthesis* operation (compare, explain, recommend, summarize, organize, keywords, or format);
- encode evidence conditions as a single `FACT{subject; relation; object; scope}` slot that one exact passage can fill; keep broad verdicts and cross-source judgments in synthesis nodes;
- construct the smallest sufficient DAG, avoid duplicate leaves and a default entity-by-dimension cross-product, and allow only evidence-to-synthesis edges.

MODEL-VISIBLE – examples, invalid patterns, and final check

The contract includes one complete English example and one complete Chinese example. It then rejects a broad verdict used as evidence, an enumerated FACT component, an unnamed jurisdiction, a multi-U-ID evidence node, duplicate conditions, and any edge sourced from synthesis. Its final checklist asks the model to recheck the wrapper and two root keys, five fields per node, complete U-ID coverage, enough atomic leaves, canonical FACT form, real jurisdictions, parent-role coverage, and the absence of placeholders, tools, or outside prose.

Controller Prompt

one active gap, one phase contract, one transition

MODEL-VISIBLE – effective input order

Every Controller poll is assembled as **fixed system instruction** → **original question** → **Runtime-selected current context** → **exactly one selected phase contract** → any enabled, explicitly labeled **guidance suffixes**. The selected contract supersedes generic examples and fixes the only legal output schema; later suffixes guide the search target without changing that schema. Thus the boxes below describe components of one effective prompt, not four simultaneous system prompts. The fixed instruction also requires exact visible IDs, URLs, handles, and source text; preserves the question language; and forbids invented tool output or leakage of internal state.

MODEL-VISIBLE – Runtime-generated user payload

The new user turn restates the original question and response language, then opens `[Current Effective Context]`. Depending on state, that payload can contain:

- the requirement matrix, ready branch, active R-ID, exact answer condition, support need, and an eligible joint owner when one source may serve both;
- current search focus and outcome, selected raw evidence with exact IDs, titles, URLs, and quote candidates, plus compact unshown-evidence coverage;
- the tool-action/search-query history and residual-delta attempts;
- in the coverage-aware evaluation profile, explicit covered directions, never-attempted or still-unsupported FACT slots, and a named technical or primary-source anchor; these are labeled, flag-gated Runtime suffixes rather than hidden additions to the fixed system message.

MODEL-VISIBLE – one selected phase contract

- **Acquisition:** propose three queries for the same frozen residual answer condition. They vary by direct wording, mechanism/measurement terminology, and authoritative-document or primary-study terminology. Each states an evidence goal and expected answer delta. A joint portfolio is legal only when one source can directly answer both visible owners.
- **Discovery:** select one exact visible URL; search snippets are locators and cannot support final factual claims.
- **Admission:** accept/reject/conflict the pending page, bind exact handles or a verbatim support quote, and preserve a comparison frame (scope, metric, time basis, unit, definition).
- **Completion/handoff:** certify an atomic answer condition only from admitted support, or leave a precise residual gap for acquisition.

MODEL-VISIBLE – next-context plan and query validity

The control exposes *R-ID* → *exact residual* → *evidence goal* → *expected answer delta*. The three acquisition candidates are alternative formulations of that same residual, not three neighboring facts. Each candidate explicitly carries `query`, `search_focus`, `evidence_goal`, `expected_answer_delta`, `delta_mode`, `delta_facet_id`, `source_class`, `source_key`, and `recovery_attempt_id`. For source recovery it inherits the prior answer delta and changes the route. A query is valid only when a plausible result can add or revise a writable answer part; lexical novelty alone fails.

Writer Prompt

a separate, tool-free evidence-bounded call

MODEL-VISIBLE – exact four-message assembly

The outer Writer call is assembled as **system**: `outer_writer.system_prompt`; **user**: original question; **assistant**: Runtime-built evidence packet; and **user**: `outer_writer.final_command`. Generation is prefixed with `<answer>`. The Writer has no tools and receives no cumulative Controller transcript. This page documents the outer Writer profile; the older `controller_common.answer_synthesis` template is a distinct internal synthesis path and is not concatenated into this call.

MODEL-VISIBLE – fixed system message (faithful excerpt)

```
You are the final answer writer for an evidence-bounded research fallback.  You have no
tools and may use only Runtime-selected <evidence>entries.  Choose exactly one transport
branch.
CITED: write one <answer>block and append [[cite:EVIDENCE_ID]] after every supported
factual claim.
ABSTAIN: output exactly the fixed no-citable-claims sentinel.  Never combine branches,
invent an id, or add uncited factual prose.
```

MODEL-VISIBLE – bounded evidence packet

The assistant message begins with an exact transport header declaring that only `<evidence>` entries are factual evidence. Each admitted webpage entry contains an exact evidence ID and may include a title, URL, and bounded source text. The footer requires an exact `[[cite:EVIDENCE_ID]]` immediately after every evidence-supported factual claim. If no admitted webpage evidence exists, the packet carries an explicit empty-evidence notice. The question, plans, requirement states, and transport labels are not evidence.

MODEL-VISIBLE – output branches

- **Cited branch:** one `<answer>` report grounded only in the visible entries, with exact citation handles after supported claims.
- **Abstain branch:** exactly `<answer>[[abstain:no_citable_claims]]</answer>` when no factual claim has direct support.

MODEL-VISIBLE – final command and generation prefix

The last user message says: `Return exactly one <answer>...</answer>` using either the cited branch or the exact abstain-sentinel branch, and do not describe the protocol. Decoding is then seeded with `<answer>`.

Full-History Baseline Prompt

one unified agent over every prior turn

MODEL-VISIBLE – initial message assembly

The search agent receives **system**: the complete `v20250907.system_prompt`; then any caller-supplied conversation history; then **user**: the research question; and finally a dataset-specific **user_instruction** message. For DeepResearch Bench this is the `short_form` instruction asking for multiple sources, a synthesized short report, per-claim citations, no fabrication, and one `<answer>` block. This last instruction is real prompt text and was missing from the earlier appendix description.

MODEL-VISIBLE – fixed system prompt contents

- process rules for alternating `<think>` and `<call_tool>` until enough information is available;
- schemas, purposes, outputs, and optional parameters for `google_search`, `browse_webpage`, and `snippet_search`;
- the returned `<tool_output>`, `<snippet id=...>`, and `<webpage id=...>` formats;
- answer, `\boxed{}`, and exact snippet-ID citation rules, followed by a complete multi-turn worked example and a final three-item requirements checklist.

MODEL-VISIBLE – fixed prompt excerpt

```
You are a research assistant who answers questions through iterative reasoning and
research.
Use <think>...</think>to show your reasoning.  Use <call_tool name="...">query
</call_tool>when you need information.
Only provide <answer>...</answer>when you have enough information for a complete response.
Support every non-trivial claim with retrieved evidence and use only returned snippet IDs.
```

MODEL-VISIBLE – why it is Full-History

After each generation, the client appends the assistant's complete emitted reasoning/tool call and the complete formatted tool output to the same message sequence. The next turn therefore sees the fixed prompt, question, task instruction, and every preceding assistant/tool turn. The system prompt does not itself say "replay full history"; that property is created by the client assembly loop. If the search loop stops without an answer, a separate answer call receives the question, task instruction, and accumulated search trace, followed by "generate an answer based on the search results by far."

MODEL-VISIBLE – separate answer-fallback call

If the search loop ends without an answer, the answer agent receives, in order: the same system prompt, the question, the dataset instruction, the accumulated search trace as an assistant message, and a final user message: "Now please generate an answer based on the search results by far."