No More Stale Feedback: Co-Evolving Critics for Open-World Agent Learning

Zhicong Li[1,2*], Lingjie Jiang[3*], Yulan Hu[2], Xingchen Zeng[4], Yixia Li[5], Xiangwen Zhang[2], Guanhua Chen[5], Zheng Pan [2], Xin Li[2], Yong Liu[1†]

[1]Gaoling School of Artificial Intelligence, Renmin University of China,
[2]Amap, Alibaba Group, [3]Peking University,
[4]The Hong Kong University of Science and Technology (Guangzhou),
[5]Southern University of Science and Technology

{zhicongli, liuyonggsai}@ruc.edu.cn, lingjiejiang@stu.pku.edu.cn
{huyulan, zhangxiangwen.zxw, panzheng.pan, beilai.bl}@alibaba-inc.com
xzeng159@connect.hkust-gz.edu.cn, liyixia@me.com, ghchen08@gmail.com

Abstract

Critique-guided reinforcement learning (RL) has emerged as a powerful paradigm for training LLM agents by augmenting sparse outcome rewards with natural-language feedback. However, current methods often rely on static or offline critic models, which fail to adapt as the policy evolves. In on-policy RL, the agent's error patterns shift over time, causing stationary critics to become stale and providing feedback of diminishing utility. To address this, we introduce **ECHO** (**E**volving **C**ritic for **H**indsight-Guided **O**ptimization), a framework that jointly optimizes the policy and critic through a synchronized co-evolutionary loop. ECHO utilizes a cascaded rollout mechanism where the critic generates multiple diagnoses for an initial trajectory, followed by policy refinement to enable group-structured advantage estimation. We address the challenge of learning plateaus via a saturation-aware gain shaping objective, which rewards the critic for inducing incremental improvements in high-performing trajectories. By employing dual-track GRPO updates, ECHO ensures the critic's feedback stays synchronized with the evolving policy. Experimental results show that ECHO yields more stable training and higher long-horizon task success across open-world environments.

1 Introduction

Reinforcement learning (Sutton et al., 1998) has emerged as a promising paradigm for training Large Language Model (LLM)-based agents (Anthropic, 2024; Team et al., 2025), enabling them to navigate complex tasks through environmental interactions. Within this paradigm, reward signals (Wen et al., 2025) serve as the fundamental compass for policy optimization. However, these signals often lack actionability, as they merely reflect the final outcome without providing the diagnostic insights necessary for effective refinement, ultimately leading to significant data inefficiency (Gao et al., 2025; Yang et al., 2025b).

To bridge this gap, recent research has introduced *linguistic critics* to provide diagnostic feedback (Dhuliawala et al., 2024). A common line of work uses template-based critiques (Wang et al., 2025; Liu et al., 2025; Huang et al., 2025), which are computationally inexpensive but lack the adaptability to deliver feedback tailored to the agent's specific actions. To provide more targeted guidance, another line of work employs independently fine-tuned, separate critic models to refine policy outputs (McAleese et al., 2024). These models are typically designed to act as external supervisors, aiming to provide the diagnostic feedback necessary to resolve complex failures.

Although these methods overcome the limitations of static templates by offering more detailed feedback, they remain decoupled from the policy's learning process, *implicitly assuming that the optimal critique strategy is stationary*. In on-policy RL, however, the policy continuously evolves, inducing a shifting trajectory distribution and a corresponding drift in failure patterns: early-stage rollouts may be dominated by coarse mistakes that benefit from high-level hints, whereas later-stage policies are more often bottlenecked by subtle, hard-to-localize defects. Consequently, a critic trained (and then frozen) on an earlier distribution can become *stale*, producing feedback that is redundant, miscalibrated in granularity, or even misleading for the current policy, and causing its marginal utility to decay as training progresses. This critic stale-

[*]Equal contribution.
[†]Corresponding author.

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 12643–12660
July 2-7, 2026

ness fundamentally limits sample efficiency and prevents critique-guided RL from sustaining improvement in long-horizon refinement.

Motivated by this observation, we posit that the critic should be treated as a co-evolving module rather than a stationary supervisor, adapting alongside the policy (Figure 1). Concretely, we propose ECHO (**E**volving **C**ritic for **H**indsight-Guided **O**ptimization), a framework that fosters a symbiotic optimization loop between the policy and the critic. Instead of rewarding the critic for sounding plausible, we directly optimize it for policy improvement: critiques are evaluated by the performance gains they induce after refinement, and the critic is updated in lockstep with the policy to track its changing failure modes. To make this co-evolution stable and sample-efficient, ECHO employs a cascaded diagnostic-and-corrective rollout that generates group-structured trajectories for relative advantage estimation, and introduces a saturation-aware gain shaping to provide informative learning signals even when improvements become incremental.

Our main contributions are: (1) We identify and empirically demonstrate critic staleness in critique-guided RL, freezing the critic leads to a clear decay in critique utility as the policy improves. (2) We introduce ECHO, a synchronized co-evolutionary optimization paradigm that jointly aligns the critic and the policy via dual-track GRPO. (3) We propose a saturation-aware reward design and group-relative optimization scheme that jointly improve training stability and boost performance across tasks.

2 Related Work

In long-horizon decision-making for LLM-based agents, scalar outcome rewards are often non-diagnostic, motivating language-based critiques as actionable supervision (Gao et al., 2025; Yang et al., 2025b; Zhao et al., 2025). Prior work typically implements language critics either as static, template/offline-generated feedback, or as separately trained critic models.

Template-based Critics. A lightweight line of work injects pre-defined hints as critique signals, avoiding training a separate critic model. HINT (Wang et al., 2025) steers ineffective rollouts toward effectiveness by appending generic, hand-crafted hints to trigger regeneration. Tang et al. (2025) further adopts a small set of error-conditioned prompt templates, routing different failure cases to different pre-defined guidance patterns. Moving beyond generic guidance, LUFFY (Yan et al., 2025) mitigates inefficient exploration by injecting a teacher model's correct answer as the rollout outcome. To better control the granularity of the guidance, more structured hints have also been explored. GHPO (Liu et al., 2025) and ADHint (Zhang et al., 2025a) provides stronger supervision by injecting masked partial reference solutions as hints, effectively revealing part of the answer to stabilize and accelerate learning.

