DualResearch: Entropy-Gated Dual-Graph Retrieval for Answer Reconstruction in Deep Research Agents

Anonymous ACL submission

Abstract

Deep Research Agents (DR-Agents) extend LLMs with multi-step retrieval, tool use, and intermediate computation to solve complex problems. However, their execution logs often contain redundant context, noisy evidence, and brittle dependencies among intermediate decisions. As a result, final answers can be sensitive to irrelevant or weakly verified information. We propose DualResearch, a log-based post-processing framework for answer reconstruction. Given a query and a completed DR-Agent log, DualResearch normalizes the log into event records and constructs two evidence graphs: a breadth semantic graph for declarative evidence and a depth causal graph for procedural provenance. It then retrieves relevant breadth nodes and depth paths, applies entropy-gated fusion to select complementary declarative and procedural evidence, and reconstructs the answer from this structured evidence. Experiments on HLE, GAIA, and GPQA show that DualResearch improves over the underlying DR-Agents and log-reuse baselines, indicating that structured reuse of execution logs can produce more accurate and traceable answers.

1 Introduction

Large language models (LLMs) have shown strong capabilities in complex problem solving (Achiam et al., 2023; Zhang et al., 2025), yet they still struggle to organize evidence reliably across long reasoning contexts (Li et al., 2024; Huang and Chang, 2023; Ahmed and Devanbu, 2023). When a task requires multiple pieces of evidence, intermediate computations, or cross-step consistency, a single generation pass may mix relevant and irrelevant context, miss verified intermediate results, or produce answers that are only weakly grounded in the available evidence (Shojaee et al., 2025). These limitations create a need for systems that can actively gather evidence, invoke tools, and maintain intermediate reasoning states over extended problem-solving processes.

This need has led to a set of approaches commonly referred to as Deep Research Agents (DR-Agents) (Jones, 2025; Hu et al., 2025b,a; Team et al., 2025). DR-Agents extend LLMs with retrieval, tool use, and multi-step execution (Schick et al., 2023), allowing them to gather external evidence, run tools, and update intermediate conclusions over a longer reasoning process. Figure 1 shows a representative case: when answering the question *"Which of these Turing Machines halts after the most number of steps and what is the number of steps?"*, a DR-Agent may retrieve transition tables, generate simulation code, execute simulations, and compare the resulting halting steps.

However, as DR-Agents proceed through multi-step retrieval, tool use, and intermediate reasoning, their execution processes often collect redundant context, conflicting intermediate results, loosely grounded evidence spans, and brittle dependencies between tool outputs and later conclusions (Liu et al., 2024a,b). These artifacts can make the final answer sensitive to irrelevant context or weakly verified evidence (Huang et al., 2025). At the same time, the execution process produces reusable traces, retrieved facts, tool outputs, intermediate artifacts, and validation signals. Completed logs record these traces and therefore provide the input for post-hoc answer reconstruction.

A natural baseline is to reuse the completed log as flat text or retrieved chunks (Lewis et al., 2020). However, this representation obscures the distinction between declarative facts and procedural provenance, and it makes it difficult to track which intermediate results were produced, consumed, and verified during execution (Trivedi et al., 2023). As a result, answer reconstruction can remain sensitive to noisy log segments and weakly supported reasoning paths.

We propose **DualResearch**, a plug-and-play post-processing framework for DR-Agents. Given

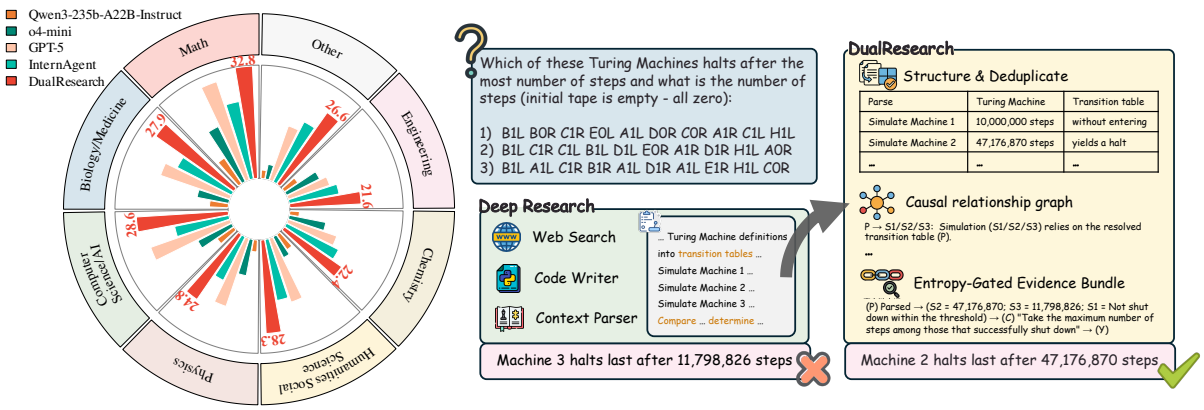

Figure 1: Overview and motivating case for DualResearch. Left: performance comparison on the HLE benchmark, where DualResearch outperforms strong baselines across diverse domains. Right: a case on the "Turing Machine halting steps" problem, where a DR-Agent reaches an incorrect conclusion because of noisy retrieval and missing execution constraints, while DualResearch retrieves breadth semantic nodes and depth causal paths, applies entropy-gated pair fusion, and reconstructs the correct answer from the selected evidence bundle.

a query and a completed DR-Agent log, DualResearch reconstructs the final answer by constructing two evidence graphs from the log. The breadth semantic graph stores declarative evidence as entities, definitions, fact spans, and formulas, while the depth causal graph stores procedural provenance as actions, artifacts, and validators connected by execution dependencies. DualResearch retrieves breadth nodes and depth paths, applies entropy-gated pair fusion to select a semantically aligned node–path evidence pair, and converts the selected pair into a compact evidence bundle for answer reconstruction. This design enables DualResearch to reuse completed DR-Agent logs through structured evidence selection while keeping both declarative support and procedural provenance available for answer reconstruction.

In summary, our contributions are as follows:

1. We propose DualResearch, a plug-and-play post-processing framework for DR-Agents that converts completed execution logs into structured evidence for answer reconstruction.

2. We introduce a dual-graph retrieval and entropy-gated fusion mechanism that combines breadth semantic nodes for declarative evidence with depth causal paths for procedural provenance.

3. Experiments on HLE, GAIA, and GPQA show that DualResearch improves answer accuracy over the underlying DR-Agent and controlled log-reuse baselines while maintaining traceable evidence support.

2 Related Work

Deep Research Agents (DR-Agents) extend search-oriented and tool-augmented language models to long-horizon problem solving. Recent work studies DR-Agents as a general paradigm in which LLM agents decompose tasks, retrieve external evidence, invoke tools, and consolidate intermediate results over multiple steps (Huang et al., 2025). Systems such as OpenAI Deep Research (OpenAI, 2025a) and Gemini Deep Research (DeepMind, 2024) combine retrieval with reasoning to generate evidence-grounded reports from heterogeneous sources. Multi-agent systems such as OWL (Hu et al., 2025a) organize agents into collaborative workflows, while InternAgent (Team et al., 2025) builds closed-loop workflows for iterative hypothesis generation and experimentation. WebThinker (Li et al., 2025) emphasizes dynamic web-based reasoning, and SFR-DR (Nguyen et al., 2025) studies reinforcement learning for autonomous single-agent research. These systems improve search, tool use, and long-horizon reasoning in LLM agents.

However, completed execution logs from these systems are often long, heterogeneous, and noisy. Existing DR-Agents mainly focus on generating or executing reasoning trajectories, whereas DualResearch focuses on reusing the completed trajectory after execution. It is therefore a post-processing framework for DR-Agents rather than a replacement agent.

