RIVO: LIGHTWEIGHT ADAPTIVE REROUTING FOR DEEP RESEARCH AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Deep Research agents rely on long-horizon reasoning and iterative tool use to solve complex information-seeking tasks. However, adaptive decision-making over these trajectories remains challenging, as agents may stop prematurely when further evidence is still needed, continue unnecessarily after sufficient evidence has been gathered, or fail to redirect unproductive trajectories. Existing solutions either require costly model retraining and dense turn-level supervision, or rely solely on heuristic control rules that are not directly optimized for downstream research utility. This motivates a lightweight, utility-aware alternative for deep research control. We formulate adaptive rerouting in long-horizon Deep Research as a state-conditioned trajectory correction problem. Based on this view, we introduce RIVO, a lightweight plug-in framework for adaptive trajectory correction. The agent first estimates its research state from lightweight state signals and determines whether trajectory redirection is needed. Once triggered, the plug-in modifies the Host token logits to steer generation toward a more promising research direction. Theoretical analysis shows that, under a value-alignment condition and bounded local curvature, a sufficiently small logit intervention can locally improve the expected downstream research utility. A curated dataset, BrowseCore, with a paired local corpus is further constructed for efficient and controlled Deep Research experimentation. Building on this resource, the plug-in is trained from downstream outcomes, enabling it to learn corrections that improve subsequent research and final task performance. Extensive experiments across multiple Deep Research benchmarks and control settings demonstrate the effectiveness of our method, showing that lightweight logit intervention can improve long-horizon research without retraining the underlying Host model[1].

1 INTRODUCTION

Large language models (LLMs) have evolved from passive text generators into autonomous agents that interleave reasoning with external tool use, such as web search and knowledge retrieval (Alzubi et al., 2025; Li et al., 2026b;c). Building on this capability, *Deep Research* agents perform long-horizon, iterative retrieval and reasoning to solve complex information-seeking tasks that require multi-hop evidence acquisition and synthesis (Dong et al., 2026; Li et al., 2026d). These agents are increasingly useful for applications such as web-scale question answering and patient diagnosis and clinical decision support (Wang et al., 2026c). However, current Deep Research systems still suffer from poor search control, as agents may continue searching after sufficient evidence has been collected, stop prematurely when further evidence is still needed, or remain on an unproductive research path instead of redirecting toward more useful evidence (Soudani et al., 2026). This highlights a key limitation of current Deep Research agents. Long-horizon research requires reliable decisions on when to stop, when to reroute, and how to redirect the search toward more promising directions.

A natural way to improve research control is to further train the underlying Deep Research model. Supervised fine-tuning can teach explicit intervention behaviors, but it requires high-quality trajectories that specify not only when the agent should stop or reroute, but also how the research trajectory should be redirected. Collecting such supervision becomes increasingly difficult as research trajectories

[1]The code and data are available at `https://anonymous.4open.science/r/RIVO-56CC/README.md`

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.

extend to dozens or even hundreds of interaction steps (Deng et al., 2024a). Reinforcement learning offers a more flexible alternative by optimizing search behavior from task rewards, yet long-horizon optimization introduces sparse and delayed feedback, difficult credit assignment, and substantial training cost due to repeated reasoning and retrieval rollouts (Wang et al., 2026a). These costs become particularly significant when each step invokes large local retrievers or external APIs, and the resulting policy is typically coupled to a specific research backbone, requiring re-optimization when the underlying system changes. Training-free approaches avoid such retraining by relying on prompting or predefined internal and retrieval signals (Yao et al., 2023; Li et al., 2026a). However, these heuristic or proxy signals are not directly optimized for downstream research reward, and therefore cannot explicitly learn the long-term utility of an intervention, whether the agent should stop, whether rerouting is beneficial, and, critically, which alternative search direction should be pursued. These limitations naturally raise the following question: *Can we learn a lightweight, plug-and-play research controller that improves decision-making in long-horizon Deep Research without retraining the underlying agent or requiring substantial computational resources?*

Motivated by recent advances in inference-time logit reweighting for frozen language models (Hiranandani et al., 2025), we formulate adaptive control in long-horizon Deep Research as a *state-conditioned trajectory correction* problem. Based on this formulation, we introduce Reward-Learned Logit Intervention for Value Optimization (RIVO), a lightweight plug-in framework for adaptive trajectory correction. Rather than retraining the underlying research model, RIVO estimates the current research state from lightweight state signals to determine whether intervention is needed. When trajectory redirection is triggered, the plug-in applies a token-level correction to the frozen model logits, steering generation toward a more promising research direction. RIVO is trained from downstream trajectory outcomes, allowing it to learn interventions that improve subsequent research and final task performance.

The key component of RIVO is a sparse logit intervention for trajectory correction. A lightweight plug-in generates a low-rank residual correction that is sparsely applied to the Top-K selected logit dimensions, while the Host model, i.e., the underlying Deep Research model, remains frozen. This intervention directly changes selected token preferences and redirects subsequent retrieval without end-to-end retraining. To support model development and controlled evaluation, we construct BrowseCore, a Deep Research dataset with a paired local corpus. Across the evaluated Deep Research benchmarks, RIVO outperforms strong baselines by up to 4.36 points. Further ablation studies validate the key components of RIVO. For example, on BrowseCore, removing the reroute predictor reduces the Success Rate by 5.67 points, while removing the logit intervention leads to a 2.48-point drop. Finally, theoretical analysis studies the effect of logit intervention and shows that sufficiently small, value-aligned interventions can locally improve expected downstream research utility.

- We construct a curated question answer dataset paired with a local corpus to support efficient and controlled training and evaluation of long-horizon Deep Research agents.
- We formulate adaptive rerouting in long-horizon Deep Research as a *state-conditioned trajectory correction* problem and introduce RIVO, a lightweight plug-in framework for targeted trajectory redirection. Theoretical analysis shows that sufficiently small, value-aligned logit interventions can locally improve expected downstream research utility.
- We conduct extensive experiments across multiple Deep Research benchmarks and control settings. The results demonstrate the effectiveness of RIVO in improving research performance while avoiding end-to-end retraining of the underlying research model.

2 Related Work

Deep Research Agents and Research Control. Recent Deep Research agents, including LiteResearcher-4B (Li et al., 2026d), Marco-DeepResearch-8B (Zhu et al., 2026), and Tongyi-DeepResearch-30B-A3B (Team, 2025), develop long-horizon research capabilities through supervised and reward-based post-training. Despite strong overall performance, these agents still exhibit limitations in intermediate research control: they may terminate at suboptimal points, continue after meaningful progress has stalled, or fail to redirect unproductive trajectories (Soudani et al., 2026). Recent work has begun to study such decisions. CaRT (Liu et al., 2025) learns when information gathering should terminate from counterfactual trajectories, while LSP (Liu & Wang, 2025) studies

adaptive early stopping based on reasoning convergence. However, these methods are not designed for long-horizon Deep Research. Extending such control requires either fine-grained supervision over long trajectories or repeated long-horizon optimization, motivating lightweight control without retraining the full research model.

Plug-in Adaptation for LLM Agents. Several approaches improve agent behavior through inference-time control. ReAct (Yao et al., 2023) interleaves reasoning and environment actions through prompting, while ASAG (Li et al., 2026a) uses attention states to adaptively terminate unnecessary reasoning. RAAC (Soudani et al., 2026) further introduces retrieval-aware signals to detect stagnation and adapt long-horizon Deep Research trajectories. In contrast, PPDPP (Deng et al., 2024b) learns a lightweight plug-in to guide the policy of a frozen LLM, but targets proactive dialogue rather than long-horizon research. A complementary line of work directly intervenes in model logits: ThinkMerge (Wang et al., 2026b) combines parallel reasoning trajectories through logit averaging, while SWAI (An et al., 2026) steers generation through training-free logit modification. Our method combines these directions by learning a lightweight plug-in that modifies frozen model logits only when trajectory rerouting is needed.

Training Resources for Deep Research. Existing Deep Research benchmarks provide limited resources for training long-horizon research control. BrowseComp-Plus (Chen et al., 2025) and DRBench (Abaskohi et al., 2026) are primarily designed for evaluation and do not provide dedicated training sets for learning trajectory-level interventions. OpenResearcher (Li et al., 2026e) instead releases 96K synthetically generated long-horizon research trajectories together with an approximately 11B-token retrieval corpus. While this provides a valuable training resource, its large-scale retrieval environment can be computationally expensive for repeated policy optimization, particularly when coupled with a large dense retriever. To support efficient and controlled training of research intervention, we construct a compact training resource that pairs high-quality question–answer examples with a dedicated local corpus, enabling repeated long-horizon rollouts without requiring large-scale retrieval infrastructure.

3 Problem Setup

We consider a long-horizon Deep Research setting in which an LLM agent interacts with a retrieval environment over multiple research turns. Given an initial user question q_0, the agent maintains an interaction history $\mathcal{F}_{t-1}$ containing previous reasoning, tool calls, retrieved evidence, and tool observations. At each turn $t = 1, 2, \ldots, T$, the frozen Host observes q_0 and $\mathcal{F}_{t-1}$ and produces either a tool action a_t or a final answer. A tool action is executed by the environment and returns an observation $o_t = \mathcal{E}(a_t)$, which is appended to the interaction history. A complete research episode is represented as $\tau = \{(a_1, o_1), (a_2, o_2), \ldots, (a_T, o_T)\}$, followed by a final answer. The environment assigns a task-level reward $R(\tau)$ according to the quality of the final research outcome. Among trajectory-level control decisions, in this work, we mainly study adaptive rerouting. At such a state, the agent generates a new retrieval direction q_t, which changes the subsequent research trajectory. The quality of a rerouting decision is determined by its downstream effect rather than by the reroute query alone. We quantify the effect of a reroute query q_t on subsequent research as

$$Q_t(q_t) = \mathbb{E}\left[R(\tau_{1:T}) \mid q_t, \mathcal{F}_{t-1}\right], \tag{1}$$

where $Q_t(q_t)$ represents the expected research utility after selecting q_t at turn t. Our objective is to learn a lightweight trajectory correction mechanism that increases this utility while keeping the host model frozen.

4 RIVO

4.1 The Challenge of Efficient Research Control

Improving long-horizon research control through supervised fine-tuning or reinforcement learning can be computationally expensive. Supervised approaches require fine-grained intervention trajectories, while reward-based optimization depends on repeated long-horizon reasoning and retrieval rollouts with sparse and delayed feedback. Training-free methods avoid model retraining, but typically rely on

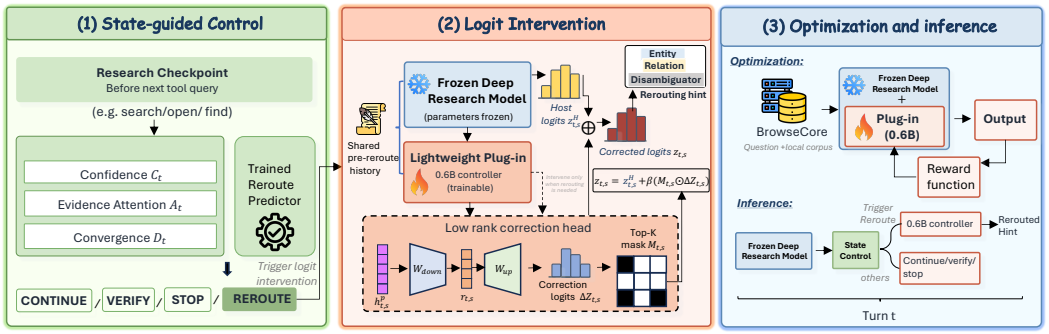

Figure 1: Overview of **RIVO**. RIVO consists of three stages: state-guided control determines whether the current Deep Research trajectory should continue, stop, verify or reroute; logit intervention uses a lightweight plug-in to modify selected Host logits when rerouting is triggered; and trajectory-level optimization trains the plug-in while keeping the Host model frozen. At inference time, the learned intervention is activated only when rerouting is required.

predefined heuristics or proxy signals that are not directly optimized for downstream research utility. A lightweight plug-in offers a more efficient alternative, but requiring a small model to independently generate complex research directions places substantial generative burden on the controller. We instead retain the frozen Host as the primary generator and use the plug-in to apply residual corrections in logit space. State-guided control determines when intervention is necessary, while the plug-in learns how to modify the Host generation from downstream trajectory-level outcomes.

Motivated by this view, we propose RIVO, a lightweight framework for state-conditioned trajectory correction. As illustrated in Figure 1, RIVO follows a simple pipeline with three components. First, State-Guided Control monitors the evolving research state using lightweight signals and determines whether the current trajectory should stop, continue, or be redirected. Second, when rerouting is triggered, Logit Intervention activates a lightweight plug-in that applies a sparse residual correction to the frozen Host logits, steering generation toward a new retrieval direction without updating the Host parameters. Third, Optimization and Inference trains the plug-in from downstream trajectory-level feedback and selectively applies the learned intervention during inference. Together, these components allow RIVO to adaptively correct unproductive research trajectories while keeping the Host model frozen and limiting intervention to states where rerouting is needed.

4.2 State-Guided Control

Figure 1 summarizes the state-guided control process in RIVO. At each retrieval checkpoint, RIVO evaluates the current research state using three complementary signals: confidence C_t, convergence D_t, and evidence attention A_t. Confidence identifies states that are sufficiently reliable to stop, convergence determines whether low-confidence states require further verification, and evidence attention distinguishes between continuing the current trajectory and redirecting it. When the state-guided controller selects CONTINUE, a Trained Reroute Predictor performs an additional learned check and may promote the decision to REROUTE. Once rerouting is triggered by either component, the plug-in applies token-level logit correction to the frozen Host to redirect subsequent retrieval.

