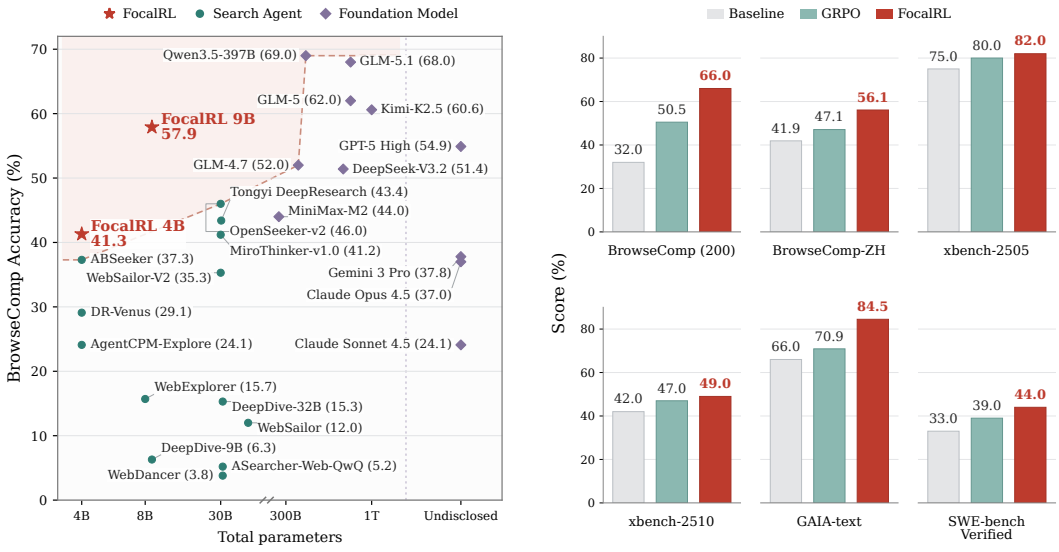
FOCAL CREDIT ASSIGNMENT FOR ULTRA LONG-HORIZON AGENTIC REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Training long-horizon large language model (LLM) agents with outcome-based reinforcement learning poses a credit-assignment challenge. GRPO assigns the same trajectory-level advantage to all actions, despite their different contributions to task success. Through a policy-gradient analysis, we identify two sources of mismatch between this trajectory-level advantage and the ideal action-specific credit: value shifts inherited from preceding history and stochasticity in subsequent interactions. Empirical analysis further reveals that sharp value changes are concentrated at a small fraction of actions. Motivated by these observations, we propose **FocalRL**, which tackles long-horizon credit assignment through local action repair. Specifically, an LLM-based locator first identifies consequential errors in failed trajectories and generates rubrics specifying the desired corrections. Then, we restore the interaction state before each identified error and sample bounded local continuations, which are evaluated against the repair rubrics. We combine these rubric-scored local groups with outcome-scored full-trajectory groups for policy optimization, providing targeted supervision that mitigates history value offsets and reduces the influence of downstream stochasticity. Experimental results show that FocalRL outperforms GRPO and most related methods under same rollout budget on representative agentic search and coding benchmarks, including BrowseComp, GAIA, and SWE-bench Verified. Notably, our deep research agent built on Qwen3.5-9B achieves **57.9**% accuracy on BrowseComp, setting a new state-of-the-art among sub-10B models. Our code is available at `anonymous.4open.science/r/FocalRL-5DE0`.



(a) Performance comparison on BrowseComp across models of various scales.

(b) Comparison of FocalRL with GRPO.

Figure 1: **FocalRL performance on deep research and software engineering.** In (b), baselines are SFT-tuned Qwen3.5-9B for deep research and vanilla Qwen3.5-4B for SWE-bench.

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.

1 INTRODUCTION

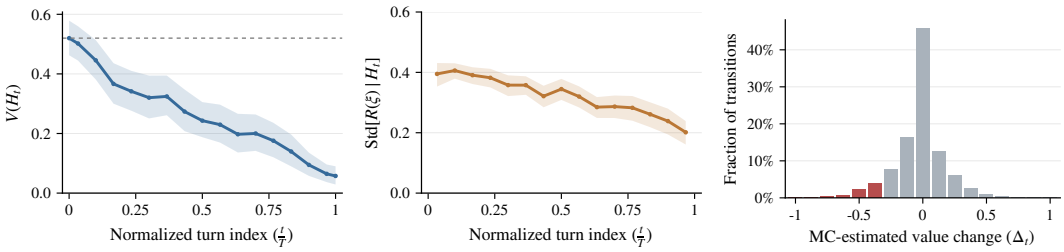
Large language models (LLMs) are increasingly deployed as agents that solve complex tasks through iterative interaction with environments. Among them, two prominent applications are deep research and software engineering: deep research agents (OpenAI, 2025; Team et al., 2025c) iteratively search, synthesize, and verify information from the web, while software-engineering agents (Yang et al., 2024) inspect repositories, execute commands, and test modifications. As a result, an agent trajectory can span hundreds of interaction turns and accumulate contexts of up to 256K tokens.

However, widely used reinforcement learning algorithms such as Group Relative Policy Optimization (GRPO) (Shao et al., 2024; Yu et al., 2026) rely solely on binary outcome rewards to train such ultra long-horizon agents, resulting in very sparse learning signals. For one thing, GRPO computes trajectory advantages based on reward differences within each group. In long-horizon tasks, trajectories in the same group often all succeed or all fail, yielding no effective learning signal. For example, when training software engineering agents with standard GRPO, we observed that nearly 30% of training groups had zero reward variance and thus provided no effective policy gradient signal. For another, GRPO assigns the same trajectory-level advantage to hundreds of actions in a trajectory, despite their vastly different contributions to task completion.

In this work, we revisit this problem from a policy gradient perspective. We decompose GRPO's trajectory-level advantage into three components (Eq. 2) : the ideal credit for each action, a history value offset induced by preceding interactions, and rollout noise arising from subsequent interactions. The latter two terms can cause the trajectory-level advantage to deviate from the ideal action-level credit. Empirically, mean state values along failed trajectories progressively decline as interactions accumulate (Figure 2(a)). This suggests that failed trajectories accumulate detrimental context: even a correct action at the current state may receive negative trajectory-level credit due to preceding mistakes. Meanwhile, stochasticity in policy sampling and environment dynamics introduces greater uncertainty over longer horizons. A current correct action may be undermined by subsequent errors, while an erroneous one may be rescued by later corrections. Consequently, earlier states exhibit greater outcome variance (Figure 2(b)). **As the credit-assignment horizon grows, the outcome rewards provide increasingly sparse supervision and obscure the contributions of individual actions**. Furthermore, Figure 2(c) shows that **the credit of actions are highly uneven**: most actions induce little change in value, while substantial value drops are concentrated among a small number of decisions. Therefore, long-horizon agentic RL requires precise credit assignment at consequential decisions without costly stepwise value estimation. Existing approaches incur expensive stepwise rollouts (Kazemnejad et al., 2025; Yu et al., 2025), rely on repeated states that are scarce in realistic agent trajectories (Feng et al., 2025; He et al., 2026), or select branching points heuristically while retaining terminal rewards (Dong et al., 2026; Ji et al., 2026), leaving this need insufficiently addressed.

To this end, we propose **FocalRL**, which focuses policy optimization on pivotal decision points. The key idea is to diagnose the errors underlying substantial value drops and use these diagnoses to guide targeted policy optimization. Specifically, we build local rollout groups at these decision points, combining bounded continuations with targeted rewards to train the policy to correct the diagnosed errors. This leads to three components. **Where to intervene.** To avoid massive Monte Carlo rollouts, we use an LLM-based locator to identify a consequential error and generate a repair rubric. The identified error determines the intervention point, while the rubric specifies the local repair objective. **How to rollout.** We restore the context before the identified error and sample an additional local group, with each continuation limited to several turns. In this way, using a shared history mitigates history-dependent value offsets in group-relative comparison, while limiting continuation length shortens the evaluation horizon to reduce rollout noise. **How to evaluate.** Since local continuations usually end before task completion, we evaluate them against repair rubrics instead of requiring the final answer. The resulting reward measures whether the agent avoids the diagnosed error and takes a valid correction, providing verifiable and targeted supervision for local rollout. Finally, we perform GRPO updates using both full rollout groups with terminal outcome rewards and local rollout groups with rubric-based rewards.