Retrieval-Augmented Generation (RAG) grounds LLM outputs by retrieving external evidence from a corpus (Ram et al., 2023; Fan

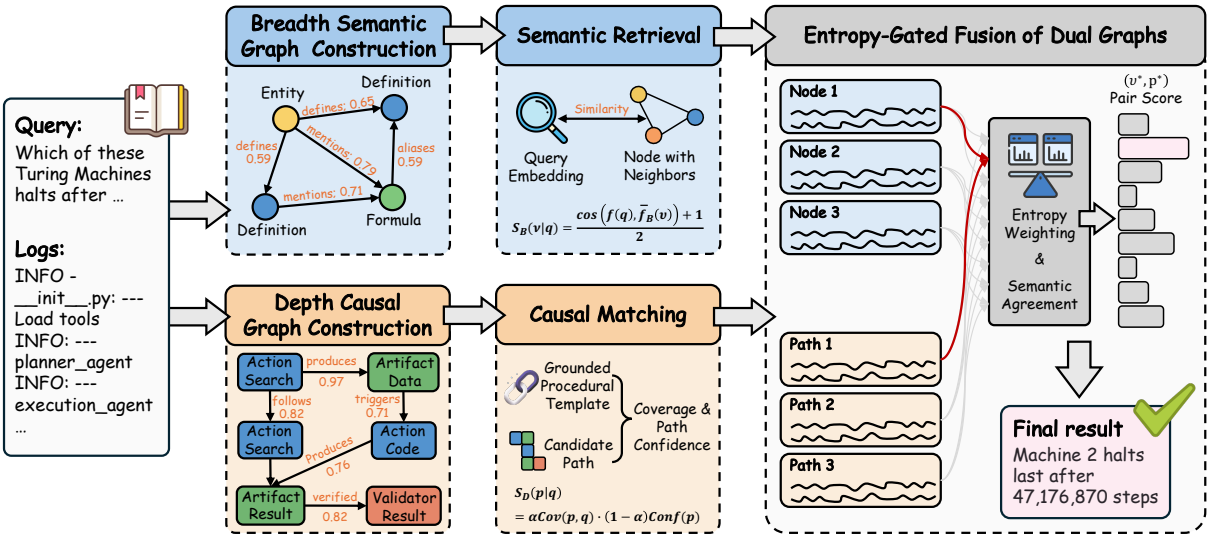

Figure 2: Overview of DualResearch. From a query and completed execution logs, DualResearch builds a breadth semantic graph for declarative evidence and a depth causal graph for procedural provenance. The breadth layer retrieves nodes via neighborhood-smoothed semantic similarity S_B, while the depth layer retrieves causal paths using procedural coverage and path confidence S_D. Entropy-gated pair fusion with semantic agreement selects (v^*, p^*), which is converted into an evidence bundle for answer reconstruction.

et al., 2024). Standard RAG retrieves top-ranked text chunks from a vector index. This works well for short-hop fact lookup, but it can fragment evidence and ignore the structure of the reasoning process (Gao et al., 2022, 2023; Chan et al., 2024; Yu et al., 2024). Graph-based and dual-pathway RAG methods further organize entities, claims, and textual evidence through structured retrieval, which improves evidence connectivity and global context modeling (Edge et al., 2024).

These methods are primarily designed for static corpora or knowledge bases. DR-Agent logs differ because they include retrieved text, tool invocations, intermediate artifacts, computed results, and validation signals. Recent dual-pathway and graph-based RAG methods combine structured relations with unstructured textual evidence, but both channels usually describe factual content from a corpus (Guo et al., 2024; Gao et al., 2025; Xu et al., 2024). By contrast, DualResearch operates on execution logs, where part of the evidence describes what was retrieved and another part describes how intermediate results were produced, checked, and used. In this sense, DualResearch is closer to completed-log reconstruction than to conventional corpus-level retrieval.

3 Method

As shown in Figure 2, given a query q and the completed execution log L of a DR-Agent, DualResearch first normalizes the log into event records and then constructs a breadth semantic graph for declarative evidence and a depth causal graph for procedural provenance. It retrieves breadth nodes and depth paths, fuses them into a node–path evidence pair, and reconstructs the final answer from the selected evidence bundle.

3.1 Graph Construction Pipeline

DR-Agents often expose execution logs in different formats, so we first map available log spans into normalized event records. Each record keeps the minimal fields needed for graph construction, including the role, source span, tool metadata, produced outputs, execution status, variables, and validator information when available. The parser only structures information already present in the log; it does not perform new external retrieval or inject facts outside the log. Full parsing rules, prompts, and schema fields are given in Appendix A.

From the normalized events, a prompted extraction module constructs two typed graphs:

$$G_B = (V_B, E_B, R_B), \quad G_D = (V_D, E_D, R_D), \tag{1}$$

where V_B, V_D are node sets, E_B, E_D are typed edge sets, and R_B, R_D store the extractor-assigned confidence scores for breadth and depth edges, respectively. The extractor follows a fixed schema and outputs typed nodes, typed edges, evidence spans, and edge confidence scores. Each node is stored as a structured record with fields such as node ID, type, label, and evidence span.

The breadth graph G_B stores declarative evidence. Its nodes are ENTITY, DEFINITION, FACTSPAN, and FORMULA; its edges include semantic and evidential links such as `mentions`, `defines`, `aliases`, `supports`, and `part_of`. The depth graph G_D stores procedural provenance. Its nodes are ACTION, ARTIFACT, and VALIDATOR; its directed edges encode execution relations such as `triggers`, `produces`, `consumes`, `verifies`, and `follows`.

Each extracted edge is assigned a confidence score in $(0, 1]$, stored as an edge attribute and used by the retrieval stages below. We denote the confidence of an edge e by $r_B(e)$ if $e \in E_B$, and by $r_D(e)$ if $e \in E_D$. When a breadth edge explicitly connects two nodes v and u, we use $r_B(v, u)$ as shorthand for the confidence of that edge.

This separation determines the retrieval targets used below. The breadth channel retrieves nodes because G_B organizes declarative units such as entities, definitions, facts, and formulas, while the depth channel retrieves paths because G_D organizes execution dependencies among actions, artifacts, and validators.

3.2 Retrieval on Breadth Semantic Graph

For the breadth semantic graph G_B, DualResearch ranks nodes by query-conditioned relevance. Retrieval is topology-aware: each node representation is smoothed with reliable supportive one-hop neighbors before it is matched to the query. Let $\mathcal{N}_B(v)$ denote the supportive one-hop neighbors of v, and let $r_B(v, u)$ be the confidence score of the semantic edge between v and u.

The smoothed node representation is

$$\bar{f}_B(v) = \frac{f(v) + \sum_{u \in \mathcal{N}_B(v)} r_B(v, u) f(u)}{1 + \sum_{u \in \mathcal{N}_B(v)} r_B(v, u)}. \quad (2)$$

The breadth relevance score is

$$S_B(v \mid q) = \frac{\cos\left(f(q), \bar{f}_B(v)\right) + 1}{2}. \quad (3)$$

We rank all breadth nodes by $S_B(v \mid q)$ and retain the top-K_B nodes as $\mathcal{V}_q$.

3.3 Retrieval on Depth Causal Graph

For the depth causal graph G_D, DualResearch retrieves ordered causal paths rather than isolated nodes. To obtain candidate paths, we start from ARTIFACT nodes because they usually correspond to intermediate results or candidate conclusions in the original logging process. Therefore, all artifact nodes can serve as path anchors V_D^{Art}.

Starting from each artifact node, we trace backward along directed dependency edges in G_D. Tracing stops when the current node has no parent or reaches another artifact node. When a node has multiple parents, the traversal branches along each dependency. This artifact-bounded tracing yields an initial pool of candidate causal segments:

$$\widetilde{\mathcal{P}} = \text{BackTrace}(G_D, V_D^{\text{Art}}). \quad (4)$$

From the candidate path pool, DualResearch selects paths for the query according to two criteria. A useful path should cover the procedural steps required by the query. Its dependency edges should also be reliable according to the confidence scores in the depth graph. We therefore construct a query-specific procedural template $O(q)$ and use it to measure how well each candidate path matches the required procedure.