Trained Reroute Predictor. We collect successful multi-turn Deep Research trajectories and extract a turn-level state $s_t = (q, h_{<t}, a_t)$, where q is the original question, $h_{<t}$ is the preceding research history, and a_t is the current tool action together with its result. A capable language model annotates each state as CONTINUE or REROUTE based on the accumulated evidence, current research direction, and gold answer. This process yields 29,252 turn-level examples from 1,475 questions. We split the data by question ID into 26,351 training examples and 2,901 test examples, ensuring that all turns from the same question remain in a single partition. We use the training split to fine-tune Qwen3-0.6B-Base as a lightweight reroute predictor, denoted by P. An action-prediction head produces a distribution $p_P(\cdot \mid s_t)$ over $\{\text{CONTINUE}, \text{REROUTE}\}$, where $p_P(\text{REROUTE} \mid s_t)$ denotes the predicted rerouting probability. The same plug-in backbone is later used with a separate low-rank residual correction head for logit intervention. Detailed annotation rules, prompts, and data statistics are provided in Appendix E.

Research-state estimation. At each retrieval checkpoint, RIVO estimates the current research state using three lightweight signals. Provisional-answer confidence C_t measures the Host's confidence in its current answer using a fixed answer probe. Evidence attention compares the attention assigned to the current and preceding retrieval segments, while attention convergence measures the change in attention entropy relative to the initial retrieval state. We define

$$C_t = \frac{1}{m}\sum_{i=1}^{m} p_H(y_i \mid X_t, I, y_{<i}), \qquad \Delta A_t = A_t^{\text{cur}} - A_t^{\text{prev}}, \qquad D_t = \frac{H_t - H_1}{|H_1|}. \tag{2}$$

Here, X_t denotes the research dialogue at checkpoint t, I is a fixed answer-probe suffix, A_t denotes the attention assigned to retrieved evidence, and H_t denotes the aggregated attention entropy. Larger C_t indicates higher answer confidence, $\Delta A_t < 0$ indicates that newly retrieved evidence receives less attention than preceding evidence, and $D_t < \tau_H$ indicates that the attention state has converged. Detailed definitions and computation are provided in Appendix B.2.

State-guided control. The three signals determine the trajectory-level control action. A high-confidence provisional answer triggers STOP. When confidence remains low but the attention state has converged, RIVO invokes VERIFY. For a low-confidence and unconverged state, decreasing evidence attention indicates an unproductive retrieval direction and triggers REROUTE. Otherwise, the current research trajectory CONTINUEs:

$$d_t = \begin{cases} \textsc{Stop}, & C_t > \lambda, \\ \textsc{Verify}, & C_t \le \lambda \wedge D_t \le \tau_H, \\ \textsc{Reroute}, & C_t \le \lambda \wedge D_t > \tau_H \wedge \Delta A_t < 0, \\ \textsc{Continue}, & \text{otherwise.} \end{cases} \tag{3}$$

For low-confidence states initially assigned CONTINUE, the trained reroute predictor P provides an additional learned check. The decision is promoted to REROUTE when $\hat{y}_t^P = \textsc{Reroute}$. Once REROUTE is triggered, the logit-intervention module is activated to redirect subsequent retrieval.

4.3 LOGIT INTERVENTION

Once REROUTE is triggered by the above state-guided control mechanism, RIVO activates the logit-intervention head attached to P. The head is trained from downstream trajectory-level outcomes to learn residual corrections to the Host token logits. During inference, the learned correction is applied while the Host remains frozen, redirecting subsequent retrieval.

Low-rank logit correction. At reroute step t and token position s, let $\mathbf{z}_{t,s}^H \in \mathbb{R}^{|\mathcal{V}|}$ denote the pre-softmax logits produced by the frozen Host, and let $\mathbf{h}_{t,s}^P$ denote the hidden representation produced by the plug-in. We project the plug-in representation through a low-rank correction head,

$$\mathbf{r}_{t,s} = W_{\text{down}}\mathbf{h}_{t,s}^P, \qquad \Delta\mathbf{z}_{t,s} = W_{\text{up}}\mathbf{r}_{t,s}. \tag{4}$$

To keep the intervention sparse, we retain only the K largest-magnitude entries of $\Delta\mathbf{z}_{t,s}$, where t denotes the Deep Research turn at which rerouting is applied, s denotes the token position within the generated reroute query, and j indexes candidate tokens $x_j \in \mathcal{V}$ in the vocabulary. Let $M_{t,s}$ denote the corresponding Top-K mask, The corrected Host logits and resulting token policy are

$$\mathbf{z}_{t,s} = \mathbf{z}_{t,s}^H + \beta\left(M_{t,s} \odot \Delta\mathbf{z}_{t,s}(\theta)\right), \qquad \pi_{\beta,\theta}\left(x_j \mid \mathcal{F}_{t,s-1}\right) = \text{softmax}\left(\mathbf{z}_{t,s}\right)_j. \tag{5}$$

Here, β controls the intervention strength. In this way, RIVO changes only selected token preferences while leaving the Host parameters unchanged.

Theoretical justification. We provide a local analysis of how the learned logit intervention affects the expected downstream value of a rerouting decision. Specifically, we characterize the first-order change in trajectory-level reward induced by perturbing the Host logits, and derive a sufficient condition under which the intervention improves the expected downstream value. Let $q_t = (x_{t,1}, \ldots, x_{t,L_t})$ denote the complete reroute query generated when rerouting is triggered at turn t. Here, q_t denotes a general textual representation of the rerouting direction; in RIVO, it is instantiated as a structured

rerouting hint that is subsequently rewritten by the frozen Host into a search query. By autoregressive factorization,

$$\pi_{\beta,\theta}(q_t \mid \mathcal{F}_{t-1}) = \prod_{s=1}^{L_t} \pi_{\beta,\theta}(x_{t,s} \mid \mathcal{F}_{t,s-1}). \tag{6}$$

When $\beta = 0$, the logit intervention vanishes and the policy reduces to the original Host policy, i.e., $\pi_{0,\theta} = \pi_H$. For a complete reroute query q_t, we define its expected downstream research utility as $Q_t(q_t) = \mathbb{E}\left[R(\tau_{1:T}) \mid q_t, \mathcal{F}_{t-1}\right]$. The corresponding expected utility under intervention strength β is $J_t(\beta) = \mathbb{E}_{q_t \sim \pi_{\beta,\theta}(\cdot|\mathcal{F}_{t-1})}\left[Q_t(q_t)\right]$. We further define the first-order query-level probability shift as

$$S_t(\theta; q_t) = \left.\frac{\partial}{\partial \beta} \log \pi_{\beta,\theta}(q_t \mid \mathcal{F}_{t-1})\right|_{\beta=0}. \tag{7}$$

Then,

$$\left.\frac{\partial J_t(\beta)}{\partial \beta}\right|_{\beta=0} = \mathrm{Cov}_{q_t \sim \pi_H(\cdot|\mathcal{F}_{t-1})}\left(Q_t(q_t), S_t(\theta; q_t)\right). \tag{8}$$

Theorem 1 (Local Improvement under Value-Aligned Logit Intervention)**.** *Let*

$$\Gamma_t := \mathrm{Cov}_{q_t \sim \pi_H(\cdot|\mathcal{F}_{t-1})}\left(Q_t(q_t), S_t(\theta; q_t)\right). \tag{9}$$

Suppose the intervention direction is locally aligned with downstream value, such that $\beta\Gamma_t > 0$, and there exists $B_J > 0$ satisfying $|J_t''(u)| \le B_J$ for all u between 0 and β. Then

$$J_t(\beta) - J_t(0) \ge \beta\Gamma_t - \frac{B_J}{2}\beta^2. \tag{10}$$

Consequently, for any nonzero β satisfying $\beta\Gamma_t > 0$, $0 < |\beta| < \frac{2|\Gamma_t|}{B_J}$, we have $J_t(\beta) > J_t(0)$.

The theorem shows that when the intervention direction is aligned with higher downstream value and its magnitude is sufficiently small, the logit intervention locally improves the expected downstream value relative to the frozen Host policy. Full assumptions, lemmas, and proofs are provided in Appendix H.

4.4 Optimization and Inference

We optimize the lightweight plug-in using trajectory-level feedback while keeping the Host model frozen. For each question q, we sample a group of G complete Deep Research trajectories, $\mathcal{T}_q = \{\tau_1, \ldots, \tau_G\}$. Whenever REROUTE is triggered, the plug-in P applies the logit intervention described in Section 4.3. Let θ_P^{logit} denote the trainable parameters of the logit-correction head of P. At token position s, the corrected Host logits are $\mathbf{z}_{t,s} = \mathbf{z}_{t,s}^H + \beta\left(M_{t,s} \odot \Delta\mathbf{z}_{t,s}(\theta_P^{\text{logit}})\right)$, where the Host logits $\mathbf{z}_{t,s}^H$ remain fixed. After each trajectory terminates, we use its downstream reward, as described in Appendix C, to compute a group-relative advantage $\hat{A}_i$ and optimize θ_P^{logit} with GRPO:

$$\mathcal{L}_{\text{GRPO}} = -\frac{1}{\sum_{i,j} m_{i,j}} \sum_{i,j} m_{i,j} \min\left(\rho_{i,j}\hat{A}_i,\ \mathrm{clip}(\rho_{i,j}, 1-\epsilon, 1+\epsilon)\hat{A}_i\right), \tag{11}$$

where $\rho_{i,j}$ is the policy ratio and $m_{i,j}$ selects trainable rerouting tokens. Gradients are applied only to θ_P^{logit}, while the Host remains frozen. At inference time, the learned correction is activated only when REROUTE is triggered. The resulting rerouting hint is rewritten by the frozen Host into a new search query, and the retrieved evidence is fed back into the next research checkpoint.

5 Experiments

Datasets. We construct BrowseCore, a curated Deep Research dataset paired with a local retrieval corpus for plug-in training and controlled evaluation. BrowseCore contains 2,870 questions for training and development, together with a test set of 359 questions, with each question associated with verified supporting evidence and hard-negative documents. We evaluate RIVO on the 359 question test set. To assess generalization beyond BrowseCore, we additionally evaluate on all 830 BrowseComp-Plus questions (Chen et al., 2025) and the 100-task DRBench benchmark (Abaskohi et al., 2026). Detailed BrowseCore construction procedures are provided in Appendix G.

Table 1: Main results on BrowseComp-Plus, BrowseCore, and DRBench. SR denotes Success Rate. For DRBench, IR, Fact, DA, RQ, and HM denote Insight Recall, Factuality, Distractor Avoidance, Report Quality, and Harmonic Mean, respectively. Higher is better for all metrics.

Deep Research Model	Method	BrowseComp-Plus	BrowseCore	DRBench				
		SR ↑	SR ↑	IR ↑	Fact ↑	DA ↑	RQ ↑	HM ↑
LiteResearcher-4B	Source	32.29	53.48	5.24	54.47	99.90	81.23	17.27
	ASAG	33.61	54.60	11.76	87.00	99.57	76.60	33.43
	RAAC	33.73	54.04	6.41	51.52	99.63	81.89	20.23
	PPDPP	32.65	55.15	3.34	73.14	99.63	76.74	11.91
	ReAct	31.57	54.32	8.76	51.34	99.60	80.79	25.64
	RIVO(OURS)	**36.14**	**59.30**	12.61	81.85	99.54	76.36	**34.88**
Marco-DeepResearch-8B	Source	31.70	48.75	15.95	82.49	99.08	72.74	40.54
	ASAG	29.78	50.42	12.43	74.40	99.24	74.90	34.09
	RAAC	32.53	52.00	15.21	79.00	98.82	71.58	39.02
	PPDPP	33.37	50.97	15.22	79.81	98.97	72.52	39.16
	ReAct	32.90	50.14	16.09	79.52	99.03	72.22	40.53
	RIVO (OURS)	**33.98**	**52.40**	18.56	85.55	99.29	74.40	**44.90**

Baselines and evaluation setup. We evaluate RIVO on two representative Deep Research systems, LiteResearcher-4B (Li et al., 2026d) and Marco-DeepResearch-8B (Zhu et al., 2026). We compare against four baselines for inference-time trajectory control. ReAct (Yao et al., 2023) interleaves reasoning with tool use through prompting without an explicit trajectory controller. ASAG (Li et al., 2026a) uses attention-state signals to adaptively terminate unnecessary reasoning. RAAC (Soudani et al., 2026) incorporates retrieval-aware signals to detect stagnation and adapt long-horizon research trajectories. PPDPP (Deng et al., 2024b) uses a lightweight plug-in to guide the policy of a frozen language model. For BrowseComp-Plus and BrowseCore, we follow the OpenResearcher evaluation protocol and report Success Rate (SR) using a GPT-4-based judge for final-answer correctness. For DRBench, we use the official `score_report` implementation with gpt-4o-mini as the judge model, and report Insight Recall (IR), Factuality (Fact.), Report Quality (RQ), Distractor Avoidance (DA), and Harmonic Mean (HM), with HM serving as the overall aggregate metric. All methods use Qwen3-Embedding-8B (Zhang et al., 2025) as the retriever for a controlled comparison.