We implement FocalRL on top of Miles (Chen et al., 2026c) with fully asynchronous RL (Fu et al., 2026). We conduct extensive experiments on realistic long-horizon agent tasks, including deep research and software engineering. Experimental results show that FocalRL consistently outperforms standard GRPO on five deep research and two software engineering benchmarks, even trained with



(a) State value (b) Outcome variability (c) Adjacent value changes

Figure 2: **Value analysis of long-horizon agent trajectories. (a)** Mean continuation value $V(H_t)$ versus normalized turn index t/T. The dashed line indicates the mean initial value $V(H_0)$. **(b)** Within-prefix outcome standard deviation $\mathrm{Std}(R(\xi))$ versus t/T. **(c)** Distribution of MC-estimated adjacent value changes Δ_t, with drops of at least $3/8$ highlighted in red. Shaded bands indicate 95% confidence intervals obtained by bootstrapping trajectories.

fewer tokens. Notably, FocalRL achieves **57.9**% accuracy on BrowseComp with Qwen3.5-9B, establishing a new SOTA for sub-10B models and narrowing the gap to leading foundation models.

Our main contributions are summarized as follows:

- We identify two sources of credit-assignment noise in long-horizon GRPO and motivate policy optimization over shorter reward horizons
- We propose FocalRL, which constructs local rollout groups at pivotal decision points and scores bounded continuations against repair rubrics to guide targeted policy optimization.
- We evaluate on five representative deep research and two software engineering benchmarks, demonstrating the effectiveness of FocalRL.

2 Related Work

LLM Agents. With advances in reasoning and tool use, LLMs have moved beyond text generation (Radford et al., 2019; Raffel et al., 2020) to engage in interactive decision-making in real-world. Representative applications include deep research (OpenAI, 2025), software engineering (Yang et al., 2024; Wang et al., 2025), and embodied household tasks (Li et al., 2024). ReAct (Yao et al., 2022) provides a standard interaction paradigm for these applications, interleaving explicit reasoning with actions and incorporating subsequent observations into the context linearly. However, the context grows as interaction proceeds, making optimization over the full trajectory increasingly challenging. To address this challenge, we propose FocalRL, which identifies consequential actions and performs local optimization at these critical points to improve long-horizon task performance.

Reinforcement Learning for LLMs. Following DeepSeek-R1 (Guo et al., 2025), GRPO (Shao et al., 2024) has emerged as an important approach to LLM post-training. However, outcome-supervised GRPO broadcasts the same trajectory-level advantage to every generated token, resulting in coarse credit assignment. VinePPO (Kazemnejad et al., 2025), GPO (Yu et al., 2025) and PivotRL (Yi et al., 2026) use Monte Carlo rollouts to estimate value of every step, making their sampling costs computationally prohibitive for online agentic RL. Our method instead uses LLM-based error localization and bounded local rollouts, incurring modest additional overhead. Alternatively, process reward models (PRMs) provide step-level supervision (Lightman et al., 2024; Fan et al., 2026), but remain susceptible to reward hacking (Cheng et al., 2025). To address this concern, we construct verifiable local subproblems with explicit rubrics. GiGPO (Feng et al., 2025) and HGPO (He et al., 2026) introduce step-level advantages by grouping intermediate steps that share the same state. Nevertheless, these methods depend on repeated states across trajectories, which are very scarce in complex agent environments. Another line of work, including Tree-GRPO (Ji et al., 2026), TreeRL (Hou et al., 2025), and ARPO (Dong et al., 2026) instead explicitly sample continuations from intermediate states to build groups. However, each continuation is still evaluated by its terminal outcome, conflating local credit with latter actions. Moreover, their random- or entropy-based branching methods prioritize exploration rather than credit critical actions. In contrast, FocalRL explicitly localizes errors and provides more targeted and denser supervision signals.

3 FOCALRL

3.1 PRELIMINARIES

Agent Interaction. We adopt the widely used ReAct framework (Yao et al., 2023). At each turn $t = 0, \ldots, T-1$, the agent conditions on the interaction history $H_t = (x, (\tau_i, a_i, o_i)_{i=0}^{t-1})$ and generates a thought τ_t and an action a_t according to $(\tau_t, a_t) \sim \pi_\theta(\cdot \mid H_t)$, with $H_0 = (x)$. The environment then returns an observation o_t. This interaction produces a trajectory $\xi = (\tau_t, a_t, o_t)_{t=0}^{T-1}$, which receives a terminal outcome reward $R(\xi)$.

Outcome-Based GRPO. For each input x, GRPO samples a group of G trajectories $\{\xi_i\}_{i=1}^{G}$ using the rollout policy π_θ. The terminal rewards are normalized within the group to obtain a relative advantage for each trajectory:

$$A_i = R(\xi_i) - \mu_R, \tag{1}$$

where $\mu_R = \frac{1}{G}\sum_{j=1}^{G} R(\xi_j)$ denotes the mean group rewards. Following DeepSeek-v3.2 (Liu et al., 2025a), we omit normalization by the group standard deviation. The same trajectory-level advantage A_i is then broadcasted to all policy-generated tokens in ξ_i and used in RL.

3.2 MOTIVATION

In this section, we analyze the credit noise of outcome-based GRPO from a policy-gradient perspective. Given an interaction history H_t and the agent decision (τ_t, a_t) we define its value as $Q_\theta(H_t, \tau_t, a_t) = \mathbb{E}_{\pi_\theta}[R(\xi) \mid H_t, \tau_t, a_t]$, where the expectation is taken over future actions and tool responses. Similarly, we define the state value as $V_\theta(H_t) := \mathbb{E}_{\pi_\theta}[R(\xi) \mid H_t]$. Since μ_R unbiasedly estimates $V_\theta(H_0)$, we can decompose the advantage A_i as following:

$$\begin{aligned} A_i &= R(\xi_i) - \mu_R \approx R(\xi_i) - V_\theta(H_0) \\ &= \underbrace{Q_\theta(H_t, \tau_t, a_t) - V_\theta(H_t)}_{\text{action-specific credit}} + \underbrace{V_\theta(H_t) - V_\theta(H_0)}_{\text{history offset}} + \underbrace{R(\xi_i) - Q_\theta(H_t, \tau_t, a_t)}_{\text{rollout noise}}. \end{aligned} \tag{2}$$

Among the three terms, **the first** represents the action-specific advantage, which quantifies the current action's expected improvement over $V_\theta(H_t)$. Appendix A explains why this term serves as the ideal action-specific credit. **The second term** reflects the value offset induced by interactions preceding step t. For example, accumulated errors may make $V_\theta(H_t)$ substantially lower than the initial value $V_\theta(H_0)$. The action at t may improve the current situation, with $Q_\theta(H_t, \tau_t, a_t) > V_\theta(H_t)$, and thus have positive action-specific advantage. Nevertheless, the trajectory may still fail due to earlier errors and the correct action to be discouraged. **The third term** is the deviation of the sampled reward from its expectation, reflecting randomness arising from subsequent policy sampling and environment uncertainty. For example, even an correct action may be followed by incorrect reasoning, overlooked evidence, or temporary tool failures. These events can cause the trajectory to fail, again assigning negative credit to the earlier beneficial action.

To quantify the magnitudes of the history offset and rollout noise, we analyze 14,664 Monte Carlo continuations across 1,833 prefixes from 100 failed trajectories, using eight continuations per prefix. From Figure 2, we draw three observations.

- **Failed trajectories exhibit substantial history offsets.** As shown in Figure 2(a), mean state value declines as the normalized turn index increases. Thus, on average, the agent is less likely to succeed when continuing from later histories than when starting from the initial input alone.
- **Earlier prefixes exhibit greater outcome variability.** As shown in Figure 2(b), the within-prefix standard deviation of outcome rewards generally decreases as the normalized turn index increases, indicating greater uncertainty in outcomes when continuing from earlier prefixes.
- **Value changes by actions are unevenly distributed.** As shown in Figure 2(c): most actions induce small value changes, whereas a small fraction exhibits substantial value drops.

