RUBRICATLAS: DECOUPLING RUBRIC DISCOVERY FROM FACT VALIDATION TO OPTIMIZE GENERALIST DEEP RESEARCH AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Deep research agents are increasingly deployed to solve complex tasks by dynamically searching the web, integrating multi-source evidence, and composing long-form reports. Since such reports lack unique reference answers, task-specific rubrics are crucial for steering agent capability enhancement. However, existing rubric synthesis paradigms—such as offline model comparisons or online rollout evaluations—accept newly generated criteria without independent verification. This induces severe reward hacking, encouraging policies to favor excessive verbosity, superficial reasoning, and factually ungrounded demands, while existing datasets and queries suffer from limited domain coverage. To overcome these limitations, we present **RUBRICATLAS**, a unified framework for optimizing deep research agents across diverse knowledge domains. RUBRICATLAS builds a four-level taxonomy with 234 leaf concepts for diverse task synthesis. To enforce reliable supervision, we introduce Decoupled Rubric Discovery and Validation (DRDV), which decouples depth-oriented criteria discovery from search-augmented factual verification to filter ungrounded constraints. A Memo Context Manager maintains structured findings and evidence provenance throughout training and inference, supporting context integrity over long research horizons. For policy optimization, Group reward-Decoupled Normalization Policy Optimization (GDPO) normalizes rubric-wise advantages before aggregation, preserving fine-grained reward distinctions across varying reward scales. Extensive experiments show that RUBRICATLAS, with Qwen3-14B as its backbone, matches or surpasses strong proprietary models across three evaluated deep-research benchmarks, achieving an **82.60%** win rate on DeepConsult, a **46.67** RACE score on DeepResearch Bench, and **62.15%** rubric compliance on ResearchRubrics.

1 INTRODUCTION

Deep research agents powered by large language models (LLMs) have emerged as a pivotal paradigm for solving open-ended, information-intensive tasks. Unlike conventional single-turn question-answering systems, these agents operate over extended horizons by dynamically formulating search queries, synthesizing multi-source evidence, resolving conflicting evidence from the web, and composing long-form analytical reports(Shao et al., 2024; Zheng et al., 2025; DeepResearch et al., 2025). These reports rarely admit a unique reference answer, where distinct responses can present equally accurate facts yet vary drastically in source attribution, depth of analysis, thematic angles, and overall task coverage. As a result, task-specific rubrics have become central to policy optimization by decomposing report quality into explicit criteria and translating open-ended research into quantifiable reward signals(Du et al., 2026; Gunjal et al., 2026; Shao et al., 2025).

Existing paradigms for rubric synthesis struggle to balance factual rigor with evaluative depth. Retrieval-grounded approaches derive atomic criteria from external sources, providing evidence-backed supervision for the factual content that a report should cover (Shao et al., 2025; Mei et al., 2026; Zhu et al., 2026). Although atomic, these criteria are primarily content-oriented and do not directly capture subtle analytical quality dimensions, including perspective diversity, cross-source reasoning, and structural coherence. To capture such dimensions, complementary approaches synthesize criteria by comparing responses across model capability levels or contrasting multiple online

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.

rollouts sampled from the current policy (Liu et al., 2026b; Zhang et al., 2026c; Rezaei et al., 2025; Shao et al., 2025). While multi-trace comparisons can reveal richer and more discriminative evaluation dimensions, the synthesized criteria inevitably inherit the generator's underlying biases and ungrounded priors. Crucially, when a criterion introduces a factual requirement, consensus among compared responses establishes neither the veracity of the underlying claim nor its relevance to the task. Beyond such factual errors, we observe that rubric generators often mistake surface signals of quality for analytical depth, converting verbose passages, self-asserted claims of comprehensiveness, or analyses produced from parametric knowledge without source support into positive criteria (Ren et al., 2026). Once embedded in policy optimization, these ungrounded and surface-oriented rubrics induce severe reward hacking, encouraging agent policies to favor excessive verbosity, superficial reasoning, and factually unsupported claims (Kawabata & Sugawara, 2026; Mahmoud et al., 2026; Yang et al., 2026). Meanwhile, most existing training corpora concentrate on a limited set of expert domains, thereby failing to capture the vast breadth of general open-ended research tasks.

To address these limitations, we introduce **RUBRICATLAS**, a unified framework for optimizing deep research agents through diverse task synthesis, reliable rubric supervision, and fine-grained rubric-guided policy optimization. Starting from 13 root categories, RUBRICATLAS constructs its concept hierarchy top-down through corpus-driven retrieval, where an LLM agent leverages search-mined evidence to recursively propose semantically distinct child concepts at each level, expanding the taxonomy into a four-level hierarchy of 234 fine-grained leaf concepts to synthesize open-ended tasks with grounded factual criteria across broad knowledge fields. To establish reliable supervision, *Decoupled Rubric Discovery and Validation* (DRDV) decouples depth- and perspective-oriented criterion discovery from search-augmented factual verification. A generator proposes tentative criteria from trajectory contrasts, while search-augmented validators extract atomic factual premises from each proposal and perform reverse web verification; iterative generator–validator debate retains only criteria that are factually supported and task-relevant. Together, taxonomy-guided task synthesis and DRDV produce a training corpus of 8000 high-quality query–rubric pairs spanning diverse knowledge domains. Throughout training and inference, a Memo Context Manager continuously organizes intermediate findings and retrieved evidence, preserving evidence provenance and contextual continuity over long research horizons. During policy optimization, RUBRICATLAS first performs format-oriented cold-start fine-tuning and then conducts rubric-guided reinforcement learning with *Group reward-Decoupled Normalization Policy Optimization* (GDPO), which computes a group-normalized advantage independently for each rubric dimension before weighted aggregation, preserving criterion-level distinctions obscured by aggregate-then-normalize reward schemes.

We conduct comprehensive experiments to validate the effectiveness and versatility of RUBRICATLAS. Built upon Qwen3-14B backbone, our method substantially improves performance across all three deep-research benchmarks, raising the win rate on DeepConsult from 17.40% to 82.60%, the RACE score on DeepResearch Bench from 35.70 to 46.67, and rubric compliance on ResearchRubrics from 39.21% to 62.15%. Beyond these benchmark gains, RUBRICATLAS matches or surpasses strong proprietary deep-research agents, underscoring the decisive role of high-quality rubric supervision in advancing deep research capabilities. Controlled ablations further confirm the joint necessity of our core components in eliminating ungrounded criteria preferences, preserving fine-grained reward distinctions, and maintaining long-horizon context integrity. Our contributions can be summarized as follows:

- We propose RUBRICATLAS, an end-to-end framework for optimizing deep research agents via taxonomy-guided task synthesis, search-augmented rubric verification, memory-managed reasoning, and group reward-decoupled advantage normalization. Additionally, we construct a curated dataset of 8000 high-quality query–rubric pairs across a four-level domain taxonomy.

- We validate the effectiveness of RUBRICATLAS through extensive evaluations on three open-ended deep research agent benchmarks. Compared with the corresponding baselines, RUBRICATLAS consistently improves agent performance across all metrics, achieving competitive results against leading proprietary deep-research systems.

- Controlled ablation studies analyze the individual contributions of each component, confirming that factually verified rubrics effectively curb verbosity bias, memory context management prevents information distortion over long execution horizons, and decoupled advantage normalization balances multi-objective optimization.

2 Related Work

Deep Research Agents. Research on web agents is expanding from fact-seeking tasks with readily verifiable answers to open-ended investigations that produce long-form reports without a unique reference answer (Zheng et al., 2025; Li et al., 2026c; Han et al., 2025b; Qiao et al., 2025; Shao et al., 2025; Du et al., 2026; Sharma et al., 2026). This transition empowers research agents to assist human decision-making and analytical workflows by broadening the optimization target beyond simple correctness to encompass information coverage, source attribution, multi-source synthesis, and analytical depth (Shao et al., 2024; Li et al., 2026d; Du et al., 2026; Gao et al., 2026; Li et al., 2025b; Han et al., 2025a). Existing studies differ in their emphasis on organizing research workflows versus training agent policies. Work on workflow design explicitly structures the execution process: STORM uses multi-perspective questioning to broaden exploration and construct initial outlines (Shao et al., 2024), whereas WebWeaver interleaves evidence acquisition with outline refinement and grounds section-wise writing in retrieved sources (Li et al., 2026d). Work on policy training develops search and reasoning behaviors through interactive reinforcement learning, as exemplified by DeepResearcher's policy optimization in real-world web environments and Tongyi DeepResearch's scalable agent training pipeline (Zheng et al., 2025; DeepResearch et al., 2025; Li et al., 2025a; 2026c). To scale training to complex research outputs, DR Tulu introduces evolving rubrics for long-form report quality, while QUEST synthesizes diverse tasks using unified rubric trees (Shao et al., 2025; Xie et al., 2026). Despite these advances, current paradigms remain constrained by limited task diversity with a heavy skew toward specialized domains, alongside variable rubric quality, hindering the scalable training of deep research agents.

Rubric-Guided Reinforcement Learning. Evaluating long-form research outputs against explicit, task-specific criteria extends reinforcement learning to open-ended generation (Gunjal et al., 2026; Huang et al., 2025; Li et al., 2026a). A growing body of literature investigates how to construct informative rubrics for policy optimization (Li et al., 2026b; Fu et al., 2026; Ding et al., 2026). Retrieval-grounded approaches derive criteria from agentic search results, evidence trees, or citation chains (Mei et al., 2026; Zhu et al., 2026; Zhang et al., 2026a).