To construct $O(q)$, we first identify the main procedural sub-goals needed to answer the query. Each sub-goal is then grounded to the most relevant existing node in G_D. Sub-goals that cannot be grounded to any existing depth-graph node are discarded. The retained nodes are ordered according to the sub-goal order in the query, yielding a node-level procedural template

$$O(q) = [o_1, \ldots, o_m], \qquad o_i \in V_D.$$

For a candidate path p, let $O(p)$ denote the set of depth-graph nodes contained in the path. For coverage computation, we view the procedural template $O(q)$ through its underlying node set. We define procedural coverage as the fraction of template nodes covered by the candidate path:

$$\text{Cov}(p, q) = \frac{|O(q) \cap O(p)|}{|O(q)|}. \quad (5)$$

A path obtains full coverage when it contains all nodes in $O(q)$, even if it includes additional intermediate nodes. The grounded nodes in $O(q)$ specify the procedural evidence expected by the query. They may appear at different positions in G_D, rather than forming a consecutive path segment. We define path confidence as the average confidence of its dependency edges:

$$\text{Conf}(p) = \frac{1}{|E_p|} \sum_{e \in E_p} r_D(e), \quad (6)$$

where E_p is the edge set of path p.

This simple form avoids over-penalizing longer but valid reasoning chains while still favoring paths supported by reliable dependencies. The final depth score is

$$S_D(p \mid q) = \alpha \operatorname{Cov}(p, q) + (1 - \alpha) \operatorname{Conf}(p), \quad (7)$$

where $\alpha \in [0, 1]$ controls the trade-off between procedural coverage and path reliability. We rank $\widetilde{\mathcal{P}}$ by $S_D(p \mid q)$ and retain the top-K_D paths as $\mathcal{P}_q$.

3.4 Entropy-Gated Pair Fusion of Breadth Nodes and Depth Paths

The breadth channel returns top-K_B nodes $\mathcal{V}_q$, and the depth channel returns top-K_D paths $\mathcal{P}_q$. Entropy is computed only over these retained candidate sets, rather than over all graph nodes or all possible paths. We first normalize the candidate scores within each retained set into channel-wise relevance distributions:

$$\begin{aligned} \pi_B(v \mid q) &= \frac{\exp(S_B(v \mid q))}{\sum_{v' \in \mathcal{V}_q} \exp(S_B(v' \mid q))}, \\ \pi_D(p \mid q) &= \frac{\exp(S_D(p \mid q))}{\sum_{p' \in \mathcal{P}_q} \exp(S_D(p' \mid q))}. \end{aligned} \quad (8)$$

The normalized entropies are then defined as

$$\begin{aligned} H_B(q) &= -\frac{1}{\log |\mathcal{V}_q|} \sum_{v \in \mathcal{V}_q} \pi_B(v \mid q) \log \pi_B(v \mid q), \\ H_D(q) &= -\frac{1}{\log |\mathcal{P}_q|} \sum_{p \in \mathcal{P}_q} \pi_D(p \mid q) \log \pi_D(p \mid q). \end{aligned} \quad (9)$$

If a candidate set has size one, its entropy is set to zero. Entropy measures how concentrated each retrieval channel is over its retained candidates; it does not verify factual correctness. We convert the normalized entropies into channel weights using a softmax over negative entropy:

$$\begin{aligned} w_B(q) &= \frac{\exp(-H_B(q))}{\exp(-H_B(q)) + \exp(-H_D(q))}, \\ w_D(q) &= 1 - w_B(q). \end{aligned} \quad (10)$$

The negative-entropy softmax favors the channel with more concentrated candidates, while the final selection remains grounded in cross-channel agreement and supporting evidence spans.

DualResearch scores each breadth–depth evidence pair $(v, p) \in \mathcal{V}_q \times \mathcal{P}_q$ by combining entropy-gated channel relevance with cross-channel semantic agreement. To estimate consistency, each breadth node and depth path is represented as a compact text by concatenating its available metadata and evidence, and both texts are encoded with the same encoder $f(\cdot)$ used for retrieval. For notational simplicity, we denote these textual representations still as v and p. Their agreement is computed as

$$C(v, p) = \frac{\cos\big(f(v), f(p)\big) + 1}{2}. \quad (11)$$

This score measures semantic alignment between the two evidence texts rather than factual verification.

The pair score is

$$\begin{aligned} F(v, p \mid q) = &(1 - \lambda)\big[w_B(q)\pi_B(v \mid q) \\ &+ w_D(q)\pi_D(p \mid q)\big] + \lambda C(v, p), \end{aligned} \quad (12)$$

where the first term measures entropy-gated channel relevance and the second term favors breadth nodes and depth paths that discuss semantically aligned evidence.

DualResearch selects the highest-scoring evidence pair:

$$(v^*, p^*) = \arg \max_{v \in \mathcal{V}_q,\, p \in \mathcal{P}_q} F(v, p \mid q). \quad (13)$$

The selected pair (v^*, p^*) is converted into a compact evidence bundle for final answer reconstruction. The bundle summarizes the declarative evidence from the breadth node and the procedural provenance from the depth path with their metadata, evidence spans, and confidence information. Given the original query q and this structured bundle, the LLM can generate the final answer $\hat{y}$ using only the provided evidence. This stage reconstructs the answer from selected evidence rather than performing new retrieval or additional tool execution.

4 Experiments

We evaluate DualResearch in a controlled log-reuse setting. Section 4.1 describes the setting and baselines. Section 4.2 compares DualResearch with controlled log-reuse baselines on the same completed InternAgent logs. Sections 4.3–4.7 analyze components, cross-agent generalization, stochastic stability, post-processing overhead, and a representative case study. Appendix B provides subset-level supplemental results.

Table 1: Controlled log-reuse comparison on HLE. We report accuracy (%) for each HLE subset under the Text-Only and All-Set settings. The best result in each column is bolded.

Setting	Method	Math	Bio/Med	CS/AI	Physics	Human.	Chem.	Engineer.	Other	Avg.
Text-Only	InternAgent	23.5	18.9	13.8	17.3	21.8	21.8	18.8	25.0	21.2
	ReAct	24.4	14.3	12.0	17.0	20.5	15.8	14.4	17.6	19.4
	Naive-RAG	21.4	16.4	14.9	18.7	21.5	19.4	18.0	20.6	19.6
	LightRAG	30.2	22.5	22.4	20.0	26.0	21.2	19.8	22.7	25.5
	DualResearch	**32.2**	**28.4**	**26.8**	**24.3**	**29.5**	**25.7**	**29.7**	**30.1**	**29.7**
All-Set	InternAgent	23.5	18.9	17.4	15.7	19.2	18.2	16.2	20.6	20.4
	ReAct	24.7	16.2	12.1	18.3	21.8	18.8	17.2	22.2	20.9
	Naive-RAG	21.6	17.6	14.7	19.3	23.3	**22.8**	20.3	24.4	20.7
	LightRAG	30.7	24.8	21.4	21.8	26.4	**22.8**	**23.4**	**27.3**	27.1
	DualResearch	**32.8**	**27.9**	**28.6**	**24.8**	**28.3**	22.4	21.6	26.6	**29.0**

4.1 Experimental Setup

Baseline. Our main experiments use InternAgent (Team et al., 2025) as the underlying DR-Agent. All baselines receive the same task questions and the same completed logs produced by InternAgent. **ReAct** (Yao et al., 2022) reads the completed log without new tool calls, **Naive-RAG** performs flat vector retrieval over log chunks, and **LightRAG** (Guo et al., 2024) builds a graph-based RAG index over the same log evidence. This setting isolates answer reconstruction from a completed log rather than from a new agent execution.

