Writing As Reasoning: Interleaving Drafting and Deepening for Open-Ended Deep Research

Anonymous ACL submission

Abstract

Generating deep research reports requires large-scale information acquisition and the synthesis of insight-driven analyses, posing a major challenge for current language models. Existing methods largely follow a *plan-then-write* paradigm, whose performance strongly depends on the quality of the initial outline. We propose a **Writing As Reasoning Policy (WARP)** framework, which enables models to dynamically revise outlines during report writing. In this policy, the agent alternates between ***Evidence-Based Drafting*** and ***Reasoning-Driven Deepening*** phases, jointly supporting information acquisition, knowledge refinement, and outline updating. We further introduce a **Multi-Stage Agentic Training**—including cold start, atomic skill RL, and holistic pipeline RL—that enables small models to operate the WARP effectively. Experiments on DeepResearch Bench, DeepConsult, and DeepResearch Gym show that our approach allows small models to surpass leading closed-source systems, particularly with substantial improvements in *Insight* metric.

1 Introduction

Open-ended deep research requires artificial agents to navigate vast information landscapes and synthesize their findings into coherent, insightful reports (OpenAI, 2025; Google, 2025; x.AI, 2025; Perplexity, 2025; Kimi, 2025; ByteDance, 2025). In the context of such complex inquiry, *writing* is far more than the mere transcription of retrieved data. Instead, it mirrors the *knowledge-transforming* process in cognitive psychology (Scardamalia and Bereiter, 1987): since the information landscape is initially opaque, researchers rarely execute a rigid, end-to-end plan derived from pre-existing thoughts. Rather, the act of writing serves as a reasoning mechanism itself, helping to uncover what is not yet known. Researchers frequently identify gaps, contradictions, and new directions only while drafting, indicating that effective synthesis relies on a tight coupling between planning and writing.

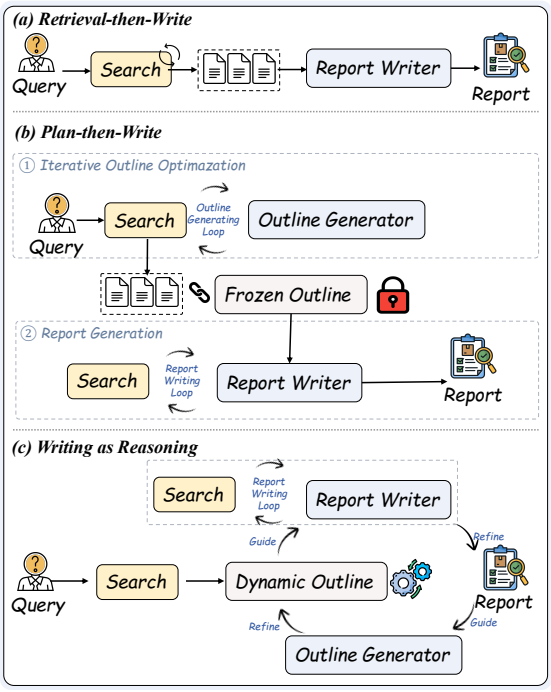

Figure 1: Comparison of different writing paradigms.

However, existing approaches have struggled to replicate this dynamic. Early methods relied on a *retrieval-then-write* paradigm (Fig. 1a) (Hu et al., 2025), where agents generated content sequentially. While flexible, this unstructured approach often deteriorates into incoherence over long horizons. To address this, recent frameworks (Fig. 1b) (Wang et al., 2024, 2025; Yan et al., 2025) like WebWeaver (Li et al., 2025c) adopted a *plan-then-write* paradigm. By freezing a comprehensive outline before writing, these systems ensure structural stability. However, this approach relies on the *assumption of initial information completeness*—a premise often fallacious in open-ended research. By treating the downstream writer merely as an executor

of a static blueprint, this dichotomy prevents the agent from capturing *emergent insights*—nuanced connections that surface only when articulating specific arguments. Consequently, such baselines encounter an *insight ceiling*, producing reports that are structurally sound but intellectually shallow.

To bridge this gap, we introduce **WARP** (Writing As Reasoning Policy), which transcends the limitations of static planning by modeling deep research as an iterative refinement loop. Instead of adhering to a fixed outline, the agent alternates between two macro-states: *Evidence-based Drafting* and *Reasoning-driven Deepening*.

Crucially, WARP operates as a dynamic policy rather than a static rule-based system. In the *Reasoning-driven Deepening* state, the agent autonomously decides whether to terminate or continue deepening, by evaluating the semantic density and logical coherence of the current draft. If deepening, it decomposes broad sections into granular inquiries and updates the outline based on the actual writing feedback, effectively mirroring the human knowledge-transforming process.

Considering the expansive action space and extended decision horizons inherent to WARP, we design a systematic training pipeline to ensure stable convergence under reasonable resource constraints. Specifically, we introduce a *trajectory pruning* strategy to rigorously filter data for higher quality supervision, and adopt a multi-stage reinforcement learning curriculum that sequentially fine-tunes local atomic actions before optimizing end-to-end performance. This approach instills a robust policy that autonomously balances research depth against computational cost, triggering recursive optimization cycles only when they yield significant informational gain.

Extensive experiments on DeepResearch Bench, Deep Consult, and DeepResearch Gym validate the efficacy of our framework, demonstrating marked improvements in report quality, particularly in the *Insight* metric. Notably, our 8B agent outperforms many closed-source deep research systems, demonstrating that specialized WARP inference diagram can enable small-scale models to rival proprietary large-scale models such as Gemini-2.5-Pro.

2 Method

In this section, we formally present WARP (Writing As Reasoning Policy). Unlike traditional pipelines (Wang et al., 2024; Shi et al., 2025) that treat planning and writing as temporally segregated stages, we formulate deep research as a unified sequential decision-making process. The core intuition of WARP is that the intermediate draft serves not merely as an output buffer, but as a *dynamic reasoning context* that should actively shape future planning. By maintaining the draft as an evolving state variable, our framework enables an *interleaved* inference loop: the agent alternates between **Evidence-Based Drafting** (filling the report based on the current plan) and **Reasoning-Driven Deepening** (expanding the plan based on the generated report). This mechanism allows the system to pivot from a broad initial skeleton to fine-grained exploration, ensuring the final report achieves both comprehensive breadth and analytical depth.

2.1 Problem Formulation

In specific, we formulate open-ended deep research as an iterative hierarchical decision-making process: At any interaction loop i, the agent observes a global state $S_i = (Q, O_i, D_i, C_i)$, comprising the user query Q, a dynamic outline O_i, the current draft D_i, and the context C_i retrieved at the current loop i. At the j-th step $t_{i,j}$ in a loop i, the agent executes an action $A_{i,j}$ selected from a defined action space: {INITIALIZE, SEARCH, WRITE, EXPAND, TERMINATE}.

This formulation unifies planning and writing: outline adjustments ($O_i \rightarrow O_{i+1}$) and content generation ($D_i \rightarrow D_{i+1}$) are treated as equivalent state transitions driven by the policy.

2.2 The WARP Inference Diagram

WARP begins with a coarse-to-fine initialization strategy designed to establish a comprehensive research scope before diving into details. Starting from the initial state S_0, the agent analyzes the query Q to generate broad search queries q_0. Upon retrieving the background context C_0, it synthesizes an initial Level-1 outline O_0:

$$O_0 \leftarrow \text{INITIALIZE}(Q, C_0). \tag{1}$$

In contrast to static planners (Li et al., 2025c) that attempt to generate a fully detailed hierarchy, our O_0 is intentionally sparse, consisting only of high-level section titles and brief writing intents. This design mitigates the risk of being ungrounded.

To maximize the final report quality, the agent operates under a unified policy π_θ that orchestrates the research trajectory. Crucially, this workflow is

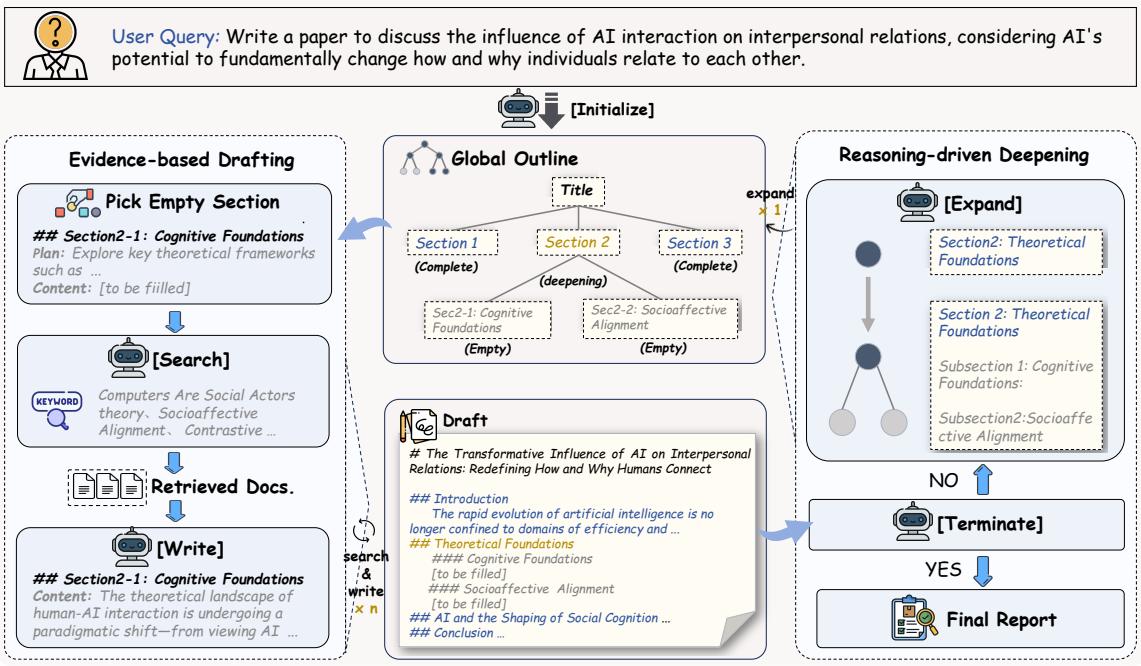

Figure 2: The WARP framework. The agent interleaves **Evidence-Based Drafting** (writing content) and **Reasoning-Driven Deepening** (updating the dynamic outline). This loop allows the agent to discover and bridge logical gaps that emerge only during the writing process.

not linear but iterative, alternating between Drafting and Deepening.