Conversely, contrastive approaches leverage response pairs to isolate uncaptured quality differentials—extracting criteria from preference contrasts (Liu et al., 2026b; Zhang et al., 2026b), policy rollouts (Rezaei et al., 2025; Ding et al., 2026), capability gaps (Zhang et al., 2026c), or dynamic context contrasts (Shao et al., 2025; Li et al., 2026a). However, unverified criteria synthesized during dynamic policy evolution can admit incorrect demands into the reward function, enabling policies to exploit verifier vulnerabilities and inflate rewards (Ren et al., 2026; Zhang et al., 2026b; Mahmoud et al., 2026; Yang et al., 2026). Additionally, multi-dimensional rewards require balanced optimization to prevent scale disparities from dominating policy updates (Tyagi et al., 2026; Li et al., 2026a; Liu et al., 2026a; Ichihara et al., 2026). RubricAtlas addresses both issues by applying a generator–validator debate mechanism to verify factual premises before reward admission and criterion-wise GDPO normalization during policy optimization.

3 Method

In this section, we detail RubricAtlas, whose overall framework is illustrated in Figure 1. We first introduce our data construction pipeline (Section 3.1), which combines taxonomy-guided synthesis for broad query diversity with Decoupled Rubric Discovery and Validation (DRDV) to construct high-quality, factually grounded query–rubric pairs (Section 3.1.2). We then present the Memo Context Manager shared by training and inference (Section 3.2), followed by our training paradigm comprising a format-oriented cold start and rubric-guided reinforcement learning (Section 3.3).

3.1 Data Construction

3.1.1 Taxonomy-Guided Task Synthesis

Recursive Taxonomy Construction. RubricAtlas constructs a hierarchical concept space to make the coverage of synthesized tasks explicit and controllable, as illustrated in Figure 2. Starting from 13 root domains spanning both specialized expert domains and general-interest knowledge,

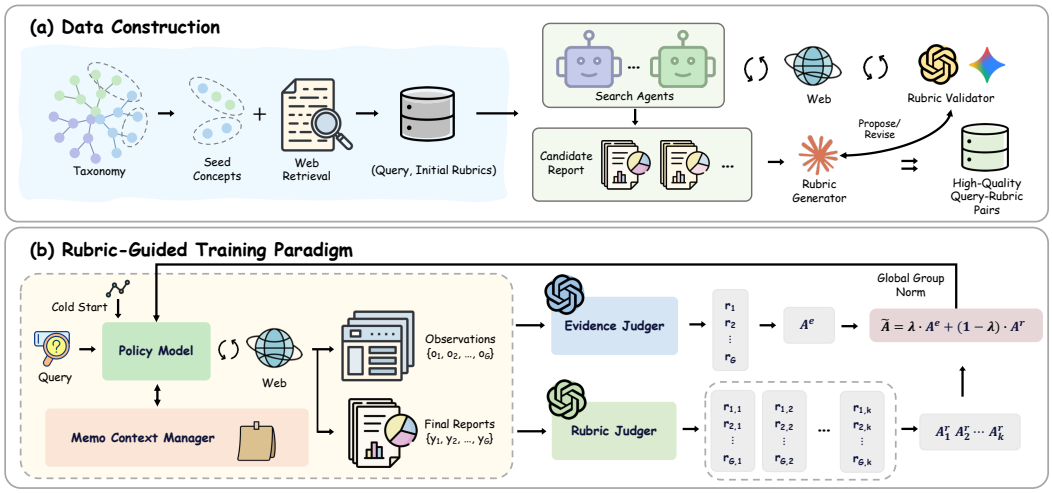

Figure 1: **Overview of the RUBRICATLAS framework.** (a) Data construction. Taxonomy-guided synthesis and DRDV produce verified query–rubric pairs. (b) Rubric-guided training paradigm. A memo-managed policy is optimized by GDPO with rubric and evidence rewards.

an autonomous search agent expands each concept node by issuing targeted web queries, discovering candidate sub-concepts, and consolidating semantically overlapping entries. The taxonomy is recursively refined from coarse-grained domains toward increasingly specific topics, ensuring that child concepts remain consistent with their parent categories while maintaining clear semantic distinctiveness among peer nodes. This recursive process yields a four-tier hierarchy across knowledge domains, where leaf nodes serve as rich semantic seeds for task synthesis, ensuring broad domain coverage and structural diversity in open-ended research queries.

Cross-Concept Task Synthesis. RUBRICATLAS synthesizes research tasks by sampling seed concepts across single topics or distinct taxonomy branches. Combining these concepts with coarse web context establishes a concrete research objective and its analytical dependencies. A search agent then executes fine-grained retrieval to ground the objective in empirical evidence, extracting entities, quantitative data, constraints, and competing claims. Conditioned on these seed concepts, objectives, and fine-grained evidence, an open-ended research query is formulated with necessary user context, analytical deliverables, and evidence requirements. Requiring the analysis of cross-concept interactions and even the reconciliation of conflicting claims across distinct web sources embeds analytical complexity through evidence-dependent reasoning rather than bloated instructions or loosely connected subquestions.

Concurrently, an LLM generates initial factual rubrics with weighted atomic criteria linked to supporting URLs and evidence spans. Since these initial rubrics focus on factual coverage without systematically capturing analytical depth or perspective diversity, the resulting query–rubric pairs proceed to DRDV for further discovery and independent validation.

3.1.2 DECOUPLED RUBRIC DISCOVERY AND VALIDATION

Report-Conditioned Rubric Discovery. Conditioned on q, its initial rubric set $\mathcal{R}_q^0$, and $\mathcal{Y}_q$, a rubric generator contrasts the reports to identify omitted dimensions of factual coverage, evidence grounding, cross-source synthesis, conflict resolution, analytical depth, and perspective diversity. These uncovered dimensions are converted into atomic candidate rubrics with proposed weight tiers that either introduce missing factual constraints or distinguish report quality based on deeper analysis. Such candidate rubrics remain tentative, as report agreement does not guarantee factual validity and generators are easily misled by verbose or self-promotional content.

Agentic Rubric Verification. To enforce factual precision and eliminate subjective bias, candidate rubrics and assigned weights are calibrated by a panel of search-agent validators. Validators systematically reject noisy or spurious criteria, including generic demands for comprehensiveness, artificial rigor, and superficial verbosity. Whether assessing factual claims or analytical perspec-

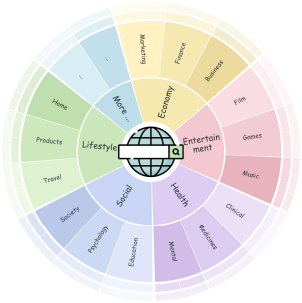

Figure 2: **Taxonomy.** Recursive concept space expansion guided by autonomous search agents.

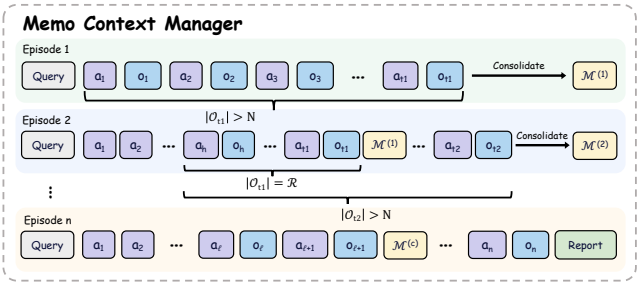

Figure 3: **Memo Context Manager.** Consolidation of long-horizon observations into aspect-indexed entries to manage context budgets.

tives, validators conduct multi-step web investigations to reverse-verify extracted entities, metrics, and source conflicts, ensuring only empirically grounded criteria enter the benchmark.

Iterative Generator–Validator Debate. When premises or weights fail validation, validators return structured feedback and counter-evidence to guide criterion recalibration or pruning. Revised rubrics undergo re-verification, entering the final set $\mathcal{R}_q$ only upon unanimous consensus. This iterative deliberation expands evaluative depth while purging low-quality and hallucinated rubrics. The final set spans source-grounded factual claims and analytical dimensions, using signed weights to incentivize desirable outputs and penalize anti-behaviors.

3.1.3 Data Statistics

The resulting taxonomy contains 345 concept nodes, including 234 leaf concepts, organized under 13 root domains across four tiers. After taxonomy-guided synthesis, DRDV, and quality filtering, we retain 8000 high-quality query–rubric pairs, with an average of 31 fine-grained rubrics per task. Detailed synthesis configurations are provided in Appendix A.

3.2 Long-Horizon Research with Memo Context Manager

Deep research requires prolonged web interactions, accumulating lengthy search histories rich in redundant clutter that distracts the policy and exhausts the context window. Sliding-window strategies discard earlier tool outputs (Tang et al., 2025), making their evidence unavailable in the active context, whereas global summarization (Wu et al., 2025; Lu et al., 2025) can obscure continuity across prior agent actions and omit fine-grained findings and provenance URLs required for factual synthesis and citation-grounded reporting. Consequently, we employ a structured Memo Context Manager throughout training and inference (Figure 3).