Implementation. All encoders use bge-small-en-v1.5 (Xiao et al., 2024). We set $K_B = K_D = 10$, $\alpha = 0.5$, and $\lambda = 0.5$ unless otherwise specified. All LLMs used by the DR-Agents and DualResearch are o4-mini (OpenAI, 2025b) and the main experiments use temperature 0. Parser rules, JSON schemas, edge-confidence rubrics, cross-agent subset tables, and stochastic run details are provided in the Appendix.

Benchmark. **GAIA** (Mialon et al., 2023) is a general artificial intelligence assistant benchmark containing 466 real-world problems that require multi-step reasoning, multimodal understanding, web browsing, and tool use. Its task design is challenging for current models. We report results on its validation set, which includes 165 problems. **Google-Proof Q&A (GPQA)** (Rein et al., 2024) is a graduate-level benchmark of 448 expert-written multiple-choice questions in biology, chemistry, and physics, designed to test advanced scientific reasoning. We adopt the GPQA-Diamond subset, which contains 198 questions curated to include only items unanimously agreed upon by domain experts but often answered incorrectly by non-experts, ensuring both reliability and difficulty. **Humanity's Last Exam (HLE)** (Phan et al., 2025) is a benchmark consisting of 2,500 expert-curated, closed-form questions across eight domains. It evaluates advanced reasoning through multiple-choice and exact-match tasks, including 2,158 text-only questions and 342 text–image questions, thereby supporting rigorous evaluation across modalities.

4.2 Comparison with Log-Reuse Baselines

Table 1 reports a comparison on HLE, where all post-processing methods reuse the same completed logs generated by InternAgent. In the Text-Only setting, ReAct and Naive-RAG remain close to InternAgent, while LightRAG improves the average accuracy from 21.2% to 25.5%. DualResearch further raises it to 29.7%, with gains across all eight subsets. The All-Set setting shows the same trend: ReAct and Naive-RAG provide only small improvements over InternAgent, LightRAG reaches 27.1%, and DualResearch obtains the best overall average of 29.0%. These results suggest that flat log reuse is unreliable, whereas structured evidence organization provides a clearer benefit.

Tables 2 and 3 show consistent results on GAIA and GPQA. On GAIA, DualResearch improves the average accuracy to 71.52%, compared with 65.45% for InternAgent and 66.67% for LightRAG. The gain is largest on Level-3, where DualResearch reaches 38.46%, suggesting that graph separation is especially useful for harder multi-step tasks. On GPQA-Diamond, DualResearch also achieves the best overall result of 87.88%, outperforming InternAgent and LightRAG.

These log-reuse experiments show that com-

Table 2: Controlled log-reuse comparison on GAIA. We report validation accuracy (%) for each official GAIA difficulty level and the overall average.

Method	Level-1	Level-2	Level-3	Avg.
InternAgent	79.25	68.60	26.92	65.45
ReAct	81.13	65.12	30.77	64.85
Naive-RAG	71.70	62.79	26.92	60.00
LightRAG	81.13	67.44	34.62	66.67
DualResearch	**84.91**	**73.26**	**38.46**	**71.52**

Table 3: Controlled log-reuse comparison on GPQA-Diamond. We report accuracy (%) for each science subset and the overall average.

Method	Bio.	Chem.	Phys.	Avg.
InternAgent	73.68	70.97	94.19	81.31
ReAct	**84.21**	73.12	90.70	81.82
Naive-RAG	57.89	67.74	88.37	75.76
LightRAG	78.95	77.42	93.02	84.34
DualResearch	**84.21**	**80.65**	**96.51**	**87.88**

pleted execution logs contain useful evidence, but simply rereading the full trajectory or retrieving flat chunks does not extract it reliably. LightRAG benefits from graph organization, but it does not explicitly model execution dependencies or verified intermediate artifacts. DualResearch improves on this by combining breadth-oriented semantic anchors with depth-oriented causal provenance through entropy-gated node–path selection, which helps preserve useful factual coverage while filtering redundant or weakly grounded log segments.

4.3 Component Analysis

Table 4: Ablation study of Eq. 12 components on HLE All-Set, GAIA, and GPQA. Breadth and Depth denote the two evidence channels, Entropy denotes entropy-gated channel weighting, and Cosine denotes cross-channel semantic agreement.

Components				Datasets		
Breadth	Depth	Entropy	Cosine	HLE	GAIA	GPQA
✘	✘	✘	✘	20.4	65.45	81.31
✔	✘	✘	✘	23.6	68.48	81.31
✘	✔	✘	✘	23.5	67.27	82.83
✔	✔	✔	✘	21.9	65.45	80.81
✔	✔	✘	✔	26.6	67.27	83.84
✔	✔	✔	✔	**29.0**	**71.52**	**87.88**

We ablate DualResearch along four components: the breadth graph, the depth graph, entropy-gated channel weighting, and cross-channel semantic agreement. The results show that the gains arise from the coordinated design of these components rather than from any single module alone.

Breadth-only and depth-only variants both improve over the underlying DR-Agent, showing that declarative evidence and procedural provenance each contribute useful signal. Removing entropy or semantic agreement reduces performance, suggesting that final evidence selection is not a simple concatenation of the two channels. The full model performs best, supporting the combined dual-graph retrieval and entropy-gated pair fusion design.

4.4 Performance across different DR-Agents

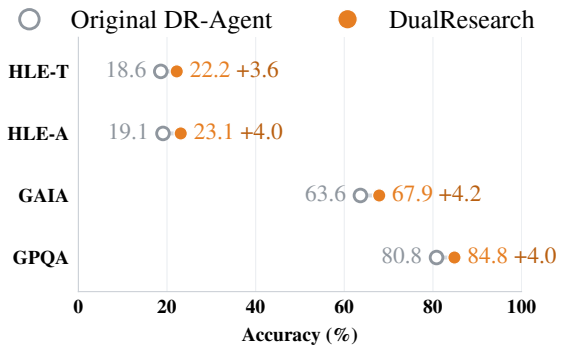

Figure 3: Performance of DualResearch as a post-processing module for OWL across three datasets. Gray markers denote the original OWL results, while orange markers denote the reconstructed results with DualResearch. HLE-T and HLE-A denote HLE Text-Only and HLE All-Set, respectively.

To test whether DualResearch depends on a specific log-producing agent, we also apply the same post-processing protocol to completed logs produced by OWL (Hu et al., 2025a). Figure 3 shows the original OWL result and the corresponding DualResearch result for each benchmark average. DualResearch yields consistent improvements over OWL logs on HLE, GAIA, and GPQA. Although these gains are smaller than those obtained with InternAgent, their consistency across datasets indicates that DualResearch generalizes beyond a single agent-specific logging format. Full subset-level OWL results are provided in Appendix B.1.

4.5 Stochastic Stability

The main experiments use temperature 0 to reduce sampling variance. To assess stochastic stability, we fix the completed InternAgent logs and rerun only the DualResearch post-processing stage five times at temperature 0.5. This setting evaluates stability under repeated reconstruction.