5.1 Main Results

Table 1 compares RIVO with ReAct Yao et al. (2023), ASAG Li et al. (2026a), RAAC Soudani et al. (2026), and PPDPP Deng et al. (2024b) across three benchmarks and two Deep Research models. RIVO achieves the strongest overall result across all six model–benchmark settings. For example, with LiteResearcher-4B, it improves over the strongest baseline by 2.41 points on BrowseComp-Plus, 4.15 points on BrowseCore, and 1.45 points in DRBench HM. The gains remain consistent when switching to Marco-DeepResearch-8B, where RIVO also improves over the strongest baseline across all three benchmarks. This cross-model pattern indicates that the benefit of RIVO is not tied to a particular Host architecture, although the magnitude of improvement varies across evaluation settings. We find that Marco-DeepResearch-8B exhibits a different performance profile from LiteResearcher-4B. Despite its larger scale, Marco-DeepResearch-8B performs worse than LiteResearcher-4B on BrowseComp-Plus and BrowseCore, but substantially better on DRBench. This cross-benchmark variation indicates that the two Hosts exhibit different performance profiles, suggesting that benchmark characteristics and model-specific research behaviors may interact in nontrivial ways. On DRBench, RIVO does not achieve the highest score on every individual metric, but it achieves the highest Harmonic Mean for both Host models. With LiteResearcher-4B, RIVO reaches an HM of 34.88, compared with 33.43 for the strongest baseline, while with Marco-DeepResearch-8B, it achieves 44.90 versus 40.54. These results show that RIVO achieves stronger overall performance through more effective trajectory correction.

5.2 Further Analysis

In this section, we systematically analyze the key design choices of RIVO, including the intervention strength β, correction scope K, state-control thresholds λ and τ_H, the contribution of individual

state signals, and the rerouting design. Additional ablations in Appendix A further examine the two functions of the 0.6B plug-in and compare alternative controller configurations, including STOP-only, REROUTE-only, standalone three-way control, and the variant without VERIFY.

Sensitivity to Intervention Strength and Scope. There are two important hyperparameters in RIVO, the intervention strength β and the correction scope K. The parameter β controls the magnitude of the residual logit correction applied to the frozen Host, while K determines how many logits are modified during rerouting. Together, they control how strongly and how broadly RIVO redirects the current research trajectory. An intervention that is too weak may fail to change the retrieval direction, whereas an overly strong or overly broad correction may unnecessarily perturb the Host distribution. We therefore study the sensitivity of RIVO to both parameters on the BrowseCore test set using LiteResearcher-4B with a fixed Qwen-Embedding-8B retriever. For questions where rerouting is triggered, we replay the trajectory from the same pre-reroute state while varying $\beta \in \{-20, -10, -5, -2, 0, 2, 5, 10, 20\}$. The question, trajectory prefix, checkpoint state, retrieval context, and random seed are kept fixed before rerouting, while the reroute query and all subsequent research steps are regenerated. As shown in Figure 2, performance varies substantially with the intervention strength. The best result is obtained at $\beta = -10$, achieving a success rate of 24, compared with 16 when $\beta = 0$. We further vary $K \in \{8, 16, 32, 64, 128, 256\}$ to examine the effect of intervention scope. Performance also changes noticeably across different values of K, with the best result achieved at $K = 128$ with a success rate of 24. These results demonstrate that the learned logit intervention in RIVO is effective, with performance improving under appropriately scaled and selective corrections to the Host logits.

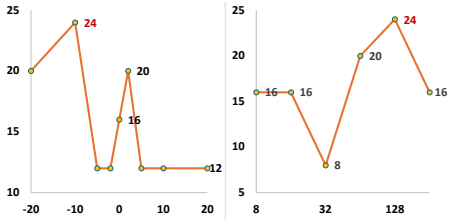

Figure 2: Sensitivity analysis of RIVO under different hyperparameter settings.

Sensitivity to State-Control Thresholds. We further analyze the two thresholds used by the state-guided controller. The confidence threshold λ determines whether the current trajectory is considered sufficiently reliable, while the convergence threshold τ_H measures whether the research state has stabilized. We evaluate different combinations of $\lambda \in \{0.90, 0.95, 0.98\}$ and $\tau_H \in \{-0.05, -0.10, -0.20\}$. As shown in Figure 3, the best performance is obtained with $\lambda = 0.95$ and $\tau_H = -0.10$, achieving an SR of 59.30. A low threshold such as $\lambda = 0.90$ may over-trust uncertain provisional answers, while a higher threshold requires stronger confidence before termination. The best performance at $\lambda = 0.95$ suggests that a moderate confidence criterion provides a better balance between premature stopping and unnecessary continued research.

λ / τ_h	-0.05	-0.10	-0.20
0.90	52.09	52.65	53.76
0.95	54.32	**59.30**	55.71
0.98	54.04	53.91	57.10

Figure 3: Sensitivity to the state-control thresholds λ and τ_H. Higher SR is better.

Ablation of State Signals. We analyze the contribution of the three state signals used by RIVO in Figure 4: confidence C^t, evidence attention A^t, and convergence D^t, denoted as C, E, and D in the figure, respectively. These signals capture complementary aspects of the current research state: C^t reflects the reliability of the current answer, A^t measures the attention assigned to retrieved evidence, and D^t indicates whether the research trajectory has converged. Among the individual signals, Evidence Attention A^t performs best with an SR of 55.03, followed by Confidence C^t at 54.47 and Convergence D^t at 53.20. Pairwise combinations generally improve performance, with $\{C^t, A^t\}$ achieving the strongest two-signal result of 56.55 and $\{C^t, D^t\}$ reaching 56.15. In contrast, $\{A^t, D^t\}$ obtains 54.32, suggesting that confidence provides an important complementary signal when combined with either evidence-based or convergence-based information. Using all three signals achieves the best performance

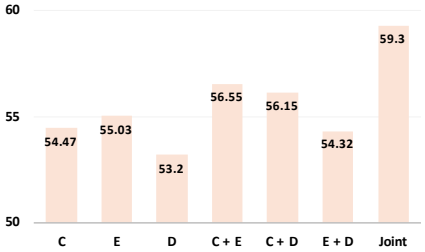

Figure 4: Ablation of state signals.

of 59.30, outperforming the best single-signal and two-signal settings by 4.27 and 2.75 points, respectively. This shows that the three signals provide complementary information for state-guided trajectory control.

Ablation of Rerouting Design. The 0.6B plug-in in RIVO has two distinct functions: a *reroute predictor*, which determines whether a current CONTINUE state should be promoted to REROUTE, and a *logit-intervention module*, which modifies the frozen Host logits after rerouting is triggered to steer the next retrieval direction. In *w/o Reroute Predictor*, the learned reroute prediction is disabled, so rerouting can only be triggered by the state-guided controller, while the logit intervention is retained. In *w/o Logit Intervention*, the reroute predictor is retained, but the residual correction to the Host logits is removed, so the Host generates the subsequent rerouting direction without learned logit modification. In *w/o Logit Intervention + Reroute Predictor*, both learned components are removed, leaving only the state-guided controller for trajectory control. As shown in Table 2, removing the reroute predictor reduces SR from 59.30 to 53.63, while removing the logit intervention reduces SR to 56.82. These results show that both deciding *when* to introduce an additional reroute and learning *how* to redirect the frozen Host contribute to the final performance.

Variant	**SR (%)**
Plug-in Function	
w/o Reroute Predictor (Pre)	53.63
w/o Logit Intervention	56.82
w/o Logit Intervention+Reroute Pre	55.71
Full RIVO	**59.30**
Rerouting Representation	
Direct Question	54.87
Structured Hint	**59.30**

Table 2: Ablation of the rerouting design.

6 CONCLUSION

We present RIVO, a lightweight framework for adaptive trajectory correction in long-horizon Deep Research. RIVO combines state-guided control with a lightweight learned reroute predictor and token-level logit intervention, enabling adaptive decisions about when trajectory redirection is needed and how the frozen Host should be redirected without retraining the Host model. We also construct BROWSECORE, a curated question–answer dataset paired with a compact local retrieval corpus for controlled training and evaluation of Deep Research agents. Experiments on BrowseComp-Plus, BrowseCore, and DRBench with two different Deep Research Hosts demonstrate performance gains over existing trajectory-control baselines across all evaluated settings, while ablation studies show that the state signals, reroute prediction, and logit intervention each contribute to the overall performance. Our theoretical analysis further shows that, under a value-alignment condition and bounded local curvature, a sufficiently small logit intervention can locally improve the expected downstream research utility. Overall, these results demonstrate that lightweight, state-conditioned logit intervention provides an effective approach to improving long-horizon Deep Research trajectories.

7 LIMITATIONS AND FUTURE DIRECTIONS

The current framework uses relatively constrained rerouting decisions and separates state-based control from the learned logit intervention. Future work could explore more adaptive rerouting strategies that support multiple interventions over a trajectory and jointly learn when to intervene, how strongly to intervene, and which parts of the Host distribution should be modified. Another promising direction is to make the retriever itself adaptive, allowing the retrieval model and trajectory controller to be jointly optimized from downstream feedback so that retrieval representations and rerouting strategies can co-adapt throughout the research process.

AI USE STATEMENT

Large language models were used only for language editing, including improving clarity and correcting grammatical issues. All technical content, analysis, and final revisions were reviewed and approved by the authors.

Ethics statement

Our study is conducted entirely on existing research benchmarks and does not involve human subjects or private user data. The proposed framework is intended for research use, and any deployment in real-world settings should account for privacy, reliability, and responsible tool use.

Reproducibility statement

We document the model configurations, training settings, hyperparameters, datasets, and evaluation procedures in the appendix. Source code and supporting resources are provided through the anonymized repository referenced in the paper.

References

Amirhossein Abaskohi, Tianyi Chen, Miguel Muñoz-Mármol, Curtis Fox, Amrutha Varshini Ramesh, Étienne Marcotte, Xing Han Lù, Nicolas Chapados, Spandana Gella, Christopher Pal, et al. Drbench: A realistic benchmark for enterprise deep research. In *International Conference on Learning Representations*, volume 2026, pp. 6727–6795, 2026.

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

Hyeseon An, Shinwoo Park, Hyundong Jin, and Yo-Sub Han. Steering language models before they speak: Logit-level interventions. *arXiv preprint arXiv:2601.10960*, 2026.

Zijian Chen, Xueguang Ma, Shengyao Zhuang, Ping Nie, Kai Zou, Andrew Liu, Joshua Green, Kshama Patel, Ruoxi Meng, Mingyi Su, et al. Browsecomp-plus: A more fair and transparent evaluation benchmark of deep-research agent. *arXiv preprint arXiv:2508.06600*, 2025.

Yang Deng, Wenxuan Zhang, Wai Lam, See-Kiong Ng, and Tat-Seng Chua. Plug-and-play policy planner for large language model powered dialogue agents. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 10058–10090, 2024a. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/29e8437db7b549160ce03d336ff66f65-Paper-Conference.pdf.

Yang Deng, Wenxuan Zhang, Wai Lam, See-Kiong Ng, and Tat-Seng Chua. Plug-and-play policy planner for large language model powered dialogue agents. In *International Conference on Learning Representations*, volume 2024, pp. 10058–10090, 2024b.

Yao Dong, Xinglin Xiao, Liwei Dong, Xinlong Jin, Zhengbo Li, Heng Zhang, Duyun Wang, and Nan Xu. S1-deepresearch: Beyond search, toward real-world long-horizon research agents. *arXiv preprint arXiv:2606.15367*, 2026.

Gaurush Hiranandani, Haolun Wu, Subhojyoti Mukherjee, and Sanmi Koyejo. Logits are all we need to adapt closed models. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 23261–23289. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/hiranandani25a.html.

Jiakai Li, Ke Qin, Rongzheng Wang, Yizhuo Ma, Qizhi Chen, Muquan Li, and Shuang Liang. Stop when further reasoning won't help: Attention-state adaptive generation in reasoning models. In *Proceedings of the 43rd International Conference on Machine Learning*, volume 306 of *Proceedings of Machine Learning Research*. PMLR, 2026a.

Mingchen Li, Jiatan Huang, Zonghai Yao, and Hong Yu. Efficient and effective internal memory retrieval for llm-based healthcare prediction. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 35895–35906, 2026b.

Mingchen Li, Hansi Zeng, Zhuo Qian, Jiatan Huang, Sunjae Kwon, Hamed Zamani, and Hong Yu. Rice-po: Turning retrieval interactions into credit signals for reasoning agents. *arXiv preprint arXiv:2605.26352*, 2026c.

Wanli Li, Bince Qu, Bo Pan, Jianyu Zhang, Zheng Liu, Pan Zhang, Wei Chen, and Bo Zhang. Literesearcher: A scalable agentic rl training framework for deep research agent. *arXiv preprint arXiv:2604.17931*, 2026d.

Zhuofeng Li, Dongfu Jiang, Xueguang Ma, Haoxiang Zhang, Ping Nie, Yuyu Zhang, Kai Zou, Jianwen Xie, Yu Zhang, and Wenhu Chen. Openresearcher: A fully open pipeline for long-horizon deep research trajectory synthesis. *arXiv preprint arXiv:2603.20278*, 2026e.

Grace Liu, Yuxiao Qu, Jeff G. Schneider, Aarti Singh, and Aviral Kumar. Cart: Teaching LLM agents to know when they know enough. *CoRR*, abs/2510.08517, 2025. doi: 10.48550/ARXIV.2510.08517. URL https://doi.org/10.48550/arXiv.2510.08517.

Xin Liu and Lu Wang. Answer convergence as a signal for early stopping in reasoning. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 17896–17907, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.904. URL https://aclanthology.org/2025.emnlp-main.904/.

Guilherme Penedo, Hynek Kydlíček, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, Thomas Wolf, et al. The fineweb datasets: Decanting the web for the finest text data at scale. *Advances in Neural Information Processing Systems*, 37:30811–30849, 2024.