At interaction turn t, the model may execute m_t tool actions. For compactness, let $\mathbf{a}_t = (a_{t,1}, \ldots, a_{t,m_t})$ and $\mathbf{o}_t = (o_{t,1}, \ldots, o_{t,m_t})$ denote the action and observation blocks at turn t, respectively. The complete interaction log retains the original action-observation structure $\mathcal{S}_t^{\text{raw}} = (q, \mathbf{a}_1, \mathbf{o}_1, \ldots, \mathbf{a}_t, \mathbf{o}_t)$. The complete log is retained for trajectory recording, whereas the policy operates on a condensed visible sequence $\mathcal{S}_t^{\text{con},(c)}$ after c consolidations. Initially, $\mathcal{S}_t^{\text{con},(0)} = \mathcal{S}_t^{\text{raw}}$ and the memo state $\mathcal{M}^{(0)} = \varnothing$. Prior to the c-th consolidation, the visible sequence is structured as:

$$\mathcal{S}_t^{\text{con},(c-1)} = \Big(q, \mathbf{a}_1, \ldots, \mathbf{a}_{h-1}, \mathbf{a}_h, o_{h,r}, \ldots, \mathbf{a}_u, \mathbf{o}_u, \mathcal{M}^{(c-1)}, \mathbf{a}_{u+1}, \mathbf{o}_{u+1}, \ldots, \mathbf{a}_t, \mathbf{o}_t\Big), \tag{1}$$

where $o_{h,r}$ denotes the earliest visible observation, and $\mathbf{o}_u$ marks the last compression boundary.

Since tool observations dominate the context length, we impose a maximum visible-observation budget N and a retained size R, where $0 < R < N$. Let $\mathcal{O}(\mathcal{S})$ denote the ordered subsequence of tool observations in a sequence $\mathcal{S}$. Once $|\mathcal{O}(\mathcal{S}_t^{\text{con},(c-1)})| > N$, the agent enters a dedicated memo-only turn and performs its c-th consolidation:

$$\mathcal{M}^{(c)} = \text{Consolidate}\Big(\mathcal{M}^{(c-1)}, \mathcal{S}_t^{\text{con},(c-1)}\Big). \tag{2}$$

Each consolidation organizes key findings, intermediate conclusions, and supporting URLs into aspect-indexed entries that can be created, updated, or removed as the investigation evolves. Assuming that the c-th consolidation is completed before interaction turn $t+1$, only after the memo update succeeds do we construct the following reasoning prefix:

$$\mathcal{S}_t^{\mathrm{con},(c)} = \left(q, \mathbf{a}_1, \ldots, \mathbf{a}_{\ell-1}, \mathbf{a}_\ell, o_{\ell,u}, \ldots, o_{\ell,m_\ell}, \mathbf{a}_{\ell+1}, \mathbf{o}_{\ell+1}, \ldots, \mathbf{a}_t, \mathbf{o}_t, \mathcal{M}^{(c)}\right), \tag{3}$$

where $|\mathcal{O}(\mathcal{S}_t^{\mathrm{con},(c)})| = R$ after the c-th consolidation. The sequence preserves every preceding tool action and the original ordering of the most recent R observations.

The resulting context preserves planning continuity, retains fine-grained evidence provenance in the memo, and exposes recent raw observations for subsequent research and final report writing.

3.3 Rubric-Guided Training Paradigm

3.3.1 Format-Oriented Cold Start

From the synthesized task pool, we curate a lightweight subset of high-quality trajectories for supervised cold-start fine-tuning. This phase instills the foundational interaction patterns required for stable long-horizon rollouts, including structured tool invocation and proactive memo consolidation, thereby supporting reliable execution of the research workflow.

3.3.2 Group-Decoupled Rubric Advantage

For each query q, the policy samples a group of G research trajectories $\{\tau_i\}_{i=1}^G$. Directly collapsing multi-aspect rubric scores into a scalar reward before group normalization obscures criterion-level gradient signals and enables dominant dimensions to swamp optimization. To ensure fine-grained credit assignment, we apply GDPO (Liu et al., 2026a) by independently standardizing each rubric dimension prior to advantage aggregation.

Specifically, an automated judge evaluates each trajectory τ_i against K criteria. Each criterion k carries a signed weight $w_k \in \mathbb{R}$, where $|w_k|$ denotes relative criticality and $\mathrm{sgn}(w_k)$ dictates whether the behavior is encouraged or penalized. Based on whether the condition in criterion k is met within the final report, the judge assigns a binary compliance reward $r_{i,k}^{\mathrm{rubric}} \in \{-1, +1\}$. Across the sampled group, GDPO computes criterion-specific advantages $A_{i,k}^{\mathrm{rubric}}$:

$$A_{i,k}^{\mathrm{rubric}} = \frac{r_{i,k}^{\mathrm{rubric}} - \mu_k}{\sigma_k + \epsilon}, \qquad \text{where } \mu_k = \frac{1}{G}\sum_{j=1}^{G} r_{j,k}^{\mathrm{rubric}}, \quad \sigma_k = \sqrt{\frac{1}{G}\sum_{j=1}^{G}(r_{j,k}^{\mathrm{rubric}} - \mu_k)^2}. \tag{4}$$

We aggregate normalized criterion advantages via $A_i^{\mathrm{rubric}} = \sum_{k=1}^{K} \bar{w}_k A_{i,k}^{\mathrm{rubric}}$ with absolute-normalized signed weights $\bar{w}_k = w_k / \sum_{\ell=1}^{K} |w_\ell|$, preserving the directional penalty of anti-behavior rubrics while maintaining advantage scale stability across queries.

However, writing-quality rubrics are optimized more readily than evidence acquisition, inducing a reward shortcut where the policy favors stylistic refinement over proactive retrieval. To counteract this shortcut, we introduce a trajectory-grounded evidence reward constructed over the K_f factual rubrics. For each such criterion k_f, an evidence judge inspects the retrieved observations in τ_i and outputs $e_{i,k_f} \in \{0, 1\}$, indicating whether the search trajectory contains substantive supporting evidence for the criterion. The evidence reward, defined as the proportion of factual rubrics supported by the search trajectory, and its group-normalized advantage are formulated as:

$$r_i^{\mathrm{evidence}} = \frac{1}{K_f}\sum_{k_f=1}^{K_f} e_{i,k_f}, \qquad A_i^{\mathrm{evidence}} = \frac{r_i^{\mathrm{evidence}} - \mu_{\mathrm{ev}}}{\sigma_{\mathrm{ev}} + \epsilon}, \tag{5}$$

To prioritize overall report quality while incorporating explicit retrieval feedback, we combine advantages via $\widetilde{A}_i = \lambda A_i^{\mathrm{rubric}} + (1-\lambda) A_i^{\mathrm{evidence}}$, where $\lambda \in [0, 1]$. To stabilize policy updates against scale variations from aggregating heterogeneous reward dimensions, we renormalize the combined

Table 1: **Main results across deep-research benchmarks.** Bold indicates the best result per category. † and ‡ denote public leaderboard entries and evaluations without the memo tool, respectively.

Model	DeepResearch Bench					DeepConsult			ResearchRubrics
	RACE	Comp.	Insight	Inst.	Read.	Win (%)	Tie (%)	Lose (%)	Compliance (%)
Proprietary Models									
GPT-5.4-High	**48.95**	**47.26**	**47.45**	**51.56**	**51.19**	**78.43**	7.60	**13.97**	62.01
Gemini-3.5-Flash	46.21	44.84	44.72	47.98	48.45	65.20	16.18	18.63	**62.94**
Open-Source Models									
GPT-OSS-120B	36.86	36.86	33.00	39.57	40.27	65.44	2.70	31.86	55.01
Tongyi DeepResearch-30B-A3B‡	40.46†	39.46†	34.44†	46.22†	44.27†	60.40	6.44	33.17	49.73
Qwen3-14B	35.70	33.22	28.77	43.39	41.12	17.40	8.09	74.51	39.21
RubricAtlas-SFT-14B	37.86	36.05	32.91	41.68	44.12	52.21	8.33	39.46	47.29
RubricAtlas-RL-14B	**46.67**	**46.66**	**42.52**	**49.81**	**50.04**	**82.60**	4.93	**12.47**	**62.15**

advantage across training batch $\mathcal{B}$ via $A_i = (\widetilde{A}_i - \mu_{\mathcal{B}})/(\sigma_{\mathcal{B}} + \epsilon)$, where $\mu_{\mathcal{B}}$ and $\sigma_{\mathcal{B}}$ denote the batch mean and standard deviation of $\widetilde{A}_i$.

Using $\rho_{i,t}$ to denote the token-level importance ratio between the current and rollout policies, we optimize the resulting advantage with the clipped group-relative objective:

$$\mathcal{J}_{\text{GDPO}}(\theta) = \mathbb{E}\left[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|\tau_i|}\sum_{t=1}^{|\tau_i|}\min(\rho_{i,t}A_i,\ \text{clip}(\rho_{i,t}, 1-\epsilon_c, 1+\epsilon_c)A_i)\right]. \tag{6}$$

4 Experiments

4.1 Experimental Setup

Training Data and Models. Our training comprises a cold-start stage followed by reinforcement learning. From the 8000 synthesized query–rubric pairs, we use 1000 queries for which GPT-OSS-120B (Agarwal et al., 2025) generates high-quality research trajectories for cold-start supervised fine-tuning. The remaining 7000 query–rubric pairs are used for rubric-guided reinforcement learning. All trainable checkpoints use Qwen3-14B (Yang et al., 2025) as the backbone.