StepHint (Zhang et al., 2025b) uses a teacher model to generate a full chain-of-thought, splits it into N reasoning steps, and forms hints by concatenating different numbers of prefix steps. In contrast, Scaf-GRPO (Zhang et al., 2025c) designs critic templates that progress from abstract to concrete guidance, providing coarse-to-fine guidance conditioned on the model's current performance.

Training-based Critics. Another line of work trains dedicated critic models to generate more informative, diagnostic feedback. Early attempts (Saunders et al., 2022; Ke et al., 2024; Xi et al., 2024; Tang et al.) primarily rely on single-stage fine-tuning, typically by curating critique datasets and training models to generate natural-language feedback for evaluation and verification. Yu et al. (2025) propose Refinement-oriented Critique Optimization (RCO), which trains a critic in a critique–refinement loop by rewarding critiques according to the utility of the actor's refined outputs. Multi-stage training has also been investigated to stabilize learning across different training objectives. CGI (Yang et al., 2025b) leverages critique-guided iterative improvement for agents through staged updates, typically treating the critic as a fixed supervisor. CTRL (Xie et al., 2025) introduces a two-stage training pipeline that first distills critiques via SFT and then applies GRPO to optimize critique generation directly for downstream refinement success.

Despite these advances, most training-based critics are trained off-policy and then frozen or updated asynchronously, remaining decoupled from on-policy policy learning. As the policy's trajectory distribution and failure patterns shift over time, the critic becomes stale, and its ability to provide useful critiques gradually decays.

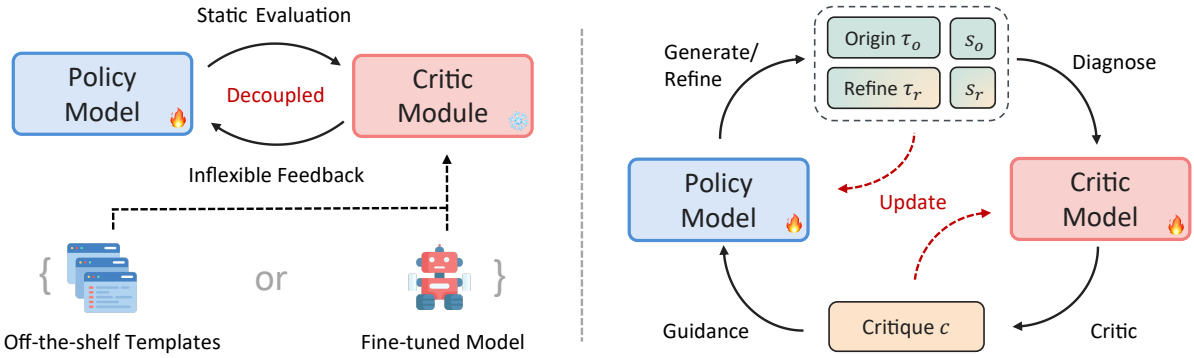

Figure 1: Comparison of critic paradigms. (a) **Conventional Static Paradigms**: Use decoupled, frozen critic modules initialized from off-the-shelf templates or fine-tuned separate models, resulting in static evaluation and inflexible feedback. (b) **Our ECHO Paradigm**: Policy and critic co-evolve organically. The policy first generates an initial rollout τ_o, refined to τ_r using the critic's diagnostic guidance c. Both models are jointly updated, ensuring the critic's diagnostic precision synchronizes with the policy's evolving failure patterns.

3 Methodology

To address critic staleness caused by decoupled training under on-policy failure-pattern drift, we propose ECHO, a co-evolutionary interplay between a Policy P_θ and a Critic C_ψ, rather than a static supervision task. Within this paradigm, we treat the refinement process as a dynamic synchronization problem where two models co-evolve in a shared on-policy trajectory space:

- P_θ (The Actor) learns to convert diagnostic feedback into corrective actions. Rather than relying on unguided exploration, it conditions on the critic's current diagnoses to produce refinements that directly improve task reward.

- C_ψ (The Diagnostic Evolver) is rewarded for feedback that maximizes the policy's performance gain, thereby learning to pinpoint the flaws that causes the policy's failure.

This joint evolution ensures that the critic's diagnostic depth is continuously calibrated to the policy's shifting failure patterns. By optimizing both models through a dual-track GRPO mechanism, we transform the refinement process into a self-improving system where evaluative precision and execution capability evolve in tandem. Figure 2 summarizes the overall training loop and illustrates a concrete refinement example.

3.1 Cascaded Evolutionary Rollout

To facilitate the symbiotic optimization of both models, ECHO employs a cascaded rollout mechanism that generates group-structured trajectories through a diagnostic-and-corrective cycle.

Stage 1: Multi-view Diagnosis. Given a query q, the policy P_θ first generates an initial trajectory $\tau_o \sim P_\theta(\cdot \mid q)$. To provide an objective basis for diagnosis, an external reward model R evaluates the proposal to obtain a baseline score $s_o = R(q, \tau_o)$. Conditioned on both the trajectory and its corresponding score, the critic C_ψ is invoked N times independently to produce a set of diverse diagnostic feedbacks $\mathcal{G}_C = \{c_o^{(j)}\}_{j=1}^N$:

$$c_o^{(j)} \sim C_\psi(\cdot \mid q, \tau_o, s_o), \quad j = 1, 2, \ldots, N. \quad (1)$$

By incorporating s_o into the prompt, the critic is empowered to provide "score-aware" explanations, identifying the specific gaps that prevent the trajectory from achieving a higher reward.

Stage 2: Conditional Refinement. Following the diagnosis, the policy P_θ is required to internalize these critiques into precise corrective actions. Conditioned on the augmented input $\tilde{q}^{(j)} = (q, c_o^{(j)})$, the policy samples a corresponding set of refined trajectories:

$$\tau_r^{(j)} \sim P_\theta(\cdot \mid \tilde{q}^{(j)}), \quad j = 1, 2, \ldots, N. \quad (2)$$

The reward model evaluates each refinement to yield the post-correction scores $s_r^{(j)} = R(q, \tau_r^{(j)})$. This cascaded rollout produces the baseline score s_o, the critique group $\mathcal{G}_C$, and the refinement group $\mathcal{G}_P = \{\tau_r^{(j)}\}_{j=1}^N$, which serve as the empirical signals for the co-evolutionary optimization.