Evidence-Based Drafting Given a tentative outline O_i, the agent executes a *retrieve-then-write* strategy to convert the structural plan into substantiated content. Unlike independent parallel generation, which often leads to fragmentation or redundancy, we enforce contextual consistency by conditioning retrieval queries on the accumulating narrative. For a specific section k, the agent first formulates query $q_{i,k}$ based on user query Q, the section's intent O_i^k, and the draft context D_i:

$$q_{i,k} \leftarrow \textsc{Search}(Q, O_i^k, D_i). \quad (2)$$

Then, the retrieval tools will acquire new content C_i^k based on the query $q_{i,k}$. This ensures that new information strictly extends the logical flow of previous sections. The agent then synthesizes the section content c_k by grounding the text in retrieved evidence to guarantee faithfulness:

$$D_i^k \leftarrow D_i^{k-1} \oplus \textsc{Write}(Q, O_i^k, D_i^{k-1}, C_i^k). \quad (3)$$

The objective is to achieve information integration—synthesizing disparate sources into a coherent argument—rather than mere aggregation. This phase focuses on writing, iteratively populating the outline to produce a new draft D_{i+1} that serves as the foundation for deeper reasoning.

Reasoning-Driven Deepening Initial outlines are inevitably constrained by the model's pre-retrieval knowledge, which often creates an *insight ceiling*: the structure may cover the breadth of the topic but fail to capture its nuanced depth. To break this ceiling, the policy π_θ periodically shifts from local drafting to global planning, treating the newly generated draft D_{i+1} as a fresh observation for reasoning and diagnosis.

Since D_{i+1} provides a concrete reasoning context, the agent can detect logical gaps or superficial arguments that were invisible during initial planning. If section k^* lacks depth, the agent generates a **local sub-sections** to decompose the topic, updating O_i and triggering a targeted drafting cycle:

$$O_{i+1} \leftarrow O_i \oplus \textsc{Expand}\{k^*\}(Q, O_i, D_{i+1}). \quad (4)$$

The process concludes only when the agent verifies that the logical chain is complete and the content depth aligns with the query's complexity:

$$End \leftarrow \textsc{Terminate}(Q, O_i, D_{i+1}). \quad (5)$$

2.3 Multi-Stage Agentic Training

While large-scale models have demonstrated strong capabilities within our WARP framework (see §3.3.1), training small-scale models (such as 8B) for open-ended research is non-trivial. The

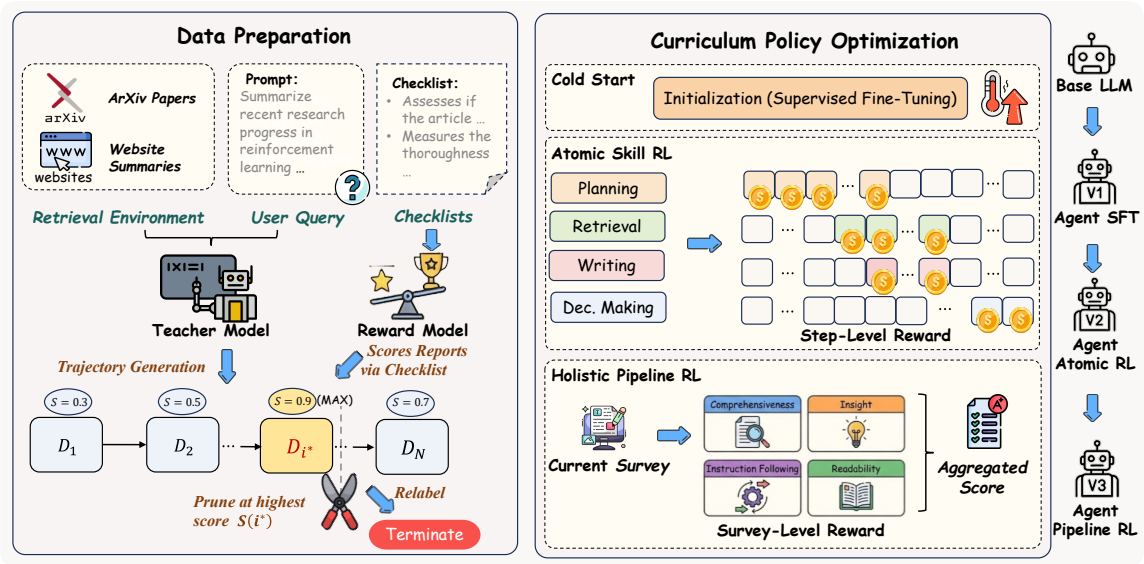

Figure 3: Overview of our multi-stage agentic training process.

Table 1: Reward definition for **Atomic Skill RL**. For each metric, we indicate whether it requires references (**Ref.**) or relies on an **LLM-Judge**. "Dec. Making" indicates the decision-making ability.

Atomic Ability	Metric Name	Need Ref.?	LLM-Judge?	Description & Reward Objective
Planning	Basic Properties	✗	✗	Section number and language consistency.
	Holistic Quality	✓	✓	Quality metrics such as guidance, logic, clarity, etc.
	Faithfulness	✗	✓	Whether the content in the plan is real and reliable.
Retrieval	Relevance Recall	✓	✗	Overlap score between retrieved docs and golden ones.
Writing	Basic Properties	✗	✗	Content length, citation number, and language consistency.
	Holistic Quality	✓	✓	Quality metrics such as relevance, coverage, depth, etc.
	Faithfulness	✗	✓	Penalizes unsupported claims.
	Citation Precision	✓	✗	Rewards citations that overlaps with golden ones.
Dec. Making	Accuracy	✓	✗	Whether terminate at the suitable time.

challenges are twofold: (1) *Ambiguous Termination*: even teacher models struggle to determine the optimal stopping point for research; (2) *Sparse Rewards*: long horizons make reward assignment difficult. To address these, we first introduce a trajectory pruning strategy in §2.3.1. Then, we propose a curriculum learning pipeline in §2.3.2.

2.3.1 Data Preparation

A critical challenge in training is the scarcity of expert trajectories that exhibit efficient decision-making. Teacher models often either expand indefinitely or terminate arbitrarily, giving rise to what we term the **optimal stopping problem**. To solve this, we introduce **trajectory pruning** strategy. Instead of cloning the teacher's termination behavior, we force the teacher to "over-expand" recursively. This generates a sequence of drafts with varying granularity $\{D_1, \ldots, D_N\}$. We then retroactively identify the optimal point i^* where the report draft D_{i^*} has the highest score. We prune the trajectory at i^*, relabeling the action to TERMINATE. This provides a supervision signal for *information saturation*, teaching the agent to stop based on report quality rather than arbitrary imitation. Details on our query construction and retrieval environment are provided in App. A.2 and App. A.3.

2.3.2 Curriculum Policy Optimization

Our optimization includes three stages, as shown in Figure 3: (1) **SFT for Cold Start:** Establishes basic instruction following and format adherence. (2) **Atomic Skill RL:** Uses teacher trajectories as anchors to master local execution and stabilize exploration. (3) **Holistic Pipeline RL:** Optimizes global report quality, enabling the agent to refine its strategy beyond the teacher's limitations.

Atomic Skill RL To tackle the reward assignment problem, we first decompose the global objective to atomic abilities: **planning** (*Initialize*, *Ex-*

pand), **retrieval** (*Search*), **writing** (*Write*), and **decision-making** (*Terminate*). Then, we design different reward functions for them, which combine execution results (e.g., basic properties, holistic quality, and faithfulness) with *reference alignment*, as shown in Table 1. This stage ensures the agent masters the "how"—producing valid plans, precise searches, and coherent paragraphs—before attempting to optimize global strategy.

Holistic Pipeline RL Local correctness (e.g., a valid paragraph) does not guarantee global coherence. Thus, the final stage shifts to end-to-end optimization to evaluate the final report quality, such as *Comprehensiveness* and *Insight*. Crucially, this stage empowers the agent to deviate from the teacher's path. By propagating the holistic report score backward, the agent learns to trigger the *deepening* only when it yields significant informational gain. This effectively refines the quality-efficiency frontier, suppressing redundant expansions that the teacher might have made.

3 Experiments

3.1 Settings

Implementation Details. We implement WARP using *MiniCPM4.1-8B* (Team et al., 2025) as the backbone for our deep research agent system. Training follows the curriculum described in §2.3, progressively scaling from atomic skills to holistic reporting. During training and inference phase, we cap the report structure at three levels and limit the number of deepening steps to 12 to ensure efficiency. Detailed hyperparameters and data statistics per stage are provided in App. B.1.