As shown in Table 5, the five DualResearch runs are tightly clustered on both HLE Text-Only and

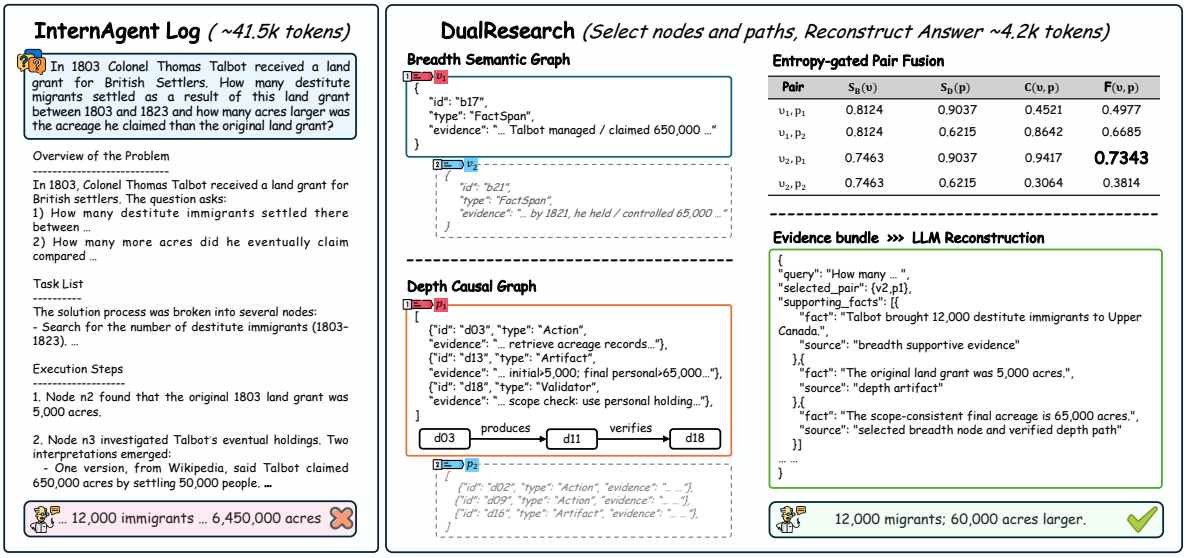

Figure 4: Node–path evidence selection from a completed InternAgent log with conflicting acreage evidence. DualResearch selects (v_2, p_1), aligning the scope-consistent 65,000-acre node with source-access and scope-check evidence before answer reconstruction.

Table 5: Stochastic stability on fixed HLE logs. InternAgent denotes the fixed score from completed logs, while Run 1–Run 5 are independent DualResearch runs. We report accuracy (%) and mean±standard deviation.

Setting	InternAgent	DualResearch Runs					Mean±Std
		Run 1	Run 2	Run 3	Run 4	Run 5	
HLE-T	21.2	28.8	28.7	28.8	28.8	28.6	**28.74±0.09**
HLE-A	20.4	29.6	29.7	29.6	29.6	29.6	**29.62±0.04**

HLE All-Set. The small variation indicates that the reconstruction results are stable under stochastic decoding when the completed logs are fixed. Full five-run subset-level results are provided in Appendix B.2.

4.6 Post-processing Overhead

Table 6 reports the incremental cost of different post-processing methods on HLE Text-Only. DualResearch adds 3.23 minutes and 44.52k tokens. Its runtime overhead is lower than Naive-RAG and LightRAG, while its token overhead is close to LightRAG. Although ReAct is cheaper, its accuracy gains are limited. These results show that DualResearch offers a favorable accuracy–cost trade-off as a lightweight log-reconstruction stage.

Table 6: Incremental post-processing overhead on HLE Text-Only. InternAgent is the base trace-generation cost, and each "+" column is the additional cost on the completed log.

Metric	InternAgent	Post-processing Overhead			
		+ ReAct	+ Naive-RAG	+ LightRAG	+ DualResearch
Time (min)	17.64	+1.21	+4.73	+3.79	**+3.23**
Token (k)	635.24	+35.26	+12.31	+46.17	**+44.52**

4.7 Case Study

Figure 4 illustrates a Talbot acreage case in which the completed log contains both a scope-consistent 65,000-acre figure and a broader 650,000-acre figure. The underlying agent can be pulled toward the broader evidence even though it does not match the question scope. DualResearch retains the relevant facts in the breadth graph and the source access, unit normalization, and scope check in the depth graph. The selected node–path pair supports the scope-consistent reconstruction: 12,000 migrants and $65{,}000 - 5{,}000 = 60{,}000$ acres.

5 Conclusion

We presented DualResearch, a log-based post-processing framework for DR-Agents. DualResearch takes completed execution logs as input and organizes them into two complementary graph structures: a breadth semantic graph for declarative evidence and a depth causal graph for procedural provenance. It then applies entropy-gated fusion to select a compact evidence bundle and reconstructs the answer. Across HLE, GAIA, and GPQA, DualResearch improves accuracy over the underlying DR-Agent and log-reuse baselines, while providing more traceable evidence support.

These results indicate that post-hoc reconstruction benefits from modeling declarative evidence and procedural provenance together. Future work will extend DualResearch to multimodal and numerical evidence, improve confidence calibration under noisy or incomplete logs.

Limitations

DualResearch depends on the quality and completeness of the completed execution log. If an underlying DR-Agent omits a required tool call, records incomplete intermediate states, or stores weak validator information, the post-processing stage cannot recover evidence that never appears in the trace. Different DR-Agents also expose different logging formats, so the raw-log parser must be adapted before the normalized event schema can be reused.

DualResearch can represent tool-backed numerical and structural evidence when such results appear in the trace, but it does not yet directly ground raw visual regions, table cells, or other multimodal artifacts outside the logged textual evidence. Future work could connect visual and tabular grounding modules to both graph layers, broaden validation tools, and study additional DR-Agent systems beyond the agents evaluated here.

Ethics Statement

This article does not involve research on human subjects or the release of a new dataset. DualResearch operates on completed execution logs, which may contain user queries, retrieved web snippets, intermediate tool outputs, and model-generated reasoning artifacts. In real deployments, such logs should be sanitized before graph construction, and sensitive identifiers should be removed or access-controlled. The graph extraction module may assign confidence to unsupported or biased evidence; therefore, confidence scores should be interpreted as evidence-support signals rather than guarantees of truth.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Toufique Ahmed and Premkumar Devanbu. 2023. Better patching using llm prompting, via self-consistency. In *2023 38th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, pages 1742–1746. IEEE.

Chi-Min Chan, Chunpu Xu, Ruibin Yuan, Hongyin Luo, Wei Xue, Yike Guo, and Jie Fu. 2024. Rq-rag: Learning to refine queries for retrieval augmented generation. *arXiv preprint arXiv:2404.00610*.

Google DeepMind. 2024. Gemini deep research. https://deepmind.google/technologies/gemini/deep-research/. Accessed: 2025-09-24.

Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, and Jonathan Larson. 2024. From local to global: A graph rag approach to query-focused summarization. *arXiv preprint arXiv:2404.16130*.

Wenqi Fan, Yujuan Ding, Liangbo Ning, Shijie Wang, Hengyun Li, Dawei Yin, Tat-Seng Chua, and Qing Li. 2024. A survey on rag meeting llms: Towards retrieval-augmented large language models. In *International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 6491–6501.

Fangqun Gao, Shu Xu, Weiyan Hao, and Tao Lu. 2025. Ka-rag: Integrating knowledge graphs and agentic retrieval-augmented generation for an intelligent educational question-answering model. *Applied Sciences*, 15(23):12547.

Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. 2022. Precise zero-shot dense retrieval without relevance labels. *arXiv preprint arXiv:2212.10496*.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, and Haofen Wang. 2023. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*.

Zirui Guo, Lianghao Xia, Yanhua Yu, Tian Ao, and Chao Huang. 2024. Lightrag: Simple and fast retrieval-augmented generation. *arXiv preprint arXiv:2410.05779*, 2(3).

Mengkang Hu, Yuhang Zhou, Wendong Fan, Yuzhou Nie, Bowei Xia, Tao Sun, Ziyu Ye, Zhaoxuan Jin, Yingru Li, Qiguang Chen, and 1 others. 2025a. Owl: Optimized workforce learning for general multi-agent assistance in real-world task automation. *arXiv preprint arXiv:2505.23885*.

Yusong Hu, Runmin Ma, Yue Fan, Jinxin Shi, Zongsheng Cao, Yuhao Zhou, Jiakang Yuan, Xiangchao Yan, Wenlong Zhang, Lei Bai, and 1 others. 2025b. Flowsearch: Advancing deep research with dynamic structured knowledge flow. *arXiv preprint arXiv:2510.08521*.