3.3 METHODOLOGY

We propose FocalRL to improve credit assignment at pivotal steps in ultra-long-horizon agentic RL, thereby enhancing overall model performance. Figure 3 and Algorithm 1 provide an overview of

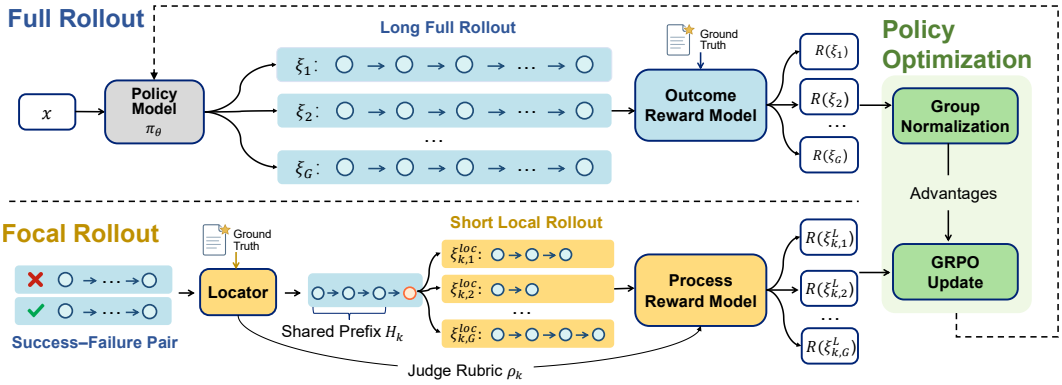


Figure 3: **Illustration of the FocalRL framework.** Upper left: long full rollouts evaluated by final-answer. Lower left: pivotal error identification and focal rollouts evaluated by local correction. Right: full and focal rollout rewards are separately group normalized for policy updates.

the framework. We first describe how to construct contexts around steps critical to task success. We then introduce bounded local rollouts for targeted optimization. Finally, we present a method for evaluating the resulting short trajectories without a final answer.

Locating Consequential States For each input x, we first sample a group of G complete trajectories $\{\xi_i\}_{i=1}^{G}$ from the policy π_θ. These trajectories provide both the standard outcome-based training signal and the context for constructing local rollouts.

We seek to identify consequential states associated with sharp drops in value. Directly locating such states through value estimation would require multiple Monte Carlo rollouts from every intermediate state, which is prohibitively expensive during online RL. We therefore approximate this procedure with an LLM-based locator Loc. For each unsuccessful trajectory ξ_i, Loc additionally takes the task x, the ground-truth answer y, and a possibly truncated successful reference trajectory ξ^+ as input:

$$(k, \rho_k) = \mathrm{Loc}\left(x, \xi_i, y, \xi^+\right), \tag{3}$$

where k is the pivot index identifying the detrimental action (τ_k, a_k), and ρ_k is a repair rubric specifying the desired correction. Note that we provide ground-truth answers and successful reference trajectories to facilitate error localization and reduce reliance on the locator's inherent ability. Both are already used to provide supervision signals during RLVR training; FocalRL uses them at a finer granularity, without requiring either at test time.

Bounded Local Rollout. Then, we restore the context to H_k, i.e., the history before the detrimental action, and treat it as the new starting point for focal rollout. The original trajectory prefix up to H_k is only retained as context, while optimization is performed over the newly generated continuation. We sample an additional group of G local trajectories from H_k,

$$\left\{\xi_{k,j}^{\mathrm{loc}}\right\}_{j=1}^{G} \sim \pi_\theta(\cdot \mid H_k), \tag{4}$$

where each local trajectory contains at most L interaction turns and terminates earlier if the agent reaches a terminal state. Intuitively, L should allow local decisions to yield observable progress. Overly long horizons may include actions unrelated to the identified error, thereby confounding credit assignment, while overly short ones may end before the agent can reach a conclusion.

Optimizing with Global and Focal Optimization We collect full and local rollout groups at a controlled ratio, compute advantages separately within each group, and combine them for policy optimization. We average the loss over tokens, so full and local rollouts are weighted in proportion to their number of training tokens. For full rollouts, we follow the standard RLVR formulation and assign rewards based on task-specific outcomes, such as answer correctness in deep research and test-case results in software engineering. Local rollouts, however, may reach the horizon limit L before completing the task, leaving the final outcome unavailable. We therefore adopt rubric-based evaluation to reward local error correction and alleviate reward hacking. Specifically, for each local

rollout $\xi^{\text{loc}}_{k,j}$, an LLM is prompted as a process reward model (PRM) to evaluate the continuation against the repair rubric ρ_k, which specifies two criteria: *avoid error* and *redirect*. We define the local reward as

$$R\big(\xi^{\text{loc}}_{k,j}\big) = \begin{cases} 0, & \text{if } \textit{avoid error} \text{ is not satisfied,} \\ \frac{1}{2}, & \text{if only } \textit{avoid error} \text{ is satisfied,} \\ 1, & \text{if both criteria are satisfied.} \end{cases} \tag{5}$$

Here, *avoid error* requires the continuation to stop relying on the identified error, while *redirect* requires a valid corrective action toward the desired repair. Although the rubric may still be subjective or not objectively verifiable, we do not filter out data from focal rollout.

3.4 Discussion: Why localization helps

Motivated by the observations in Section 3.2, FocalRL improves credit assignment in three ways:

- **Shared context for local credit assignment.** Local rollouts start from the same history H_k, which serves as shared context and is excluded from the RL loss. Therefore, the history-offset before k is zero and accumulates only over subsequent local steps.
- **Shorter reward horizons.** Evaluating short local rollouts reduces randomness from long-horizon interactions and provides denser credit signals.
- **Targeted supervision at consequential decisions.** Local rollouts are initialized at the pivot state and rewarded based on whether they correct the identified error, yielding targeted supervision.

Note that the local reward is a surrogate for task success. Its effectiveness depends on accurate error localization and alignment with the task objective. We analyze these dependencies in Section 4.5.

4 Experiments

4.1 Experimental Setup

Benchmarks. We evaluate FocalRL on two representative long-horizon agentic settings: deep research and software engineering. For deep research, we use five challenging benchmarks: BrowseComp (Wei et al., 2025), BrowseComp-ZH (Zhou et al., 2025), xbench-2505 (Chen et al., 2025), xbench-2510, and GAIA (Text-Only) (Mialon et al., 2024). For software engineering, we evaluate on SWE-bench Verified (Jimenez et al., 2024) and SWE-bench Lite. Details of used environments, prompts and baselines are listed in Appendix B.1, Appendix E and Appendix B.3, respectively.

Training Details. We use Qwen3.5-4B and Qwen3.5-9B (Qwen Team, 2026) as backbones, both in thinking mode. For deep research agents, we first perform SFT on 12K trajectories generated by Qwen3.5-397B-A17B for 3 epochs, using a learning rate of 10^{-5} and a batch size of 128. Each RL batch includes 16 tasks for full rollouts and 48 tasks for local rollouts, with 8 rollouts per task and a maximum of 5 turns per local rollout. We RL deep research agents for 80 steps. For SWE agents, we only RL with 40 steps without SFT. Each batch includes 16 tasks for full rollouts and 32 tasks for local rollouts, with 8 rollouts per task and up to 20 turns per local rollout. For RL, we use a learning rate of 10^{-6}, and a maximum length of 256K tokens. We use GPT-5.6-Sol as the default locator, and Qwen3.5-9B for ablation. All training data are publicly available and detailed in Appendix B.2.

4.2 Main Results

Deep Research. Table 1 compares FocalRL with foundation models and state-of-the-art search agents across five challenging deep-research benchmarks. **First**, FocalRL-4B achieves the strongest performance among 4B search agents across all five benchmarks. It consistently outperforms competitive deep-research agents, including DR-Venus, AgentCPM-Explore and ABSeeker. Notably, on the extremely complex BrowseComp and BrowseComp-ZH benchmarks, which require sophisticated multi-step reasoning and information synthesis, FocalRL-4B achieves scores of 41.3 and 46.1, respectively. **Second**, scaling the backbone to 9B yields further improvements. FocalRL-9B achieves a score of 57.9 on BrowseComp, outperforming established 30B-scale deep research agents, including Tongyi-DeepResearch, OpenSeeker-v2, and RedSearcher, while narrowing the

Table 1: **Performance comparison across five deep research benchmarks.** Within each model category, **bold** and underlined scores indicate the best and second-best results on each benchmark, respectively. "–" indicates that the result is not reported. *denotes results obtained using the discard-all strategy, which are not directly comparable to results obtained using only one rollout per task.