Benchmarks and Metrics. To ensure comprehensive evaluation, we test on three diverse benchmarks: (1) **DeepResearch Bench** (Du et al., 2025) (100 PhD-level scientific tasks); (2) **DeepConsult** (Dee, 2025) (102 business and financial analysis queries); and (3) **DeepResearchGym** (Coelho et al., 2025) (100 general-purpose information-seeking tasks). We adhere to the standard evaluation protocols of each benchmark, employing *Gemini-2.5-Pro*, *o3-mini*, and *GPT-4.1-mini* respectively as impartial judges.

Baselines. We compare WARP against three distinct categories of latest systems: (1) **Proprietary Systems:** Leading commercial deep research systems including OpenAI (OpenAI, 2025), Gemini (Google, 2025), and Claude (claude, 2025), and Doubao (ByteDance, 2025). (2) **Prompt-Based Frameworks:** WebWeaver (Li et al., 2025c), Enterprise DR (Prabhakar et al., 2025), and RhinoInsigh (Lei et al., 2025). (3) **Trained Open Models:** Recent open-source research agents including WebShaper (Tao et al., 2025), WebThinker (Li et al., 2025b) and DR Tulu (Shao et al., 2025).

3.2 Main Results

Our results on three benchmarks are summarized in Table 2 and Figure 4.

(1) Our WARP framework has strong performance on *Insight* and *Comprehensiveness*. Across these benchmarks, our method achieves nearly the best performance in both *Insight* and *Comprehensiveness* metrics despite using the smallest model. Specifically, on the DeepResearch Bench, it achieves an *Insight* score of 52.64 and a *Comprehensiveness* score of 50.54, surpassing Gemini-2.5-Pro-deepresearch (49.45 and 49.51, respectively). On the DeepResearch Gym, it gets the highest 100.0 score in the *Depth*, *Breadth*, and *Insightfulness* metrics. These gains stem directly from our *reasoning-driven deepening*. On one hand, the agent continuously **extracts insights from condensed intermediate drafts**, enabling deeper reasoning and synthesis. On the other hand, by revisiting intermediate outputs, it can **identify missing topics and globally assess which sections require further expansion**, resulting in broader and more balanced coverage.

(2) The *Multi-Stage Agentic Training* brings stable and comprehensive improvement. The performance of WARP-8B steadily improves from SFT to Atomic RL and finally to Pipeline RL across all metrics on these benchmarks. On DeepResearch Bench, the metric *comprehensiveness* rises from 46.24 to 50.54, *Insight* from 48.10 to 52.64, and *Readability* from 41.79 to 44.17. On DeepConsult, average score grows from 6.04 to 6.60, win rate increase from 54.17% to 57.60%, and loss rate drop from 35.54% to 28.68%. These consistent gains demonstrate that each stage of the curriculum contributes to mastering the full deep research workflow, yielding a more stable and capable agent.

(3) Small-scale agent systems can rival large-scale ones. Averaged across benchmarks, our deep research system demonstrates excellent performance. Our WARP-8B (Pipeline RL) achieves an *Overall* score of 50.11 on DeepResearch Bench,

Table 2: Performance of agent systems on DeepResearch Bench in terms of comprehensiveness (Comp.), insight, instruction-following (Inst.), readability (Read.) and DeepConsult (Avg., Win, Tie, Lose).

Agent systems	DeepResearch Bench					DeepConsult			
	Overall	Comp.	Insight	Inst.	Read.	Avg.	Win	Tie	Lose
Proprietary Deep Research Systems									
Doubao-research	44.34	44.84	40.56	47.95	44.69	5.42	29.95	40.35	29.70
Claude-research	45.00	45.34	42.79	47.58	44.66	4.60	25.00	38.89	36.11
OpenAI-deepresearch	46.45	46.46	43.73	49.39	47.22	5.00	0.00	100.00	0.00
Gemini-2.5-Pro-deepresearch	**49.71**	**49.51**	**49.45**	**50.12**	**50.00**	**6.70**	61.27	31.13	**7.60**
Prompt-Based Frameworks									
WebWeaver (Qwen3-30B-A3B)	46.77	45.15	45.78	49.21	47.34	4.57	28.65	34.90	36.46
WebWeaver (Claude-Sonnet-4)	50.58	**51.45**	50.02	50.81	49.79	**6.96**	66.86	10.47	22.67
Enterprise DR (Gemini-2.5-Pro)	49.86	49.01	50.28	50.03	49.98	6.82	**71.57**	19.12	**9.31**
RhinoInsigh (Gemini-2.5-Pro)	**50.92**	50.51	**51.45**	**51.72**	**50.00**	6.82	68.51	11.02	20.47
Trained Open Models									
WebShaper-32B	34.93	31.58	26.17	44.81	40.38	1.63	3.25	3.75	93.00
WebThinker-32B-DPO	–	39.40	35.40	46.00	**43.50**	–	–	–	–
DR Tulu-8B	–	**41.70**	**41.80**	**48.20**	41.30	–	–	–	–
Our Deep Research Systems									
WARP-8B (SFT)	46.73	46.24	48.10	47.61	41.79	6.04	54.17	10.29	35.54
WARP-8B (Atomic RL)	48.81	48.70	51.36	48.64	42.25	6.06	56.13	11.03	32.84
WARP-8B (Pipeline RL)	**50.11**	**50.54**	**52.64**	**48.87**	**44.17**	**6.60**	**57.60**	13.73	**28.68**

surpassing Gemini-2.5-Pro-deepresearch (49.71). It also attains state-of-the-art results on DeepResearch Gym, with an average score of 98.48. These results show that integrating WARP with *multi-stage agentic training* enables small models to reach the performance level of leading proprietary research systems. These findings suggest that, for deep research tasks, the primary bottleneck lies not in model size, but in the design of effective cognitive and planning processes that fully leverage a model's inherent capabilities.

3.3 Analysis

3.3.1 Does *Writing As Reasoning Policy* remain effective without training?

To assess whether our framework is intrinsically effective without training, we conduct a prompt-based comparison on DeepResearch Bench using a larger model, *Qwen3-235B-A22B-Instruct-2507* (Yang et al., 2025). We compare two policies: (1) *Plan-then-write*, where the model first constructs a detailed outline through retrieval and then generates the report from this fixed plan; and (2) WARP, which starts from an simple outline and interleaves writing with iterative deepening.

As shown in Table 3, WARP consistently outperforms the *Plan-then-write* paradigm across all metrics, with a notable gain in *Insight* (+1.19) and *Comprehensiveness* (+0.98). By using the evolving draft as a reasoning context, WARP can detect underdeveloped or ambiguous content during writing and trigger targeted *Deepening* with additional evidence, whereas *Plan-then-write* remains constrained by a static outline. This confirms that draft-aware deepening is a key driver of insight.

Table 3: Evaluation for different generation paradigms.

Paradigm	Overall	Comp.	Insight	Inst.	Read.
Plan-then-write	49.90	49.35	51.60	50.13	46.46
WARP (ours)	**50.72**	**50.33**	**52.79**	**50.32**	**47.20**

3.3.2 How *Multi-Stage Training* shapes agent actions and report structure?

In this section, we analyze how agent behavior evolves across training stages by examining statistics of its actions and report sections. Specifically, we focus on the *Write* and *Expand* actions, which directly determine the report structure.

Table 4: Evolution of action usage and hierarchical sectioning across training stages on DeepResearch Bench.

Stages	Actions		Sections		
	Write	Expand	Level-1	Level-2	Level-3
SFT	21.24	4.44	6.27	10.11	4.86
Atomic RL	36.89	8.88	6.49	14.17	16.50
Pipeline RL	39.51	8.63	6.52	15.75	17.32

As shown in Table 4, a clear behavioral shift emerges when moving from SFT to the RL-based

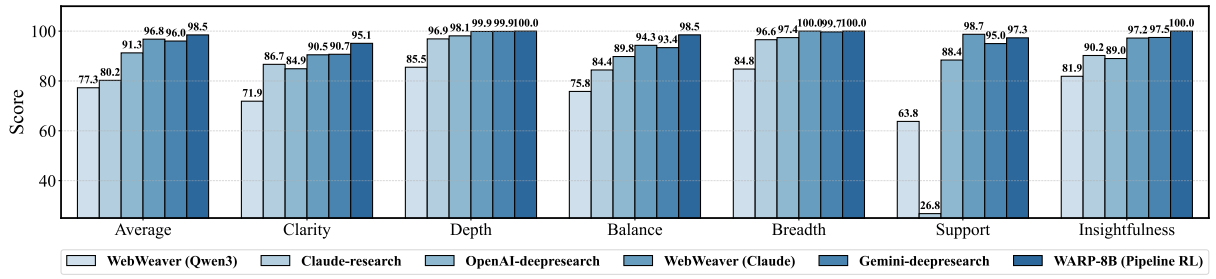

Figure 4: Performance of agent systems on DeepResearch Gym.

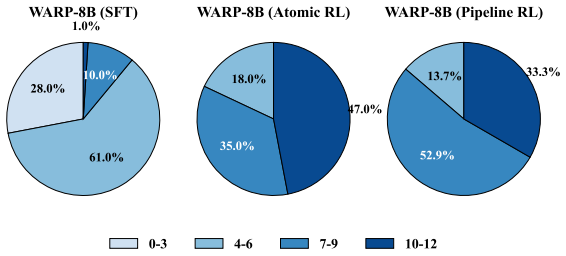

Figure 5: The *Expand* steps on DeepResearch Bench.

training stages. The frequency of ***Expand*** (Deepening) actions nearly doubles (from 4.44 to around 8.8), which in turn leads to a dramatic growth in fine-grained subsectioning (Level-3: from 4.86 to 17.32). Moreover, Figure 5 illustrates that RL training drives the agent to deepen more compared to SFT, ensuring at least 4 *Expand* steps in all cases. This trend indicates that RL training effectively equips the agent with *Reasoning-Driven Deepening*: Rather than adhering to a shallow outline as in SFT, the agent learns to identify underdeveloped parts of a draft and proactively expand them through iterative refinement, resulting in reports with substantially richer structure.