Jie Huang and Kevin Chen-Chuan Chang. 2023. Towards reasoning in large language models: A survey. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 1049–1065.

Yuxuan Huang, Yihang Chen, Haozheng Zhang, Kang Li, Meng Fang, Linyi Yang, Xiaoguang Li, Lifeng Shang, Songcen Xu, Jianye Hao, and 1 others. 2025. Deep research agents: A systematic examination and roadmap. *arXiv preprint arXiv:2506.18096*.

Nicola Jones. 2025. Openai's' deep research'tool: is it useful for scientists? *Nature*.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474.

Tianle Li, Ge Zhang, Quy Duc Do, Xiang Yue, and Wenhu Chen. 2024. Long-context llms struggle with long in-context learning. *Preprint*, arXiv:2404.02060.

Xiaoxi Li, Jiajie Jin, Guanting Dong, Hongjin Qian, Yutao Zhu, Yongkang Wu, Ji-Rong Wen, and Zhicheng Dou. 2025. Webthinker: Empowering large reasoning models with deep research capability. *arXiv preprint arXiv:2504.21776*.

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

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, and 1 others. 2024b. Agentbench: Evaluating llms as agents. In *International Conference on Learning Representations*, volume 2024, pages 52989–53046.

Grégoire Mialon, Clémentine Fourrier, Thomas Wolf, Yann LeCun, and Thomas Scialom. 2023. Gaia: a benchmark for general ai assistants. In *The Twelfth International Conference on Learning Representations*.

Xuan-Phi Nguyen, Shrey Pandit, Revanth Gangi Reddy, Austin Xu, Silvio Savarese, Caiming Xiong, and Shafiq Joty. 2025. Sfr-deepresearch: Towards effective reinforcement learning for autonomously reasoning single agents. *arXiv preprint arXiv:2509.06283*.

OpenAI. 2025a. Deepresearch. https://openai.com/research/deep-research. Accessed: 2025-09-24.

OpenAI. 2025b. Openai o3 and o4-mini system card. OpenAI website. Available: https://openai.com/index/o3-o4-mini-system-card/.

Long Phan, Alice Gatti, Ziwen Han, Nathaniel Li, Josephina Hu, Hugh Zhang, Chen Bo Calvin Zhang, Mohamed Shaaban, John Ling, Sean Shi, and 1 others. 2025. Humanity's last exam. *arXiv preprint arXiv:2501.14249*.

Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2023. In-context retrieval-augmented language models. *Transactions of the Association for Computational Linguistics (TACL)*, 11:1316–1331.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. 2024. Gpqa: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language models can teach themselves to use tools. *Advances in neural information processing systems*, 36:68539–68551.

Parshin Shojaee, Iman Mirzadeh, Keivan Alizadeh, Maxwell Horton, Samy Bengio, and Mehrdad Farajtabar. 2025. The illusion of thinking: Understanding the strengths and limitations of reasoning models via the lens of problem complexity. *arXiv preprint arXiv:2506.06941*.

NovelSeek Team, Bo Zhang, Shiyang Feng, Xiangchao Yan, Jiakang Yuan, Zhiyin Yu, Xiaohan He, Songtao Huang, Shaowei Hou, Zheng Nie, and 1 others. 2025. Novelseek: When agent becomes the scientist–building closed-loop system from hypothesis to verification. *arXiv preprint arXiv:2505.16938*.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2023. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: long papers)*, pages 10014–10037.

Shitao Xiao, Zheng Liu, Peitian Zhang, Niklas Muennighoff, Defu Lian, and Jian-Yun Nie. 2024. C-pack: Packed resources for general chinese embeddings. In *Proceedings of the 47th international ACM SIGIR conference on research and development in information retrieval*, pages 641–649.

Sheng Xu, Mike Chen, and Shuwen Chen. 2024. Enhancing retrieval-augmented generation models with knowledge graphs: Innovative practices through a dual-pathway approach. In *International conference on intelligent computing*, pages 398–409. Springer.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. 2022. React: Synergizing reasoning and acting in language models. In *The eleventh international conference on learning representations*.

Yue Yu, Wei Ping, Zihan Liu, Boxin Wang, Jiaxuan You, Chao Zhang, Mohammad Shoeybi, and Bryan Catanzaro. 2024. Rankrag: Unifying context ranking with retrieval-augmented generation in llms. *arXiv preprint arXiv:2407.02485*.

Yanbo Zhang, Sumeer A Khan, Adnan Mahmud, Huck Yang, Alexander Lavin, Michael Levin, Jeremy Frey, Jared Dunnmon, James Evans, Alan Bundy, and 1 others. 2025. Exploring the role of large language models in the scientific method: from hypothesis to discovery. *npj Artificial Intelligence*, 1(1):14.

A Reproducibility Details

This appendix records the implementation-facing details needed to reproduce DualResearch from completed execution logs. The retrieval and fusion formulas are given in the main Method section; here we focus on log normalization, graph extraction interfaces, confidence assignment, and operational settings.

A.1 Log Normalization and Graph Extraction Interface

This section describes how DualResearch converts DR-Agent logs into the structured event records used for graph construction. This step is a format-normalization stage rather than an evidence-expansion stage. It does not perform new retrieval or inject external facts.

Chunked log normalization. Completed DR-Agent logs can be long and structurally heterogeneous. We therefore parse them in fixed-size chunks to make span segmentation stable and to preserve local tool-call, output, and validation structures. Specifically, we split each completed log into 2048-token chunks with an overlap of 64 tokens. The overlap prevents tool calls, tool outputs, validator messages, or final-answer spans from being truncated at chunk boundaries.

Each chunk is parsed into normalized events using the following prompt. The parser assigns each event one of six roles: `think`, `tool_call`, `tool_output`, `code`, `check`, or `final`. It also records the original source span, approximate token range, tool metadata, output artifacts, variables, execution status, and validator information when these fields are available in the log. The normalization stage does not summarize long outputs. It preserves verbatim source spans and records token ranges, while later graph extraction selects minimal evidence spans from the normalized events.

```
<SYSTEM PROMPT>
You are a strict log-normalization engine for completed DR-Agent
traces.
Return only valid JSON. Do not add explanations. Do not infer facts
that are not explicitly present in the log chunk.

<USER PROMPT>
# Task:
Convert the given log chunk into a list of normalized events. The goal is to
segment and structure the existing trace, not to summarize it or add new
information.

## General rules:
1. Use only text that appears in LOG_CHUNK.
2. Do not perform new retrieval, rerun tools, execute code, or add external
   facts.
3. Preserve the original wording in source spans whenever possible.
4. Keep events in their original order within the chunk.
5. Split the chunk into multiple events when it contains multiple reasoning
   spans, tool calls, tool outputs, code blocks, checks, or final-answer spans.
6. If a field is unavailable in the log chunk, use an empty string, empty list,
   or "unknown" rather than guessing.
7. If a span is incomplete because it is cut by the chunk boundary, still record
   the available text and mark execution_status as "unknown" unless the status
   is explicit.
8. Do not deduplicate across chunks. Cross-chunk deduplication will be handled
   after all chunks are parsed.

## Event roles:
- think:
  Planning, intermediate reasoning, decomposition, or self-reflection recorded
  in the trace.
- tool_call:
  A structured invocation of an external tool, such as search, browser,
  calculator, Python, SymPy, RDKit, or other agent tools.
- tool_output:
  Returned observations, retrieved snippets, browser outputs, command outputs,
  code results, or other tool-produced content.
- code:
  Raw executable code blocks, scripts, or explicit variable assignments recorded
  in the trace. If a code block is wrapped as a tool invocation, record the
  invocation as tool_call and the returned result as tool_output.
- check:
  Explicit validation, verification, unit checking, consistency checking,
  contradiction checking, scope checking, or pass/fail judgments.
- final:
  Final answer draft, conclusion, or answer span produced by the DR-Agent.

## For each event, output the following fields:
- role:
  One of think, tool_call, tool_output, code, check, final.
- source_span:
  The verbatim text span from LOG_CHUNK that supports this event.
- source_token_range:
  The approximate token range of the event within the full log.
- tool_name:
  The tool name if the event is a tool call or tool output; otherwise empty.
- tool_inputs:
  The explicit tool arguments if present; otherwise an empty object.
- output_text:
  The returned or produced text if present; otherwise empty.
- artifacts:
  Computed values, retrieved spans, files, tables, object ids, or named
  intermediate outputs explicitly present in the log.
- execution_status:
  One of success, warning, failure, or unknown.
- variables:
  Explicit variable names and values if present.
- validator_status:
  One of pass, warning, fail, none, or unknown.
- validator_message:
  The explicit validation message if present; otherwise empty.

Output JSON format:
{
  "role": "<think | tool_call | tool_output | code | check | final>",
  "source_span": "<verbatim span from LOG_CHUNK>",
  "source_token_range": [start_token, end_token],
  "tool_name": "<tool name or empty string>",
  "tool_inputs": {},
  "output_text": "<verbatim output text or empty string>",
  "artifacts": [],
  "execution_status": "<success | warning | failure | unknown>",
  "variables": [],
  "validator_status": "<pass | warning | fail | none | unknown>",
  "validator_message": "<verbatim validator message or empty string>"
}

Return only the JSON object.
```