Heydar Soudani, Elizabeth Lingg, Faegheh Hasibi, and Navid Rekabsaz. When deep research agents stagnate: Enhancing reasoning with retrieval-aware agent control. In *Proceedings of the 2026 Conference on Empirical Methods in Natural Language Processing*, 2026.

Tongyi DeepResearch Team. Tongyi deepresearch: A new era of open-source ai researchers. https://github.com/Alibaba-NLP/DeepResearch, 2025.

Guoqing Wang, Sunhao Dai, Guangze Ye, Zeyu Gan, Wei Yao, Yong Deng, Xiaofeng Wu, and Zhenzhe Ying. Information gain-based policy optimization: A simple and effective approach for multi-turn search agents. In *The Fourteenth International Conference on Learning Representations*, 2026a. URL https://openreview.net/forum?id=qkWP6phrvZ.

Haonan Wang, Chao Du, Kenji Kawaguchi, and Tianyu Pang. Think in parallel, answer as one: Logit averaging for open-ended reasoning. In *International Conference on Learning Representations*, volume 2026, pp. 77445–77462, 2026b.

Zhizheng Wang, Chih-Hsuan Wei, Joey Chan, Robert Leaman, Chi-Ping Day, Chuan Wu, Mark A Knepper, Antolin Serrano Farias, Jordina Rincon-Torroella, Hasan Slika, et al. Deeper-med: Advancing deep evidence-based research in medicine through agentic ai. *arXiv preprint arXiv:2604.15456*, 2026c.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*, 2023.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, et al. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*, 2025.

Bin Zhu, Qianghuai Jia, Tian Lan, Junyang Ren, Feng Gu, Feihu Jiang, Longyue Wang, Zhao Xu, and Weihua Luo. Marco deepresearch: Unlocking efficient deep research agents via verification-centric design. *arXiv preprint arXiv:2603.28376*, 2026.

APPENDIX NAVIGATION

A Ablation Study 13

B RIVO's Adaptive Rerouting Pipeline 13

B.1 Structural Ablations 13

B.2 Lightweight State Signals 14

C Rerouting Reward Details 15

D Implementation Details 16

E Turn-Level Supervision and Data Construction 17

F Prompt 17

F.1 Deep Research System Prompt 17

F.2 LiteResearcher-4B Browser Tools 18

F.3 Turn-Level Annotation for Plug-in Training 19

F.4 Evidence Document Verification 19

F.5 Prompt Template for Clue Decomposition 19

G BrowseCore 20

G.1 Background and Motivation 20

G.2 Benchmark and Corpus Construction 23

G.2.1 Question Source and Filtering 23

G.2.2 Candidate Document Sources 23

G.2.3 Evidence Collection and Verification 23

G.2.4 Positive and Gold Document Identification 24

G.2.5 Hard-Negative Mining 24

G.2.6 Final Corpus Assembly 24

H Proof of Main Theorem 24

A ABLATION STUDY

Ablation of the 0.6B Plug-in. The 0.6B plug-in in RIVO is used in two distinct stages: first, to predict whether a current CONTINUE state should be promoted to REROUTE, and second, to modify the frozen Host logits after rerouting is triggered. We separately remove each function while keeping the remaining pipeline unchanged. Removing the 0.6B reroute predictor reduces SR from 59.30 to 53.63, indicating that the learned reroute decision contributes substantially to identifying useful intervention points. When the reroute decision is retained but the 0.6B logit intervention is removed, the system directly outputs the rerouting hint without modifying the Host logits and achieves an SR of 56.82. The degradation in both variants shows that the two uses of the 0.6B plug-in are complementary: the predictor helps determine *when* rerouting should occur, while the logit intervention controls *how* the frozen Host is redirected toward a more effective retrieval trajectory.

Controller Design Ablation. We further examine how different controller designs affect long-horizon research performance. Table 3 compares variants that restrict the lightweight controller to a single action, allow it to independently perform full three-way control, or combine it with the original control mechanism. The definitions of STOP-only, REROUTE-only, and the standalone 0.6B three-way controller are provided in Section B.1. *w/o Verify* removes the VERIFY action from the state-guided controller.

Table 3: Ablation study of different controller designs.

Controller Design	SR (%)
STOP-only	56.27
REROUTE-only	55.15
Standalone 0.6B 3-way	47.91
w/o Verify	54.32
Joint controller (Ours)	**59.30**

B **RIVO**'S ADAPTIVE REROUTING PIPELINE

B.1 STRUCTURAL ABLATIONS.

Figure 5 summarizes the control flow of RIVO and the structural variants used in our ablation study. In the full model (**1**), RIVO first evaluates the current research state using three complementary signals: confidence C_t, convergence D_t, and evidence attention A_t. Confidence determines whether the current trajectory is sufficiently reliable to terminate. For low-confidence states, convergence identifies whether the current provisional answer should be further verified, while evidence attention compares the usefulness of newly retrieved evidence with previously collected evidence to determine whether the trajectory should be rerouted or continued. If the state-guided controller selects CONTINUE, a lightweight 0.6B reroute predictor performs an additional learned check and may promote the decision to REROUTE. Once REROUTE is triggered, either by the state-guided controller or the reroute predictor, the 0.6B plug-in applies token-level logit correction to the frozen Host, guiding it toward a new retrieval direction. A CONTINUE decision instead leaves the Host trajectory unchanged and proceeds to the next research step. We construct four structural variants to isolate different parts of this control mechanism. Variant **(2)**, **STOP-only**, retains only the confidence-based stopping mechanism. Variant **(3)**, **REROUTE-only**, retains the rule-based rerouting mechanism while removing the other control branches. Variant **(4)**, **Predictor-only**, replaces the state-guided controller with the lightweight 0.6B reroute predictor. Variant **(5)**, **w/o Verify**, removes the verification branch while preserving the remaining control logic. Importantly, these variants modify only how the control decision is made. After a REROUTE or CONTINUE decision is produced, variants **(2)–(5)** follow exactly the same downstream execution as the full model: REROUTE invokes the plug-in-based logit correction followed by a new retrieval step, whereas CONTINUE proceeds with the normal Host research trajectory.

When VERIFY is selected, the current provisional answer is explicitly treated as an unverified hypothesis. The Host receives the following instruction:

> *The current provisional answer is: {answer}. Treat it as an unverified hypothesis. Check it against the retrieved primary evidence, look for contradictions, and use* `browser.open`*/*`browser.find` *(or a new search when necessary) to resolve any missing fact before answering.*

A STOP decision terminates the research process and requires the Host to return the final response in the format `<answer>\boxed{answer}</answer>`. We additionally impose a maximum

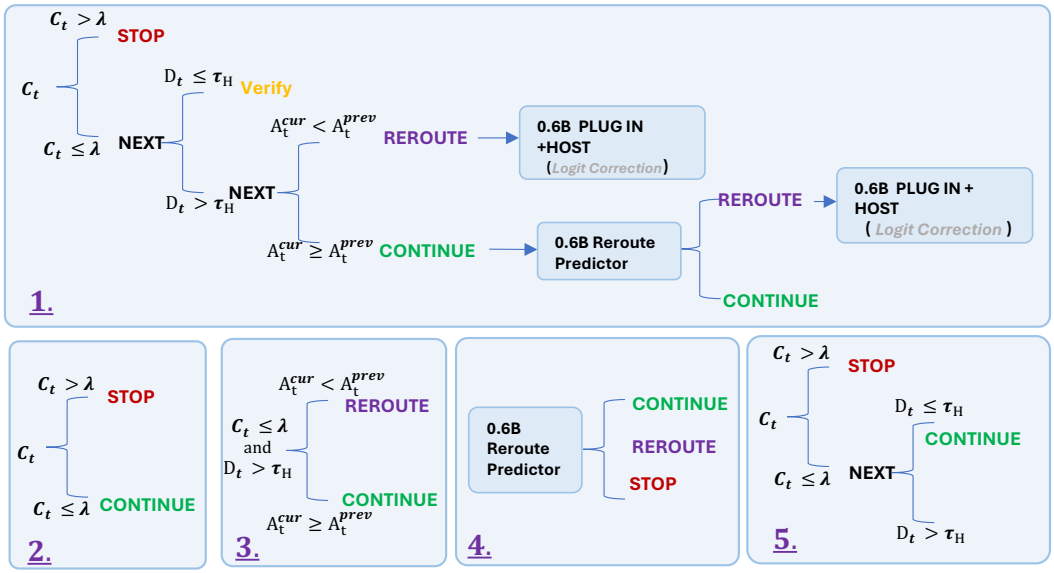

Figure 5: Structural ablations of **RIVO**.

research-turn budget. If this budget is exhausted before a normal stopping decision is reached, further tool calls are disabled and the Host is instructed to synthesize the best-supported answer from the evidence already collected:

> *The research budget is exhausted. Do not call any tool. Synthesize the best-supported final answer from the evidence already in this conversation. Return only* `<answer>...</answer>`.

B.2 State-guided control

Provisional-answer confidence. At checkpoint t, let X_t denote the serialized dialogue. We append a fixed answer-probe suffix I directly at the token level and greedily decode a provisional answer $Y_t = (y_1, \ldots, y_m)$. From the selected-token log-probabilities $\ell_i = \log p_H(y_i \mid X_t, I, y_{<i})$, we compute

$$C_t = \frac{1}{m} \sum_{i=1}^{m} \exp(\ell_i), \qquad \text{high}_t = \mathbf{1}[C_t > \lambda], \tag{12}$$

where m ends at the first complete boxed answer and $\lambda = 0.95$. C_t provides a lightweight internal estimate of the Host's confidence in its current provisional answer given the evidence available at the current retrieval state.

Evidence-attention score. We monitor the final $L = 4$ Host layers over a $W = 32$ token window. Let $a_{h,q,k}^{(l)}$ denote the attention weight from query position q to token k at head h of layer l. Since a retrieval step may return multiple pieces of evidence, we define S_t as the union of all evidence-token spans accumulated within the current retrieval segment before checkpoint t. For an evidence segment S, we compute

$$A_t(S) = \frac{1}{L} \sum_{l=1}^{L} \frac{1}{N_h Q |S|} \sum_{h} \sum_{q} \sum_{k \in S} a_{h,q,k}^{(l)}, \tag{13}$$

where N_h is the number of attention heads, Q is the number of query positions in the monitored window, and $|S|$ is the number of evidence tokens. Thus, $A_t(S)$ measures the average attention assigned by the Host to the evidence accumulated in a retrieval segment. We denote the scores of the previous and current retrieval segments as $A_t^{\text{prev}} = A_t(S_{t-1})$ and $A_t^{\text{cur}} = A_t(S_t)$, respectively. Their difference, $\Delta A_t = A_t^{\text{cur}} - A_t^{\text{prev}}$, indicates whether the newly accumulated evidence receives more or less attention than the evidence from the preceding retrieval segment.

Attention convergence. We use attention entropy to measure whether the Host has reached a more focused and stable research state. For query position q, head h, and layer l, we compute the normalized attention entropy

$$E_{h,q}^{(l)} = \frac{-\sum_{k=1}^{K} \widetilde{a}_{h,q,k}^{(l)} \log \widetilde{a}_{h,q,k}^{(l)}}{\max(\log K, 1)}, \qquad \widetilde{a} = \max(a, 10^{-12}), \tag{14}$$

where lower entropy indicates that attention is concentrated on a smaller set of tokens. We aggregate the entropy across the monitored layers, heads, and query positions as

$$H_t = \sum_{l=1}^{L} \sum_{h} \sum_{q} E_{h,q}^{(l)}, \qquad D_t = \frac{H_t - H_1}{\max(|H_1|, 10^{-12})}, \tag{15}$$

where H_1 is the entropy at the first retrieval checkpoint. Thus, D_t measures how much the attention distribution has become more concentrated relative to the initial research state. We treat the state as converged when $D_t < \tau_H$, with $\tau_H = -0.10$.

At each retrieval checkpoint, rerouting can be triggered by either the inference-time controller or the learned 0.6B plug-in. The controller triggers REROUTE when the current provisional answer has low confidence, the attention state has not converged, and the newly retrieved evidence receives less attention than the preceding evidence. If the controller instead returns a low-confidence CONTINUE, we consult the plug-in and promote the decision to REROUTE when its rerouting probability exceeds a threshold.The final control decision combines the state-guided controller and the learned plug-in trigger:

$$\begin{aligned}
g_t^{\text{stop}} &= \mathbf{1}[C_t > \lambda], \\
g_t^{\text{verify}} &= \mathbf{1}[C_t \le \lambda \wedge D_t \le \tau_H], \\
g_t^{\text{ctrl}} &= \mathbf{1}[C_t \le \lambda \wedge D_t > \tau_H \wedge \Delta A_t < 0], \\
g_t^{\text{plug}} &= \mathbf{1}[d_t = \textsc{Continue} \wedge C_t \le \lambda \wedge p_P(\textsc{Reroute} \mid s_t) \ge \tau_P \wedge G_t = 1], \\
g_t^{\text{reroute}} &= g_t^{\text{ctrl}} \vee g_t^{\text{plug}}, \\
d_t^{\text{final}} &= \begin{cases} \textsc{Stop}, & g_t^{\text{stop}} = 1, \\ \textsc{Verify}, & g_t^{\text{verify}} = 1, \\ \textsc{Reroute}, & g_t^{\text{reroute}} = 1, \\ \textsc{Continue}, & \text{otherwise.} \end{cases}
\end{aligned} \tag{16}$$

Here, $\Delta A_t = A_t^{\text{cur}} - A_t^{\text{prev}}$, and G_t indicates whether the minimum-round and cooldown constraints are satisfied. When $g_t = 1$, the plug-in correction head is activated to redirect the subsequent retrieval.