3.3.3 How does the number of deepening influence report quality?

To examine whether the model has learned an appropriate stopping policy for *deepening* and to quantify how deepening depth affects report quality, we conduct a "Forced Expansion" experiment. Specifically, during inference, we force WARP-8B (Pipeline RL) to apply the *Expand* action exactly k times, where k ranges from 0 to 15. We then compare this forced expansion curve with the actual deepening behaviors of WARP-8B at different training stages (SFT, Atomic RL, and Pipeline RL), overriding their learned termination policies. This allows us to directly evaluate report quality as a function of deepening depth and to compare the model's learned stopping behavior against the optimal expansion point.

The results in Figure 6 reveal three consistent patterns. **First**, performance increases steadily with deeper expansion and begins to plateau at around nine steps, indicating diminishing returns beyond this depth. **Second**, both *Comprehensiveness* and *Insight* rise strongly with deepening, improving by nearly 6 points from shallow to sufficiently deep regimes, confirming the importance of iterative refinement for rich and insightful reports. **Third**, different training stages exhibit distinct stopping behaviors. The SFT agent typically stops within 6 steps and rarely reaches the saturation regime, whereas the Atomic RL and Pipeline RL agents shift their stopping distributions toward 6–15 steps, closely matching the empirically optimal depth.

3.3.4 How does the *Trajectory Pruning* affect the agent training?

To address the optimal stopping problem, we introduce a *trajectory pruning* strategy to construct higher-quality training data. In this section, we isolate its impact by training SFT models using the same teacher-generated trajectories, either with or without pruning. We consider two settings: (1) w/o pruning, which directly uses the raw trajectories produced by the teacher model, and (2) with pruning, which selects the best intermediate draft in one trajectory based on reward scores, retaining only the sub-trajectory before that draft.

Table 5: Effect of trajectory pruning on SFT training.

Trajectory	Overall	Comp.	Insight	Inst.	Read.
w/o pruning	45.80	44.95	47.35	46.71	40.86
with pruning	**46.73**	**46.24**	**48.10**	**47.61**	**41.79**

As shown in Table 5, models trained on pruned trajectories consistently outperform those trained on raw teacher ones across all evaluation dimensions. This indicates that trajectory pruning effectively improves the quality of supervision. More

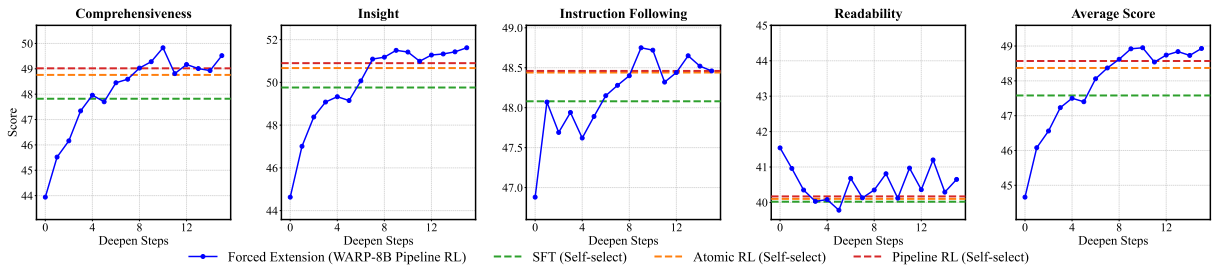

Figure 6: Mean Performance metrics per Deepen Steps on DeepResearch Bench.

importantly, these results reveal a key limitation of large teacher models: although they generate strong drafts, their termination decisions are often suboptimal. By applying reward-based selection over intermediate states, trajectory pruning filters out poorly timed stopping points and provides cleaner training signals. As a result, the student model learns a more accurate termination policy, which is crucial for effective *Reasoning-Driven Deepening*.

3.3.5 Cost Analysis

In this work, we use a small model (MiniCPM4.1-8B) as the core agent, which outperforms proprietary deep research systems. To demonstrate the cost advantage of WARP, we estimate the per-report cost based on OpenRouter pricing. As MiniCPM4.1-8B is not listed on OpenRouter, we adopt the pricing of a similarly sized model, Qwen3-8B ($0.117 per million input tokens, $0.455 per million output tokens). The average cost per report using WARP is approximately $0.148. In contrast, Gemini Deep Research costs between $2 and $5 per report. Table 6 summarizes the cost comparison against existing systems, and our system achieves a significant cost advantage.

Table 6: The cost of different deep research systems.

System	Backbone	Tokens	Cost	Time
Enterprise DR	Gemini-2.5-pro	539k	1.17	–
Langchain DR	Multi-LLMs	2,070k	0.878	–
WebWeaver	Claude-sonnet-4	763k	2.83	29 min
WebWeaver	Qwen3-30B-A3B	773k	–	18 min
Gemini DR	Gemini-2.5-pro	–	2-5	16 min
WARP-8B	MiniCPM4.1-8B	1,068k	0.148	17 min

4 Related Work

Deep information seeking is a core component of recent deep research systems (Google, 2025; OpenAI, 2025; Li et al., 2025b). Existing benchmarks and systems target either complex QA (Wei et al., 2025; Zhou et al., 2025) or long-form report generation (Du et al., 2025; Dee, 2025; Coelho et al., 2025). Our work focuses on emphasizing scalable discovery of retrieval directions to improve information completeness.

Knowledge-intensive long-form writing—such as Wikipedia entries (Shao et al., 2024a), academic surveys (Wang et al., 2024; Hu et al., 2024; Wang et al., 2025), and research reports (Li et al., 2025b,c)—requires factual reliability with citations and coverage of major subtopics. Most prior approaches adopt a *retrieve-then-generate* pipeline with a *plan-then-write* strategy, where an initial outline guides retrieval and writing (Yan et al., 2025; Li et al., 2025b; Wang et al., 2025). Thus, final quality heavily depends on the initial outline's completeness. Existing methods improve outline generation via multi-perspective discussion (Shao et al., 2024a; Jiang et al., 2024), concept-pool expansion (Li et al., 2025a), or iterative refinement (Yan et al., 2025; Li et al., 2025c). In contrast, we adopt a *writing as reasoning* policy that integrates outline expansion into the writing process, enabling progressive refinement and targeted post-hoc revision.

5 Conclusion

We present a new paradigm for open-ended deep research, **Writing As Reasoning Policy (WARP)**, which tightly integrates outline refinement with long-form writing. By iteratively alternating between content generation and structural expansion, WARP treats intermediate drafts as compact knowledge that continuously updates the plan, overcoming the rigidity of fixed outlines while reducing reliance on large-scale models. Combined with a multi-stage agentic training strategy, our compact 8B agent achieves strong performance across multiple benchmarks, surpassing leading closed-source systems. These results demonstrate that dynamic, draft-driven planning with small models is sufficient for high-quality deep research.

Limitations

Report presentation. In most existing deep research systems, including ours, tables and figures are generated inline with paragraph-level text. However, constructing tabular layouts requires a reasoning process fundamentally different from writing prose, placing heavy demands on a model's structural and formatting abilities. This coupling partly explains why agent systems based on smaller models often underperform large ones in presentation quality. A promising direction is to decouple presentation from content generation and assign it to a dedicated rendering agent, which could enable small models to achieve comparable or even superior layout quality. Moreover, current readability evaluation remains largely text-based and weakly reflects the true visual structure of rendered reports, suggesting the need for visual-modality evaluation in future work.

Information sources. Our system relies on a locally deployed textual knowledge base (e.g., arXiv abstracts and web summaries), which ensures stability and reproducibility but limits coverage and timeliness. It also lacks access to images, videos, domain-specific corpora, and personalized data. Future extensions will expand the knowledge base to support multi-modal content, local and personalized sources, and continuous updates, enabling richer and more realistic research scenarios.

References

2025. Deepconsult: A deep research benchmark for consulting/business queries. https://github.com/youdotcom-oss/ydc-deep-research-evals. GitHub repository.

ByteDance. 2025. Doubao deep research. https://www.doubao.com/chat/.

Yuxuan Chen, Dewen Guo, Sen Mei, Xinze Li, Hao Chen, Yishan Li, Yixuan Wang, Chaoyue Tang, Ruobing Wang, Dingjun Wu, and 1 others. 2025. Ultrarag: A modular and automated toolkit for adaptive retrieval-augmented generation. *arXiv preprint arXiv:2504.08761*.

Meet claude. 2025. Claude deep research. https://www.anthropic.com/claude.

João Coelho, Jingjie Ning, Jingyuan He, Kangrui Mao, Abhijay Paladugu, Pranav Setlur, Jiahe Jin, Jamie Callan, João Magalhães, Bruno Martins, and 1 others. 2025. Deepresearchgym: A free, transparent, and reproducible evaluation sandbox for deep research. *arXiv preprint arXiv:2505.19253*.

Mingxuan Du, Benfeng Xu, Chiwei Zhu, Xiaorui Wang, and Zhendong Mao. 2025. Deepresearch bench: A comprehensive benchmark for deep research agents. *arXiv preprint arXiv:2506.11763*.

Google. 2025. Gemini deep research. https://gemini.google/overview/deep-research/.

Chen Hu, Haikuo Du, Heng Wang, Lin Lin, Mingrui Chen, Peng Liu, Ruihang Miao, Tianchi Yue, Wang You, Wei Ji, and 1 others. 2025. Step-deepresearch technical report. *arXiv preprint arXiv:2512.20491*.