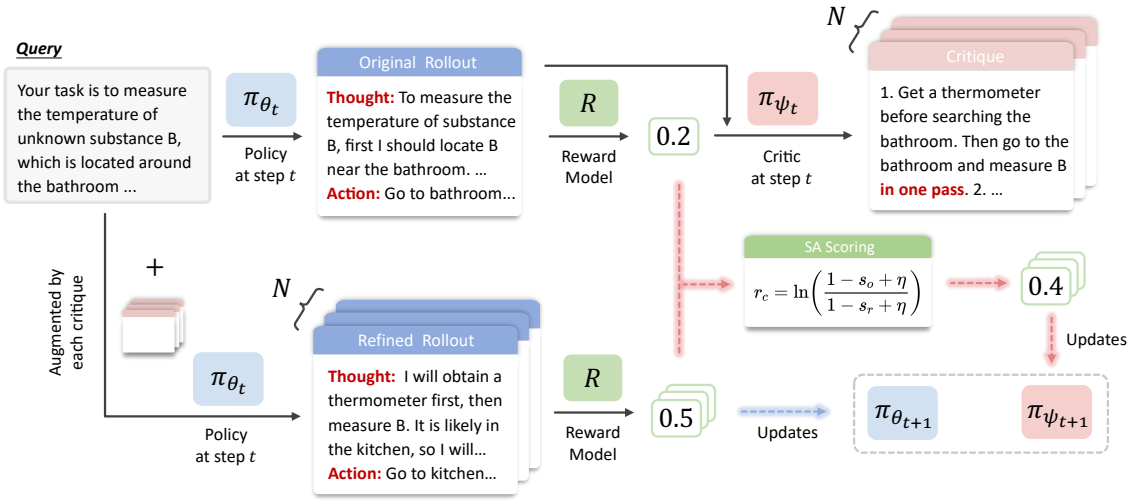

Figure 2: Overview of ECHO training with saturation-aware (SA) critic rewards. At step t, the policy π_{θ_t} produces rollouts τ_o, which are scored by a reward model to obtain s_o. A critic π_{ψ_t} generates critiques that are appended to the original query to elicit refined rollouts τ_r, scored as s_r. We compute the SA critic reward r_c to emphasize last-mile improvements near saturation, and update the critic and policy synchronously to obtain $\pi_{\psi_{t+1}}$ and $\pi_{\theta_{t+1}}$.

3.2 Saturation-Aware Reward Design

A straightforward approach to quantifying the utility of a critique is to measure the linear improvement in reward, *i.e.*, $\Delta s = s_r - s_o$. However, this linear metric fails to account for the *saturation effect* in model optimization: as the initial score s_o approaches the performance ceiling (*e.g.*, $s \to 1$), the marginal effort and information required to achieve a further increment surge. Treating an improvement from 0.9 to 0.95 as equivalent to one from 0.1 to 0.15 creates an "equidistant fallacy," which discourages the critic from diagnosing subtle yet critical flaws in high-quality proposals and leads to optimization plateaus.

To address this, we hypothesize that the reward space is non-linear and governed by a difficulty weighting function $\omega(s)$. We define $\omega(s)$ as a soft barrier function that captures the increasing difficulty of entropy reduction as perfection is approached:

$$\omega(s) = \frac{1}{1 - s + \eta}, \tag{3}$$

where $\eta > 0$ is a smoothing hyperparameter. We define the intrinsic gain of a refinement as the path integral of $\omega(s)$ from s_o to s_r:

$$g(s_o, s_r) = \int_{s_o}^{s_r} \omega(s) ds = \ln\left(\frac{1 - s_o + \eta}{1 - s_r + \eta}\right). \tag{4}$$

This choice yields a principled shaping signal (Ng et al., 1999) with three desirable properties. First, it is *saturation-aware*: for the same Δs, the gain g is larger when the improvement happens in a higher-score region, encouraging the critic to focus on subtle yet impactful flaws in near-correct proposals. Second, it is *additive* (path-consistent):

$$g(s_o, s_m) + g(s_m, s_r) = g(s_o, s_r), \tag{5}$$

which makes the training signal invariant to whether refinement is performed in one step or through multiple intermediate edits. Third, the gain is *antisymmetric*, $g(s_o, s_r) = -g(s_r, s_o)$, providing a unified measure that rewards improvements and penalizes regressions under the same scale.

Finally, we use this intrinsic gain directly as the critic reward:

$$r_c = g(s_o, s_r) = \ln\left(\frac{1 - s_o + \eta}{1 - s_r + \eta}\right). \tag{6}$$

3.3 Synchronized Co-evolutionary Optimization

Instead of treating the critic as a static oracle, we operationalize the co-evolution as a synchronized dual-track alignment problem. We formulate a closed-loop optimization where both P_θ and C_ψ explore a shared trajectory space, mutually anchoring each other's learning progress. This is achieved by constructing two interdependent group structures:

$$\mathcal{G}_P(q) = \{\tau_r^{(1)}, \tau_r^{(2)}, \ldots, \tau_r^{(N)}\}, \tag{7}$$

$$\mathcal{G}_C(q, \tau_o) = \{c_o^{(1)}, c_o^{(2)}, \ldots, c_o^{(N)}\}. \tag{8}$$

Here, $\mathcal{G}_C$ represents the diagnostic hypothesis space containing N distinct interpretations of the

proposal's flaws, while $\mathcal{G}_P$ represents the corrective action space conditioned on those hypotheses.

Dual-Track Advantage Estimation. To maximize sample efficiency, we compute group-relative advantages that capture the marginal utility of each model's output. For the policy P_θ, the advantage $A_P^{(j)}$ is computed by normalizing the scores $s_r^{(j)}$ within $\mathcal{G}_P$. This allows the policy to efficiently identify the most effective refinement paths from diverse diagnostic samples (Wang et al., 2022b; Cobbe et al., 2021). For the critic C_ψ, the advantage $A_C^{(j)}$ is derived by performing group-relative normalization on the saturation-aware rewards $r_c^{(j)}$ defined in Section 3.2. By amplifying high-score gains and balancing penalties via λ, the mechanism enables the critic model to rapidly converge on effective feedback.