C REROUTING REWARD DETAILS

We use a trajectory-level reward that combines final task success with the quality of the rerouting intervention. For trajectory τ_i, the reward is

$$R_i = R_i^{\text{outcome}} + \lambda Q_i - \gamma P_i, \tag{17}$$

where R_i^{outcome} measures final-answer correctness, Q_i measures rerouting-hint quality, and P_i penalizes invalid rerouting outputs.

Outcome reward. The outcome reward is determined by the final answer of the complete Deep Research trajectory:

$$R_i^{\text{outcome}} = \begin{cases} 1, & \text{if the final answer is judged correct,} \\ 0, & \text{otherwise.} \end{cases} \tag{18}$$

Rerouting-hint quality. Each rerouting hint consists of three fields: ENTITY, RELATION, and DISAMBIGUATOR. We evaluate the generated hint according to seven criteria:

1. **Format validity**: the output follows the required three-field format;
2. **English usage**: the generated fields are written in English;
3. **No self-talk**: the output does not contain reasoning traces, instructions, or meta-language;
4. **Entity anchoring**: the ENTITY field is grounded in the question or the observed research history;
5. **Relation validity**: the RELATION field provides a meaningful relation for redirecting retrieval;
6. **Disambiguation validity**: the DISAMBIGUATOR provides additional information that narrows the search direction;
7. **Field distinctness**: the three fields provide non-redundant information.

Let $c_{i,k} \in \{0, 1\}$ indicate whether the k-th criterion is satisfied. We compute

$$S_i = \sum_{k=1}^{7} w_k c_{i,k}, \qquad \sum_{k=1}^{7} w_k = 1, \tag{19}$$

where w_k denotes the fixed importance weight of criterion k. We use uniform weights $w_k = 1/7$ for all seven criteria.

The rerouting-quality reward is

$$Q_i = \begin{cases} 1, & \text{if all criteria are satisfied,} \\ -1 + 0.5S_i, & \text{if the generated hint violates any criterion,} \\ 0, & \text{if no rerouting hint is generated.} \end{cases} \tag{20}$$

Thus, valid hints receive a positive reward, while invalid hints receive a graded negative reward according to the satisfied quality criteria.

Invalid-hint penalty. We additionally define

$$P_i = \begin{cases} 1, & \text{if a generated rerouting hint is invalid,} \\ 0, & \text{otherwise.} \end{cases} \tag{21}$$

No invalid-hint penalty is applied when no rerouting hint is generated.

For the configuration used in our main experiments, we set $\lambda = 2.0, \gamma = 2.0$.

When no rerouting hint is generated, $Q_i = P_i = 0$, and the reward reduces to $R_i = R_i^{\text{outcome}}$

D IMPLEMENTATION DETAILS

For action prediction, we fine-tuned Qwen3-0.6B-Base using LoRA-based supervised fine-tuning (SFT) with two action labels: CONTINUE, and REROUTE. We set the maximum sequence length to 8,192 tokens and trained for 3 epochs using AdamW with a learning rate of 2×10^{-4}, a cosine learning-rate schedule, and a warmup ratio of 0.03. For LoRA, we used a rank of 16, a scaling factor of 32, and a dropout rate of 0.05. Training was conducted on 4 GPUs with a per-device batch size of 1 and 4 gradient accumulation steps, resulting in an effective batch size of 16. To alleviate class imbalance among the action labels, we applied class-aware sampling with an exponent of 0.5.

We trained a Qwen3-0.6B control-and-correction plugin while keeping the Deep Research Host frozen. The plugin was initialized from a two-class SFT controller and adapted using LoRA with rank 16, scaling 32, and dropout 0.05. We used GRPO with five trajectories per question, a training batch size of 8, and a clipping ratio of 0.2. The correction head was optimized using AdamW with a learning rate of 10^{-4}, zero weight decay, and a maximum gradient norm of 1.0. The SFT-initialized controller parameters were kept fixed during GRPO training. No KL or entropy regularization was applied. The structured route hint used token budgets of 16, 12, and 12 for the ENTITY, RELATION, and DISAMBIGUATOR fields, with temperature 1.0, top-$p = 0.95$, and correction Top-$K = 128$. We trained on 2,870 questions, used 359 questions for testing.

We used Qwen/Qwen3-Embedding-8B as the fixed dense retriever. The maximum research horizon was set to 80 rounds, with an input budget of 23,000 tokens and a maximum model context length of 32,768 tokens. For state-based action decisions, we used the host model itself to extract attention-based signals and confidence scores. We set the confidence threshold to 0.95 and the entropy-delta threshold to -0.10, using a 32-token decoding attention window and monitoring the final four layers. It generated an ENTITY ; RELATION ; DISAMBIGUATOR route hint with token budgets of 16, 12, and 12, respectively, using correction Top-$K = 128$. The frozen Host rewrote this hint into a browser.search query using temperature 0.7, a maximum of 32 generated tokens, and a 16-word query limit.

E Turn-Level Supervision and Data Construction

Turn-Level Supervision. We collect multi-turn deep research trajectories from a pretrained deep-research agent and retain trajectories with correct final answers. Upon completion of each deep-research turn t, we extract a state $s_t = (q, h_{<t}, a_t)$, where q is the original question, $h_{<t}$ is the deep research history preceding the current turn, and a_t is the current action together with its result. We extract a_t from the final tool call of turn t, such as `search`, `open`, or `find`. A capable language model, such as GPT-4.1, evaluates the current action a_t in the context of the question q and the accumulated deep-research history $h_{<t}$, using the gold answer as the annotation target. It first determines whether the cumulative evidence after a_t supports all essential parts of the gold answer. If so, it sets $z_t^{\text{stop}} = 1$. Otherwise, it sets $z_t^{\text{stop}} = 0$ and evaluates whether the search direction represented by a_t remains productive given the preceding history. It sets $z_t^{\text{reroute}} = 1$ if the direction is unproductive and requires a substantial change, and $z_t^{\text{reroute}} = 0$ if the direction remains productive and should be continued. A single failed tool call is insufficient to trigger rerouting. When $z_t^{\text{stop}} = 1$, we set $z_t^{\text{reroute}} = 0$. The resulting label is:

$$y_t = \textsc{Reroute} \text{ if } z_t^{\text{stop}} = 0 \wedge z_t^{\text{reroute}} = 1; \quad y_t = \textsc{Continue} \text{ o.w.}. \tag{22}$$

Appendix F reports the trajectory-generation model and annotation prompt. Using these annotations, we initialize the plug-in from Qwen3-0.6B-Base and fine-tune it on the turn-level action labels. Given the current state $s_t = (q, h_{<t}, a_t)$, the plug-in generates exactly one label from $\{\textsc{Continue}, \textsc{Reroute}\}$. Training details are provided in Appendix D.

Training Data Construction. We use **Tongyi-DeepResearch-30B-A3B** to generate complete multi-turn deep-research trajectories for 3,229 questions. To avoid test leakage, trajectories corresponding to the 359 BrowseCore test questions are excluded from controller training. We then use GPT-4.1 to evaluate the correctness of each final answer and retain 1,575 correct trajectories, comprising 32,351 turns ending with actions such as `search`, `open`, or `find`. For each turn t, GPT-4.1 receives the original question, the gold answer, the accumulated deep-research trajectory through the current action result, and the current action type. The gold answer is used only as the annotation target and is not treated as evidence. The judge first determines whether the accumulated evidence is sufficient to answer the question and then, when the evidence is insufficient, determines whether the current search direction remains productive. After annotation-quality filtering, 1,475 questions have complete turn-level annotations, yielding 29,252 examples. We split the data by question ID into 26,351 training examples and 2,901 test examples, ensuring that all turns associated with the same question remain in a single partition. The complete annotation prompt is provided at Figure 8. Turn-level action annotation with GPT-4.1 cost approximately $1,800.

F Prompt

F.1 Deep Research System Prompt

Figure 6 presents an example system prompt used by LiteResearcher-4B in our experiments. The prompt specifies the agent role, tool-use protocol, reasoning format, and final answer format. It illustrates how the Deep Research agent is instructed to iteratively reason, interact with browser tools, and return a final answer after sufficient information has been collected.

System Prompt for LiteResearcher

You are a **Deep AI Research Assistant**.

The question is a complex question that requires **deep research** to answer.
The runtime provides browser tools for searching information and inspecting retrieved webpages.

You do not need to answer the question immediately. You should first reason about the research plan or determine what to search next.

Output Format
Your output must follow one of the following two formats.

```
<think>
YOUR THINKING PROCESS
</think>
<answer>
YOUR ANSWER AFTER GETTING ENOUGH INFORMATION
</answer>
```

or

```
<think>
YOUR THINKING PROCESS
</think>
<tool_call>
YOUR TOOL CALL WITH CORRECT FORMAT
</tool_call>
```

The runtime provides the available browser function signatures separately. Use only the exact function names and argument schemas supplied by the runtime. Do not invent aliases or rely on any other tool interface.

For each function call, return a JSON object with the function name and arguments inside `<tool_call></tool_call>` tags.

```
<tool_call>
{
   "name":  <function-name>,
   "arguments":  <args-json-object>
}
</tool_call>
```

You should always follow the above formats strictly.
Only output the final answer in words, numbers, or a short phrase inside the `<answer></answer>` tag, without explanations or extra information. For yes-or-no questions, output only `yes` or `no`.

Current date: {current_date}

Figure 6: System prompt used by LiteResearcher during Deep Research. The model iteratively reasons, invokes browser tools, and produces a final answer after sufficient evidence has been collected.

F.2 Browser Tools

Figure 7 presents an example of the browser tool interfaces provided to LiteResearcher-4B in our experiments. The runtime exposes tools for web search, webpage inspection, and in-page information lookup, enabling the agent to iteratively gather evidence during long-horizon research.

Browser Tool Specification for LiteResearcher-4B

The runtime provides three browser tools for external information retrieval and webpage inspection.

browser.search

Searches for information related to a query and returns the top retrieved results with titles, URLs, and summaries.

```
{
  "query": <string>,
  "topn": <integer, default=10>
}
```

browser.open

Opens a retrieved link or a fully qualified URL and optionally displays a specified region of the page.

```
{
  "id": <integer or string>,
  "cursor": <integer>,
  "loc": <integer>,
  "num_lines": <integer>,
  "view_source": <boolean>,
  "source": <string>
}
```

browser.find

Finds exact matches of a text pattern within the current or specified webpage.

```
{
  "pattern": <string>,
  "cursor": <integer>
}
```

Figure 7: Browser tool specifications provided to LiteResearcher-4B. The agent uses `browser.search` for retrieval, `browser.open` for webpage inspection, and `browser.find` for locating specific information within a page.

F.3 Turn-Level Annotation for Plug-in Training

Figure 8 shows the prompt template used to annotate each deep-research turn with one of two labels: CONTINUE or REROUTE, providing turn-level supervision for training the plug-in model.

F.4 Evidence Document Verification

Figure 9 presents the prompt template used to verify candidate evidence documents. For each document, the annotator first determines whether it SUPPORTS, is RELATED-BUT-INSUFFICIENT to, or is IRRELEVANT to the target clue. It then determines whether the document contains sufficient evidence to answer the original question. For positive cases, the shortest supporting evidence span is also extracted.

F.5 Prompt Template for Clue Decomposition

Figure 10 shows the prompt used for clue decomposition.

Prompt Template for Turn-Level Action Annotation

You are a strict evidence-sufficiency judge.

Evaluate {case_count} independent cases. For each case, use only the evidence in that case's cumulative history. Never transfer information between cases and do not use outside knowledge. A ground-truth answer is only the evaluation target; it is not itself evidence.

Judgment A: Evidence sufficiency.

- **STOP = 1:** The evidence collected so far contains sufficient information to support all essential claims in the ground-truth answer. The question can therefore be answered correctly without further retrieval.
- **STOP = 0:** At least one essential part of the ground-truth answer is unsupported, unclear, ambiguous, or requires additional retrieval.

An unsupported guess or claim made by the assistant does not count as evidence.

Judgment B: Search-direction assessment.

If **STOP = 0**, evaluate the current search direction:

- **REROUTE = 1:** The current search direction is no longer productive. The agent should switch to a substantially different direction to obtain the missing evidence.
- **REROUTE = 0:** The current search direction remains productive, and further retrieval along the same direction may provide the missing evidence.

A single failed tool call is not sufficient to trigger a reroute when the broader search direction remains promising.

Label Definitions

- **REROUTE:** $\mathrm{STOP} = 0$ and $\mathrm{REROUTE} = 1$.
- **CONTINUE:** $\mathrm{STOP} = 0$ and $\mathrm{REROUTE} = 0$. The available evidence is insufficient, but the current search direction remains productive and should be continued.

If **STOP = 1**, **REROUTE** must be 0, because no further retrieval is needed.

Output Format

Return only a JSON array in `case_id` order, with no Markdown or explanation:

```
[{"case_id":0,"stop":0,"reroute":0}]
```

The array must contain exactly {case_count} objects, one per case.

Input Cases

{cases}

Figure 8: Prompt template used to annotate each deep-research turn with CONTINUE, or REROUTE.