Yuntong Hu, Zhuofeng Li, Zheng Zhang, Chen Ling, Raasikh Kanjiani, Boxin Zhao, and Liang Zhao. 2024. Hireview: Hierarchical taxonomy-driven automatic literature review generation.

Yucheng Jiang, Yijia Shao, Dekun Ma, Sina J Semnani, and Monica S Lam. 2024. Into the unknown unknowns: Engaged human learning through participation in language model agent conversations. *arXiv preprint arXiv:2408.15232*.

Kimi. 2025. Kimi-researcher: End-to-end rl training for emerging agentic capabilities. https://moonshotai.github.io/Kimi-Researcher/.

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

Derek Li, Jiaming Zhou, Leo Maxime Brunswic, Abbas Ghaddar, Qianyi Sun, Liheng Ma, Yu Luo, Dong Li, Mark Coates, Jianye Hao, and 1 others. 2025a. Omni-thinker: Scaling multi-task rl in llms with hybrid reward and task scheduling. *arXiv preprint arXiv:2507.14783*.

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

Zijian Li, Xin Guan, Bo Zhang, Shen Huang, Houquan Zhou, Shaopeng Lai, Ming Yan, Yong Jiang, Pengjun Xie, Fei Huang, and 1 others. 2025c. Webweaver: Structuring web-scale evidence with dynamic outlines for open-ended deep research. *arXiv preprint arXiv:2509.13312*.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Wei Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. Factscore: Fine-grained atomic evaluation of factual precision in long form text generation. *arXiv preprint arXiv:2305.14251*.

OpenAI. 2025. Deep research. https://openai.com/index/introducing-deep-research/.

Perplexity. 2025. Perplexity deep research. https://www.perplexity.ai/hub/blog/introducing-perplexity-deep-research.

Akshara Prabhakar, Roshan Ram, Zixiang Chen, Silvio Savarese, Frank Wang, Caiming Xiong, Huan Wang, and Weiran Yao. 2025. Enterprise deep research: Steerable multi-agent deep research for enterprise analytics. *arXiv preprint arXiv:2510.17797*.

Marlene Scardamalia and Carl Bereiter. 1987. Knowledge telling and knowledge transforming in written composition. *Advances in applied psycholinguistics*, 2:142–175.

Rulin Shao, Akari Asai, Shannon Zejiang Shen, Hamish Ivison, Varsha Kishore, Jingming Zhuo, Xinran Zhao, Molly Park, Samuel G Finlayson, David Sontag, and 1 others. 2025. Dr tulu: Reinforcement learning with evolving rubrics for deep research. *arXiv preprint arXiv:2511.19399*.

Yijia Shao, Yucheng Jiang, Theodore A Kanell, Peter Xu, Omar Khattab, and Monica S Lam. 2024a. Assisting in writing wikipedia-like articles from scratch with large language models. *arXiv preprint arXiv:2402.14207*.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, and 1 others. 2024b. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Xiaofeng Shi, Qian Kou, Yuduo Li, Ning Tang, Jinxin Xie, Longbin Yu, Songjing Wang, and Hua Zhou. 2025. Scisage: A multi-agent framework for high-quality scientific survey generation. *arXiv preprint arXiv:2506.12689*.

Zhengwei Tao, Jialong Wu, Wenbiao Yin, Junkai Zhang, Baixuan Li, Haiyang Shen, Kuan Li, Liwen Zhang, Xinyu Wang, Yong Jiang, and 1 others. 2025. Webshaper: Agentically data synthesizing via information-seeking formalization. *arXiv preprint arXiv:2507.15061*.

MiniCPM Team, Chaojun Xiao, Yuxuan Li, Xu Han, Yuzhuo Bai, Jie Cai, Haotian Chen, Wentong Chen, Xin Cong, Ganqu Cui, and 1 others. 2025. Minicpm4: Ultra-efficient llms on end devices. *arXiv preprint arXiv:2506.07900*.

Haoyu Wang, Yujia Fu, Zhu Zhang, Shuo Wang, Zirui Ren, Xiaorong Wang, Zhili Li, Chaoqun He, Bo An, Zhiyuan Liu, and 1 others. 2025. Llm mapreduce-v2: Entropy-driven convolutional test-time scaling for generating long-form articles from extremely long resources. *arXiv preprint arXiv:2504.05732*.

Yidong Wang, Qi Guo, Wenjin Yao, Hongbo Zhang, Xin Zhang, Zhen Wu, Meishan Zhang, Xinyu Dai, Qingsong Wen, Wei Ye, and 1 others. 2024. Autosurvey: Large language models can automatically write surveys. *Advances in neural information processing systems*, 37:115119–115145.

Jason Wei, Zhiqing Sun, Spencer Papay, Scott McKinney, Jeffrey Han, Isa Fulford, Hyung Won Chung, Alex Tachard Passos, William Fedus, and Amelia Glaese. 2025. Browsecomp: A simple yet challenging benchmark for browsing agents. *Preprint*, arXiv:2504.12516.

x.AI. 2025. Grok 3 beta — the age of reasoning agents.

Zekun Xi, Wenbiao Yin, Jizhan Fang, Jialong Wu, Runnan Fang, Ningyu Zhang, Jiang Yong, Pengjun Xie, Fei Huang, and Huajun Chen. 2025. Omnithink: Expanding knowledge boundaries in machine writing through thinking. *arXiv preprint arXiv:2501.09751*.

Xiangchao Yan, Shiyang Feng, Jiakang Yuan, Renqiu Xia, Bin Wang, Bo Zhang, and Lei Bai. 2025. Surveyforge: On the outline heuristics, memory-driven generation, and multi-dimensional evaluation for automated survey writing. *arXiv preprint arXiv:2503.04629*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Peilin Zhou, Bruce Leon, Xiang Ying, Can Zhang, Yifan Shao, Qichen Ye, Dading Chong, Zhiling Jin, Chenxuan Xie, Meng Cao, and 1 others. 2025. Browsecomp-zh: Benchmarking web browsing ability of large language models in chinese. *arXiv preprint arXiv:2504.19314*.

A Method Details

A.1 The Prompts in WARP Framework

In our WARP framework, there are five actions in all three stages: *Initialize*, *Search*, *Write*, *Expand(Deepen)*, and *Terminate*. In the ***Initialization*** stage, the agent generates the initial Level-1 outline with writing plans by the *search* and *initialize* actions. The prompt for the ***search*** is shown in Figure 8, and the prompt for the ***initialize*** is shown in Figure 7. Then, in the ***Evidence-Based Drafting*** stage, the agent writes the paragraphs by the *search* and *write* actions. The prompt for the ***write*** is shown in Figure 9. After that, in the ***Reasoning-Driven Deepening*** stage, the agent will decision whether to expand a section for more details by the ***expand*** action or to end the total process directly by the ***terminate*** action. The prompt for both actions is shown in Figure 10.

A.2 User Query Construction

We constructed a dataset of approximately 2000 user queries with corresponding scoring checklists to support the multi-stage training process. Of these, around 700 queries are focused on specialized academic survey topics, while the remaining 1300 address general research reporting topics.

```
You are a professional report generation expert, skilled at creating high-quality report outlines.
Now, you need to analyze the users question and provide a simple article outline structure (only
top-level sections).

** User Query **
[user query]

** Latest Retrieved Information **
[current information]


## Notes
1. The outline must be comprehensive, logically sound, and aligned with the users stated
preferences and requirements.
2. The output language must match the language of the users query.

** Available Actions **
- initialize: Generate the top-level section outline along with an appropriate title.

## Action Format:
<action> {"name": "initialize", "title": "...", "sections": [{"title": "...", "plan": "..."}, {"
title": "...", "plan": "..."}, ...]} </action>

** Output Format **
<thought> Provide a detailed reasoning process </thought>
<action> Action (in JSON format) </action>

Please output strictly according to the specified format.
```

Figure 7: The LLM prompt for the *initialize* action.

For the academic survey queries, we employed a *reverse question construction* approach: we first selected 700 surveys from ArXiv and then used a large model to generate user questions based on these articles. The prompts used for this process are available in Figure 11. For general research reporting queries, we selected 1300 real questions.

In addition, we constructed **a preference checklist** inspired by DeepResearch Bench (Du et al., 2025) for each user query. A large model was used to generate weighted scores for different evaluation aspects based on the existing questions. The final report score for a query is computed as a weighted sum across these aspects. The method for checklist generation is same as DeepResearch Bench. Out of the 2000 queries, approximately 1500 were used to **construct trajectory data** for SFT and single-step RL, while the remaining 500 were directly used for end-to-end RL training.

A.3 Retrieval Environment Setup

We constructed a local database containing approximately 2.86 million documents to serve as the agent's retrieval environment. This database supports both trajectory data collection and interactive RL training. Among these documents, roughly 2.71 million are abstracts of papers from ArXiv, sourced via Kaggle[1] . The remaining 150k documents come from general web pages, for which we employed *Gemini 2.0-Flash* to generate concise summaries while controlling for document length and quality.

To ensure efficient retrieval, we built a vector database. Specifically, all documents were vectorized using the embedding model `MiniCPM-Embedding-Light` [2] and indexed with Faiss[3]. The pipeline is implemented via `UltraRAG`(Chen et al., 2025), an open-source library for constructiong Retrieval-Augmented Generation (RAG) systems.

A.4 Trajectory Data Construction

Base on the user queries in §A.2 and the retrieval environment in §A.3, we collected 1,500 actual execution trajectories within our WARP framework.