Training and Inference Setups. We conduct cold-start supervised fine-tuning using ms-swift (Zhao et al., 2025) and reinforcement learning using verl (Sheng et al., 2025), initializing the latter from the cold-start checkpoint. Unless otherwise specified, we set the rubric–evidence balance to $\lambda = 0.8$, assigning weights of 0.8 and 0.2 to the rubric and trajectory-grounded evidence advantages, respectively. For controlled comparisons, locally executed models otherwise share the same ReAct agent scaffold (Yao et al., 2022) and tools, including memo management, web search, webpage retrieval, scholarly search, and Python execution. They also share the same tool-call budget, maximum report length, and evaluation protocol. Detailed training hyperparameters and compute configurations are provided in Appendix B.

Benchmarks and Models. We evaluate on three benchmarks covering expert-authored research questions, consulting and business scenarios, and broadly scoped open-ended prompts: DeepResearch Bench (Du et al., 2026) reports overall RACE and its four dimensions, DeepConsult (DeepConsult, 2025) reports pairwise win, tie, and lose rates against the benchmark reference, ResearchRubrics (Sharma et al., 2026) reports binary rubric compliance. We follow each benchmark's official report-evaluation protocol; detailed benchmark statistics, judge configurations, and evaluation dimensions are provided in Appendix C. Our main comparison includes two frontier proprietary models, GPT-5.4 with high reasoning effort (OpenAI, 2026) and Gemini-3.5-Flash (DeepMind, 2026); the large open-source GPT-OSS-120B model; the specialized deep-research agent Tongyi DeepResearch-30B-A3B (DeepResearch et al., 2025); and the training backbone Qwen3-14B. We evaluate two checkpoints of our method: RubricAtlas-SFT-14B, obtained after format-oriented cold-start, and RubricAtlas-RL-14B, further optimized with GDPO.

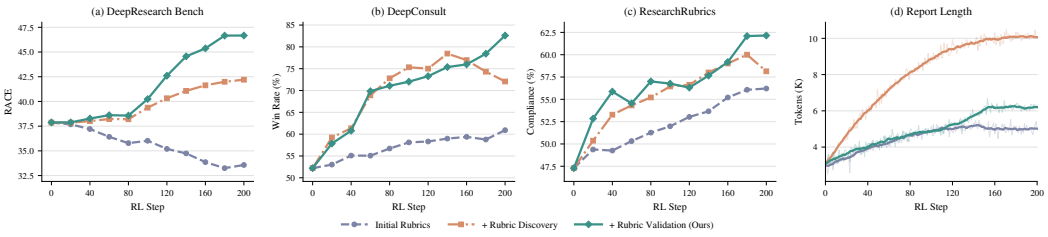

Figure 4: Downstream performance and output length across RL training steps for three rubric-construction variants across DeepResearch Bench, DeepConsult, and ResearchRubrics.

4.2 Main Results

As shown in Table 1, RubricAtlas-RL-14B achieves the strongest performance among open-source models across all three benchmarks. It consistently surpasses general and specialized baselines, including the substantially larger GPT-OSS-120B and search-optimized Tongyi DeepResearch. Tongyi DeepResearch produces comparatively brief reports with limited fine-grained analysis, as it is optimized primarily for verifiable QA tasks. GPT-OSS-120B exhibits format drift and incomplete deliverables, with protracted reasoning traces destabilizing long-horizon execution. In contrast, RubricAtlas-RL-14B reliably follows requested structures, completes deliverables, and sustains fine-grained multi-perspective analysis.

RubricAtlas-RL-14B is also competitive with frontier proprietary models. The model excels at structuring broad research questions, quantitatively synthesizing multi-source evidence, and translating insights into actionable recommendations within coherent, table-supported reports. These strengths are most pronounced on DeepConsult, where it achieves the highest win rate. On DeepResearch Bench and ResearchRubrics, it slightly trails leading proprietary systems in analytical insight and rubric compliance, respectively, while remaining broadly competitive overall.

The progression from Qwen3-14B through SFT to RL demonstrates a clear staged learning trajectory. Cold-start SFT equips the model with basic long-horizon workflows and active memo management, improving stability and report completion. However, imitation learning alone remains insufficient for thorough evidence acquisition and analytical rigor. Subsequent rubric-guided RL drives proactive exploration and evidence grounding, yielding consistent gains across all three benchmarks.

4.3 Effectiveness of Decoupled Rubric Discovery and Validation

To isolate the individual contributions of report-conditioned discovery and independent validation, we compare three rubric-construction variants initialized from the same cold-start checkpoint and optimized under an identical reinforcement learning budget: (1) *Initial Rubrics*, using only source-grounded factual criteria from task synthesis ($\approx$ 14 rubrics per query), (2) *Rubric Discovery*, augments initial sets via unvalidated report-conditioned generation ($\approx$ 39 rubrics per query), (3) *Rubric Validation* (ours), filtering unverified criteria to retain $\approx$ 31 validated rubrics per query.

Figure 4 tracks downstream performance dynamics and output verbosity across RL steps. Training solely with initial rubrics yields marginal gains in win rate and compliance while RACE performance steadily degrades. Unfiltered rubric discovery substantially boosts performance metrics in early training steps, confirming the utility of analytical and perspective-diverse criteria beyond basic factual coverage, yet it ultimately constrains peak performance during later optimization. Moreover, unfiltered discovery induces severe verbosity explosion, causing report lengths to breach generation budgets and degrade synthesis quality. In contrast, DRDV prevents reward overfitting, achieving superior accuracy across all three benchmarks while maintaining concise, high-density outputs.

4.4 Effectiveness of the Memo Context Manager

To evaluate the Memo Context Manager, we conduct ablation experiments across GPT-5.4, GPT-OSS-120B, and RubricAtlas-RL-14B under identical prompt templates, tool suites, and context budgets. As detailed in Table 2, standard sliding windows suffer from context eviction, forcing models into redundant search loops to recover forgotten evidence. By persistently consolidating structured

Table 2: Effect of the Memo Context Manager. DRB, DC, and RR denote DeepResearch Bench RACE, DeepConsult win rate, and ResearchRubrics compliance, respectively.

Model	Memo	Tool Calls ↓	Citation ↑	DRB ↑	DC ↑	RR ↑
GPT-5.4	w/o	60.6	2.27	46.81	77.70	61.31
	w/	**21.3**	**9.09**	**48.95**	**78.43**	**62.01**
GPT-OSS-120B	w/o	35.8	13.47	36.39	61.27	53.77
	w/	**22.5**	**17.26**	**36.77**	**65.44**	**55.01**
RubricAtlas-RL-14B	w/o	16.3	12.13	43.82	67.88	58.29
	w/	**15.6**	**22.94**	**46.67**	**82.60**	**62.15**

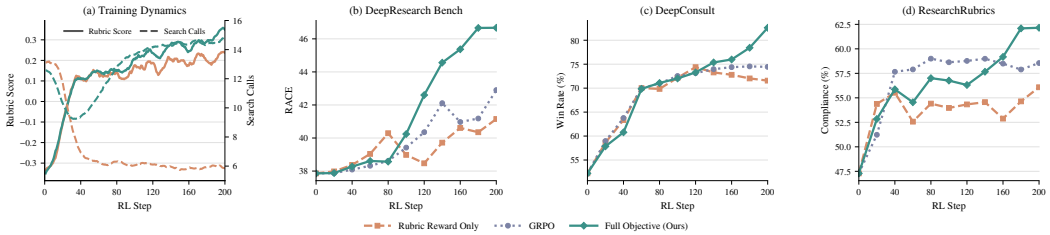

Figure 5: **Optimization ablations.** (a) Training dynamics and (b–d) downstream benchmark performance across optimization variants.

findings and provenance prior to eviction, the memo manager mitigates this inefficiency, drastically reducing tool overhead while elevating evidence citation density across all three architectures. The sharp reduction in tool calls for general-purpose LLMs confirms that persistent memory effectively counters context truncation, whereas the performance gains for RubricAtlas-RL-14B demonstrate its synergistic compatibility with rubric-guided policy optimization.

4.5 Effectiveness of the Trajectory-Grounded Evidence Reward

The rubric-only variant sets $\lambda = 1$, whereas the full objective uses $\lambda = 0.8$ together with the trajectory-grounded evidence reward described in Section 3.3. Figure 5(a) shows that rubric-only optimization improves rubric score while suppressing search, revealing a shortcut toward writing-level compliance without sufficient retrieval. Trajectory-grounded supervision instead sustains search activity, while the corresponding downstream curves in Figure 5(b–d) continue improving after the rubric-only variant plateaus.

4.6 GDPO versus GRPO

We compare GDPO with aggregate-reward GRPO across all three benchmarks. As shown in Figure 5(b–d), GRPO improves early but later plateaus or regresses, whereas GDPO continues improving. This result indicates that criterion-wise normalization better preserves heterogeneous reward signals during later optimization.

5 Conclusion

We introduced RubricAtlas, a unified framework that combines taxonomy-guided task synthesis, independently validated rubric discovery, long-horizon context management, and fine-grained rubric-guided optimization for deep research agents. Built on Qwen3-14B, RubricAtlas achieves the strongest open-source results across three deep-research benchmarks, surpassing proprietary systems on DeepConsult while remaining competitive on DeepResearch Bench and ResearchRubrics. Controlled ablations further demonstrate the complementary benefits of DRDV, memo management, trajectory-grounded evidence rewards, and GDPO. These results highlight reliable, fine-grained rubric supervision as an effective foundation for improving evidence-grounded deep research.