G BROWSECORE

To support controlled and reproducible retrieval experiments for Deep Research agents, we construct **BrowseCore**, a compact benchmark with manually verified evidence. BrowseCore is built from OpenResearcher question–answer pairs while substantially reducing the scale of the retrieval corpus and strengthening evidence quality through explicit verification. For each retained question, we identify supporting and gold documents, mine challenging hard negatives, and ensure that the verified evidence is sufficient to derive the reference answer. The resulting benchmark contains dedicated training and development splits and is designed to enable efficient analysis of retrieval behavior and long-horizon Deep Research trajectories.

G.1 BACKGROUND AND MOTIVATION

Existing Deep Research benchmarks often rely on large-scale retrieval corpora, making controlled retrieval analysis computationally expensive. Among them, OpenResearcher provides a representative

Prompt Template for Evidence Document Verification

You are labeling candidate documents for a multi-hop question answering dataset.

Original question: {original_question}
Ground-truth answer: {original_answer}
Target clue: {target_subquestion}
Candidate documents: {documents_json}

For each candidate document, make two judgments.

Judgment A: Clue-level evidence assessment.
Classify the document with respect to the target clue:

- **SUPPORTS:** The document provides explicit or clearly inferable evidence that answers the target clue.
- **RELATED_BUT_INSUFFICIENT:** The document is relevant to the topic, entity, or clue, but does not provide sufficient evidence to answer the target clue.
- **IRRELEVANT:** The document does not meaningfully help answer the target clue.

Judgment B: Original-question evidence assessment.
Determine whether this single document contains sufficient evidence to answer the original question with the given ground-truth answer.
Set `can_answer_original=true` only if the document contains sufficient explicit or logically inferable evidence to derive the final answer.
A gold document is not defined solely by exact string matching. The answer may be implicit, paraphrased, or expressed in a different surface form. If the document only contains the answer string without establishing its relation to the original question, set `can_answer_original=false`.

Important Rules

- Do not use outside knowledge.
- Do not treat shared keywords, similar topics, or partial entity matches as sufficient evidence.
- If the label is `SUPPORTS`, identify the shortest supporting evidence span.
- If `can_answer_original=true`, identify the shortest span supporting the final answer.
- Preserve the original `document_id` for every candidate document.

Output Format

```
{ "results": [ { "document_id": "...",
"label": "SUPPORTS" | "RELATED_BUT_INSUFFICIENT" |
"IRRELEVANT",
"evidence_span": "...",
"rationale": "...",
"can_answer_original": true | false,
"answer_evidence_span": "...",
"answer_rationale": "..." } ] }
```

Figure 9: Prompt template for automated evidence-document verification.

open benchmark with large-scale web documents and question–answer pairs, which we use as the starting point for constructing BrowseCore.

OpenResearcher provides approximately 10K gold documents for 6K questions in an open benchmark specifically designed to assess whether deep-research AI agents can interactively and creatively navigate the web to find complex, hard-to-find information. To approximate real-world web coverage and reflect realistic search complexity, it collects 15 million documents (approximately 10 trillion tokens) from FineWeb (Penedo et al., 2024). For efficient large-scale dense retrieval, each document is embedded with Qwen3-Embedding-8B (Zhang et al., 2025) and indexed using FAISS; the resulting corpus.pkl file is 113.65 GB.

Prompt Template for Clue Decomposition

You are decomposing a complex question into **atomic factual clues** for evidence retrieval.

You are given an original question and its reference answer. Your task is to extract the individual factual conditions in the question that the reference answer must satisfy.

Original question: {question}
Reference answer: {answer}

Rules

- Generate the **smallest complete set of clues** needed to cover the conditions in the original question.
- Each clue must express only **one factual condition or relation**.
- Write each clue as a **self-contained declarative statement**.
- Preserve important constraints such as **dates, locations, author roles, affiliations, quantities, titles, and relationships**.
- Use only information stated in the original question or reference answer.
- Do not invent facts, aliases, entities, or supporting evidence.
- Do not generate retrieval queries, explanations, URLs, or evidence spans.
- Return no text outside the JSON object.

Output Format

```
{  "clues":  [
   {
     "clue_id":  "C1",
     "clue":  "A short, self-contained factual
condition."
   }
]
}
```

Figure 10: Prompt template for atomic clue decomposition. Given an original question and its reference answer, the model decomposes the information need into a minimal set of self-contained factual clues for subsequent evidence retrieval and verification.

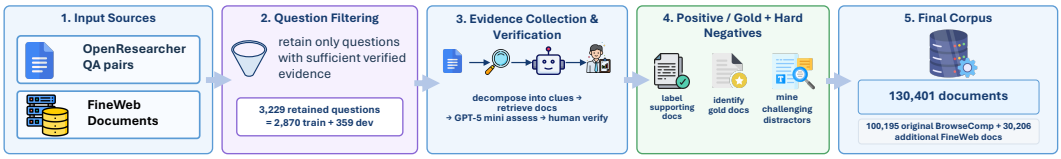

Figure 11: Overview of our benchmark and corpus construction pipeline. We start from OpenResearcher question–answer pairs, collect candidate evidence through clue-level retrieval, and combine GPT-5 mini assessment with human verification to filter unverifiable questions. Verified supporting documents are labeled as positive and gold evidence, while high-ranking non-supporting documents are retained as hard negatives. The final corpus augments the BrowseComp-Plus retrieval corpus with these verified evidence documents and mined distractors, resulting in a compact and efficient benchmark for controlled Deep Research experiments.

Although this setup is effective and widely used for end-to-end evaluation of integrated deep-research systems, it introduces several limitations for controlled retrieval research. During gold-document construction, OpenResearcher concatenates the question and reference answer to form the search query in order to improve recall, and then retrieves web content through the Serper API. However, the retrieved candidate documents are not exhaustively verified through human evaluation. As a result, the gold set may be incomplete or contain weakly validated evidence documents. In addition, the extremely large corpus makes experiments computationally expensive and less suitable for fine-

grained analysis of retrieval behavior and research insights. Moreover, the retrieved documents are not systematically verified for whether they actually contain sufficient evidence to support the reference answer. Therefore, in this work, we aim to construct a high-quality and easy-to-use benchmark with a compact, carefully verified retrieval corpus.

G.2 Benchmark and Corpus Construction

As shown in Figure 11, we construct BrowseCore from question–answer pairs in OpenResearcher through a four-stage pipeline consisting of question filtering, evidence collection and verification, hard-negative mining, and final corpus assembly. Rather than directly treating automatically retrieved documents as gold evidence, we explicitly verify whether the retrieved documents support the decomposed clues and whether the resulting evidence is sufficient to derive the reference answer.

Firstly, we adopt a clue-level verification protocol. For each question, the question is first decomposed into self-contained clues, and candidate documents are retrieved for each clue using Qwen3-Embedding-8B. Preliminary relevance judgments and evidence spans are then generated automatically with GPT-5 mini, followed by final human verification. Annotators verify that each retained clue is supported by explicit textual evidence and that the complete set of verified evidence is sufficient for a human to derive the reference answer. If sufficient evidence cannot be identified, the clues are revised and additional retrieval rounds are performed, for up to five rounds in total. Questions for which adequate supporting evidence still cannot be established after five rounds are removed from the benchmark. This process yields a compact and carefully verified retrieval corpus that is easier to index and retrieve from, while remaining suitable for controlled analysis of retrieval quality and long-horizon Deep Research trajectories.

G.2.1 Question Source and Filtering

We begin with question–answer pairs from the training set of OpenResearcher. For each question, we first attempt to collect sufficient supporting evidence through automated retrieval and human verification. Questions are retained only when the retrieved evidence can be verified to support the key clues and is sufficient for a human to derive the reference answer. If adequate evidence cannot be identified after iterative retrieval and manual review, the question is removed from the benchmark. This filtering process results in 3,229 retained questions, including 2,870 training questions and 359 development questions.

G.2.2 Candidate Document Sources

To collect evidence, we search over the document corpus associated with OpenResearcher, which contains approximately 15 million FineWeb documents, together with the original retrieval corpus used for the BrowseComp questions. These documents serve as the candidate pool from which supporting and distracting documents are identified.

G.2.3 Evidence Collection and Verification

Following BrowseComp-Plus, we first decompose each question into a set of self-contained clues, using the prompt shown in Figure 10, that jointly lead to the reference answer. For each clue, we use Qwen3-Embedding-8B to retrieve the top five candidate documents from the corpus. We then use GPT-5 mini to automatically assess whether each retrieved document supports the corresponding clue, using the prompt shown in Figure 9. The model distinguishes supporting evidence from documents that are merely related or irrelevant, and extracts the shortest supporting evidence span whenever applicable. Human annotators subsequently review the automatically generated relevance labels and evidence spans. For each question, annotators are required to verify that every clue is justified by explicit textual evidence and that the complete set of verified clues is sufficient for a human to derive the reference answer.

Specifically, annotators are asked to:

- confirm that each clue is supported by the retrieved documents and label the exact text spans that justify the clue; and

- determine whether the combination of verified clues and supporting evidence is sufficient to answer the entire question correctly.

If the initial retrieval results do not provide sufficient evidence, annotators revise the decomposed clues and perform additional rounds of retrieval. This process is repeated for up to five rounds. If sufficient supporting evidence still cannot be identified after five retrieval rounds, the question is considered unverifiable and removed from the benchmark.

G.2.4 Positive and Gold Document Identification

Based on the evidence verification results, documents that provide sufficient support for at least one decomposed clue are labeled as positive evidence documents. Among these positives, annotators further identify documents that contain sufficient information to directly derive or verify the final reference answer; we designate these as gold documents.

Following BrowseComp-Plus, gold documents are not restricted to documents containing the reference answer as an exact substring. A document may still be considered gold when the answer is expressed implicitly, paraphrased, or in a different surface form, as long as the document provides sufficient evidence to derive the final answer. This distinction enables us to separately evaluate retrieval of useful intermediate evidence and retrieval of documents that directly support the final answer.

G.2.5 Hard-Negative Mining

To maintain a manageable corpus while preserving retrieval difficulty, we further mine hard-negative documents as realistic distractors. Following BrowseComp-Plus, we use GPT-5 mini to decompose each original question into simpler, self-contained sub-queries. For each sub-query, Qwen3-Embedding-8B retrieves the top 10 candidate documents from the corpus.

Because these candidates are retrieved using individual clues rather than the complete set of constraints in the original question, they are often semantically relevant to part of the information need but insufficient for answering the full question. We therefore remove all documents previously verified as positive or gold evidence and retain the remaining high-ranking, non-supporting documents as hard negatives. These documents provide challenging distractors that are topically related to the query while lacking sufficient evidence to derive the reference answer.

G.2.6 Final Corpus Assembly

For each question, we augment the original BrowseComp-Plus retrieval corpus with our newly verified positive and gold evidence documents, together with the mined hard-negative documents. This construction preserves the retrieval difficulty of BrowseComp-Plus while ensuring that each retained question has a manually verified evidence path to the reference answer. The resulting benchmark therefore provides both reliable evidence supervision and a compact retrieval setting suitable for controlled analysis of retrieval quality, rerouting behavior, and long-horizon Deep Research trajectories. The final retrieval corpus contains 130,401 documents in total, including 100,195 documents from the original BrowseComp corpus and 30,206 additional documents collected and verified from OpenResearcher. The corpus occupies approximately 1.8 GB on disk, providing a substantially more compact retrieval environment for controlled Deep Research experiments. Note: We use GPT-5 mini for clue decomposition and preliminary evidence assessment, followed by final human verification conducted by four Ph.D. student annotators.

H Proof of Main Theorem

We establish the theoretical properties of the RIVO logit intervention, showing how token-level corrections can shift the generation distribution toward higher-value reroute queries. By modeling the logit modification as a function of the Plugin parameters θ, we analyze how trajectory-level reward learning induces changes in reroute-query probabilities and, under suitable conditions, improves the expected downstream value of the generated queries.

Consider a Deep Research turn t at which REROUTE is triggered. Let $\mathcal{F}_{t-1}$ denote the research history available before generating the reroute query at turn t, and let $\mathcal{F}_{t,s-1}$ denote the corresponding history before generating token position s of that query. Let $\theta \in \Theta \subseteq \mathbb{R}^d$ denote the trainable Plugin parameters, and let $\mathbf{z}_{t,s}^H$ denote the vocabulary-level logits produced by the frozen Host at token position s. For each candidate token $x_j \in \mathcal{V}$, the Plugin produces an effective logit modification $g_{t,s,j}(\theta)$, which modifies the Host distribution used to generate the reroute query. Before presenting the main theoretical result, we first introduce the assumptions required for the analysis.