We chose *Qwen3-235B-A22B-Instruct-2507* as the teacher model, and use the prompts in §A.1. Despite their scale, current large language models

[1] https://www.kaggle.com/api/v1/datasets/download/Cornell-University/arxiv
[2] https://huggingface.co/openbmb/MiniCPM-Embedding-Light
[3] https://github.com/facebookresearch/faiss

```
You are a searcher within a multi-agent system consisting of "Analyst-Searcher-Writer". You must
perform retrieval based on instructions from the "Analyst". Carefully select the most accurate
search keywords and strictly adhere to the specified output format.
You should focus on the user's query and the current article outline to determine the most
relevant keywords for searching. You can give one or less to five keywords. The content should be
in the same language as the user's query.

** User Query **
[user query]


** Current Article Outline **
[current outline]

** Analyst's Instruction **
[current instruction]

## Action Example:
<action> {"name": "search", "keywords": [keyword-1, keyword-2, ..]} </action>


** Output Format **
<thought> Your reasoning process </thought>
<action> Action (in JSON format) </action>

Please output strictly according to the specified format.
```

Figure 8: The LLM prompt for the *search* action.

still struggle with high-level decision-making and cannot reliably determine when to stop. To address this, we introduce an **trajectory pruning** strategy. The gathering process here slightly differs from the standard inference phase of the WARP framework: during each ***Reasoning-Driven Deepening*** stage, we explicitly force the agent to select a position for outline deepening, instead of allowing the model to autonomously decide whether to expand or terminate. This ensures that the agent continuously expands the outline and revises the report until a maximum of 12 expansions is reached. All the results are scored by the survey-level reward mentioned in §A.6.2, and the highest-scoring result is selected as the endpoint of the trajectory.

For each user query, we collect a single high-quality execution trajectory. In total, 1500 trajectories were obtained corresponding to the 1500 user queries. Among these, 1200 trajectories were used as SFT data for cold-start training, while the remaining 300 were reserved for atomic skill RL.

A.5 Action Data Distribution

For a complete trajectory collected in Section A.4, it typically contains one *initialize* action, one *termination* action, several *expand* actions, and many *search* and *write* actions. These actions correspond to four core agent capabilities: *planning*, *retrieval*, *writing*, and *decision-making*. However, the natural distribution of actions is highly imbalanced with respect to training needs: the more critical and challenging abilities—such as *planning* and *decision-making*—are underrepresented, while easier-to-learn abilities—such as *searching* and *writing*—dominate the trajectories.

To address this issue, we introduce an **action-level balanced sampling** strategy that increases the sampling probability of more important and difficult actions and decreases that of easier ones, thereby providing more effective supervision for training the agent's core capabilities.

From the 1,500 collected trajectories, we obtained approximately 100k actions in total. When grouped by user query type, actions from academic review tasks and general report tasks followed an approximate ratio of 3:5. We used about 33k actions for cold-start training (SFT) and about 5k actions for atomic skill RL, with the remaining data discarded.

A.6 Reward System

In Reinforcement Learning (RL) training, the design of the reward functions is crucial, often directly impacting training efficiency and stability. In this section, we introduce our reward system from two aspects. First, in Section A.6.1, we introduce

our different ability-specific reward functions for four abilities: **planning, retrieval, writing**, and **decision-making**, primarily used for single-step RL training. These functions are used for optimizing five actions: *initialize, expand-plan, search, write*, and *terminate*, as shown in Table 7. Second, in Section A.6.2, we introduce report-level reward design, which scores the overall quality of the generated report, primarily used for end-to-end RL training.

We use GRPO (Shao et al., 2024b) as our RL algorithm for both Atomic Skill RL and Holistic Pipeline RL. Notably, we set the KL divergence coefficient to 0 (i.e., no KL penalty), allowing the agent to explore freely beyond the reference policy. To mitigate noise from LLM-based judgment, we evaluate each reward aspect across multiple dimensions and average the scores, which stabilizes the reward signal.

A.6.1 Rewards for Action Optimization

In this section, we introduce our reward design for action optimization, aimed at improving the agent's four core capabilities. For LLM as Judgement, the judgment model we use is Qwen2.5-72B-Instruct.

(1) Planning capability. Both *Initialize* and *Expand* actions reflect the planning ability, and we calculate the rewards for these two actions by evaluating their generated one-level outlines (with detailed writing plans for each section). We use **basic properties**, **holistic quality**, and **faithfulness** to evaluate the results.

Outline Basic Properties mainly evaluates whether the number of sections in the outline is reasonable, and judges the language consistency. We constrain the number of subsections in a section to between 2 to 7. Meanwhile, we control the language consistency by character statistics.

Outline Holistic Quality is evaluated by LLM for three aspects, following the setting of OmniThink (Xi et al., 2025). The scoring criteria are in Table 10 and the prompt Figure 12:

- ***Guidance for Content Generation*** Does the outline effectively guide content generation, ensuring comprehensive coverage of the topic?
- ***Hierarchical Clarity*** Does the outline clearly define a hierarchy of topics and subtopics, with a logical, diverse structure that is easy to understand?
- ***Logical Coherence*** Does the outline logically organize topics and subtopics, ensuring a smooth and natural flow of ideas with clear logical transitions?

Outline Faithfulness (Min et al., 2023) verifys whether the content in the plan is real and reliable.

For the final reward computation of planning actions (*Initialize* and *Expand*), if the basic properties are not met, the reward is 0. Otherwise:

$$R_{\text{plan}} = 1.5 \times R_{\text{holistic}} + 3 \times R_{\text{faithful}} \quad (6)$$

where $R_{\text{holistic}} \in [1, 5]$ is the average holistic quality score and $R_{\text{faithful}} \in [0, 1]$ is the faithfulness score.

(2) Retrieval capbility. *Search* action reflects the retrieval ability, and we calculate the rewards for the action by evaluating its generated search keywords. We use **recall score** to evaluate them.

Recall Score is calculated by the comparison of the retrieved documents and the golden ones.

(3) Writing capability. The *Write* action reflects the writing ability, and we calculate the rewards for the action by evaluating its generated paragraphs. We use **basic properties**, **holistic quality**, **faithfulness**, and **citation precision** to evaluate them.

Paragraph Basic Properties mainly evaluates whether the length and the number of citations of a paragraph are reasonable, and judges the language consistency. We constrain the length of a paragraph to between 100 to 2000 tokens. Then, we constrain the citation number of a paragraph to between 0 to 12. Finally, we control the language consistency by character statistics.

Paragraph Holistic Quality is evaluated by LLM for four aspects, according to STORM (Shao et al., 2024a), with the prompt Figure 13 and the critria Table 11 :

- ***Relevance*** How effectively does the report maintain relevance and focus, given the dynamic nature of the discourse?
- ***Coverage*** Does the article provide an in-depth exploration of the topic and have good coverage?
- ***Depth*** How thoroughly does the report explore the initial topic and its related areas, reflecting the dynamic discourse?

Table 7: The mapping between actions and the four agent abilities.

Agent Ability	Action Name	Action Parameters
planning	initialize	{"title": "...", "sections": [{"title": "...", "plan": "..."}, {"title": "...", "plan": "...",}, ...]}
	expand	{"position": "section-x.y.z", "content": "...", "subsections": [{"title": "...", "plan": "..."}, ...]}
retrieval	search	{"keywords": [keyword-1, keyword-2, ..,]}
writing	write	{"position": "section-x.y.z", "title": "...", "content": "..."}
decision-making	terminate	{}

- ***Novelty*** Does the report cover novel aspects that relate to the user's initial intent but are not directly derived from it?

Paragraph Faithfulness checks whether the fact claims are consistent with citations.

Citation Precision is calculated by the comparison of the selected citations and the golden ones. We first check the citation hallucination phenomenon, any hallucination will make the score as 0. Then we employ the F1 Score of the citations between the generated content with the golden one.

For the final reward computation of the *Write* action, if the basic properties are not met or hallucinated citations exist, the reward is 0. Otherwise:

$$R_{\text{write}} = 2\times R_{\text{holistic}} + 3\times R_{\text{faithful}} + 3\times R_{\text{citation}} \quad (7)$$

where $R_{\text{holistic}} \in [1, 5]$ is the average holistic quality score, $R_{\text{faithful}} \in [0, 1]$ is the faithfulness score, and $R_{\text{citation}} \in [0, 1]$ is the citation precision score.

(4) Decision-making capbility. The *terminate* action reflects the agent's high-level decision-making ability, and its reward is computed by evaluating the correctness of this decision. We use **accuracy** as the metric: if the agent's decision matches the reference answer, the reward is 1.0; otherwise, it is 0.0.

A.6.2 Rewards for Pipeline Optimization

Unlike the action-level RL phase, the process-level RL phase no longer has a reference answer to constrain the agent's exploration, and the training data only contains user questions. We will directly evaluate the final result of the entire process (the generated report) from several aspects, instead of scoring the intermediate actions. We hope that in this phase, the agent can explore more freely, generate more diverse results, and ultimately surpass the capabilities of the teacher model. This section will detail our evaluation of the final result from four aspects: comprehensiveness, insight, instruction-following, and readability. The judgment model we use is *Qwen3-32B* (Yang et al., 2025).

B Experiments Details

B.1 Training Details

We adopt a three-stage training pipeline consisting of cold start training (SFT), atomic skill RL (single-step RL), and holistic pipeline RL (end-to-end RL). All our experiments are run on 8 A100 GPUs, and the training settings are shown in Table 8.

Cold-Start Training For cold-start, we collect approximately 33k action-level samples from 1,200 trajectories. The model is trained using SFT with a learning rate of 1.5e-5 and batch size as 32 for 4 epochs, taking about 2 days to complete.