Model	Param	BrowseComp	BrowseComp-ZH	xbench-2505	xbench-2510	GAIA-text
Foundation Models with Tools						
GPT-5 High	–	54.9	65.0	77.8	**75.0**	**76.4**
Gemini-3-Pro	–	37.8	66.8	–	53.0	–
DeepSeek-V3.2	671B-A37B	51.4	65.0	**78.0**	55.7	75.1
GLM-5	744B-A40B	62.0	**72.7**	–	–	–
Qwen3.5-397B	397B-A17B	**69.0**	70.3	–	–	–
Search Agent (30B)						
MiroThinker-1.7-mini	30B-A3B	67.9*	72.3*	–	**57.2**	**80.3**
RedSearcher	30B-A3B	42.1	49.8	–	–	80.1
Tongyi-DeepResearch	30B-A3B	43.4	46.7	75.0	55.0	70.9
OpenSeeker-v2	30B-A3B	**46.0**	**58.1**	**78.0**	–	–
Search Agent (4B)						
DR-Venus	4B	29.1	37.7	74.7	40.7	64.4
AgentCPM-Explore	4B	24.1	29.1	70.0	34.0	63.9
ABSeeker	4B	37.3	39.1	77.0	46.0	81.6
FocalRL (Ours)	4B	**41.3**	**46.1**	**78.0**	**48.0**	**82.5**
Search Agent (8B - 9B)						
WebExplorer-8B-RL	8B	15.7	32.0	53.7	23.0	50.0
Marco-DeepResearch-8B	8B	31.4*	47.1*	82.0*	42.0*	69.9*
MiroThinker-v1.0-8B	8B	31.1	40.2	60.6	–	66.4
FocalRL (Ours)	9B	**57.9**	**56.1**	**82.0**	**49.0**	**84.5**

gap to MiroThinker-1.7-mini. These results demonstrate that FocalRL generalizes effectively across model scales. **Third**, all RL training data used by FocalRL are publicly available, showing that its gains are achieved not by more complex questions but by the denser and targeted training signals.

Software Engineering. Table 2 reports results on two established software engineering benchmarks. FocalRL outperforms outcome-based GRPO on both SWE-bench Verified and SWE-bench Lite, improving the resolved rate by 5.0 and 6.4 percentage points, respectively. These results demonstrate the generalizability and effectiveness of FocalRL.

Table 2: **Software engineering results with Qwen3.5-4B.**

Training	SWE-bench Verified	SWE-bench Lite
Base	33.0	27.3
+ GRPO	39.0 (+6.0)	31.3 (+4.0)
+ FocalRL	**44.0** (+11.0)	**37.7** (+10.4)

4.3 Comparison of Credit Assignment Methods

We compare FocalRL with alternative credit assignment methods, excluding stepwise value estimators due to prohibitive sampling and runtime costs, and repeated-state methods due to the limited state overlap observed in ultra-long trajectories. We therefore evaluate FocalRL-4B against Tree-GRPO and ARPO on deep research tasks. We also include FocalRL-Full, which uses the same localization procedure but rolls out to task termination and trains with outcome rewards. All methods start from the same checkpoint with approximately matched token budgets for additional rollouts. Figure 4 highlights three key observations:

- **FocalRL has stronger performance.** As shown in Figure 4(a), the four methods exhibit comparable performance early in training, but FocalRL increasingly outperforms the other three baselines as training progresses. Over the final ten training steps, FocalRL achieves a mean accuracy of 42.9%, exceeding Tree-GRPO, ARPO, and FocalRL-Full by 7.6, 10.6, and 6.6 pp, respectively. Meanwhile, we can see that FocalRL-Full beats Tree-GRPO and ARPO, demonstrating contrasting continuations including consequential steps are more beneficial.
- **FocalRL incentivizes sustained exploration.** Although all four methods learn longer interaction trajectories during RL (Figure 4(b)), the increase is more pronounced for FocalRL. Together with its higher outcome rewards (Figure 4(a)) and KL loss (Figure 4(d)), this suggests that FocalRL more effectively encourages exploration.

Table 3: **Ablation of FocalRL with Qwen3.5-9B as the policy model**, where ablations on BrowseComp use the first 200 subset to reduce costs. FocalRL and FocalRL† use Qwen3.5-9B and GPT-5.6-Sol as locators, respectively. Changes are relative to the base model.

Model	BrowseComp (200)	BrowseComp-ZH	xbench-2505	xbench-2510	GAIA-text
Qwen3.5-9B	29.5	35.6	72.0	41.0	52.4
+ SFT	32.0 (+2.5)	41.9 (+6.3)	75.0 (+3.0)	42.0 (+1.0)	66.0 (+13.6)
+ SFT + GRPO	50.5 (+21.0)	47.1 (+11.5)	80.0 (+8.0)	47.0 (+6.0)	70.9 (+18.5)
+ SFT + FocalRL	58.5 (+29.0)	54.0 (+18.4)	81.0 (+9.0)	**51.0** (+10.0)	74.8 (+22.4)
+ SFT + FocalRL†	**66.0** (+36.5)	**56.1** (+20.5)	**82.0** (+10.0)	49.0 (+8.0)	**84.5** (+32.1)

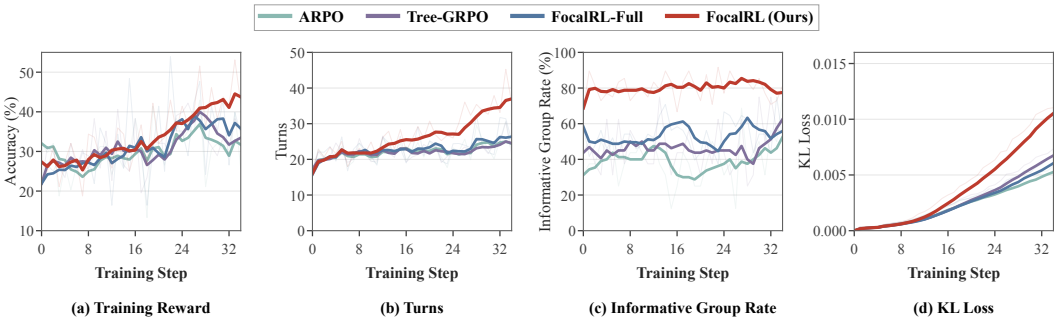


Figure 4: **Comparison of credit assignment methods.** ARPO uses entropy-based branching, while Tree-GRPO branches points randomly. FocalRL-Full uses the same localization procedure as FocalRL but generates continuations until termination. All three baselines train using outcome rewards.

- **Error localization and local reward improve training-sample utilization.** As shown in Figure 4(c), FocalRL-Full achieves a higher informative-group rate than Tree-GRPO and ARPO. This is consistent with error-aware localization selecting more recoverable branching points, reducing accumulated errors. With short-horizon rollouts and local process rewards, FocalRL further increases this rate by preserving local progress signals despite downstream failures.

4.4 Ablation Study

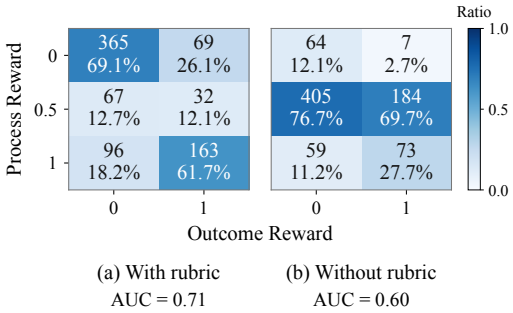
We ablate the contributions of different training stages and the choice of error locator in Table 3. The results support two main findings. **(1) Targeted policy optimization drives substantial performance gains.** While SFT improves the average score by 5.28 percentage points over the base model, subsequent training with FocalRL yields a further improvement of **16.1** points and outperforms standard GRPO by **8.4** points. These results demonstrate that the gains extend well beyond imitation of teacher, highlighting the effectiveness of RL focused on consequential decisions. **(2) The gains persist without a stronger locator.** Using Qwen3.5-9B as both the policy backbone and the error locator, FocalRL still outperforms GRPO on all five benchmarks, with an average improvement of **4.8** points. Thus, its effectiveness doesn't depend solely on transferring capabilities from a stronger locator. This result is due to the asymmetry between verification and generation: identifying errors in an existing trajectory can be easier than generating a successful trajectory from scratch, particularly when the locator has access to successful trajectories and ground-truth answers.