Graph extraction. After log normalization, the graph extractor builds either the breadth semantic graph or the depth causal graph from the normalized event sequence. The same prompt template is used for both graph layers, with the `MODE` field controlling the allowed node and edge types.

```
<SYSTEM PROMPT>
You are a strict information-extraction engine for DR-Agent execution logs.
Return only valid JSON that matches the schema. Do not add explanations.

<USER PROMPT>
# Task:
Extract a graph G = (V, E) for downstream retrieval and auditing.

## MODE:
{BREADTH | DEPTH}

## QUESTION:
{question}

## NORMALIZED_EVENTS:
{normalized_events}

## General rules:
1. Use only information explicitly supported by NORMALIZED_EVENTS.
2. Do not introduce external facts, new retrieval results, or new tool outputs.
3. Every node and edge must include:
   (i) a short label,
   (ii) supporting event ids,
   (iii) minimal evidence spans copied verbatim from NORMALIZED_EVENTS,
   (iv) a confidence score in (0, 1] measuring support strength in the log.
4. Evidence spans should be concise and preserve the original wording.
5. Deduplicate nodes that refer to the same entity, artifact, value, or step.
6. If the log evidence is conflicting or incomplete, keep the item only when it
   may affect answer reconstruction, and assign a lower confidence score.
7. Do not use outside knowledge to complete missing evidence.

## Confidence rubric:
```

```
- 0.90--1.00:
  Directly stated, tool-produced, code-produced, or explicitly validated, and
  internally consistent.
- 0.70--0.89:
  Clearly supported by one or more log spans, with only minor ambiguity.
- 0.40--0.69:
  Partially supported, paraphrased, or dependent on indirect evidence.
- 0.10--0.39:
  Speculative, conflicting, incomplete, or weakly grounded.

## If MODE = BREADTH:
- Allowed node types:
  Entity, Definition, FactSpan, Formula.
- Allowed edge types:
  mentions, defines, aliases, supports, contradicts, part_of.
- Extract stable declarative evidence from retrieved text, factual summaries,
  formulas, definitions, entity relations, and explicitly stated claims.
- Exclude procedural execution steps unless they state reusable facts.
- Use FactSpan for concrete claims, values, dates, or evidence snippets.
- Use Formula for mathematical, symbolic, or structured expressions.

## If MODE = DEPTH:
- Allowed node types:
  Action, Artifact, Validator.
- Allowed edge types:
  triggers, produces, consumes, verifies, follows.
- Extract procedural provenance from tool calls, code execution, intermediate
  outputs, computed values, scope checks, consistency checks, and final-answer
  dependencies.
- Use Action for tool calls, searches, browser operations, code execution,
  parsing, computation, comparison, or answer-revision steps.
- Use Artifact for retrieved snippets, parsed tables, computed values, generated
  files, intermediate conclusions, or final candidate answers.
- Use Validator for explicit checks, unit tests, consistency checks, scope
  checks, constraint checks, or contradiction checks.
- Represent computations as Action -> Artifact -> Validator structures whenever
  the corresponding command, result, and check appear in the normalized events.

Output JSON format:
{
  "mode": "<BREADTH | DEPTH>",
  "question": "<copy QUESTION>",
  "nodes": [
    {
      "id": "n1",
      "type": "<allowed node type>",
      "label": "<short label>",
      "canonical": "<canonical name or empty string>",
      "evidence": ["<verbatim span copied from NORMALIZED_EVENTS>"],
      "confidence": 0.85
    }
  ],
  "edges": [
    {
      "source": "n1",
      "target": "n2",
      "type": "<allowed edge type>",
      "evidence": ["<verbatim span copied from NORMALIZED_EVENTS>"],
      "confidence": 0.85
    }
  ]
}

Return only the JSON object.
```

The confidence score assigned by the extractor measures how strongly the completed log supports an extracted node or edge. It is used for auditing, schema filtering, breadth neighborhood smoothing, and depth path scoring. It should not be interpreted as an independent guarantee of factual correctness.

A.2 Implementation Details

DualResearch performs no additional external retrieval or tool execution during post-processing. All evidence used for graph construction, retrieval, and answer reconstruction must appear in the completed log after normalization.

Retrieval implementation. Breadth retrieval applies only one-hop smoothing over supportive neighbors. This keeps neighborhood support while avoiding multi-hop oversmoothing, and low-confidence edges contribute less through their r_B weights. Depth retrieval uses artifact-bounded backtracing: candidate paths start from artifact nodes and trace backward along dependency edges until a root node or another artifact node is reached. If no query sub-goal can be grounded to an existing depth node, the coverage term is set to zero and paths are ranked by dependency confidence. If no artifact-bounded path is available, the bundle falls back to the retained breadth evidence with an empty depth path.

Table 7: Cross-agent evaluation on GAIA with InternAgent and OWL logs.

Method	Level-1	Level-2	Level-3	Avg.
InternAgent	79.25	68.60	26.92	65.45
DualResearch on InternAgent	84.91	73.26	38.46	71.52
OWL	75.47	66.28	30.77	63.64
DualResearch on OWL	81.13	69.77	34.62	67.88

Table 8: Cross-agent evaluation on GPQA with InternAgent and OWL logs.

Method	Bio.	Chem.	Phys.	Avg.
InternAgent	73.68	70.97	94.19	81.31
DualResearch on InternAgent	84.21	80.65	96.51	87.88
OWL	63.16	73.12	93.02	80.81
DualResearch on OWL	73.68	76.34	96.51	84.85

B Additional Experimental Results

B.1 Cross-Agent Generalization with OWL

Tables 7–9 provide the subset-level results behind Figure 3. DualResearch uses the same post-processing protocol for InternAgent and OWL logs.

B.2 Stochastic Stability

The main experiments use temperature 0 to minimize sampling variance. To report stochastic variation without rerunning the underlying DR-Agent, we fix the same completed InternAgent logs and run five replications of only the DualResearch post-processing stage on HLE at temperature 0.5. Table 10 provides the subset-level results behind the stability summary in Table 5.

Analysis of the landscape tables. Tables 11–13 place the controlled DualResearch results in a broader landscape. On GAIA, DualResearch with o4-mini reaches an average accuracy of 71.52, improving over InternAgent with o4-mini by 6.07 points and ranking highest among the listed systems; the gain is consistent across Level-1, Level-2, and Level-3, although OpenAI Deep Research remains higher on the most difficult Level-3 subset. On GPQA, DualResearch obtains the best listed average score of 87.88, exceeding InternAgent by 6.57 points and matching or surpassing

Table 9: Cross-agent evaluation on HLE with InternAgent and OWL logs. We report accuracy (%) on each HLE subset under Text-Only and All-Set settings.