Atomic Skill RL We further perform single-step RL using approximately 5150 action-level samples from 300 trajectories. We set the learning rate to 2.5e-6, batch size to 8, rollout number to 8, and train for 200 optimization steps, taking about 2 days to complete.

Holistic Pipeline RL Finally, we conduct end-to-end RL on 500 user queries, optimizing the entire report generation pipeline jointly. The learning rate remains 1e-6, with a batch size of 8, rollout number of 4, and a total of 50 training steps, taking about 4 days to complete.

Implementation Details of Baselines The teacher agent used for trajectory data collection (§A.4) is built upon *Qwen3-235B-A22B-Instruct-2507* as its backbone. For the trained open models compared in our experiments, the backbone architectures are as follows: Qwen3-8B is used for DR-Tulu (Shao et al., 2025), QwQ-32B for WebShaper (Tao et al., 2025), and QwQ-32B for WebThinker (Li et al., 2025b). The results for DR-Tulu and WebThinker are reproduced from the DR-Tulu paper, while the WebShaper results are taken from the WebWeaver (Li et al., 2025c) paper.

B.2 Metrics Details

We conducted evaluations on three benchmarks: DeepResearch Bench (Du et al., 2025), DeepCon-

Table 8: The training settings for different stages.

Parameters	SFT	Single-Step RL	End-to-End RL
user queries	1,200	300	500
train samples	33,292	5150	500
learning rate	1.5e-5	2.5e-6	1e-6
batch size	32	8	8
rollout number	–	8	4
train epochs	4	–	–
train steps	–	200	50

sult (Dee, 2025), and DeepResearch Gym (Coelho et al., 2025).

DeepResearch Bench It consists of 100 PhD-level research tasks spanning 22 academic domains. It adopts the RACE and FACT evaluation frameworks. RACE assesses Comprehensiveness, Insight/Depth, Instruction Following, and Readability, while FACT evaluates effective citations per report and citation reliability. We evaluate it by *Gemini-2.5-Pro*.

Deep Consult It includes 102 queries from business and consulting scenarios. Evaluation is conducted via pairwise comparisons against an *OpenAI-DeepSearch* baseline, reporting win, tie, and loss rates, together with average quality scores on instruction following, comprehensiveness, completeness, and writing quality. We evaluate it by *o3-mini-2025-01-31*.

DeepResearch Gym It is built on the Researchy Questions dataset. Following WebWeaver (Li et al., 2025c), we sample 100 queries from the top 1,000 test queries and evaluate the report quality in six aspects: clarity, depth, balance, breadth, support, and insightfulness. We evaluate it by *GPT-4.1-mini-20250414*.

B.3 Analysis of Judge Model Choice

A potential concern is whether the choice of judge model might affect the relative ranking of systems, particularly when some baselines share the same backbone as the judge. In the original DeepResearch Bench paper (Du et al., 2025) (Section 4.3.4), the authors conduct a comparative analysis involving human evaluators alongside different judge models—Gemini-2.5-Pro, GPT-o3, GPT-o4-mini, and Claude-3.7-Sonnet. Four metrics are used for evaluation: PAR (Pairwise Agreement Rate), OPC (Overall Pearson Correlation), FAP (Filtered Average Pearson), and FAS (Filtered Average Spearman). Among them, Gemini-2.5-Pro achieves the highest overall score of 72.56, as shown in Table 9.

Beyond DeepResearch Bench, we employ GPT-o3-mini as the judge in DeepConsult, and GPT-4.1 as the judge in DeepResearch Gym. In our observations, we do not find evidence that a judge model assigns higher scores to models from the same series versus those from different series. Therefore, to maintain consistency with prior work, we retain the default judge for each benchmark.

Table 9: Comparison of different judge LLMs on DeepResearch Bench (from Du et al. 2025).

Judge LLM	PAR	OPC	FAP	FAS	Overall
Gemini-2.5-Pro	71.33	99.54	60.24	59.12	**72.56**
o3	68.11	96.22	57.64	52.36	68.58
o4-mini	70.89	97.06	59.54	59.02	71.63
Claude-3.7-Sonnet	70.78	96.53	58.22	63.61	72.28

```
You are a writer operating within a multi-agent system consisting of "Analyzer-Searcher-Writer".
Based on instructions from the "Analyzer", the current writing status, and the most recently
retrieved information, you are to compose a new paragraph while ensuring logical coherence and
accurate citation of facts.
You should give a paragraph with breadth and depth, ensuring it is informative and engaging. You
are encouraged to incorporate examples, **tables**, code snippets, and other elements to enhance
the content. But don't write other sections or chapters that are not assigned to you.
You'd better give analytical and comparative content, not just a summary of facts. Please
attention the coherence and logical flow of the entire article and the other sections.

You can extract the claims from the retrieved information, and design how to write the paragraph
based on the claims in the thought process.
**BE FAITHFUL! Make sure all your claims, especially the facts and the numbers, can be supported
by the retrieved information, with your citations. Don't add any claims can't be supported by your
 citations. All the facts or data in your claims should can be found in the retrieved information
you cited.**
**You should ensure that the content you write is not redundant with other sections.**
**And you should strictly follow the citation format like  \\cite{bibkey} or \\cite{bibkey1,
bibkey2..} for any referenced information. The content should be in the same language as the user'
s query.**


PLEASE JUST OUTPUT THE CONTENT IN ANALYZER'S INSTRUCTION, DO NOT OUTPUT OTHER SECTIONS.

THE OUTPUT SHOULD BE IN THE SAME LANGUAGE AS THE USER'S QUERY.


** User Query **
[user query]

** Current Article Summary **

[current survey]

** Analyzers Instruction **
[current instruction]

** Retrieved Information **
[current information]

## Action Example:
<action> content </action>


** Output Format **
<thought> Your thought process </thought>
<action> Your Content (in Markdown format) (include BIBKEY for citations within the content) </
action>

Please strictly follow the specified output format.
```

Figure 9: The LLM prompt for the *write* action.

```
You are a professional report-generation expert skilled at crafting high-quality report outlines.
Based on the the users stated preferences, you must now determine whether any section requires
expansion into subsections.

## Important Notes:
1. Select only the single section or subsection most in need of expansion.
2. If no expansion is needed, output a "terminate" (no operation) action.
3. If you think the expendsion is necessary to make the article more comprehensive or insightful,
feel free to expand it.
4. Make sure the new subsections aren't redundant or overly detailed with other sections. If it's
too detailed or redundant with other sections, just terminate it.
5. Make sure the new subsections are relevant and coherent with other sections.
6. You can only expand the section in 1 level and 2 level, do not expand the section in 3 level or
 more.
7. Please don't extend the section that is already extended.
8. Just extend one hierarchy level at a time, the subsections you give should not have more than
one hierarchy level.
8. The output language must match the language of the users query.


** User Query **
[user query]

** Current Full Report **
[current survey]

** Available Actions **
- extend-plan: Expand a section by adding subsections (e.g., section-1 to section-1.1, section-1.2,
 section-1.3).
- terminate: No operation.


## Action Format:
<action> {"name": "expand", "position": "section-x.y.z", "subsections": [{"title": "...", "plan":
"..."}, {"title": "...", "plan": "..."}, ...]} </action>
<action> {"name": "terminate"} </action>

** Output Format **
<thought> Provide a detailed reasoning process </thought>
<action> Action (in JSON format) </action>


Please output strictly according to the specified format.
```

Figure 10: The LLM prompt for the *expand* and *terminate* actions.

```
You are a Instruction-writing expert. Below is a Survey title. Your task is to infer the
Instruction the user might have.

Survey Title:
{title}

Before crafting the Query, analyze the following:
1. **Avoid using exact titles.**
2. Use domain-specific keywords, synonyms.

First, output your analysis starting with "Thought:", simulating the Survey authors thought
process.
Then, generate one or more Queries based on the analysis, starting with "Instruction:". DONT
OUTPUT EXPLANATIONS AFTER THE Instruction.
```

Figure 11: The LLM prompt for reverse question (user query) construction.

Rating	Description
	Guide
Score 1	The outline fails to guide content generation, omitting significant aspects of the topic or providing insufficient direction.
Score 2	The outline provides limited guidance, covering some key areas but lacking depth or completeness in addressing the topic.
Score 3	The outline provides moderate guidance for content generation, addressing most key areas but leaving some gaps or ambiguities.
Score 4	The outline effectively guides content generation, covering all significant aspects with clear direction, though minor refinements could enhance comprehensiveness.
Score 5	The outline is exemplary in guiding content generation, thoroughly addressing all aspects of the topic with clear, detailed direction and no significant gaps.
	Hierarchical
Score 1	The outline exhibits no discernible hierarchical structure. Topics and subtopics are jumbled together without logical separation or clear levels, making it nearly impossible to follow or identify any organization.
Score 2	The outline attempts to establish a hierarchy but fails to maintain logical consistency. Main topics and subtopics are frequently misclassified, and the structure is overly rigid or disjointed. Subtopics may be missing, misplaced, or redundant, making it hard to grasp the intent of the structure.
Score 3	The outline demonstrates a basic level of logical coherence. Most topics follow a general sequence, but some sections feel forced, with weak or unclear transitions. There are small jumps in logic, causing slight confusion or loss of flow at certain points.
Score 4	The outline displays a clear, logical, and diverse hierarchical structure. Main topics are distinct, and subtopics are properly nested. While most elements are well-placed, there may be minor redundancies or opportunities to introduce more diverse formats for subtopics. Slight adjustments could achieve better precision and variety in style.
Score 5	The outline showcases an exceptional, flawless hierarchical structure. Each main topic is distinct, and subtopics are logically nested with absolute clarity and stylistic diversity. The outline demonstrates flexibility in structure and organization, adapting its style where appropriate for the content and logic. No further refinement is necessary.
	Coherence
Score 1	The outline is highly disjointed and incoherent. Topics and subtopics appear in a random, unordered manner, with no logical flow or sense of progression. Major conceptual gaps and illogical jumps are present throughout the structure.
Score 2	The outline shows some attempt at logical organization, but it contains frequent inconsistencies, abrupt shifts, or logical missteps. Topics and subtopics are misaligned or lack proper transitions, making the reader work hard to follow the structure.
Score 3	The outline demonstrates a basic level of logical coherence. Most topics follow a general sequence, but some sections feel forced, with weak or unclear transitions. There are small jumps in logic, causing slight confusion or loss of flow at certain points.
Score 4	The outline exhibits a strong sense of logical flow, with ideas presented in a mostly smooth and connected manner. Transitions between topics and subtopics are clear, but a few minor adjustments could make the flow more seamless or natural. The logic is sound, but room for refinement exists.
Score 5	The outline achieves exceptional logical coherence. Each topic and subtopic follows a deliberate, thoughtful progression, with clear, natural, and intuitive transitions. The reader experiences a seamless flow of ideas, and no adjustments are required to improve logical consistency or flow.