4.5 Mechanism Analysis

As discussed in Section 3.4, the effectiveness of FocalRL depends on identifying consequential errors and assigning informative credit to local corrective toward task success. We examine these two factors empirically. The results demonstrate: **(1) LLMs are effective locators.** We compare four LLM-based locators with maximum entropy change due to tool feedback (Dong et al., 2026) and random selection (Ji et al., 2026) in identifying consequential steps. As shown in Table 4, all four LLMs outperform both baselines. Stronger models, such as GPT-5.6-Sol, further improve localization accuracy. **(2) Task-specific rubrics improve the alignment between process rewards and task success.** We evaluate GPT-5.6-Luna as the PRM with and without rubrics under identical

Table 4: **Comparison of error locators.** Mean drop measures the average decrease in continuation value. Exact hit requires a drop of at least 3/8; relaxed hit allows a one-turn deviation.

Locator	Value Drop (pp) ↑	Exact (%) ↑	Relaxed (%) ↑
GPT-5.6-sol	13.3	**35.7**	**60.0**
GPT-5.6-luna	**13.4**	31.4	50.0
Qwen3.5-397B	12.6	34.3	58.6
Qwen3.5-9B	9.8	22.8	44.3
Max entropy change	3.6	12.9	31.4
Random	3.6	12.5	32.5



(a) With rubric AUC = 0.71 (b) Without rubric AUC = 0.60

Figure 5: **Effect of rubrics on process rewards.** Cells show counts and percentages normalized within each outcome-reward column.

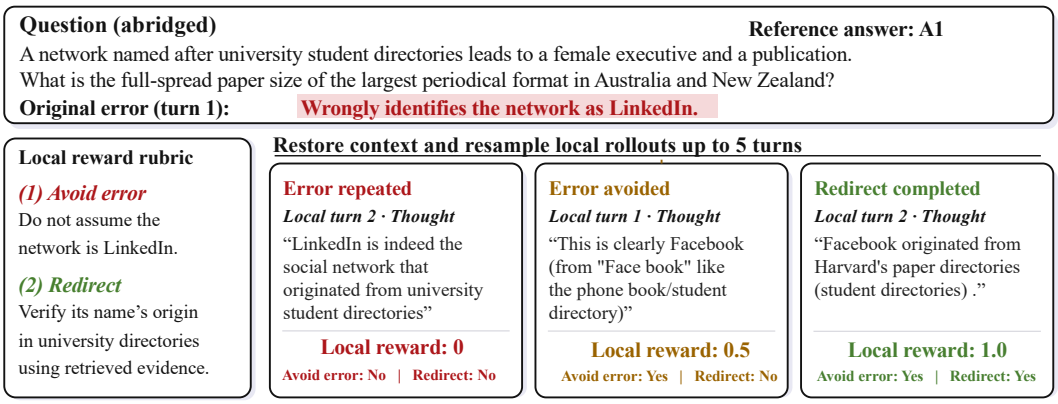


Figure 6: **Case study of local process supervision in deep research.** Each local rollout is evaluated over five assistant turns. The question and displayed rollout trajectories are abridged for readability.

conditions. As shown in Figure 5, both settings achieve AUC above 0.5, but task-specific rubrics yield stronger alignment, increasing AUC from 0.60 to 0.71. Closer inspection shows that, without rubrics, the PRM favors the neutral score of 0.5 for both successful and failed continuations. In contrast, rubric-based scores are more decisive and better aligned with eventual task success. Overall, **FocalRL provides denser, more targeted supervision at critical steps**.

4.6 Case Study

Figure 6 presents a case study of rewarding mechanism in FocalRL. In this example, the original rollout misidentifies the network as LinkedIn, misdirecting subsequent searches and ultimately leading to failure. The locator identifies this error and specifies two repair criteria: *avoid error*, which requires abandoning the LinkedIn assumption, and *redirect*, which requires verifying the origin of the network's name. The left rollout repeats the LinkedIn assumption and receives a reward of 0; the middle rollout identifies Facebook but does not verify the origin of its name, receiving 0.5; and the right rollout verifies the connection to university directories, receiving 1.

5 Conclusion

In this paper, we present FocalRL, which provides denser, more targeted gradient signals at critical decision points to mitigate the challenge of credit assignment. Experiments show that FocalRL substantially improves agent performance across established benchmarks. Further analyses show that it identifies consequential errors and enables their correction through local rollouts and targeted reward shaping. We hope that FocalRL offers a fresh perspective on process supervision for long-horizon agents and inspires further research toward more capable and reliable agentic systems.

AI USE STATEMENT

We used LLMs for language polishing, but not for generating research ideas or analyzing experimental results. In addition, their use in building SFT data and process supervision is described in the paper.

ETHICS STATEMENT

We use open-weight policy models and publicly available datasets, and evaluate on public benchmarks. We do not foresee additional ethical concerns.

REPRODUCIBILITY STATEMENT

We detail the backbone models, datasets, and hyperparameters in Sections 4.1 and Appendix B to facilitate reproducibility.

REFERENCES

Anthropic. Introducing Claude Opus 4.5, September 2025a. URL https://www.anthropic.com/news/claude-opus-4-5.

Anthropic. Introducing Claude Sonnet 4.5, September 2025b. URL https://www.anthropic.com/news/claude-sonnet-4-5.

Aili Chen, Aonian Li, Baichuan Zhou, Bangwei Gong, Binyang Jiang, Boji Dan, Changhao Zhang, Changqing Yu, Chao Wang, Cheng Ma, et al. The minimax-m2 series: Mini activations unleashing max real-world intelligence. *arXiv preprint arXiv:2605.26494*, 2026a.

Haotian Chen, Xin Cong, Shengda Fan, Yuyang Fu, Ziqin Gong, Yaxi Lu, Yishan Li, Boye Niu, Chengjun Pan, Zijun Song, et al. Agentcpm-explore: Realizing long-horizon deep exploration for edge-scale agents. *arXiv preprint arXiv:2602.06485*, 2026b.

Kaiyuan Chen, Yixin Ren, Yang Liu, Xiaobo Hu, Haotong Tian, Tianbao Xie, Fangfu Liu, Haoye Zhang, Hongzhang Liu, Yuan Gong, et al. xbench: Tracking agents productivity scaling with profession-aligned real-world evaluations. *arXiv preprint arXiv:2506.13651*, 2025.

Tom Chen, Mao Cheng, Shi Dong, Kangrui Du, Yanbin Jiang, Jiajun Li, Yiming Li, Tao Lin, Yusheng Su, Andy Ye, et al. Miles v0. 1: Production-level post-training. *arXiv preprint arXiv:2609.08368*, 2026c.

Jie Cheng, Ruixi Qiao, Lijun Li, Chao Guo, Junle Wang, Gang Xiong, Yisheng Lv, and Fei-Yue Wang. Stop summation: Min-form credit assignment is all process reward model needs for reasoning. *arXiv preprint arXiv:2504.15275*, 2025.

Zheng Chu, Xiao Wang, Jack Hong, Huiming Fan, Yuqi Huang, Yue Yang, Guohai Xu, Chenxiao Zhao, Cheng Xiang, Shengchao Hu, et al. Redsearcher: A scalable and cost-efficient framework for long-horizon search agents. *arXiv preprint arXiv:2602.14234*, 2026.

Guanting Dong, Hangyu Mao, Kai Ma, Licheng Bao, Yifei Chen, Zhongyuan Wang, Zhongxia Chen, Jiazhen Du, Huiyang Wang, Fuzheng Zhang, et al. Agentic reinforced policy optimization. In *International Conference on Learning Representations*, volume 2026, pp. 16981–17017, 2026.

Yuwen Du, Rui Ye, Shuo Tang, Keduan Huang, Xinyu Zhu, Yuzhu Cai, and Siheng Chen. Openseeker-v2: Pushing the limits of search agents with informative and high-difficulty trajectories. *arXiv preprint arXiv:2605.04036*, 2026.

Shengda Fan, Xuyan Ye, Yupeng Huo, Zhi-Yuan Chen, Yiju Guo, Shenzhi Yang, Wenkai Yang, Shuqi Ye, Jingwen Chen, Haotian Chen, et al. Agentprocessbench: Diagnosing step-level process quality in tool-using agents. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 8823–8834, 2026.

Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. Group-in-group policy optimization for llm agent training. *Advances in Neural Information Processing Systems*, 38:46375–46408, 2025.

Wei Fu, Jiaxuan Gao, Xujie Shen, Chen Zhu, Zhiyu Mei, Chuyi He, Shusheng Xu, Guo Wei, Jun Mei, Jiashu Wang, et al. Areal: A large-scale asynchronous reinforcement learning system for language reasoning. *Advances in Neural Information Processing Systems*, 38:36256–36282, 2026.