Assumption 1 (Bounded Logit Intervention)**.** *Let $g_{t,s,j}(\theta)$ denote the effective Plugin-induced logit correction applied to candidate token $x_j \in \mathcal{V}$ at token position s of the reroute query generated at Deep Research turn t. We assume that there exists a constant $G > 0$ such that $|g_{t,s,j}(\theta)| \le G$ for all relevant Deep Research turns t, token positions s, and candidate tokens $x_j \in \mathcal{V}$. We further assume that the reroute-query length is bounded by $L_t \le L_{\max}$ and that the downstream value is bounded as $|Q_t(q_t)| \le Q_{\max}$.*

Assumption 2 (Value Alignment with Downstream Return)**.** *Let $q_t = (x_{t,1}, \dots, x_{t,L_t})$ denote the complete reroute query generated at Deep Research turn t. We define its downstream value as*

$$Q_t(q_t) = \mathbb{E}\left[R(\tau_{1:T}) \mid q_t, \mathcal{F}_{t-1}\right], \tag{23}$$

where $\tau_{1:T}$ denotes the complete Deep Research trajectory. We further define the first-order query-level probability shift as

$$S_t(\theta; q_t) = \left.\frac{\partial}{\partial \beta} \log \pi_{\beta,\theta}(q_t \mid \mathcal{F}_{t-1})\right|_{\beta=0}. \tag{24}$$

We assume that the chosen intervention direction is locally aligned with downstream value, such that

$$\beta \ \mathrm{Cov}_{q_t \sim \pi_H(\cdot|\mathcal{F}_{t-1})}\left(Q_t(q_t), S_t(\theta; q_t)\right) > 0. \tag{25}$$

Lemma 1 (Query-Distribution Shift under Logit Intervention)**.** *Consider a reroute query $q_t = (x_{t,1}, \dots, x_{t,L_t})$ generated at Deep Research turn t when rerouting is triggered. At token position s, let $\mathbf{z}_{t,s}^H$ denote the logits produced by the frozen Host model, and let $g_{t,s,j}(\theta)$ denote the Plugin-induced logit modification for candidate token $x_j \in \mathcal{V}$. For an intervention strength $\beta \in \mathbb{R}$, define the induced token policy as*

$$\pi_{\beta,\theta}(x_j \mid \mathcal{F}_{t,s-1}) = \frac{\exp\left(z_{t,s,j}^H + \beta g_{t,s,j}(\theta)\right)}{\sum_{k \in \mathcal{V}} \exp\left(z_{t,s,k}^H + \beta g_{t,s,k}(\theta)\right)}. \tag{26}$$

Then $\pi_{\beta,\theta}$ is differentiable with respect to β, and

$$\frac{\partial}{\partial \beta} \log \pi_{\beta,\theta}(x_j \mid \mathcal{F}_{t,s-1}) = g_{t,s,j}(\theta) - \mathbb{E}_{x_k \sim \pi_{\beta,\theta}}\left[g_{t,s,k}(\theta) \mid \mathcal{F}_{t,s-1}\right]. \tag{27}$$

Equivalently,

$$\frac{\partial}{\partial \beta} \pi_{\beta,\theta}(x_j \mid \mathcal{F}_{t,s-1}) = \pi_{\beta,\theta}(x_j \mid \mathcal{F}_{t,s-1}) \left(g_{t,s,j}(\theta) - \mathbb{E}_{x_k \sim \pi_{\beta,\theta}}[g_{t,s,k}(\theta)]\right). \tag{28}$$

Consequently, the probability of the complete reroute query satisfies

$$\pi_{\beta,\theta}(q_t \mid \mathcal{F}_{t-1}) = \prod_{s=1}^{L_t} \pi_{\beta,\theta}(x_{t,s} \mid \mathcal{F}_{t,s-1}), \tag{29}$$

and its first-order probability shift is characterized by

$$S_t(\theta; q_t) := \left.\frac{\partial}{\partial \beta} \log \pi_{\beta,\theta}(q_t \mid \mathcal{F}_{t-1})\right|_{\beta=0}. \tag{30}$$

Proof. Token-level probability shift. For a candidate token $x_j \in \mathcal{V}$, the intervened token policy is

$$\pi_{\beta,\theta}(x_j \mid \mathcal{F}_{t,s-1}) = \frac{\exp\left(z_{t,s,j}^H + \beta g_{t,s,j}(\theta)\right)}{\sum_{k \in \mathcal{V}} \exp\left(z_{t,s,k}^H + \beta g_{t,s,k}(\theta)\right)}. \tag{31}$$

Taking logarithms gives

$$\log \pi_{\beta,\theta}(x_j \mid \mathcal{F}_{t,s-1}) = z^H_{t,s,j} + \beta g_{t,s,j}(\theta) - \log \sum_{k\in\mathcal{V}} \exp\big(z^H_{t,s,k} + \beta g_{t,s,k}(\theta)\big)\,. \tag{32}$$

Differentiating with respect to β yields

$$\frac{\partial}{\partial\beta} \log \pi_{\beta,\theta}(x_j \mid \mathcal{F}_{t,s-1}) = g_{t,s,j}(\theta) - \sum_{k\in\mathcal{V}} \pi_{\beta,\theta}(x_k \mid \mathcal{F}_{t,s-1}) g_{t,s,k}(\theta). \tag{33}$$

This derivative characterizes the local sensitivity of the token probability to the intervention strength β, indicating the direction and rate of probability change with respect to β. The second term is the probability-weighted expectation of the Plugin-induced logit modification:

$$\sum_{k\in\mathcal{V}} \pi_{\beta,\theta}(x_k \mid \mathcal{F}_{t,s-1}) g_{t,s,k}(\theta) = \mathbb{E}_{x_k\sim\pi_{\beta,\theta}}\left[g_{t,s,k}(\theta) \mid \mathcal{F}_{t,s-1}\right]. \tag{34}$$

Hence,

$$\frac{\partial}{\partial\beta} \log \pi_{\beta,\theta}(x_j \mid \mathcal{F}_{t,s-1}) = g_{t,s,j}(\theta) - \mathbb{E}_{x_k\sim\pi_{\beta,\theta}}\left[g_{t,s,k}(\theta) \mid \mathcal{F}_{t,s-1}\right]. \tag{35}$$

Using

$$\frac{\partial}{\partial\beta}\pi_{\beta,\theta} = \pi_{\beta,\theta}\frac{\partial}{\partial\beta}\log \pi_{\beta,\theta}, \tag{36}$$

we obtain

$$\frac{\partial}{\partial\beta}\pi_{\beta,\theta}(x_j \mid \mathcal{F}_{t,s-1}) = \pi_{\beta,\theta}(x_j \mid \mathcal{F}_{t,s-1})\left(g_{t,s,j}(\theta) - \mathbb{E}_{x_k\sim\pi_{\beta,\theta}}\left[g_{t,s,k}(\theta) \mid \mathcal{F}_{t,s-1}\right]\right). \tag{37}$$

This identity gives a direct interpretation of the probability shift. For brevity, let $\mathbb{E}[g]$ denote the probability-weighted average Plugin-induced logit correction over the vocabulary. Then $g_{t,s,j}(\theta) - \mathbb{E}[g]$ represents the relative correction assigned to token x_j with respect to the current token distribution. Consequently, the change in its probability is determined by both its current probability $\pi_{\beta,\theta}(x_j)$ and this relative correction. In particular, $g_{t,s,j}(\theta) > \mathbb{E}[g]$ implies $\partial\pi_{\beta,\theta}(x_j)/\partial\beta > 0$, whereas $g_{t,s,j}(\theta) < \mathbb{E}[g]$ implies $\partial\pi_{\beta,\theta}(x_j)/\partial\beta < 0$. Thus, a positive change in β shifts probability mass toward tokens receiving above-average logit corrections, whereas a negative change in β reverses this direction.

Query-level probability shift. Let

$$q_t = (x_{t,1}, x_{t,2}, \ldots, x_{t,L_t}) \tag{38}$$

denote a complete reroute query. By autoregressive factorization,

$$\pi_{\beta,\theta}(q_t \mid \mathcal{F}_{t-1}) = \prod_{s=1}^{L_t} \pi_{\beta,\theta}(x_{t,s} \mid \mathcal{F}_{t,s-1}). \tag{39}$$

Taking logarithms,

$$\log \pi_{\beta,\theta}(q_t \mid \mathcal{F}_{t-1}) = \sum_{s=1}^{L_t} \log \pi_{\beta,\theta}(x_{t,s} \mid \mathcal{F}_{t,s-1}). \tag{40}$$

Differentiating with respect to β gives

$$\frac{\partial}{\partial\beta} \log \pi_{\beta,\theta}(q_t \mid \mathcal{F}_{t-1}) = \sum_{s=1}^{L_t} \frac{\partial}{\partial\beta} \log \pi_{\beta,\theta}(x_{t,s} \mid \mathcal{F}_{t,s-1}). \tag{41}$$

Applying the token-level result yields

$$\frac{\partial}{\partial\beta} \log \pi_{\beta,\theta}(q_t \mid \mathcal{F}_{t-1}) = \sum_{s=1}^{L_t} \left[g_{t,s,x_{t,s}}(\theta) - \mathbb{E}_{x_k\sim\pi_{\beta,\theta}}\left[g_{t,s,k}(\theta) \mid \mathcal{F}_{t,s-1}\right]\right]. \tag{42}$$

First-order shift around the host policy. At $\beta = 0$, the intervention vanishes and $\pi_{0,\theta} = \pi_H$. Therefore,

$$\frac{\partial}{\partial \beta} \log \pi_{\beta,\theta}(q_t \mid \mathcal{F}_{t-1})\Big|_{\beta=0} = \sum_{s=1}^{L_t} \left[g_{t,s,x_{t,s}}(\theta) - \mathbb{E}_{x_k \sim \pi_H} \left[g_{t,s,k}(\theta) \mid \mathcal{F}_{t,s-1} \right] \right]. \tag{43}$$

We define the resulting first-order query-level shift as

$$S_t(\theta; q_t) := \frac{\partial}{\partial \beta} \log \pi_{\beta,\theta}(q_t \mid \mathcal{F}_{t-1})\Big|_{\beta=0}. \tag{44}$$

This establishes how the token-level logit intervention induces a first-order shift in the probability of a complete reroute query around the frozen host policy.

□

Lemma 2 (First-Order Downstream Return Characterization)**.** *Fix the current history $\mathcal{F}_{t-1}$ and the downstream continuation policy. Define*

$$J_t(\beta) := \mathbb{E}_{q_t \sim \pi_{\beta,\theta}(\cdot|\mathcal{F}_{t-1})} \left[Q_t(q_t) \right], \tag{45}$$

where

$$Q_t(q_t) := \mathbb{E} \left[R(\tau_{1:T}) \mid \mathcal{F}_{t-1}, q_t \right]. \tag{46}$$

Here, $J_t(\beta)$ denotes the expected terminal task reward when a logit intervention of strength β is applied to the reroute query at Deep Research turn t, conditioned on the current history $\mathcal{F}_{t-1}$, and the agent subsequently completes the remaining Deep Research trajectory. $R(\tau_{1:T})$ denotes the terminal task reward evaluated after the complete Deep Research trajectory terminates. Thus, $Q_t(q_t)$ measures the expected final task reward obtained after selecting reroute query q_t and completing the remaining retrieval and reasoning process under the fixed downstream continuation policy.

Then, For a general intervention strength β, define the query-level score as

$$S_{t,\beta}(q_t) := \frac{\partial}{\partial \beta} \log \pi_{\beta,\theta}(q_t \mid \mathcal{F}_{t-1}). \tag{47}$$

Then,

$$\frac{\partial J_t(\beta)}{\partial \beta} = \mathbb{E}_{q_t \sim \pi_{\beta,\theta}(\cdot|\mathcal{F}_{t-1})} \left[Q_t(q_t) S_{t,\beta}(q_t) \right]. \tag{48}$$

At $\beta = 0$, we have $S_{t,0}(q_t) = S_t(\theta; q_t)$ and $\pi_{0,\theta} = \pi_H$. Therefore,

$$\frac{\partial J_t(\beta)}{\partial \beta}\Big|_{\beta=0} = \mathbb{E}_{q_t \sim \pi_H(\cdot|\mathcal{F}_{t-1})} \left[Q_t(q_t) S_t(\theta; q_t) \right]. \tag{49}$$

Moreover,

$$\mathbb{E}_{q_t \sim \pi_H(\cdot|\mathcal{F}_{t-1})} \left[S_t(\theta; q_t) \right] = 0. \tag{50}$$

Therefore,

$$\boxed{\frac{\partial J_t(\beta)}{\partial \beta}\Big|_{\beta=0} = \mathrm{Cov}_{q_t \sim \pi_H(\cdot|\mathcal{F}_{t-1})} \left(Q_t(q_t), S_t(\theta; q_t) \right)}. \tag{51}$$

Proof. Since the current history $\mathcal{F}_{t-1}$ is fixed, expanding the expectation over all possible reroute queries gives

$$J_t(\beta) = \sum_{q_t} \pi_{\beta,\theta} \left(q_t \mid \mathcal{F}_{t-1} \right) Q_t(q_t). \tag{52}$$

Under Assumption 1, the relevant quantities are bounded, allowing differentiation to be interchanged with the expectation over reroute queries. Because the downstream continuation policy is fixed, $Q_t(q_t)$ does not depend on β. Differentiating with respect to β therefore gives

$$\frac{\partial J_t(\beta)}{\partial \beta} = \sum_{q_t} Q_t(q_t) \frac{\partial}{\partial \beta} \pi_{\beta,\theta} \left(q_t \mid \mathcal{F}_{t-1} \right). \tag{53}$$

Using the identity

$$\frac{\partial}{\partial\beta}\pi_{\beta,\theta}\left(q_t \mid \mathcal{F}_{t-1}\right) = \pi_{\beta,\theta}\left(q_t \mid \mathcal{F}_{t-1}\right)\frac{\partial}{\partial\beta}\log\pi_{\beta,\theta}\left(q_t \mid \mathcal{F}_{t-1}\right), \tag{54}$$

we obtain

$$\frac{\partial J_t(\beta)}{\partial\beta} = \mathbb{E}_{q_t\sim\pi_{\beta,\theta}(\cdot|\mathcal{F}_{t-1})}\left[Q_t(q_t)\frac{\partial}{\partial\beta}\log\pi_{\beta,\theta}\left(q_t \mid \mathcal{F}_{t-1}\right)\right]. \tag{55}$$