AI USE STATEMENT

Generative AI tools were used solely for language editing and polishing. The authors reviewed and verified all technical claims, citations, experimental results, and released artifacts, and take full responsibility for the final content of this paper.

ETHICS STATEMENT

Our pipeline retrieves public web content to construct research tasks and evaluation rubrics. A release must respect source licenses, robots directives, privacy requirements, and content-removal requests. We will exclude private or sensitive personal information, document the temporal and geographic biases of the retrieved sources, and release page-derived evidence only when redistribution is permitted. Because synthetic rubrics can encode source and model biases, we treat automatic judgments as imperfect supervision and include independent human auditing.

REFERENCES

Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K Arora, Yu Bai, Bowen Baker, Haiming Bao, et al. gpt-oss-120b & gpt-oss-20b model card. *arXiv preprint arXiv:2508.10925*, 2025.

Anthropic. Introducing Claude Opus 4.6, February 2026. URL https://www.anthropic.com/news/claude-opus-4-6.

DeepConsult. Deepconsult: A deep research benchmark for consulting and business queries, 2025. URL https://github.com/Su-Sea/ydc-deep-research-evals.

DeepMind. Gemini 3.5 flash model card, 2026. URL https://deepmind.google/models/model-cards/gemini-3-5-flash.

Tongyi DeepResearch, 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*, 2025.

Hongxin Ding, Baixiang Huang, Yue Fang, Weibin Liao, Zheng Li, Jinyang Zhang, Zhijing Wu, Junfeng Zhao, and Yasha Wang. Evorubrics: Dynamic rubrics as rewards via adversarial co-evolution for llm reinforcement learning. *arXiv preprint arXiv:2606.23038*, 2026.

Mingxuan Du, Benfeng Xu, Chiwei Zhu, Licheng Zhang, Xiaorui Wang, and Zhendong Mao. Deep-research bench: A comprehensive benchmark for deep research agents. In *Proceedings of the International Conference on Learning Representations*, pp. 42414–42448, 2026.

Dazhi Fu, Jiuding Yang, Yiwen Guo, and Jicong Fan. Many voices, one reward: Multi-role rubric generation for llm judging and reward modeling. *arXiv preprint arXiv:2607.01830*, 2026.

Yiwen Gao, Ruochen Zhao, Yang Deng, and Wenxuan Zhang. Dr-arena: an automated evaluation framework for deep research agents. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics*, pp. 27130–27152, 2026.

Anisha Gunjal, Anthony Wang, Elaine Lau, Vaskar Nath, Yunzhong He, Bing Liu, and Sean Hendryx. Rubrics as rewards: Reinforcement learning beyond verifiable domains. In *Proceedings of the International Conference on Learning Representations*, volume 2026, pp. 127924–127945, 2026.

Janghoon Han, Heegyu Kim, Changho Lee, Dahm Lee, Min Hyung Park, Hosung Song, Stanley Jungkyu Choi, Moontae Lee, and Honglak Lee. Deer: A benchmark for evaluating deep research agents on expert report generation. *arXiv preprint arXiv:2512.17776*, 2025a.

Rujun Han, Yanfei Chen, Zoey CuiZhu, Lesly Miculicich, Guan Sun, Yuanjun Bi, Weiming Wen, Hui Wan, Chunfeng Wen, Solène Maître, et al. Deep researcher with test-time diffusion. *arXiv preprint arXiv:2507.16075*, 2025b.

Zenan Huang, Yihong Zhuang, Guoshan Lu, Zeyu Qin, Haokai Xu, Tianyu Zhao, Ru Peng, Jiaqi Hu, Zhanming Shen, Xiaomeng Hu, et al. Reinforcement learning with rubric anchors. *arXiv preprint arXiv:2508.12790*, 2025.

Yuki Ichihara, Yuu Jinnai, Tetsuro Morimura, Mitsuki Sakamoto, Ryota Mitsuhashi, and Eiji Uchibe. Mo-grpo: Mitigating reward hacking of group relative policy optimization on multi-objective problems. *Transactions of the Association for Computational Linguistics*, 14:1867–1882, 2026.

Akira Kawabata and Saku Sugawara. C2: Scalable rubric-augmented reward modeling from binary preferences. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics*, pp. 11397–11424, 2026.

Gaotang Li, Bhavana Dalvi Mishra, Zifeng Wang, Jun Yan, Yanfei Chen, Chun-Liang Li, Long T Le, Rujun Han, George Lee, Hanghang Tong, et al. Rubricem: Meta-rl with rubric-guided policy decomposition beyond verifiable rewards. *arXiv preprint arXiv:2605.10899*, 2026a.

Kuan Li, Zhongwang Zhang, Huifeng Yin, Liwen Zhang, Litu Ou, Jialong Wu, Wenbiao Yin, Baixuan Li, Zhengwei Tao, Xinyu Wang, et al. Websailor: Navigating super-human reasoning for web agent. *arXiv preprint arXiv:2507.02592*, 2025a.

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

Sunzhu Li, Jiale Zhao, Huimin Ren, Zhenlin Wei, Yang Zhou, Jingwen Yang, Shunyu Liu, Kaike Zhang, and Chen Wei. Rubrichub: A comprehensive and highly discriminative rubric dataset via automated coarse-to-fine generation. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 31320–31344, 2026b.

Xiaoxi Li, Jiajie Jin, Guanting Dong, Hongjin Qian, Yongkang Wu, Ji-Rong Wen, Yutao Zhu, and Zhicheng Dou. Webthinker: Empowering large reasoning models with deep research capability. In *Proceedings of the Advances in Neural Information Processing Systems*, pp. 120091–120131, 2026c.

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 *Proceedings of the International Conference on Learning Representations*, pp. 4173–4217, 2026d.

Shih-Yang Liu, Xin Dong, Ximing Lu, Shizhe Diao, Peter Belcak, Mingjie Liu, Min-Hung Chen, Hongxu Yin, Yu-Chiang Frank Wang, Kwang-Ting Cheng, et al. Gdpo: Group reward-decoupled normalization policy optimization for multi-reward rl optimization. *arXiv preprint arXiv:2601.05242*, 2026a.

Tianci Liu, Ran Xu, Tony Yu, Ilgee Hong, Carl Yang, Tuo Zhao, and Haoyu Wang. Openrubrics: Towards scalable synthetic rubric generation for reward modeling and llm alignment. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics*, pp. 17417–17437, 2026b.

Miao Lu, Weiwei Sun, Weihua Du, Zhan Ling, Xuesong Yao, Kang Liu, and Jiecao Chen. Scaling llm multi-turn rl with end-to-end summarization-based context management. *arXiv preprint arXiv:2510.06727*, 2025.

Anas Mahmoud, MohammadHossein Rezaei, Zihao Wang, Anisha Gunjal, Bing Liu, and Yunzhong He. Reward hacking in rubric-based reinforcement learning. *arXiv preprint arXiv:2605.12474*, 2026.

Wangyi Mei, Zhouhong Gu, Zhenhan Bai, Yin Cai, Lefan Zhang, Zhenxin Ding, Bo Chen, Yan Gao, Yi Wu, Yao Hu, et al. Deep research as rubric for reinforcement learning. *arXiv preprint arXiv:2606.01091*, 2026.

OpenAI. OpenAI o3 and o4-mini System Card, April 2025. URL `https://openai.com/index/o3-o4-mini-system-card/`.

OpenAI. Gpt-5.4 thinking system card, 2026. URL https://deploymentsafety.openai.com/gpt-5-4-thinking.

Zile Qiao, Guoxin Chen, Xuanzhong Chen, Donglei Yu, Wenbiao Yin, Xinyu Wang, Zhen Zhang, Baixuan Li, Huifeng Yin, Kuan Li, et al. Webresearcher: Unleashing unbounded reasoning capability in long-horizon agents. *arXiv preprint arXiv:2509.13309*, 2025.

Qingnan Ren, Zhen Fang, Shiting Huang, Yu Zeng, Lin Chen, Zehui Chen, and Feng Zhao. Unbiased principles, robust rewards. In *Proceedings of the International Conference on Machine Learning*, 2026.

MohammadHossein Rezaei, Robert Vacareanu, Zihao Wang, Clinton Wang, Bing Liu, Yunzhong He, and Afra Feyza Akyürek. Online rubrics elicitation from pairwise comparisons. *arXiv preprint arXiv:2510.07284*, 2025.

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.

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

Manasi Sharma, Chen Bo Calvin Zhang, Chaithanya Bandi, Clinton Wang, Ankit Aich, Huy Nghiem, Tahseen Rabbani, Ye Htet, Brian Jang, Sumana Basu, et al. Researchrubrics: A benchmark of prompts and rubrics for evaluating deep research agents. In *Proceedings of the International Conference on Learning Representations*, pp. 90447–90472, 2026.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. In *Proceedings of the Twentieth European Conference on Computer Systems*, pp. 1279–1297, 2025.

Qiaoyu Tang, Hao Xiang, Le Yu, Bowen Yu, Yaojie Lu, Xianpei Han, Le Sun, WenJuan Zhang, Pengbo Wang, Shixuan Liu, et al. Beyond turn limits: Training deep search agents with dynamic context window. *arXiv preprint arXiv:2510.08276*, 2025.

Utkarsh Tyagi, Xingang Guo, MohammadHossein Rezaei, Daniel George, Anas Mahmoud, Jackson Lee, Bing Liu, and Yunzhong He. Not every rubric teaches equally: Policy-aware rubric rewards for rlvr. *arXiv preprint arXiv:2605.20164*, 2026.