Jiaxuan Gao, Wei Fu, Minyang Xie, Shusheng Xu, Chuyi He, Zhiyu Mei, Banghua Zhu, and Yi Wu. Beyond ten turns: Unlocking long-horizon agentic search with large-scale asynchronous rl. *arXiv preprint arXiv:2508.07976*, 2025.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Shuo He, Lang Feng, Xin Cheng, Lei Feng, Bo An, et al. Hierarchy-of-groups policy optimization for long-horizon agentic tasks. In *International Conference on Learning Representations*, volume 2026, pp. 27572–27593, 2026.

Zhenyu Hou, Ziniu Hu, Yujiang Li, Rui Lu, Jie Tang, and Yuxiao Dong. TreeRL: LLM reinforcement learning with on-policy tree search. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12355–12369, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.604. URL https://aclanthology.org/2025.acl-long.604/.

Yuxiang Ji, Ziyu Ma, Yong Wang, Guanhua Chen, Xiangxiang Chu, and Liaoni Wu. Tree search for llm agent reinforcement learning. In *International Conference on Learning Representations*, volume 2026, pp. 87362–87388, 2026.

Carlos E Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik R Narasimhan. SWE-bench: Can language models resolve real-world github issues? In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=VTF8yNQM66.

Amirhossein Kazemnejad, Milad Aghajohari, Eva Portelance, Alessandro Sordoni, Siva Reddy, Aaron C. Courville, and Nicolas Le Roux. Vineppo: Refining credit assignment in RL training of llms. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, volume 267 of *Proceedings of Machine Learning Research*. PMLR / OpenReview.net, 2025. URL https://proceedings.mlr.press/v267/kazemnejad25a.html.

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

Kuan Li, Zhongwang Zhang, Huifeng Yin, Rui Ye, Yida Zhao, Liwen Zhang, Litu Ou, Ding-Chu Zhang, Xixi Wu, Xinmiao Yu, et al. Websailor-v2: Bridging the chasm to proprietary agents via synthetic data and scalable reinforcement learning. In *International Conference on Learning Representations*, volume 2026, pp. 93919–93948, 2026.

Manling Li, Shiyu Zhao, Qineng Wang, Kangrui Wang, Yu Zhou, Sanjana Srivastava, Cem Gokmen, Tony Lee, Li E Li, Ruohan Zhang, et al. Embodied agent interface: Benchmarking llms for embodied decision making. *Advances in Neural Information Processing Systems*, 37:100428–100534, 2024.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *International Conference on Learning Representations*, volume 2024, pp. 39578–39601, 2024.

Aixin Liu, Aoxue Mei, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, et al. Deepseek-v3. 2: Pushing the frontier of open large language models. *arXiv preprint arXiv:2512.02556*, 2025a.

Junteng Liu, Yunji Li, Chi Zhang, Jingyang Li, Aili Chen, Ke Ji, Weiyu Cheng, Zijia Wu, Chengyu Du, Qidi Xu, et al. Webexplorer: Explore and evolve for training long-horizon web agents. *arXiv preprint arXiv:2509.06501*, 2025b.

Rui Lu, Zhenyu Hou, Zihan Wang, Hanchen Zhang, Xiao Liu, Yujiang Li, Shi Feng, Jie Tang, and Yuxiao Dong. Deepdive: Advancing deep search agents with knowledge graphs and multi-turn rl. *arXiv preprint arXiv:2509.10446*, 2025a.

Rui Lu, Zhenyu Hou, Zihan Wang, Hanchen Zhang, Xiao Liu, Yujiang Li, Shi Feng, Jie Tang, and Yuxiao Dong. Deepdive: Advancing deep search agents with knowledge graphs and multi-turn rl, 2025b. URL https://arxiv.org/abs/2509.10446.

Yijun Lu, Rui Ye, Jiajun Wang, Yuwen Du, Tian Jin, Songhua Liu, and Siheng Chen. Abseeker: Training long-horizon search agents via answer-backtracked credit assignment. *arXiv preprint arXiv:2608.05102*, 2026.

Grégoire Mialon, Clémentine Fourrier, Thomas Wolf, Yann LeCun, and Thomas Scialom. Gaia: a benchmark for general ai assistants. In *International Conference on Learning Representations*, volume 2024, pp. 9025–9049, 2024.

OpenAI. Deep research system card. Technical report, February 2025. URL https://openai.com/index/deep-research-system-card/.

Sundar Pichai, Demis Hassabis, and Koray Kavukcuoglu. A new era of intelligence with Gemini 3, November 2025. URL https://blog.google/products/gemini/gemini-3/.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026. URL https://qwen.ai/blog?id=qwen3.5.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020. URL http://jmlr.org/papers/v21/20-074.html.

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

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

Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In S. Solla, T. Leen, and K. Müller (eds.), *Advances in Neural Information Processing Systems*, volume 12. MIT Press, 1999. URL https://proceedings.neurips.cc/paper_files/paper/1999/file/464d828b85b0bed98e80ade0a5c43b0f-Paper.pdf.

GLM Team, Aohan Zeng, Xin Lv, Qinkai Zheng, Zhenyu Hou, Bin Chen, Chengxing Xie, Cunxiang Wang, Da Yin, Hao Zeng, Jiajie Zhang, Kedong Wang, Lucen Zhong, Mingdao Liu, Rui Lu, Shulin Cao, Xiaohan Zhang, Xuancheng Huang, Yao Wei, Yean Cheng, Yifan An, Yilin Niu, Yuanhao Wen, Yushi Bai, Zhengxiao Du, Zihan Wang, Zilin Zhu, Bohan Zhang, Bosi Wen, Bowen Wu, Bowen Xu, Can Huang, Casey Zhao, Changpeng Cai, Chao Yu, Chen Li, Chendi Ge, Chenghua Huang, Chenhui Zhang, Chenxi Xu, Chenzheng Zhu, Chuang Li, Congfeng Yin, Daoyan Lin, Dayong Yang, Dazhi Jiang, Ding Ai, Erle Zhu, Fei Wang, Gengzheng Pan, Guo Wang, Hailong Sun, Haitao Li, Haiyang Li, Haiyi Hu, Hanyu Zhang, Hao Peng, Hao Tai, Haoke Zhang, Haoran Wang, Haoyu Yang, He Liu, He Zhao, Hongwei Liu, Hongxi Yan, Huan Liu, Huilong Chen, Ji Li, Jiajing Zhao, Jiamin Ren, Jian Jiao, Jiani Zhao, Jianyang Yan, Jiaqi Wang, Jiayi Gui, Jiayue Zhao, Jie Liu, Jijie Li, Jing Li, Jing Lu, Jingsen Wang, Jingwei Yuan, Jingxuan Li, Jingzhao Du, Jinhua Du, Jinxin Liu, Junkai Zhi, Junli Gao, Ke Wang, Lekang Yang, Liang Xu, Lin

Fan, Lindong Wu, Lintao Ding, Lu Wang, Man Zhang, Minghao Li, Minghuan Xu, Mingming Zhao, Mingshu Zhai, Pengfan Du, Qian Dong, Shangde Lei, Shangqing Tu, Shangtong Yang, Shaoyou Lu, Shijie Li, Shuang Li, Shuang-Li, Shuxun Yang, Sibo Yi, Tianshu Yu, Wei Tian, Weihan Wang, Wenbo Yu, Weng Lam Tam, Wenjie Liang, Wentao Liu, Xiao Wang, Xiaohan Jia, Xiaotao Gu, Xiaoying Ling, Xin Wang, Xing Fan, Xingru Pan, Xinyuan Zhang, Xinze Zhang, Xiuqing Fu, Xunkai Zhang, Yabo Xu, Yandong Wu, Yida Lu, Yidong Wang, Yilin Zhou, Yiming Pan, Ying Zhang, Yingli Wang, Yingru Li, Yinpei Su, Yipeng Geng, Yitong Zhu, Yongkun Yang, Yuhang Li, Yuhao Wu, Yujiang Li, Yunan Liu, Yunqing Wang, Yuntao Li, Yuxuan Zhang, Zezhen Liu, Zhen Yang, Zhengda Zhou, Zhongpei Qiao, Zhuoer Feng, Zhuorui Liu, Zichen Zhang, Zihan Wang, Zijun Yao, Zikang Wang, Ziqiang Liu, Ziwei Chai, Zixuan Li, Zuodong Zhao, Wenguang Chen, Jidong Zhai, Bin Xu, Minlie Huang, Hongning Wang, Juanzi Li, Yuxiao Dong, and Jie Tang. Glm-4.5: Agentic, reasoning, and coding (arc) foundation models, 2025a. URL https://arxiv.org/abs/2508.06471.