Setting	Method	Math	Bio/Med	CS/AI	Physics	Human.	Chem.	Engineer.	Other	Avg.
Text-Only	InternAgent	23.5	18.9	13.8	17.3	21.8	21.8	18.8	25.0	21.2
	DualResearch on InternAgent	32.2	28.4	26.8	24.3	29.5	25.7	29.7	30.1	29.7
	OWL	22.1	13.2	14.5	17.4	19.6	16.4	13.5	18.0	18.6
	DualResearch on OWL	26.8	18.2	16.2	18.3	20.5	20.0	16.2	22.3	22.2
All-Set	InternAgent	23.5	18.9	17.4	15.7	19.2	18.2	16.2	20.6	20.4
	DualResearch on InternAgent	32.8	27.9	28.6	24.8	28.3	22.4	21.6	26.6	29.0
	OWL	22.1	14.0	11.2	14.9	20.7	19.8	17.2	22.7	19.1
	DualResearch on OWL	26.3	20.7	14.7	18.8	23.3	22.8	17.2	25.6	23.1

Table 10: Five-run subset-level HLE stability results for DualResearch at temperature 0.5

Setting	Run	Math	Bio/Med	CS/AI	Physics	Human.	Chem.	Engineer.	Other	Avg.
Text-Only	R1	32.6	27.5	28.2	24.8	27.9	21.8	22.5	26.6	28.8
	R2	32.2	27.1	30.3	27.0	27.9	20.0	19.8	26.2	28.7
	R3	32.4	28.6	27.4	23.5	28.3	23.6	22.5	27.5	28.8
	R4	33.2	26.1	29.5	23.9	28.8	20.6	19.8	26.6	28.8
	R5	32.2	28.2	29.0	23.5	28.3	23.6	21.6	25.3	28.6
	Mean±Std	32.52±0.42	27.50±0.96	28.88±1.13	24.54±1.48	28.24±0.38	21.92±1.65	21.24±1.32	26.44±0.80	**28.74±0.09**
All-Set	R1	31.5	29.3	26.8	24.8	29.5	28.7	31.3	28.4	29.6
	R2	32.4	27.0	25.4	24.3	31.1	27.7	32.8	28.4	29.7
	R3	31.9	27.9	27.7	24.8	30.1	23.8	32.8	29.0	29.6
	R4	31.8	27.5	26.8	25.2	28.5	26.7	29.7	31.3	29.6
	R5	32.4	27.9	26.8	23.3	29.0	26.7	28.1	30.1	29.6
	Mean±Std	32.00±0.39	27.92±0.86	26.70±0.82	24.48±0.73	29.64±0.99	26.72±1.82	30.94±2.01	29.44±1.26	**29.62±0.04**

the strongest direct-model baselines on all three subject subsets. On HLE, DualResearch with o4-mini improves the InternAgent baseline from 21.2 to 29.7 in the Text-Only setting and from 20.4 to 29.0 in the All-Set setting. The HLE table also shows that the same reconstruction procedure improves the Qwen3-235B InternAgent logs, but the absolute scores remain lower than the o4-mini setting, suggesting that DualResearch benefits from stronger underlying logs while still adding value through post-processing. These landscape numbers therefore support the main trend observed in the controlled experiments, while remaining descriptive rather than protocol-controlled comparisons.

Table 11: Landscape results on **GAIA**. These results provide context and are not controlled head-to-head comparisons. Methods marked † are direct model evaluations, and those marked ‡ are agent-based.

Method	Accuracy in each subset (%) ↑			
	Level-1	Level-2	Level-3	Avg.
Qwen3-235B †	15.09	3.49	3.84	6.67
o4-mini †	28.30	12.79	7.69	16.97
DeepSeek-R1-671B †	33.96	13.95	3.84	18.74
OpenAI Deep Research ‡	74.29	69.06	**47.60**	67.36
InternAgent (o4-mini) ‡	79.25	68.60	26.92	65.45
DualResearch (o4-mini) ‡	**84.91**	**73.26**	38.46	**71.52**

Table 12: Landscape results on **GPQA**. These results provide context and are not controlled head-to-head comparisons. Methods marked † are direct model evaluations, and those marked ‡ are agent-based.

Method	Accuracy in each subset (%) ↑			
	Bio	Chem	Phys	Avg.
DeepSeek-R1-671B †	63.16	76.34	91.86	82.32
o4-mini †	78.95	63.44	94.19	78.28
GPT-5 †	**84.21**	76.34	95.35	85.35
WebThinker-32B-RL ‡	78.90	50.50	90.70	70.70
InternAgent (o4-mini) ‡	73.68	70.97	94.19	81.31
DualResearch (o4-mini) ‡	**84.21**	**80.65**	**96.51**	**87.88**

Table 13: Landscape results on **HLE**. These results provide context and are not controlled head-to-head comparisons. Methods marked † are direct model evaluations, and those marked ‡ are agent-based. Here, "Qwen3-235B" denotes "Qwen3-235B-A22B-Instruct."

	Method	Accuracy in each subset (%) ↑								
		Math	Bio/Med	CS/AI	Physics	Human.	Chem.	Engineer.	Other	Avg.
Text-Only	QwQ-32B †	12.6	14.0	7.9	4.0	6.0	13.3	5.3	4.4	9.6
	DeepSeek-R1-671B †	9.3	8.6	7.4	5.8	11.0	5.6	10.3	7.5	8.6
	Qwen3-235B †	11.4	9.0	8.5	5.5	8.3	4.9	7.8	6.3	9.2
	o4-mini †	19.7	9.9	13.4	13.4	9.8	6.9	9.4	6.8	14.5
	GPT-5 †	31.3	21.2	25.5	23.3	21.8	18.8	10.9	19.3	25.9
	WebThinker-32B-RL ‡	16.7	25.6	2.0	12.7	18.0	**26.7**	15.8	15.6	15.8
	SFR-DR-20B ‡	-	-	-	-	-	-	-	-	28.7
	Tongyi ‡	-	-	-	-	-	-	-	-	**32.9**
	Kimi-Research ‡	-	-	-	-	-	-	-	-	26.9
	InternAgent (Qwen3-235B) ‡	13.5	15.3	11.6	6.9	16.1	8.9	15.6	17.6	13.3
	InternAgent (o4-mini) ‡	23.5	18.9	13.8	17.3	21.8	21.8	18.8	25.0	21.2
	DualResearch (Qwen3-235B) ‡	16.9	18.9	13.9	10.9	19.7	21.8	21.9	19.9	17.1
	DualResearch (o4-mini) ‡	**32.2**	**28.4**	**26.8**	**24.3**	**29.5**	25.7	**29.7**	**30.1**	29.7
All-Set	Qwen3-235B †	11.1	7.9	8.3	6.1	7.8	5.5	7.2	5.2	8.6
	o4-mini †	19.0	11.4	12.9	12.6	9.1	12.7	12.6	6.9	14.3
	GPT-5 †	31.0	22.1	24.9	21.7	20.6	16.4	14.4	18.0	24.8
	OpenAI Deep Research ‡	-	-	-	-	-	-	-	-	26.6
	Gemini Deep Research ‡	-	-	-	-	-	-	-	-	26.9
	InternAgent (Qwen3-235B) ‡	13.0	12.5	10.8	7.4	14.2	7.9	9.0	13.7	11.9
	InternAgent (o4-mini) ‡	23.5	18.9	17.4	15.7	19.2	18.2	16.2	20.6	20.4
	DualResearch (Qwen3-235B) ‡	16.2	15.0	12.8	9.6	17.4	13.3	12.6	15.0	14.8
	DualResearch (o4-mini) ‡	**32.8**	**27.9**	**28.6**	**24.8**	**28.3**	**22.4**	**21.6**	**26.6**	**29.0**