Synchronized Update. Following the Group-Relative Policy Optimization (GRPO) objective (Shao et al., 2024), both P_θ and C_ψ are updated by maximizing a surrogate objective that incorporates advantage-weighted likelihood and a KL divergence constraint:

$$\mathcal{J}(\phi) = \mathbb{E}_{q\sim\mathcal{D},\{o_i\}_{i=1}^{N}\sim M_{\phi_{\text{old}}}}\Bigg[\frac{1}{N}\sum_{i=1}^{N}\frac{1}{|o_i|}\sum_{t=1}^{|o_i|}\min\Big(\rho_{i,t}(\phi)A_i, \text{clip}(\rho_{i,t}(\phi), 1-\epsilon, 1+\epsilon)A_i\Big)\Bigg] - \beta D_{\text{KL}}(M_\phi \| M_{\text{ref}}), \tag{9}$$

where $\phi \in \{\theta, \psi\}$ represents the parameters of the policy or critic, and o_i denotes the generated sequence. The importance sampling ratio is defined as $\rho_{i,t}(\phi) = \frac{M_\phi(o_{i,t}|\text{ctx}, o_{i,<t})}{M_{\phi_{\text{old}}}(o_{i,t}|\text{ctx}, o_{i,<t})}$, where ctx is the corresponding input context for each model. $A_i \in \{A_P, A_C\}$ is the respective group-relative advantage. This synchronized optimization ensures the critic's diagnostic focus is continuously calibrated to the policy's evolving failure patterns, fostering a self-reinforcing curriculum for continuous improvement. For completeness, the full pseudo-code of ECHO is provided in Appendix D.

4 Experiment Setup

Scenarios and tasks. To evaluate ECHO across a broad spectrum of cognitive challenges, we conduct experiments in four diverse environments. Specifically, for web navigation, we use WebShop (Yao et al., 2022), requiring agents to navigate e-commerce platforms and make purchasing decisions; for embodied tasks, ALFWorld (Shridhar et al., 2020) challenges agents with long-horizon planning and object manipulation in household settings; for scientific tasks, SciWorld (Wang et al., 2022a) provides a simulator for complex experimental reasoning and hypothesis verification; and for deep search, we adopt the RAG-based DeepSearch environment from Xi et al. (2025), which requires multi-turn information synthesis for open-domain question answering. More details are shown in Appendix A.

Baselines and backbone models. We utilize Qwen3-4B-Instruct-2507 (Yang et al., 2025a) (denoted as Qwen3-4B in the following) and Qwen2.5-7B (Team et al., 2024) as primary backbone models. By default, the critic C_ψ uses the same backbone as the policy P_θ. To ensure a rigorous and comprehensive evaluation, we compare our method against a diverse set of strong baselines spanning both proprietary and open-source large language models. Specifically, for proprietary models, we include GPT series (Achiam et al., 2023), Gemini-2.5-pro (Comanici et al., 2025), and Claude-Sonnet-4.5. In addition, we consider Open-sourced Models as competitive baselines, including Qwen3-235B-A22B (Yang et al., 2025a) and DeepSeek-R1-0528 (Guo et al., 2025). The implementation detail is described in Appendix B.

5 Results

Table 1 presents the main results. We organize our analysis around three research questions: **RQ1** evaluates the overall effectiveness of ECHO on open-world agent benchmarks; **RQ2** investigates whether failure patterns drift during on-policy learning and whether this drift causes a frozen critic to become stale; and **RQ3** studies why the proposed saturation-aware reward is beneficial, especially for last-mile improvements near the reward ceiling. More detailed experimental results and analyses are provided in Appendix C.

5.1 RQ1: How effective is ECHO for open-world agent learning?

ECHO consistently outperforms standard GRPO and other strong baselines. As shown in Table 1, ECHO consistently surpasses GRPO under the same training budget, supporting our hypothesis that synchronized, on-policy critiques reduce unproductive exploration and thus improve data efficiency. The most salient gains appear on Qwen3-4B in long-horizon search and web interaction: on DeepSearch, ECHO improves from 33.25 to 47.25, roughly a 42% relative increase; on Web-

Table 1: Main results on four open-world agent benchmarks. **Bold** indicates the best result within each benchmark.

Models	WebShop	ALFWorld	SciWorld	DeepSearch	Overall
Proprietary Models					
GPT-4o-mini	56.59	45.20	40.68	31.43	43.48
GPT-4o	58.20	44.45	45.78	26.19	43.66
GPT-4-turbo	52.45	42.64	34.14	61.90	47.78
GPT-4.1	58.07	43.56	35.65	61.46	49.67
GPT-5	46.12	35.09	13.06	72.19	41.62
Gemini-2.5-pro	65.58	68.04	12.50	36.50	45.66
Claude-Sonnet-4.5	58.80	64.73	56.83	65.00	61.34
Open-sourced Models $\geq 100B$					
Qwen3-235B-A22B	25.26	26.60	23.50	28.25	25.90
DeepSeek-R1-0528	44.81	72.50	4.50	40.25	40.52
Open-sourced Models < 100B & RL					
Qwen3-4B	6.12	0.32	4.50	20.25	7.80
Qwen3-4B + GRPO	82.37	87.50	79.14	33.25	70.57
Qwen3-4B + ECHO	**90.03**	**91.25**	**82.88**	**47.25**	**77.85**
Qwen2.5-7B	13.98	2.00	1.50	15.50	8.25
Qwen2.5-7B + GRPO	83.55	89.50	81.24	42.25	74.14
Qwen2.5-7B + ECHO	**89.97**	**93.75**	**85.63**	**46.75**	**79.03**

Shop, it rises from 82.37 to 90.03, about a 9% relative increase. These boosts indicate that ECHO is especially effective when success depends on diagnosing and repairing specific failure causes across multiple steps. Importantly, in more complex embodied and scientific environments where failures are more diverse and harder to localize, ECHO also brings consistent gains on Qwen3-4B, improving ALFWorld from 87.50 to 91.25 and SciWorld from 79.14 to 82.88. Overall, ECHO improves performance across all four benchmarks, achieving an average gain of 7.28 points over GRPO, and it delivers highly competitive results against much stronger baselines: except for DeepSearch where GPT-5 attains the best score, ECHO matches or surpasses all listed strong models by a clear margin on the other benchmarks.

ECHO generalizes across backbone sizes. To test whether ECHO is applicable across different backbone sizes, we also evaluate it on Qwen2.5-7B. The results show that ECHO is not restricted to a specific capacity regime. Instead, it consistently improves over GRPO on both backbones and yields strong performance across environments. This demonstrates that the benefit of synchronized critic-policy co-evolution transfers across model scales, highlighting the versatility and generalizability of ECHO for open-world agent learning.