Table 10: The plan scoring criteria rating scale 1-5.

```
###Task Description:
An instruction (might include an Input inside it), a response to evaluate, a reference answer that
 gets a score of 5, and a score rubric representing a evaluation criteria are given.
1. Identify the major and minor errors in this Response. Write a detailed list of the errors in
the response strictly based on the given score rubric, not evaluating in general.
2. After writing the list of errors, write a score that is an integer between 1 and 5. You should
refer to the score rubric.
3. The output format should look as follows: "(write the list of errors for criteria) [RESULT] (an
 integer number between 1 and 5)"
4. Please do not generate any other opening, closing, and explanations.
5. Please be fair, don't hesitate to give a low score like 1 or 2.
6. Note that Major errors refer to actual errors that affects the task severely, may change the
meaning of the output, and Minor errors refer to smaller imperfections, and purely subjective
opinions about the output.

###The instruction to evaluate:
{instruction}

###Response to evaluate:
{response}

###Reference Answer (Score 5):
{reference_answer}

###Score Rubrics:
{rubric}

###Feedback:
```

Figure 12: The Outline quality reward prompt template.

```
Here is an academic survey about the topic "[TOPIC]":
---
[SURVEY]
---

<instruction>
Please evaluate this survey about the topic "[TOPIC]" based on the criterion provided below,
identify the major and minor errors in this survey, and give a score from 1 to 5 according to the
score description:
---
Criterion Description: [Criterion Description]
---
Score 1 Description: [Score 1 Description]
Score 2 Description: [Score 2 Description]
Score 3 Description: [Score 3 Description]
Score 4 Description: [Score 4 Description]
Score 5 Description: [Score 5 Description]
---
Note that Major errors refer to actual errors that affect the task severely, may change the
meaning of the output, and Minor errors refer to smaller imperfections, and purely subjective
opinions about the output.
There may be multiple errors or no errors in the output.
After listing the errors, then, please score the survey with 1 to 5.
Return the score without any other information at the end of the output.
```

Figure 13: The Content quality reward prompt template.

Rating	Description
	Relevance
Score 1	Very poor focus; discourse diverges significantly from the initial topic and intent with many irrelevant detours.
Score 2	Poor focus; some relevant information, but many sections diverge from the initial topic.
Score 3	Moderate focus; mostly stays on topic with occasional digressions that still provide useful information.
Score 4	Good focus; maintains relevance and focus throughout the discourse with minor divergences that add value.
Score 5	Excellent focus; consistently relevant and focused discourse, even when exploring divergent but highly pertinent aspects.
	Coverage
Score 1	Severely lacking; offers little to no coverage of the topic's primary aspects, resulting in a very narrow perspective.
Score 2	Partial coverage; includes some of the topic's main aspects but misses others, resulting in an incomplete portrayal
Score 3	Acceptable breadth; covers most main aspects, though it may stray into minor unnecessary details or overlook some relevant points.
Score 4	Good coverage; achieves broad coverage of the topic, hitting on all major points with minimal extraneous information.
Score 5	Exemplary in breadth; delivers outstanding coverage, thoroughly detailing all crucial aspects of the topic without including irrelevant information.
	Depth
Score 1	Very superficial; provides only a basic overview with significant gaps in exploration.
Score 2	Superficial; offers some detail but leaves many important aspects unexplored.
Score 3	Moderate depth; covers key aspects but may lack detailed exploration in some areas.
Score 4	Good depth; explores most aspects in detail with minor gaps.
Score 5	Excellent depth; thoroughly explores all relevant aspects with comprehensive detail, reflecting a deep and dynamic discourse.
	Novelty
score 1	Lacks novelty; the report strictly follows the user's initial intent with no additional insights.
score 2	Minimal novelty; includes few new aspects but they are not significantly related to the initial intent.
score 3	Moderate novelty; introduces some new aspects that are somewhat related to the initial intent.
score 4	Good novelty; covers several new aspects that enhance the understanding of the initial intent.
score 5	Excellent novelty; introduces numerous new aspects that are highly relevant and significantly enrich the initial intent.

Table 11: The content scoring criteria rating scale 1-5.

C Additional Experiments and Analysis

C.1 Generalization Across Model Scales

To demonstrate the robustness of WARP beyond the 8B scale, we conduct additional experiments with both smaller and larger backbones.

Smaller Scale. We train a 4B model using *Qwen3-4B-Instruct-2507* as the backbone and apply the same multi-stage training pipeline. As shown in Table 12, even at the 4B scale, our approach achieves performance comparable to the closed-source deep research system Gemini-2.5-Pro-deepresearch, demonstrating the efficiency of our training pipeline.

Table 12: Performance of WARP-4B (Qwen3-4B backbone) at different training stages on DeepResearch Bench.

Agent System	Overall	Comp.	Insight	Inst.	Read.
OpenAI DeepResearch	46.45	46.46	43.73	49.39	47.22
Gemini DeepResearch	49.71	49.51	49.45	50.12	50.00
WARP-4B (SFT)	47.96	47.64	49.64	47.95	43.93
WARP-4B (Atomic RL)	48.97	48.99	52.04	48.32	41.70
WARP-4B (Pipeline RL)	49.21	49.66	52.30	48.36	41.06

Larger Scale and API-Based Models. We further evaluate the WARP framework by applying it directly to *Gemini-2.5-Pro* via API during inference (without training). As shown in Table 13, our framework remains effective when deployed on API-based large language models, with consistent gains over Gemini DeepResearch, confirming that the WARP paradigm is not overfitting to a single backbone.

Table 13: Performance of WARP applied to larger and API-based models on DeepResearch Bench.

Agent System	Overall	Comp.	Insight	Inst.	Read.
Gemini DeepResearch	49.71	49.51	49.45	50.12	50.00
WARP (Qwen3-235B-A22B)	50.72	50.33	52.79	50.32	47.20
WARP (Gemini-2.5-Pro)	**50.87**	49.43	**53.37**	**50.47**	49.17

C.2 Necessity of Action Balancing

In §A.5, we introduce an action-level balanced sampling strategy. To validate its necessity, we compare training with and without this strategy using MiniCPM4.1-8B as the backbone. As shown in Table 14, employing the action balancing strategy significantly outperforms using the original unbalanced data. Table 15 shows the action counts before and after balancing: the original data is dominated by *Search* and *Write* actions, while the balanced version prunes redundant instances to give critical planning and decision-making actions more relative weight.

Table 14: Effect of action balancing on SFT training (DeepResearch Bench).

Data	Overall	Comp.	Insight	Inst.	Read.
w/ action balancing	**46.73**	**46.24**	**48.10**	**47.61**	**41.79**
w/o action balancing	44.45	43.89	45.02	46.54	39.46

Table 15: SFT action counts with and without balancing.

Setting	Sum	Search	Write	Init.	Expand	Term.
Balanced	33,292	2,856	23,000	1,158	5,236	1,042
Unbalanced	93,698	43,601	42,661	1,158	5,236	1,042

C.3 Necessity of Multi-Stage Training

To verify that each stage in our training curriculum is indispensable, we conduct an ablation experiment where we skip the *Atomic Skill RL* stage and directly apply *Pipeline RL* after SFT. As shown in Table 16, skipping Atomic RL leads to substantially lower performance compared to the full three-stage pipeline. Specifically, the model that skips Atomic RL achieves an *Overall* score of only 47.15, compared to 50.11 for the full pipeline. This confirms that Atomic RL provides essential skill-level training that the holistic Pipeline RL alone cannot replace: without first mastering atomic actions (planning, retrieval, writing), the agent lacks the foundation needed for effective end-to-end optimization.

Table 16: Ablation: skipping Atomic RL on DeepResearch Bench.

Agent System	Overall	Comp.	Insight	Inst.	Read.
WARP-8B (SFT)	46.73	46.24	48.10	47.61	41.79
WARP-8B (Atomic RL)	48.81	48.70	51.36	48.64	42.25
WARP-8B (Pipeline RL)	**50.11**	**50.54**	**52.64**	**48.87**	**44.17**
WARP-8B (Pipeline RL, skip Atomic RL)	47.15	46.79	49.11	47.47	41.02

D The Usage of LLMs

We use LLMs (e.g., ChatGPT) to help polish the paragraphs of our paper.