The expectation above is a probability-weighted average over all possible reroute queries, where each query q_t is weighted by its current probability $\pi_{\beta,\theta}(q_t \mid \mathcal{F}_{t-1})$. Evaluating at $\beta = 0$, where $\pi_{0,\theta} = \pi_H$, and using the definition of $S_t(\theta; q_t)$ from Lemma 1, yields

$$\left.\frac{\partial J_t(\beta)}{\partial\beta}\right|_{\beta=0} = \mathbb{E}_{q_t\sim\pi_H(\cdot|\mathcal{F}_{t-1})}\left[Q_t(q_t)S_t(\theta; q_t)\right]. \tag{56}$$

Next, the query-level score has zero expectation under the host policy:

$$\mathbb{E}_{q_t\sim\pi_H(\cdot|\mathcal{F}_{t-1})}\left[S_t(\theta; q_t)\right] = \sum_{q_t}\pi_H\left(q_t \mid \mathcal{F}_{t-1}\right)\left.\frac{\partial}{\partial\beta}\log\pi_{\beta,\theta}\left(q_t \mid \mathcal{F}_{t-1}\right)\right|_{\beta=0} \tag{57}$$

$$= \sum_{q_t}\left.\frac{\partial}{\partial\beta}\pi_{\beta,\theta}\left(q_t \mid \mathcal{F}_{t-1}\right)\right|_{\beta=0} \tag{58}$$

$$= \left.\frac{\partial}{\partial\beta}\sum_{q_t}\pi_{\beta,\theta}\left(q_t \mid \mathcal{F}_{t-1}\right)\right|_{\beta=0} \tag{59}$$

$$= \frac{\partial}{\partial\beta}1 = 0. \tag{60}$$

Therefore, by the definition of covariance,

$$\begin{aligned}&\mathrm{Cov}_{q_t\sim\pi_H(\cdot|\mathcal{F}_{t-1})}\left(Q_t(q_t), S_t(\theta; q_t)\right)\\ &\quad= \mathbb{E}_{q_t\sim\pi_H(\cdot|\mathcal{F}_{t-1})}\left[Q_t(q_t)S_t(\theta; q_t)\right]\\ &\qquad- \mathbb{E}_{q_t\sim\pi_H(\cdot|\mathcal{F}_{t-1})}\left[Q_t(q_t)\right]\mathbb{E}_{q_t\sim\pi_H(\cdot|\mathcal{F}_{t-1})}\left[S_t(\theta; q_t)\right].\end{aligned} \tag{61}$$

Since

$$\mathbb{E}_{q_t\sim\pi_H(\cdot|\mathcal{F}_{t-1})}\left[S_t(\theta; q_t)\right] = 0, \tag{62}$$

it follows that

$$\mathrm{Cov}_{q_t\sim\pi_H(\cdot|\mathcal{F}_{t-1})}\left(Q_t(q_t), S_t(\theta; q_t)\right) = \mathbb{E}_{q_t\sim\pi_H(\cdot|\mathcal{F}_{t-1})}\left[Q_t(q_t)S_t(\theta; q_t)\right]. \tag{63}$$

Combining the above identities gives

$$\boxed{\left.\frac{\partial J_t(\beta)}{\partial\beta}\right|_{\beta=0} = \mathrm{Cov}_{q_t\sim\pi_H(\cdot|\mathcal{F}_{t-1})}\left(Q_t(q_t), S_t(\theta; q_t)\right)}. \tag{64}$$

The left-hand side, $\partial J_t(\beta)/\partial\beta|_{\beta=0}$, measures the first-order change in the expected final task performance when the logit intervention is infinitesimally activated from the host policy. The right-hand side, $\mathrm{Cov}(Q_t(q_t), S_t(\theta; q_t))$, measures the alignment between the final value of a reroute query and the direction in which the Plugin shifts its probability. Here, $S_t(\theta; q_t)$ denotes the first-order shift score of the complete reroute query q_t, defined as the derivative of its log-probability with respect to the intervention strength β at $\beta = 0$. Intuitively, $S_t(\theta; q_t) > 0$ means that activating the Plugin tends to increase the probability of generating q_t, whereas $S_t(\theta; q_t) < 0$ means that the query is relatively suppressed. Its magnitude reflects how strongly the probability of q_t responds to the logit intervention. The left-hand side, $\partial J_t(\beta)/\partial\beta|_{\beta=0}$, measures the first-order change in expected downstream performance as the intervention strength moves away from the frozen Host policy. The covariance on the right-hand side measures the alignment between the downstream value of a reroute query and its first-order probability shift. Specifically, $S_t(\theta; q_t) > 0$ means that the probability of

q_t increases as β increases from zero, whereas $S_t(\theta; q_t) < 0$ means that it decreases. Therefore, the sign of $\mathrm{Cov}(Q_t, S_t)$ determines the locally improving direction of the intervention. A positive covariance favors increasing β, whereas a negative covariance favors decreasing β. Consequently, an intervention direction satisfying $\beta\ \mathrm{Cov}(Q_t, S_t) > 0$ produces a positive first-order change in expected downstream value. This completes the proof. □

Lemma 3 (Bounded Second-Order Variation of Downstream Return)**.** *Assume that the Plugin-induced logit modification is uniformly bounded,*

$$|g_{t,s,j}(\theta)| \le G, \tag{65}$$

for all token positions s, candidate tokens x_j, and relevant Plugin parameters θ. Assume further that the reroute-query length is bounded by $L_t \le L_{\max}$ and that

$$|Q_t(q_t)| \le Q_{\max}. \tag{66}$$

Then there exists a finite constant $B_J > 0$ such that

$$\left|\frac{\partial^2 J_t(\beta)}{\partial \beta^2}\right| \le B_J \tag{67}$$

for β in a neighborhood of zero. In particular, one may take

$$B_J = Q_{\max} G^2 \left(4L_{\max}^2 + L_{\max}\right). \tag{68}$$

Proof. Bounding the query-level score. Define the query-level score at intervention strength β as

$$S_{t,\beta}(q_t) := \frac{\partial}{\partial \beta} \log \pi_{\beta,\theta}(q_t \mid \mathcal{F}_{t-1}). \tag{69}$$

From Lemma 1,

$$S_{t,\beta}(q_t) = \sum_{s=1}^{L_t} \left[g_{t,s,x_{t,s}}(\theta) - \mathbb{E}_{x_k \sim \pi_{\beta,\theta}}\left[g_{t,s,k}(\theta) \mid \mathcal{F}_{t,s-1}\right]\right]. \tag{70}$$

Since $|g_{t,s,j}(\theta)| \le G$, we have

$$\left|g_{t,s,x_{t,s}}(\theta) - \mathbb{E}_{x_k \sim \pi_{\beta,\theta}}\left[g_{t,s,k}(\theta) \mid \mathcal{F}_{t,s-1}\right]\right| \le 2G. \tag{71}$$

Therefore,

$$\begin{aligned} |S_{t,\beta}(q_t)| &= \left|\sum_{s=1}^{L_t} \left[g_{t,s,x_{t,s}}(\theta) - \mathbb{E}_{x_k \sim \pi_{\beta,\theta}}\left[g_{t,s,k}(\theta) \mid \mathcal{F}_{t,s-1}\right]\right]\right| \\ &\le \sum_{s=1}^{L_t} \left|g_{t,s,x_{t,s}}(\theta) - \mathbb{E}_{x_k \sim \pi_{\beta,\theta}}\left[g_{t,s,k}(\theta) \mid \mathcal{F}_{t,s-1}\right]\right| \\ &\le \sum_{s=1}^{L_t} 2G \\ &= 2L_t G \\ &\le 2L_{\max} G. \end{aligned} \tag{72}$$

Bounding the derivative of the query-level score.

For each token position s, define

$$\mu_{t,s}(\beta) := \mathbb{E}_{x_k \sim \pi_{\beta,\theta}}\left[g_{t,s,k}(\theta) \mid \mathcal{F}_{t,s-1}\right]. \tag{73}$$

Since $g_{t,s,k}(\theta)$ does not explicitly depend on β,

$$\begin{aligned} \frac{\partial}{\partial \beta}\mu_{t,s}(\beta) &= \sum_{k \in \mathcal{V}} g_{t,s,k}(\theta) \frac{\partial}{\partial \beta} \pi_{\beta,\theta}\left(x_k \mid \mathcal{F}_{t,s-1}\right) \\ &= \sum_{k \in \mathcal{V}} \pi_{\beta,\theta}\left(x_k \mid \mathcal{F}_{t,s-1}\right) g_{t,s,k}(\theta)\left(g_{t,s,k}(\theta) - \mu_{t,s}(\beta)\right) \\ &= \mathbb{E}_{x_k \sim \pi_{\beta,\theta}}\left[g_{t,s,k}(\theta)^2 \mid \mathcal{F}_{t,s-1}\right] - \mu_{t,s}(\beta)^2 \\ &= \mathrm{Var}_{x_k \sim \pi_{\beta,\theta}}\left(g_{t,s,k}(\theta) \mid \mathcal{F}_{t,s-1}\right). \end{aligned} \tag{74}$$

Since $|g_{t,s,k}(\theta)| \le G$,

$$\mathrm{Var}_{x_k \sim \pi_{\beta,\theta}} \left(g_{t,s,k}(\theta) \mid \mathcal{F}_{t,s-1} \right) \le G^2. \tag{75}$$

Therefore,

$$\frac{\partial}{\partial \beta} S_{t,\beta}(q_t) = -\sum_{s=1}^{L_t} \mathrm{Var}_{x_k \sim \pi_{\beta,\theta}} \left(g_{t,s,k}(\theta) \mid \mathcal{F}_{t,s-1} \right). \tag{76}$$

Hence,

$$\left| \frac{\partial}{\partial \beta} S_{t,\beta}(q_t) \right| \le L_t G^2 \le L_{\max} G^2. \tag{77}$$

Bounding the second-order variation of the downstream return. From Lemma 2,

$$\frac{\partial J_t(\beta)}{\partial \beta} = \mathbb{E}_{q_t \sim \pi_{\beta,\theta}} \left[Q_t(q_t) S_{t,\beta}(q_t) \mid \mathcal{F}_{t-1} \right]. \tag{78}$$

Differentiating once more with respect to β gives

$$\frac{\partial^2 J_t(\beta)}{\partial \beta^2} = \mathbb{E}_{q_t \sim \pi_{\beta,\theta}} \left[Q_t(q_t) \left(S_{t,\beta}(q_t)^2 + \frac{\partial}{\partial \beta} S_{t,\beta}(q_t) \right) \middle| \mathcal{F}_{t-1} \right]. \tag{79}$$

Using $|Q_t(q_t)| \le Q_{\max}$ together with the bounds above,

$$\begin{aligned} \left| \frac{\partial^2 J_t(\beta)}{\partial \beta^2} \right| &= \left| \mathbb{E}_{q_t \sim \pi_{\beta,\theta}} \left[Q_t(q_t) \left(S_{t,\beta}(q_t)^2 + \frac{\partial}{\partial \beta} S_{t,\beta}(q_t) \right) \right] \right| \\ &\le \mathbb{E}_{q_t \sim \pi_{\beta,\theta}} \left[|Q_t(q_t)| \left(|S_{t,\beta}(q_t)|^2 + \left| \frac{\partial}{\partial \beta} S_{t,\beta}(q_t) \right| \right) \right] \\ &\le Q_{\max} \left((2 L_{\max} G)^2 + L_{\max} G^2 \right) \\ &= Q_{\max} \left(4 L_{\max}^2 G^2 + L_{\max} G^2 \right) \\ &= Q_{\max} G^2 \left(4 L_{\max}^2 + L_{\max} \right). \end{aligned} \tag{80}$$

Thus, defining

$$B_J := Q_{\max} G^2 \left(4 L_{\max}^2 + L_{\max} \right), \tag{81}$$

we obtain

$$\boxed{\left| \frac{\partial^2 J_t(\beta)}{\partial \beta^2} \right| \le B_J}. \tag{82}$$

This establishes the bounded second-order variation of the expected downstream return and completes the proof. □

Proof of Theorem 1. Let

$$C_t := \mathrm{Cov}_{q_t \sim \pi_H(\cdot \mid \mathcal{F}_{t-1})} \left(Q_t(q_t), S_t(\theta; q_t) \right). \tag{83}$$

By Lemma 2,

$$J_t'(0) = C_t. \tag{84}$$

By Assumption 2, the chosen intervention direction satisfies

$$\beta C_t > 0. \tag{85}$$

We now apply a second-order Taylor expansion of $J_t(\beta)$ around $\beta = 0$. For sufficiently small nonzero β, Taylor's theorem with Lagrange remainder gives

$$J_t(\beta) = J_t(0) + \beta J_t'(0) + \frac{\beta^2}{2} J_t''(\xi), \tag{86}$$

for some ξ lying between 0 and β.

Therefore,

$$J_t(\beta) - J_t(0) = \beta C_t + \frac{\beta^2}{2} J_t''(\xi). \tag{87}$$

By Lemma 3,

$$|J_t''(\xi)| \le B_J, \tag{88}$$

and hence

$$J_t''(\xi) \ge -B_J. \tag{89}$$

It follows that

$$\boxed{J_t(\beta) - J_t(0) \ge \beta C_t - \frac{B_J}{2}\beta^2}. \tag{90}$$

Since $\beta C_t > 0$, the right-hand side is strictly positive whenever

$$0 < |\beta| < \frac{2|C_t|}{B_J}. \tag{91}$$

Therefore,

$$\boxed{J_t(\beta) > J_t(0)}. \tag{92}$$

Thus, when the intervention direction is locally aligned with downstream value, a sufficiently small logit intervention improves the expected downstream value relative to the frozen Host policy. □