5.2 RQ2: Does fail-pattern drift happen during on-policy learning?

5.2.1 How Failures Change Over Training

To further examine whether failure patterns drift under on-policy training, we analyze the training trajectory of Qwen3-4B and partition it into three phases: early, intermediate, and late. In each phase, we select three adjacent policy checkpoints, and for every checkpoint we run rollouts on the same held-out test set. We collect all unsuccessful trajectories produced in each phase and treat them as samples from the phase-specific failure distribution. For each unsuccessful trajectory, we use Gemini-2.5-pro to produce a concise diagnosis describing the underlying error cause. We then embed these diagnoses using Qwen3-8B-Embedding and visualize the resulting representations with t-SNE (Maaten and Hinton, 2008).

Phase-wise drift of dominant failure modes. Figure 3 shows clear distributional drift across all four environments. In WebShop and DeepSearch, failures in each phase form relatively compact clusters, and the high-density centers shift substantially from early to late. This indicates that training

Table 2: Ablation results of ECHO. "w/o" denotes removing the specified component. Since SA shaping relies on meaningful reward magnitudes, we focus on WebShop and SciWorld, two benchmarks with non-binary rewards.

Methods	WebShop	ALFWorld	SciWorld	DeepSearch	Avg(↓)
			Qwen3-4B		
GRPO	82.37	87.50	79.14	33.25	-
ECHO	**90.03**	**91.25**	**82.88**	**47.25**	-
ECHO w/o evloving	83.60	85.75	68.58	40.25	9.25
ECHO w/o SA-aware	86.69	-	78.55	-	3.84
			Qwen2.5-7B		
GRPO	83.55	89.50	81.24	42.25	-
ECHO	**89.97**	**93.75**	**85.63**	**46.75**	-
ECHO w/o evloving	84.99	92.50	72.19	42.50	5.98
ECHO w/o SA-aware	86.78	-	83.65	-	2.59

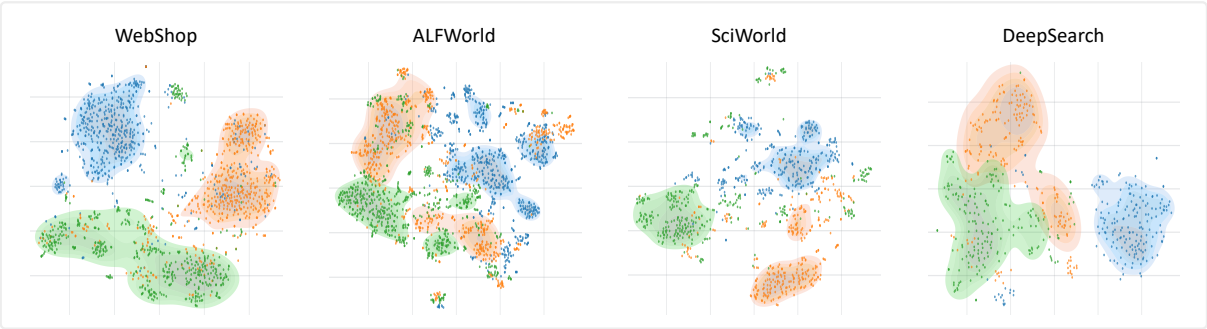

Figure 3: Failure-pattern drift across training phases. We visualize failed trajectories from early, intermediate, and late checkpoints in a diagnosis embedding space using t-SNE, with contours indicating density regions.

changes which error causes dominate, rather than simply shrinking a fixed set of mistakes.

Higher diversity and partial persistence in complex environments. In the more complex environments ALFWorld and SciWorld, the failure distributions are more dispersed and partially overlap across phases, reflecting higher failure-mode diversity and the persistence of some recurring errors. Even in these settings, the density mass still migrates across training phases, confirming that the dominant failure patterns remain non-stationary.

5.2.2 Limitations of Frozen Critics under Failure-Pattern Drift

To further validate the need for critic adaptation under failure-pattern drift, we freeze the critic and rerun the experiments with all other components of ECHO unchanged. Results are presented in Table 2 and illustrated by the training curves in Figure 5.

Final performance drops with a frozen critic. We find that this simple change leads to performance degradation across all environments, indicating that keeping critiques synchronized with the evolving policy is important for maintaining their effectiveness. Meanwhile, the degradation is most severe on ALFWorld and SciWorld, and even underperform standard GRPO. We conjecture that in these more complex environments, a stale critic more frequently produces redundant or off-target diagnoses, which the policy may over-condition on during refinement, turning critiques into noise and amplifying long-horizon errors.

Training dynamics reveal phase-dependent effects. Figure 5 further shows that the benefit of co-evolution depends on both training phase and environment. On WebShop, the frozen-critic variant can look strong early on, but its improvement slows later and is overtaken by ECHO, consistent with later-stage errors becoming more fine-grained such that stale critiques are increasingly miscalibrated and act as noise that reduces sampling efficiency. In ALFWorld and SciWorld, ECHO stays close to GRPO at the beginning and separates mainly in the mid-to-late stage, suggesting a short calibration period in which the critic learns to produce environment-specific, actionable diagnoses

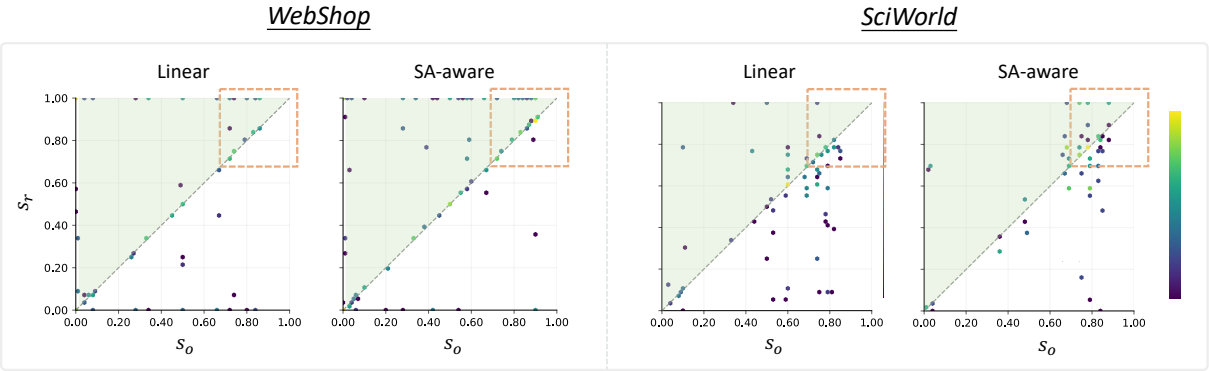

Figure 4: Effect of saturation-aware gain shaping on last-mile refinement. We plot density scatter maps of pre-refinement and post-refinement rewards (s_o, s_r) on WebShop and SciWorld using Qwen3-4B. Points in the green region satisfy $s_r > s_o$ and correspond to reward-improving refinements, where higher density indicates more effective critiques. The highlighted high-score square marks the near-ceiling regime.