Xixi Wu, Kuan Li, Yida Zhao, Liwen Zhang, Litu Ou, Huifeng Yin, Zhongwang Zhang, Xinmiao Yu, Dingchu Zhang, Yong Jiang, et al. Resum: Unlocking long-horizon search intelligence via context summarization. *arXiv preprint arXiv:2509.13313*, 2025.

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

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Minglai Yang, Xinyu Guo, Utkarsh Tyagi, Mian Zhang, Razvan Dumitru, Sunjie Hou, Yunzhong He, Daniel Yue Zhang, and Ying Liu. Rubric dropout: A simple way to mitigate reward hacking in rubric-as-reward rl. *arXiv preprint arXiv:2608.11669*, 2026.

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.

Jiajie Zhang, Xin Lv, Ling Feng, Lei Hou, and Juanzi Li. Chaining the evidence: Robust reinforcement learning for deep search agents with citation-aware rubric rewards. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics*, pp. 20744–20765, 2026a.

Junkai Zhang, Zihao Wang, Lin Gui, Swarnashree Mysore Sathyendra, Jaehwan Jeong, Victor Veitch, Wei Wang, Yunzhong He, Bing Liu, and Lifeng Jin. Chasing the tail: Effective rubric-based reward modeling for large language model post-training. In *Proceedings of the International Conference on Learning Representations*, pp. 133430–133457, 2026b.

Rongzhi Zhang, Rui Feng, Zhihan Zhang, Jingfeng Yang, Qingyu Yin, Xin Liu, Zixuan Zhang, Priyanka Nigam, Bing Yin, Tuo Zhao, et al. Qubric: Co-designing queries and rubrics for rl beyond verifiable rewards. *arXiv preprint arXiv:2606.03968*, 2026c.

Yuze Zhao, Jintao Huang, Jinghan Hu, Xingjun Wang, Yunlin Mao, Daoze Zhang, Zeyinzi Jiang, Zhikai Wu, Baole Ai, Ang Wang, et al. Swift: a scalable lightweight infrastructure for fine-tuning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 29733–29735, 2025.

Yuxiang Zheng, Dayuan Fu, Xiangkun Hu, Xiaojie Cai, Lyumanshan Ye, Pengrui Lu, and Pengfei Liu. Deepresearcher: Scaling deep research via reinforcement learning in real-world environments. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 414–431, 2025.

Minghang Zhu, Chuyang Wei, Junhao Xu, Yilin Cheng, Zhumin Chen, and Jiyan He. Deeprubric: Evidence-tree rubric supervision for efficient reinforcement learning of deep research agents. *arXiv preprint arXiv:2606.17029*, 2026.

A Data Construction Details

This section provides the prompts and model configurations used throughout taxonomy construction, concept search, task and initial-rubric synthesis, and DRDV.

A.1 Taxonomy and Concept Discovery

The taxonomy contains 345 concept nodes, including 234 leaf concepts, organized into four tiers under 13 root domains. We provide the complete recursive taxonomy-expansion prompt, including instructions for child-concept generation, semantic-distinctiveness constraints, and stopping criteria.

We use Claude Opus 4.6 (Anthropic, 2026) for web-grounded concept discovery and recursive graph construction, and GPT-5.4 to deduplicate proposed concepts against existing graph nodes.

Expand Node Prompt

You are building a source-grounded concept graph. Research the supplied concept and decide whether it should be expanded into more specific concepts.

Concept case: ${case_json}
Limits: ${limits_json}

Use web search before answering. Investigate the concept's scope, common subdivisions, specialist terminology, and possible boundary cases. Read the underlying pages rather than relying on search snippets.

A proposed concept should:
- be narrower than the current concept;
- be a direct and useful refinement, not a remote descendant;
- have a conventional name recognized by specialists;
- be distinct from aliases and nearby graph nodes;
- be supported by the sources you read.

Do not propose examples, named instances, procedural steps, applications, or concepts that merely co-occur with the current concept. Keep the proposals at a similar granularity and prefer a coherent set over a long miscellaneous list.

Use decision: "expand" when valid refinements exist. Use "stop" when further subdivision would only produce examples or low-level details. Use "review" when the evidence is insufficient, conflicting, or the supplied graph placement appears wrong. Failure to find results is not by itself a reason to stop.

Return only one valid JSON object:

```
{
  "decision": "expand|stop|review",
  "reason": "brief explanation of the decision",
  "proposals": [
    {
      "name": "canonical concept name",
      "scope": "what this concept covers",
      "why": "why it is a direct refinement",
      "sources": ["https://..."]
    }
  ],
  "connections": [
    {
      "name": "associated concept that is not a refinement",
      "why": "how it is connected",
```

```
      "sources": ["https://..."]
    }
  ]
}
```

Requirements:
- `expand` requires at least one proposal.
- `stop` and `review` require an empty `proposals` list.
- Every proposal must cite at least one page that supports its placement.
- `connections` is optional in substance: use an empty list when none are useful, and never include aliases, ancestors, or accepted proposals.
- Respect the proposal limit supplied in `limits_json`.
- Write names and explanations in the language requested by `case_json`, preserving established technical terms where appropriate.
- Output JSON only, without comments, Markdown fences, or extra fields.

Dedup Node Prompt

Please determine whether the proposed concept is already represented by one of the existing graph nodes below.

Proposed concept: ${proposed_node_json}
Potential matches: ${candidate_nodes_json}
Relevant graph context: ${graph_context_json}

Compare the concepts by meaning, scope, and granularity rather than name similarity alone.

Treat two records as the same concept only when they have essentially the same meaning and granularity. Synonyms, translations, acronyms, spelling variants, and trivial rephrasings may be the same concept.

Do not merge concepts when one is broader or narrower, when their scopes only overlap, when they are merely associated, or when similar names have different meanings. A shared domain or parent is not evidence of identity. If the same concept appears under a different graph branch, reuse the existing node rather than creating a duplicate.

Choose:
- `reuse`: exactly one existing node is the same concept;
- `create`: no existing node is the same concept;
- `review`: the information is insufficient or multiple nodes appear to match;
- `reject`: the proposal is not a coherent, nameable concept.

Return only one valid JSON object:

```
{
  "decision": "reuse|create|review|reject",
  "match_id": null,
  "reason": "brief semantic justification"
}
```

For `reuse`, replace `match_id` with the matching node ID. For every other decision, keep it as JSON `null`. Output JSON only, without comments, Markdown fences, or extra fields.

A.2 Task and Source-Grounded Rubric Synthesis

We use GPT-5.4 to conduct fine-grained web retrieval and collect a broad set of source materials. Conditioned on the sampled concepts and retrieved evidence, Claude Opus 4.6 synthesizes open-ended research queries and initial factual rubrics.

Search Materials Prompt

Please collect source material that can support a research task involving the following seed concept or concepts.

Seed concepts: ${concepts_json}
Existing concept context: ${concept_context_json}

The input may contain one seed or multiple seeds. For one seed, identify concrete entities that enable a focused, in-depth task within that concept. For multiple seeds, identify entities or situations through which the concepts can be studied together. Useful entities include organizations, methods, systems, datasets, policies, products, studies, locations, or time periods. Search every seed individually and, when there are multiple seeds, search their meaningful combinations as well.

Prioritize material containing specific facts, measurements, methods, assumptions, operational constraints, comparisons, or competing claims. Avoid collecting many generic definitions or pages that merely mention the concepts together.

Each retained item must contain a clear claim, a short evidence span from the page, and its source URL. Preserve exact quantities, units, dates, entities, and conditions. Use separate items for claims that disagree so the conflict remains visible.

Return only one valid JSON object:

```
{
  "decision": "ready|insufficient",
  "entities": [
    {
      "name": "specific entity",
      "role": "why this entity is useful for connecting the seed
concepts"
    }
  ],
  "materials": [
    {
      "id": "M1",
      "type": "fact|quantity|method|assumption|constraint|comparison
|competing_claim",
      "claim": "one concrete claim",
      "evidence": "short supporting passage from the page",
      "url": "https://...",
      "concepts": ["seed concept supported by this item"]
    }
  ],
  "gaps": ["missing evidence or unresolved search direction"]
}
```

Use `ready` only when the material contains enough concrete, source-diverse evidence to formulate a nontrivial research task. With multiple seeds, the material must also establish a substantive connection among them. Otherwise use `insufficient` and explain the missing evidence in `gaps`. Output JSON only, without comments, Markdown fences, or extra fields.

Synthesize Query Prompt

Please decide whether the collected material can support a high-quality open-ended research task and, if so, write the task.

Seed concepts: ${concepts_json}
Collected material: ${materials_json}
Task requirements: ${query_requirements_json}

Accept the material only when it supports:
- a concrete research objective rather than a broad topic;
- focused analytical depth for one seed, or genuine interaction among multiple seeds;
- conclusions that require combining multiple sources;
- specific analytical deliverables grounded in the supplied evidence;
- a task that cannot be answered by copying one page or listing disconnected facts.

For multiple seeds, reject the material when the concepts only co-occur without a meaningful analytical relationship. For any number of seeds, reject when the evidence is too thin, the task would be a simple lookup, or the requested analysis would require unsupported assumptions.

When accepted, write one self-contained user query. Include only the context, deliverables, and evidence expectations needed to define the task. Do not mention internal material IDs, the taxonomy, or the synthesis process. Do not reveal the expected answer in the question.