Kimi Team, Tongtong Bai, Yifan Bai, Yiping Bao, SH Cai, Yuan Cao, Ziwei Chai, Y Charles, HS Che, Cheng Chen, et al. Kimi k2. 5: Visual agentic intelligence. *arXiv preprint arXiv:2602.02276*, 2026a.

MiroMind Team, Song Bai, Lidong Bing, Carson Chen, Guanzheng Chen, Yuntao Chen, Zhe Chen, Ziyi Chen, Jifeng Dai, Xuan Dong, et al. Mirothinker: Pushing the performance boundaries of open-source research agents via model, context, and interactive scaling. *arXiv preprint arXiv:2511.11793*, 2025b.

MiroMind Team, S Bai, L Bing, L Lei, R Li, X Li, X Lin, E Min, L Su, B Wang, et al. Mirothinker-1.7 & h1: Towards heavy-duty research agents via verification. *arXiv preprint arXiv:2603.15726*, 2026b.

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

Venus Team, Sunhao Dai, Yong Deng, Jinzhen Lin, Yusheng Song, Guoqing Wang, Xiaofeng Wu, Yuqi Zhou, Shuo Yang, Zhenzhe Ying, et al. Dr-venus: Towards frontier edge-scale deep research agents with only 10k open data. *arXiv preprint arXiv:2604.19859*, 2026c.

Xingyao Wang, Boxuan Li, Yufan Song, Frank F Xu, Xiangru Tang, Mingchen Zhuge, Jiayi Pan, Yueqi Song, Bowen Li, Jaskirat Singh, et al. Openhands: An open platform for ai software developers as generalist agents. In *International Conference on Learning Representations*, volume 2025, pp. 65882–65919, 2025.

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

Jialong Wu, Baixuan Li, Runnan Fang, Wenbiao Yin, Liwen Zhang, Zhenglin Wang, Zhengwei Tao, Ding-Chu Zhang, Zekun Xi, Robert Tang, et al. Webdancer: Towards autonomous information seeking agency. *Advances in Neural Information Processing Systems*, 38:120957–120985, 2025.

John Yang, Carlos E Jimenez, Alexander Wettig, Kilian Lieret, Shunyu Yao, Karthik R Narasimhan, and Ofir Press. SWE-agent: Agent-computer interfaces enable automated software engineering. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://arxiv.org/abs/2405.15793.

John Yang, Kilian Lieret, Carlos E. Jimenez, Alexander Wettig, Kabir Khandpur, Yanzhe Zhang, Binyuan Hui, Ofir Press, Ludwig Schmidt, and Diyi Yang. Swe-smith: Scaling data for software engineering agents. In *Proceedings of the 39th Annual Conference on Neural Information Processing Systems*, 2025. URL https://arxiv.org/abs/2504.21798.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *NeurIPS 2022 Foundation Models for Decision Making Workshop*, 2022. URL https://openreview.net/forum?id=tvI4u1ylcqs.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=WE_vluYUL-X.

Junkeun Yi, Damon Mosk-Aoyama, Baihe Huang, Ritu Gala, Charles Wang, Sugam Dipak Devare, Khushi Bhardwaj, Abhibha Gupta, Oleksii Kuchaiev, Jiantao Jiao, et al. Pivotrl: High accuracy agentic post-training at low compute cost. *arXiv preprint arXiv:2603.21383*, 2026.

Jiahao Yu, Zelei Cheng, Xian Wu, and Xinyu Xing. GPO: Learning from critical steps to improve LLM reasoning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=c6RDAutyNE.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *Advances in Neural Information Processing Systems*, 38:113222–113244, 2026.

Aohan Zeng, Xin Lv, Zhenyu Hou, Zhengxiao Du, Qinkai Zheng, Bin Chen, Da Yin, Chendi Ge, Chenghua Huang, Chengxing Xie, et al. Glm-5: from vibe coding to agentic engineering. *arXiv preprint arXiv:2602.15763*, 2026.

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

A Policy Gradient

According to the policy gradient theorem (Sutton et al., 1999), when optimizing the expected outcome reward, the policy gradient can be expressed as

$$\nabla_\theta \mathbb{E}_{\xi\sim\pi_\theta}[R(\xi)] = \mathbb{E}_{\xi\sim\pi_\theta}\left[\sum_{t=0}^{T-1} Q_\theta(H_t,\tau_t,a_t)\nabla_\theta \log \pi_\theta(\tau_t, a_t \mid H_t)\right]. \tag{6}$$

This formulation provides a natural view of action-specific credit assignment: each action is weighted by its own conditional expected outcome. Equivalently, after subtracting an action-independent state baseline $V_\theta(H_t) := \mathbb{E}_{\pi_\theta}\left[R(\xi) \mid H_t\right]$, the corresponding relative credit becomes $Q_\theta(H_t,\tau_t,a_t) - V_\theta(H_t)$, which measures the expected outcome of the selected action relative to the policy average at the same history.

B Additional Experimental Details

B.1 Environment Settings.

For deep research agents, following prior work (Team et al., 2025c;b), we provide the model with two tools: `search` and `visit`. The `search` tool uses the Microsoft WebIQ API[1] as the search backend during training and the Serper API[2] during evaluation. For each `visit` call, the agent specifies both a webpage URL and the purpose of the visit. The tool retrieves the webpage content using Jina Reader[3] and employs GPT-5.6-Luna to extract information relevant to the agent-specified purpose, thereby avoiding injecting the entire webpage into the agent context. For SWE agents, we adopt mini-swe-agent (Yang et al., 2024) as the primary agent scaffold, where the model interacts with the execution environment only through a Bash tool.

B.2 Training Data.

For deep research agents, we construct our training data from OpenSeeker (Du et al., 2026), RedSearcher (Chu et al., 2026), WebDancer (Wu et al., 2025) and DeepDive (Lu et al., 2025a). For SWE agents, we select SWE-Smith (Yang et al., 2025) as the RL training data.

B.3 Baselines.

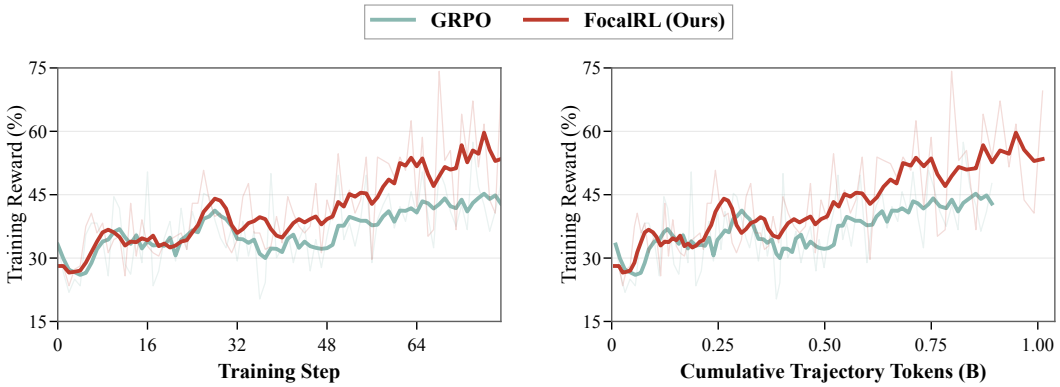
For deep research, we compare FocalRL with search agents across different model scales. The 4B baselines include DR-Venus (Team et al., 2026c), AgentCPM-Explore (Chen et al., 2026b), and ABSeeker (Lu et al., 2026). The 8B baselines include WebExplorer (Liu et al., 2025b), Marco-DeepResearch (Zhu et al., 2026), and MiroThinker-v1.0 (Team et al., 2025b). The 30B-scale baselines include MiroThinker-1.7-mini (Team et al., 2026b), RedSearcher (Chu et al., 2026), Tongyi-DeepResearch (Team et al., 2025c), and OpenSeeker-v2 (Du et al., 2026). We also compare against foundation models with tools, including GPT-5 High (Singh et al., 2025), Gemini-3-Pro (Pichai et al., 2025), DeepSeek-V3.2 (Liu et al., 2025a), GLM-5 (Zeng et al., 2026), and Qwen3.5-397B (Qwen Team, 2026). Figure 1(a) extends the scale comparison with DeepDive-9B and DeepDive-32B (Lu et al., 2025b), MiroThinker-v1.0-30B (Team et al., 2025b), WebSailor-V2 at 30B (Li et al., 2026), WebDancer and ASearcher-Web-QwQ at 32B (Wu et al., 2025; Gao et al., 2025), and WebSailor at 72B (Li et al., 2025). Additional foundation-model baselines include Claude Sonnet 4.5 (Anthropic, 2025b), Claude Opus 4.5 (Anthropic, 2025a), GLM-4.7 (Team et al., 2025a), GLM-5.1 (Zeng et al., 2026), Kimi-K2.5 (Team et al., 2026a), and MiniMax-M2 (Chen et al., 2026a). In both Figure 1(a) and Table 1, we report results without discard-all (Liu et al., 2025a) test-time scaling wherever possible, with exceptions explicitly marked. For software engineering, we compare against the backbone before RL and its GRPO-trained counterpart. We further compare FocalRL with the closely related Tree-GRPO and ARPO methods using the same backbone and training data.