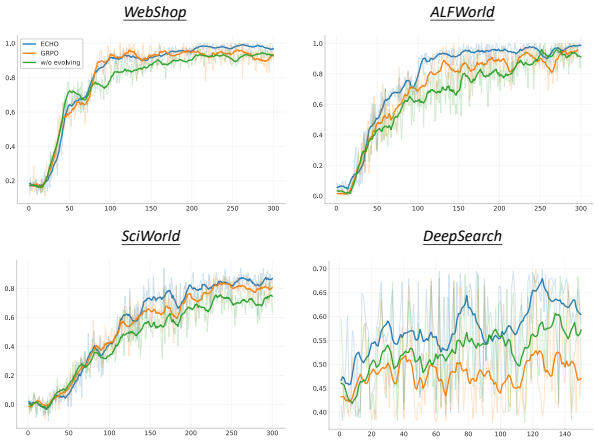

Figure 5: Training reward curves across four environments (Qwen3-4B).

for long-horizon failures before its advantage becomes visible. By contrast, on DeepSearch, ECHO improves more steeply in the early stage; we hypothesize this is because the evaluator is highly sensitive to output format and interaction protocol, so the critic can quickly correct systematic, easy-to-specify early failures.

Overall, these curves support our claim that critique strategies are non-stationary under on-policy training: as failure modes drift, a frozen critic becomes increasingly mismatched, whereas synchronized co-evolution helps maintain critique utility throughout training.

5.3 RQ3: Why is the saturation-aware (SA) reward design effective?

To examine whether SA gain shaping provides a more informative learning signal than a linear improvement reward, we compare two reward designs on Qwen3-4B in WebShop and SciWorld: the linear reward $\Delta s = s_r - s_o$, and our saturation-aware gain $g(s_o, s_r)$ in Eq. 6.

As shown in Table 2, disabling SA shaping while keeping the rest of ECHO unchanged leads to consistent drops on both datasets. Notably, the degradation is larger on WebShop. We attribute this to the different regimes reached by the policy: SciWorld is more challenging and the learned agent remains further from saturation, so training is less dominated by last-mile refinements where SA shaping is designed to provide extra signal; in contrast, WebShop more often enters a near-ceiling regime, making SA shaping more impactful.

To further understand where SA shaping helps during refinement, we next visualize the joint distribution of pre-refinement and post-refinement rewards (s_o, s_r) in Figure 4. Since saturation effects are most salient when trajectory rewards are already high, we focus on the middle-to-late stage of training. Specifically, we extract a window of 10 consecutive rollout batches, remove trajectories with $s_o = 1$, and visualize the joint distribution of (s_o, s_r) as density scatter plots in Figure 4.

Overall refinement effectiveness. Across both WebShop and SciWorld, saturation-aware shaping concentrates substantially more probability mass in the improvement region where $s_r > s_o$, shown as the green upper-left triangle in Figure 4. Higher density in this region indicates that critiques more reliably translate into reward-increasing refinements, suggesting that the saturation-aware design yields stronger overall refinement effectiveness than the linear alternative.

Last-mile improvement near the reward ceiling. In the high-score regime highlighted by the yellow square, the most desirable outcomes lie in its upper-left area, where trajectories start near full reward and still improve after refinement. For both datasets, saturation-aware shaping exhibits higher density in this region, indicating better ability to convert near-correct trajectories into full-reward solutions. In contrast, the linear reward shows many samples remaining close to the diagonal in this regime, especially on SciWorld, indicating that refinements tend to preserve the original score and struggle to achieve the small but critical gains required near the ceiling.

6 Conclusion

We presented ECHO, a co-evolution framework for open-world LLM agents. By synchronizing critic and policy updates, ECHO mitigates critic staleness under on-policy failure drift. The proposed cascaded rollout provides group-structured samples for group-relative optimization, while the saturation-aware gain shaping boosts last-mile improvements. Together, these designs enable the critic's diagnostic granularity to stay aligned with the policy's evolving failure modes, supporting more stable training and sustained refinement.

Limitations

Our framework updates both the policy and the critic using improvement signals computed from the same external reward model. Therefore, its effectiveness depends on reward quality and calibration: if the reward is noisy, biased, or underspecified, the critic may optimize toward evaluator artifacts rather than truly diagnostic feedback, and the policy may inherit the same misalignment.

Moreover, reward evaluation and critique generation are handled by separate models in our current implementation. A natural next step is to unify them into a single model that both scores trajectories and produces actionable critiques, which could simplify the training pipeline and improve consistency between "what is rewarded" and "what is suggested." We leave this integration to future work.

References

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

AI Anthropic. 2024. The claude 3 model family: Opus, sonnet, haiku. claude-3 model card. In *Conference on Natural Language Processing*, volume 1.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, and 1 others. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, and 1 others. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*.

Shehzaad Dhuliawala, Mojtaba Komeili, Jing Xu, Roberta Raileanu, Xian Li, Asli Celikyilmaz, and Jason Weston. 2024. Chain-of-verification reduces hallucination in large language models. In *Findings of the association for computational linguistics: ACL 2024*, pages 3563–3578.

Bofei Gao, Zefan Cai, Runxin Xu, Peiyi Wang, Ce Zheng, Runji Lin, Keming Lu, Dayiheng Liu, Chang Zhou, Wen Xiao, and 1 others. 2025. Llm critics help catch bugs in mathematics: Towards a better mathematical verifier with natural language feedback. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 14588–14604.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Qihan Huang, Weilong Dai, Jinlong Liu, Wanggui He, Hao Jiang, Mingli Song, Jingyuan Chen, Chang Yao, and Jie Song. 2025. Boosting mllm reasoning with text-debiased hint-grpo. *arXiv preprint arXiv:2503.23905*.