Return only one valid JSON object:

```
{
  "decision": "accept|reject",
  "reason":  "brief explanation of why the material does or does not work",
  "objective": "the central research objective, or null",
  "query": "the final user-facing research query, or null",
  "material_ids": ["M1"]
}
```

For `reject`, set `objective` and `query` to JSON `null`, and list no `material_ids`. For `accept`, include only material IDs that are genuinely needed by the task. Output JSON only, without comments, Markdown fences, or extra fields.

Initial Rubric Prompt

Please create the initial factual rubrics for the research query using only the supplied source material.

Research query: ${query_text}
Source material: ${materials_json}
Rubric settings: ${rubric_settings_json}

Each rubric must describe one factual requirement that a strong answer should satisfy. It must be relevant to the query, independently judgeable, and directly supported by the supplied evidence.

Do not create criteria for writing style, generic comprehensiveness, verbosity, or unsupported analytical preferences. Do not combine several requirements into one criterion. Do not reward facts merely because they appear in the material; include them only when they help answer the query.

When sources disagree, write a criterion requiring the answer to identify or reconcile the disagreement rather than treating one disputed claim as settled. Assign weights according to analytical importance using the weight scheme in the settings: central requirements should receive more weight than supporting context.

Return only one valid JSON object:

```
{
  "decision": "ready|review",
  "reason": "brief explanation of the result",
  "rubrics": [
    {
      "criterion": "one atomic, factually testable requirement",
      "weight": 1,
      "sources": [
        {
          "url": "https://...",
          "evidence": "supporting evidence span"
        }
      ]
    }
  ]
}
```

Use `review` with an empty `rubrics` list when the material does not adequately support factual evaluation of the query. Follow the rubric count and allowed weights in `rubric_settings_json`. Output JSON only, without comments, Markdown fences, or extra fields.

A.3 DRDV Configuration

For each query, candidate reports are independently generated by o3 (OpenAI, 2025), GPT-5.4, and Gemini-3.5-Flash. Claude Opus 4.6 serves as the rubric generator, while GPT-5.4 and Gemini-3.5-Flash serve as independent search-agent validators.

A.4 Rubric-Generator Prompt

We include the complete prompt for contrasting candidate reports, identifying dimensions omitted by the initial rubric set, atomizing newly discovered criteria, and proposing signed weight tiers.

Rubric-Generator Prompt

Please discover task-specific rubrics that are missing from the initial rubric set.

Research query: ${query_text}
Initial source-grounded rubrics: ${initial_rubrics_json}
Candidate research reports: ${candidate_reports_json}
Rubrics already accepted in earlier debate rounds: ${accepted_rubrics_json}
Validator feedback from the previous round: ${validator_feedback_json}

Use the reports as a comparison panel. Examine where their quality diverges and turn the meaningful differences into new rubric candidates. Compare them pairwise as well as against the initial rubrics.

Search broadly across these dimensions:
- factual coverage: entities, quantities, events, constraints, assumptions, methods, or relationships covered by stronger reports and missed or mishandled by others;
- perspective diversity: stakeholders, competing explanations, alternative interpretations, trade-offs, counterarguments, and implications;
- evidence use: source quality, citation support, cross-source integration, treatment of uncertainty, and reconciliation of conflicting claims;
- analytical quality: comparison, causal reasoning, methodological scrutiny, limitations,

boundary conditions, and synthesis into a supported conclusion;
- writing quality: organization, precision, clarity, coherence, and whether presentation choices make the requested analysis easier or harder to follow;
- report-specific strengths and failures that reveal additional ways to distinguish a strong answer from a weak one.

To increase diversity, inspect the reports through complementary lenses when they are relevant to the query:
- scale: macro-level systems, policies, markets, or social effects; meso-level organizations, communities, and institutions; micro-level actors, mechanisms, components, or decisions;
- time: historical causes, current conditions, short-term effects, long-term consequences, and future scenarios;
- stakeholder: different affected groups, decision-makers, implementers, beneficiaries, and critics;
- context: geography, industry, culture, regulation, operating environment, or deployment setting;
- analytical frame: technical, economic, political, social, ethical, scientific, legal, operational, and environmental perspectives;
- reasoning mode: description, comparison, causal explanation, mechanism analysis, risk assessment, trade-off analysis, and recommendation.

These lenses are discovery directions rather than a checklist. Convert a lens into a rubric only when the reports reveal a concrete and task-relevant quality difference.

Treat every discovered fact, perspective, and quality dimension as a candidate for the validators to verify in the next stage. Aim for broad discovery while keeping each proposed rubric:
- tied to a concrete difference observed across the reports;
- specific to the research query;
- focused on one independently judgeable requirement;
- observable in the final answer;
- distinct from the initial and already accepted rubrics.

Positive rubrics describe content or reasoning that should be present. Negative rubrics describe a concrete harmful behavior that should be penalized; phrase them so that the criterion is satisfied exactly when that behavior appears. For example, use "The answer states X without evidence", rather than "The answer should avoid stating X without evidence".

Assign one integer weight:
- `+8` to `+10`: necessary point – omitting or getting it wrong directly undermines task completion;
- `+4` to `+7`: important point – supports a core conclusion, key reasoning step, evidence requirement, or essential explanation;
- `+1` to `+3`: additional point – useful but nonessential context, extension, recommendation, or supporting detail;
- `−1` to `−10`: pitfall – an observable error or harmful behavior, with larger magnitude for more severe damage to correctness or task completion.

Weight importance, not novelty, length, or frequency across reports.

When validator feedback is nonempty, return the candidates still under consideration. A revised candidate keeps its original ID; a newly discovered candidate receives a fresh ID. Apply the feedback to both criterion wording and weight. Rubrics already accepted in earlier rounds remain final and are not part of this response.

Return only one valid JSON object:

```
{
  "rubrics": [
```

```
    {
      "id": "R1",
      "criterion": "one atomic and judgeable requirement",
      "weight": 8,
      "rationale": "the report contrast that motivated this
rubric and why it matters"
    }
  ]
}
```

An empty list is valid when no defensible rubric is missing. Respect the count limit in `generation_settings_json`. Output JSON only, without comments, Markdown fences, or extra fields.

A.5 Search-Agent Validator Prompt

We include the complete independent-validation prompt, web-search instructions, evidence-return schema, criterion-level verdict format, and weight-calibration rules.

Search-Agent Validator Prompt

Please independently validate the proposed rubric and its weight.

Research query: ${query_text}
Proposed rubric: ${rubric_json}
Existing and already accepted rubrics: ${existing_rubrics_json}
Previously collected source material: ${source_materials_json}

Use web search to verify the rubric rather than trusting the generator, candidate reports, or supplied source material. Try to falsify the rubric before accepting it.

Check all of the following:
1. Relevance: the rubric materially affects fulfillment of the research query.
2. Grounding: factual claims, named entities, quantities, methods, assumptions, or source conflicts in the rubric are corroborated by credible evidence.
3. Objectivity: a judge can decide whether the final answer satisfies the rubric without relying on taste or hidden reasoning.
4. Atomicity: the rubric contains one requirement rather than several independently satisfiable requirements.
5. Distinctness: it does not duplicate, paraphrase, or merely negate an existing rubric.
6. Substantiveness: it rewards or penalizes meaningful content or reasoning, not verbosity, formatting, confidence, or generic writing quality.
7. Weight: the proposed weight matches the consequence of satisfying or violating the criterion.

For a high-level analytical criterion, verify that the perspective is genuinely warranted by the query and external evidence. For a conflict-resolution criterion, verify that the underlying source conflict actually exists. For a negative rubric, verify that the behavior is concrete, observable, and harmful; calibrate its magnitude by severity. A negative criterion must describe the presence of the undesirable behavior, not compliance with an instruction to avoid it.

Use the following weight ranges:
- `+8` to `+10`: necessary for successful task completion;
- `+4` to `+7`: important to a core conclusion or key reasoning step;
- `+1` to `+3`: useful but nonessential;
- `−1` to `−10`: harmful behavior, with larger magnitude for greater harm.

Choose:
- `accept`: the criterion and proposed weight are both valid;
- `revise`: the core idea is valid, but the wording, scope, atomicity, or weight should change;
- `reject`: the criterion is unsupported, irrelevant, subjective, redundant, unjudgeable, or otherwise unsuitable.

Return only one valid JSON object:

```
{
  "id": "the proposed rubric ID",
  "verdict": "accept|revise|reject",
  "reason": "specific feedback for the generator",
  "weight": 8,
  "sources": [
    {
      "url": "https://...",
      "evidence": "short evidence span supporting the verdict"
    }
  ]
}
```

For `accept`, keep the proposed weight. For `revise`, return the recommended weight and explain every required change in `reason`. For `reject`, set `weight` to JSON `null`. Include independent sources whenever the verdict depends on factual or analytical grounding; an empty source list is allowed only when rejection follows solely from irrelevance, duplication, or malformed wording. Output JSON only, without comments, Markdown fences, or extra fields.

B Training and Inference Details

B.1 Training Hyperparameters

Table 3 reports the cold-start SFT configurations, while Table 4 summarizes the rollout and optimization configurations for reinforcement learning.

Table 3: Key configurations and hyperparameters for cold-start SFT.