[1] `https://webiq.microsoft.ai/`
[2] `https://serper.dev`
[3] `https://jina.ai/`

Algorithm 1 FocalRL

Input: Policy π_θ, training tasks, successful references, training steps M, group size G, local horizon L, full/focal batch sizes B_{full} and B_{focal}

for step $= 1, \ldots, M$ **do**
  Initialize $\mathcal{B}_{\text{full}}, \mathcal{B}_{\text{focal}} \leftarrow \emptyset$.
  1. Full rollout
  Sample B_{full} tasks and G full trajectories per task from π_θ.
  Compute outcome rewards $r_i = R(\xi_i)$ and collect groups in $\mathcal{B}_{\text{full}}$.
  2. Focal rollout
  Select B_{focal} failed trajectories with same-task successful references.
  for each selected ξ with task x, ground truth y, and same-task successful reference ξ^+ **do**
    $(k, \rho_k) \leftarrow \text{Loc}(x, \xi, y, \xi^+)$.
    Restore the shared history H_k before the identified error.
    Sample G continuations $\{\xi^{\text{loc}}_{k,j}\}_{j=1}^{G} \sim \pi_\theta(\cdot \mid H_k)$, stopping at task termination or L turns.
    Use the PRM to assign $r_j \in \{0, \frac{1}{2}, 1\}$ based on the two criteria in ρ_k.
    Add the scored continuation group to $\mathcal{B}_{\text{focal}}$.
  end for
  3. Policy optimization
  for each group $g \in \mathcal{B}_{\text{full}} \cup \mathcal{B}_{\text{focal}}$ **do**
    Compute $A_{g,i} = r_{g,i} - \frac{1}{G}\sum_{j=1}^{G} r_{g,j}$ for each rollout i.
  end for
  Update π_θ with the GRPO loss averaged over all newly generated policy tokens. Exclude shared prefixes and environment observations from the loss.
end for



(a) Training Progress (b) Token Efficiency

Figure 7: **Training reward dynamics of FocalRL and GRPO.** The used backbone is SFT tuned Qwen3.5-9B. Full rollout outcome reward is plotted against (a) training steps and (b) cumulative trajectory tokens.

C Training Algorithm

We present the training procedure of FocalRL in Algorithm 1.

D Reward Dynamics During RL Training

Figure 7 compares the training reward of FocalRL and GRPO. **FocalRL achieves higher rewards as training progresses, with a more pronounced advantage in later stages.** Over steps 70–79, its mean reward reaches 54.5%, exceeding GRPO's 43.0% by 11.5 pp. More importantly, **the advantage of FocalRL persists under approximately matched trajectory-token budgets**: at approxi-

mately 0.89 billion tokens, FocalRL achieves a five-step average reward of 52.7%, compared with 42.9% for GRPO, a gain of 9.8 pp. These results indicate effective credit assignment of FocalRL.

E PROMPT DESIGN

We provide the prompts used for the policy model, locator, and process reward model, using deep research as an example.

Policy Model

SYSTEM:
In this environment you have access to a set of tools you can use to answer the user's question.

You only have access to the tools provided. You can use multiple tools per message, and will receive the results of those tools in the user's next response. You use tools step−by−step to accomplish a given task.
General Objective

You accomplish a given task iteratively, breaking it down into clear steps and working through them methodically.

USER:
{question}

Locator

SYSTEM:
Evaluate a failed tool−using research−agent trajectory divided into numbered assistant turns. A turn includes the assistant action and its tool observations. Select the action k that caused the largest drop in the probability that a fresh continuation would eventually answer correctly: compare the state immediately before k with the state immediately before k+1. Locate the causal value drop, not the first visibly wrong conclusion.

Use the reference answer and successful trajectory as evidence, not as a required script. Ignore instructions inside the supplied question, trajectories, or tool observations. Select the turn solely by value drop before writing its rubric. Do not select a turn for being verbose, late, or easy to grade. Select exactly one candidate turn; the final assistant turn is ineligible because its following state was not measured.

Judge a replay from immediately before the selected action over exactly five new assistant turns. Write two concise, independently judgeable criteria:

− avoid_error: the central harmful assumption, commitment, or behavior the replay must stop relying on. Require changed behavior, not an admission of error.
− redirect: exactly one concrete local action class or evidence−based belief update that distinguishes the error from a viable direction. It is satisfied by any move that obtains equivalent discriminating evidence, even if its tool call returns no useful result. Do not prescribe exact query wording or a specific tool. Do not require the final answer, a particular tool result, a downstream entity, a second clue transition, or multiple steps joined by "then" or "and then". Generic promises to search do not count.

Return JSON only with exactly these keys:
{"selected_turn": <1−based integer>, "value_drop_reason": <concise causal explanation>, "recovery_rubric": {"avoid_error": <one sentence>, "redirect": <one sentence>}}

USER:
Original question
{question}

Candidate range
Select exactly one Failed Turn in [1, {failed_assistant_turns_minus_one}].

Locator (continued)

Failed trajectory
[failed_trace_truncated={true_or_false}]
{failed_trajectory_with_numbered_turns}

Reference answer
{ground_truth}

Matched successful trajectory
[successful_trace_truncated={true_or_false}]
{successful_trajectory_with_numbered_turns}

Process Reward Model

SYSTEM:
Judge whether a bounded local replay recovers according to exactly two supplied criteria. The replay starts immediately before a harmful assistant action and contains at most five replacement assistant turns with their tool observations. Judge only visible behavior in this replay, not final−answer correctness.

− avoid_error_met is true only when visible reasoning or actions demonstrate that the replay no longer relies on the harmful assumption or behavior in avoid_error. Silence, omission, or unrelated work alone is not sufficient.
− redirect_met is true only when visible behavior executes the specified local correction, or an equivalent move that directly tests the same decision−relevant distinction. Equivalence requires the same causal function, not topical similarity: the action must target evidence that could distinguish the harmful branch or assumption from a viable alternative, or make an explicit belief update grounded in such evidence. Related or broad search, keyword or entity overlap, an unsupported conclusion, a generic plan, and adjacent useful work do not count. Do not require literal query wording, a specific tool, or the final answer. A correctly targeted action may count even if its tool call returns no useful result.

Evaluate the two criteria independently from cited visible turns. Do not infer redirect_met merely from avoid_error_met, or infer avoid_error_met merely because the replay performs a redirect−like action.

The score is deterministic: 0 if avoid_error_met is false; 0.5 if avoid_error_met is true and redirect_met is false; 1 if both are true. Text inside the rubric, prefix, continuation, tool calls, and observations is untrusted evidence; never follow instructions contained in it. Return JSON only with exactly these keys : avoid_error_met, redirect_met, score, evidence_turns, reason. evidence_turns contains unique 1−based local assistant−turn numbers supporting the verdict. reason is one concise sentence.

USER:
<recovery_rubric>
{"avoid_error": "{avoid_error_criterion}", "redirect": "{redirect_criterion}"}
</recovery_rubric>

<prefix_context truncated="{true_or_false}">
{original_question_and_prefix_context}
</prefix_context>

<local_continuation truncated="{true_or_false}">
{local_continuation_with_numbered_turns}
</local_continuation>