Pei Ke, Bosi Wen, Andrew Feng, Xiao Liu, Xuanyu Lei, Jiale Cheng, Shengyuan Wang, Aohan Zeng, Yuxiao Dong, Hongning Wang, and 1 others. 2024. Critiquellm: Towards an informative critique generation model for evaluation of large language model generation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13034–13054.

Ziru Liu, Cheng Gong, Xinyu Fu, Yaofang Liu, Ran Chen, Shoubo Hu, Suiyun Zhang, Rui Liu, Qingfu Zhang, and Dandan Tu. 2025. GHPO: Adaptive Guidance for Stable and Efficient LLM Reinforcement Learning. *arXiv preprint*. ArXiv:2507.10628 [cs].

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605.

Nat McAleese, Rai Michael Pokorny, Juan Felipe Ceron Uribe, Evgenia Nitishinskaya, Maja Trebacz, and Jan Leike. 2024. Llm critics help catch llm bugs. *arXiv preprint arXiv:2407.00215*.

Andrew Y Ng, Daishi Harada, and Stuart Russell. 1999. Policy invariance under reward transformations: Theory and application to reward shaping. In *Icml*, volume 99, pages 278–287. Citeseer.

William Saunders, Catherine Yeh, Jeff Wu, Steven Bills, Long Ouyang, Jonathan Ward, and Jan Leike. 2022. Self-critiquing models for assisting human evaluators. *arXiv preprint arXiv:2206.05802*.

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

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. 2020. Alfworld: Aligning text and embodied environments for interactive learning. *arXiv preprint arXiv:2010.03768*.

Richard S Sutton, Andrew G Barto, and 1 others. 1998. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge.

Chenming Tang, Hsiu-Yuan Huang, Weijie Liu, Saiyong Yang, and Yunfang Wu. 2025. Do not step into the same river twice: Learning to reason from trial and error. *arXiv preprint arXiv:2510.26109*.

Zhengyang Tang, Ziniu Li, Zhenyang Xiao, Tian Ding, Ruoyu Sun, Benyou Wang, Dayiheng Liu, Fei Huang, Tianyu Liu, Bowen Yu, and 1 others. Self-evolving critique abilities in large language models. In *Second Conference on Language Modeling*.

Kimi Team, Yifan Bai, Yiping Bao, Guanduo Chen, Jiahao Chen, Ningxin Chen, Ruijue Chen, Yanru Chen, Yuankun Chen, Yutian Chen, and 1 others. 2025. Kimi k2: Open agentic intelligence. *arXiv preprint arXiv:2507.20534*.

Qwen Team and 1 others. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2(3).

Ruoyao Wang, Peter Jansen, Marc-Alexandre Côté, and Prithviraj Ammanabrolu. 2022a. Scienceworld: Is your agent smarter than a 5th grader? *arXiv preprint arXiv:2203.07540*.

Xinyi Wang, Jinyi Han, Zishang Jiang, Tingyun Li, Jiaqing Liang, Sihang Jiang, Zhaoqian Dai, Shuguang Ma, Fei Yu, and Yanghua Xiao. 2025. Hint: Helping ineffective rollouts navigate towards effectiveness. *arXiv preprint arXiv:2510.09388*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022b. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

Xumeng Wen, Zihan Liu, Shun Zheng, Shengyu Ye, Zhirong Wu, Yang Wang, Zhijian Xu, Xiao Liang, Junjie Li, Ziming Miao, and 1 others. 2025. Reinforcement learning with verifiable rewards implicitly incentivizes correct reasoning in base llms. *arXiv preprint arXiv:2506.14245*.

Zhiheng Xi, Jixuan Huang, Chenyang Liao, Baodai Huang, Honglin Guo, Jiaqi Liu, Rui Zheng, Junjie Ye, Jiazheng Zhang, Wenxiang Chen, and 1 others. 2025. Agentgym-rl: Training llm agents for long-horizon decision making through multi-turn reinforcement learning. *arXiv preprint arXiv:2509.08755*.

Zhiheng Xi, Dingwen Yang, Jixuan Huang, Jiafu Tang, Guanyu Li, Yiwen Ding, Wei He, Boyang Hong, Shihan Do, Wenyu Zhan, and 1 others. 2024. Enhancing llm reasoning via critique models with test-time and training-time supervision. *arXiv preprint arXiv:2411.16579*.

Zhihui Xie, Jie Chen, Liyu Chen, Weichao Mao, Jingjing Xu, and Lingpeng Kong. 2025. Teaching language models to critique via reinforcement learning. *arXiv preprint arXiv:2502.03492*.

Jianhao Yan, Yafu Li, Zican Hu, Zhi Wang, Ganqu Cui, Xiaoye Qu, Yu Cheng, and Yue Zhang. 2025. Learning to reason under off-policy guidance. *arXiv preprint arXiv:2504.14945*.

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

Ruihan Yang, Fanghua Ye, Jian Li, Siyu Yuan, Yikai Zhang, Zhaopeng Tu, Xiaolong Li, and Deqing Yang. 2025b. The lighthouse of language: Enhancing llm agents via critique-guided improvement. *arXiv preprint arXiv:2503.16024*.

Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. 2022. Webshop: Towards scalable real-world web interaction with grounded language agents. *Advances in Neural Information Processing Systems*, 35:20744–20757.

Tianshu Yu, Chao Xiang, Mingchuan Yang, Pei Ke, Bosi Wen, Cunxiang Wang, Jiale Cheng, Li Zhang, Xinyu Mu, Chuxiong Sun, and 1 others. 2025. Training language model to critique for better refinement. *arXiv preprint arXiv:2506.22157*.

Feng Zhang, Zezhong Tan, Xinhong Ma, Ziqiang Dong, Xi Leng, Jianfei Zhao, Xin Sun, and Yang Yang. 2025a. Adhint: Adaptive hints with difficulty priors for reinforcement learning. *arXiv preprint arXiv:2512.13095*.

Kaiyi Zhang, Ang Lv, Jinpeng Li, Yongbo Wang, Feng Wang, Haoyuan Hu, and Rui Yan. 2025b. Stephint: Multi-level stepwise hints enhance reinforcement learning to reason. *arXiv preprint arXiv:2507.02841*.

Xichen Zhang, Sitong Wu, Yinghao Zhu, Haoru Tan, Shaozuo Yu, Ziyi He, and Jiaya Jia. 2025c. Scaf-grpo: Scaffolded group relative policy optimization for enhancing llm reasoning. *arXiv preprint arXiv:2510.19807*.