Name	Value
Backbone	Qwen3-14B
Training trajectories	1,000
Trajectory generator	GPT-OSS-120B
Framework	ms-swift
Trajectory selection	Rubric score > 0.6
Fine-tuning type	Full
Learning rate	5.0×10^{-5}
LR scheduler type	Cosine
Global batch size	64
Training epochs	2
Maximum sequence length	65,536

B.2 Reward-Judge Prompts

We use GPT-OSS-120B to implement both reward judges during reinforcement learning.

B.2.1 Rubric-Compliance Judge

Given a research query, its task-specific rubrics, and the final report, the rubric judge independently determines whether each criterion is satisfied and returns a binary compliance score.

Table 4: Key configurations and hyperparameters for rubric-guided RL.

Name	Value
Initialization	RubricAtlas-SFT-14B
Training queries	7,000
Framework	verl
Algorithm	GDPO
Maximum turns	35
Rollout group size G	8
Train batch size	64
Mini-batch size	256
Rollout temperature	1.0
Policy learning rate	2.0×10^{-6}
Optimization steps	200
Lower clipping ratio	0.20
Upper clipping ratio	0.28
Rubric–evidence balance λ	0.8
Maximum response length	16,384
Judge model	GPT-OSS-120B

Rubric-Compliance Judge Prompt

Please evaluate the final report against the supplied rubric.

Research query: ${query_text}
Rubric: ${rubric_json}
Final report: ${final_report}

Use only the final report and the information contained in the rubric. Check whether the report actually satisfies the criterion.

Judge only whether the condition written in the rubric is satisfied:
- pass: the report satisfies the condition;
- fail: the report does not satisfy the condition.

Ignore the rubric's weight and whether it is positive or negative. Do not reinterpret, invert, or compensate for its sign.

Apply the rubric at its stated scope. A partially satisfied atomic criterion receives fail. Accept equivalent wording and organization when the substantive requirement is met. For criteria involving comparison, synthesis, causality, uncertainty, or source conflict, require the corresponding analysis rather than a list of disconnected facts.

Return only one valid JSON object:

```
{
  "verdict": "pass|fail",
  "reason": "brief criterion-specific justification",
  "report_evidence": "short excerpt or precise location in the
report, or null"
}
```

Do not compute a score, weighted result, or overall report rating. Output JSON only, without comments, Markdown fences, or extra fields.

B.2.2 TRAJECTORY-GROUNDED EVIDENCE JUDGE

For each factual rubric, the evidence judge examines the retrieved observations in the search trajectory and determines whether they contain substantive supporting evidence, without relying on the final report alone.

Trajectory-Grounded Evidence Judge Prompt

Please determine whether the content described by the factual rubric is present in the supplied search observations.

Research query: ${query_text}
Factual rubric: ${factual_rubric_json}
Search observations: ${search_observations_json}

Inspect all observations and choose:
- present: the observations contain the information described by the rubric;
- not_present: that information does not appear in the observations.

Judge semantic content rather than exact wording. The observation must contain the facts needed by the rubric, not merely mention the same topic. Use only the supplied observations, ignore rubric weights and signs, external knowledge, and information that could have been inferred but was not actually retrieved.

Return only one valid JSON object:

```
{
  "verdict": "present|not_present",
  "observation_ids": ["observation containing the support"],
  "reason": "brief explanation of the match"
}
```

A not_present result must have an empty `"observation_ids"` list. Output JSON only, without comments, Markdown fences, or extra fields.

B.3 AGENT INTERFACE

We use a common ReAct agent interface for cold-start trajectory generation, reinforcement-learning rollouts, and evaluation inference. The agent has access to memo management, web search, webpage retrieval, scholarly search, and Python execution.

B.3.1 RESEARCH-AGENT SYSTEM PROMPT

Research-Agent System Prompt

You are a Deep Research Agent. Your task is to conduct exhaustive information gathering to construct a **professional, comprehensive, and well-structured report** about a user-provided topic.
At each step, you should first perform detailed reasoning, and enclose the reasoning process within `<think> </think>` tags, then proceed with a tool call or response.
When you have synthesized all necessary information, provide your answer as follows: `<think>` reasoning process here `</think>` Your comprehensive final answer here.

B.3.2 TOOL SCHEMAS

We provide the complete JSON schemas for all agent-facing tools, including their arguments, required fields, and return formats.

Tool Schema Definitions

```
Memo_Tool_Schema = {
  "type": "function",
  "function": {
    "name": "Memo",
    "description": "A persistent scratchpad. Use it to jot down key findings and intermediate conclusions to build a global knowledge structure and prevent forgetting. Decompose the problem into multiple aspects and maintain one memo entry per aspect. Use write to create a new entry; use update to overwrite an existing entry; use delete to remove obsolete entries. Do NOT call this tool proactively; only use it when explicitly asked to consolidate findings.",
    "parameters": {
      "type": "object",
      "properties": {
        "entries": {"type": "array", "description": "A list of memo operations."}
      },
      "required": ["entries"]
    }
  }
}

Scholar_Schema = {
  "type": "function",
  "function": {
    "name": "ScholarSearch",
    "description": "Queries Google Scholar to retrieve peer-reviewed literature, domain-relevant publications, and bibliographic evidence.",
    "parameters": {
      "type": "object",
      "properties": {
        "query": {"type": "array", "description": "A list of search queries.", "items": {"type": "string"}}
      },
      "required": ["query"]
    }
  }
}

Fetch_Schema = {
  "type": "function",
  "function": {
    "name": "WebFetch",
    "description": "A link browsing tool that can open various types of URLs, such as webpages and PDFs, and extract and summarize relevant information based on the given requirements.",
    "parameters": {
      "type": "object",
      "properties": {
        "url": {"type": "string", "description": "The target URL, starting with http."},
        "description": {"type": "string", "description": "Describe the specific information you want to extract from the current URL."}
      },
      "required": ["url", "description"]
    }
  }
}
```

Search_Schema = {
"type": "function",
"function": {
"name": "WebSearch",
"description": "A web search tool that takes a list of search queries and returns a list of webpages with corresponding summaries for each query. Queries should be concise and clear; complex questions should be decomposed into multiple simpler queries and searched step by step.",
"parameters": {
"type": "object",
"properties": {
"query": {"type": "array", "items": {"type": "string"}, "minItems": 1, "maxItems": 5, "description": "A list of search queries containing at least 1 and at most 5 questions."}
},
"required": ["query"]
}
}
}

Code_Schema = {
"type": "function",
"function": {
"name": "CodeInterpreter",
"description": "Call this tool when you need to achieve a goal by executing Python code, such as performing arithmetic calculations, data analysis, or text processing. It runs the provided code block and returns the execution results.",
"parameters": {
"type": "object",
"properties": {
"code": {"type": "string", "description": "The code block to be executed must be provided in the exact format: {code_content}."}
},
"required": ["code"]
}
}
}

B.4 Memo Context Manager

The Memo Context Manager is applied consistently during cold-start trajectory generation, reinforcement learning, and inference. We set the visible-observation budget to $N = 10$ and the retained size to $R = 5$. When $|\mathcal{O}(\mathcal{S}^{\text{con}})| > N$, the agent first consolidates the visible history into the memo; after a successful update, older observations are removed and only the most recent R observations remain visible.

B.4.1 Memo-Consolidation Prompt

We include the complete prompt issued when the visible-observation budget is exceeded, restricting the agent to a dedicated memo-only turn that consolidates key findings before older observations are evicted.

Memo-Consolidation Prompt

Your conversation history is about to be compressed. Please use the Memo tool NOW to consolidate any important findings, intermediate conclusions, or evidence you have gathered so far

as a draft for your final report. You are ONLY allowed to call the Memo tool. Do NOT call any other tool.

B.4.2 MEMO-INJECTION PROMPT

We include the template used to re-inject the updated aspect-indexed memo into the condensed visible context after a successful consolidation.

Memo-Injection Prompt

Active Working Memory

{topic_1}
{content_1}

{topic_2}
{content_2}

...

C BENCHMARK AND EVALUATION DETAILS

C.1 BENCHMARK OVERVIEW

We follow the official data and report-evaluation protocols for all three benchmarks.

C.1.1 DEEPRESEARCH BENCH

DeepResearch Bench contains 100 expert-authored tasks spanning 22 domains. Following its official protocol, Gemini-2.5-Pro evaluates each report and produces an overall RACE score together with comprehensiveness, insight, instruction following, and readability. We use the official task set, judge prompt, score aggregation, and parsing procedure.

C.1.2 DEEPCONSULT

DeepConsult contains 102 consulting and business queries and evaluates each candidate report through pairwise comparison against a reference report. Following the official protocol, o3-mini assesses instruction following, comprehensiveness, completeness, and writing quality. To mitigate position bias, each pair is judged in both answer orders, with three trials per order, and the resulting decisions are aggregated into win, tie, and lose rates.

C.1.3 RESEARCHRUBRICS

ResearchRubrics contains 101 open-ended prompts paired with 2,593 expert-written criteria. Following its official protocol, Gemini-2.5-Pro independently judges whether each criterion is satisfied by the final report, and performance is reported as binary rubric compliance.

C.2 EVALUATION SETTINGS

Unless otherwise specified, all baselines are evaluated with the same tool suite and tool schemas described in Appendix B.3. Each run permits up to 60 interaction turns and a maximum generation length of 16384 tokens per turn. Memo-enabled systems use the same compression configuration as during training, with $N = 10$ and $R = 5$. We perform three independent runs for every evaluation and report the average result to improve reliability.