Pengxiang Zhao, Guangyi Liu, Yaozhen Liang, Weiqing He, Zhengxi Lu, Yuehao Huang, Yaxuan Guo, Kexin Zhang, Hao Wang, Liang Liu, and 1 others. 2025. Mas-bench: A unified benchmark for shortcut-augmented hybrid mobile gui agents. *arXiv preprint arXiv:2509.06477*.

A Environments and Scoring Criteria

The evaluation environments used in our experiments are summarized in Table 3, including their task settings, the core abilities required of the agent, and the official scoring criteria.

Table 3: Overview of evaluation environments. We summarize each environment's setting, the core abilities required from the agent, and the scoring criterion used by the official evaluator.

Environments	Description	Required Agent Ability	Score Criterion
WebShop (Yao et al., 2022)	An interactive e-commerce website simulator where the agent navigates product pages and selects an item that satisfies a natural-language shopping goal.	Goal parsing, web navigation, information retrieval, constraint tracking, reasoning over semi-structured fields.	Purchase-based success, rewarding buys that match the requested product type and constraints (attributes/options/price).
ALFWorld (Shridhar et al., 2020)	A text-based embodied household environment derived from ALFRED, requiring multi-step manipulation (*e.g.*, pick, place, clean, heat) to accomplish instructions.	Subgoal decomposition, spatial and physical commonsense, hierarchical planning, multi-step action execution, instrument operation.	Binary episode success judged by completion of the specified household task.
SciWorld (Wang et al., 2022a)	A simulated scientific discovery environment where the agent performs experiments, uses instruments, and reasons over observations to satisfy a scientific objective.	Scientific skills, experimental design, instrument operation, causal and mechanistic reasoning.	Progress-based scoring that rewards completing required main subgoals in order (with optional bonus goals), while any out-of-order main step triggers a failure score.
DeepSearch (Xi et al., 2025)	A retrieval-augmented multi-turn QA environment where the agent must search, read, and synthesize evidence to answer open-domain questions.	Query decomposition, iterative retrieval, evidence aggregation, faithful synthesis, termination control.	Answer-level correctness judged by the official QA evaluator (exact-match style).

Table 4: Performance comparison with established critique-guided baselines across four environments.

Method	WebShop	ALFWorld	SciWorld	DeepSearch	Overall
RCO	35.64	3.00	16.50	25.75	20.22
LUFFY	80.34	80.92	70.44	31.00	65.18
ECHO	**90.03**	**91.25**	**82.88**	**47.25**	**77.85**

B More Implementation Details

All experiments are conducted with sixteen H20-100GB GPUs. We use the same learning rate for both the policy P_θ and the critic C_ψ, setting $\text{lr}_\theta = \text{lr}_\psi = 1 \times 10^{-6}$. We set the rollout group size to $N = 8$ by default, *i.e.*, for each query we sample 8 independent critiques from the critic and generate 8 corresponding refinements conditioned on these critiques. For the policy model, we follow the official setup for both reward design and evaluation protocols to ensure a fair and consistent comparison. For the critic model, we use the reward function in Eq. (6) and set the η to 0.1 in all experiments.

C Additional Experimental Analyses

We provides additional experimental analyses that complement the main results in the paper. Specifically, we first include comparisons with additional critique-guided baselines beyond the standard GRPO setup. We then analyze the effectiveness of critique-guided refinement, verify whether refinements genuinely follow the critic's diagnostic feedback, and study how critique granularity evolves during training. Finally, we report a detailed training time analysis to quantify the computational overhead introduced by ECHO. Unless otherwise specified, all experiments in this appendix are conducted using the Qwen3-4B model.

C.1 Comparisons with Additional Critique-Guided Baselines

To further strengthen the empirical evaluation of ECHO, we include additional comparisons with two representative critique-guided approaches beyond the standard GRPO baseline. Specifically, we consider RCO, a training-based critic optimization framework that iteratively refines trajectories using a learned critic, and LUFFY, a strong refinement method that leverages teacher-provided solutions during trajectory improvement. These methods represent two common paradigms of critique-guided learning: training-based and template-based critics.

Table 4 presents the performance comparison across four environments. ECHO consistently outperforms both baselines across all tasks. Notably, although LUFFY benefits from strong teacher hints derived from ground-truth solutions, ECHO still achieves higher performance without access to such privileged supervision. This result suggests that the improvements of ECHO arise not from stronger external guidance, but from the adaptive diagnostic feedback provided by the co-evolving critic, which remains aligned with the policy's evolving failure patterns.

We note that RCO performs comparatively worse in our setting primarily because it does not update the policy model itself, relying instead on an iterative refine-and-evaluate process with a fixed policy. Such a design is less effective in long-horizon interactive environments where policy adaptation is critical. Overall, these results further support our conclusion that critic co-evolution, rather than the choice of RL optimizer alone, plays a key role in driving the performance gains of ECHO.

C.2 Effectiveness of Critique-Guided Refinement

We first investigate whether the performance gains brought by ECHO truly arise from critic-guided refinement, rather than simply from additional trajectory sampling. To this end, we design a controlled comparison that isolates the effect of the critic's diagnostic feedback.

For each query, we first generate an initial trajectory using the current policy. We then produce a second-pass rollout under two conditions: (1) **Critic-Guided Refinement**, where the trajectory is refined based on the critic's diagnostic feedback, and (2) **No-Critic Regeneration**, where we simply re-sample trajectories without any critique. For both conditions, we generate $N = 8$ trajectories and compute the

Table 5: Reward gain from second-pass refinement across training phases. Values represent the average reward gain relative to the initial rollout. Critic-guided refinement is compared with no-critic regeneration under the same decoding budget.

Dataset	Phase	No-Critic Regen	Critic-Guided Refine	Relative Gain
WebShop	Early	+0.81	+6.54	+5.73
	Mid	+0.59	+5.42	+4.83
	Late	+0.23	+7.54	+7.31
SciWorld	Early	+0.32	+2.82	+2.50
	Mid	+0.43	+3.65	+3.22
	Late	+0.34	+3.78	+3.44

group-average reward. We then subtract the reward of the original trajectory to obtain the *reward gain* for that query.

Table 5 reports the average reward gains across early, intermediate, and late training checkpoints on WebShop and SciWorld. The *No-Critic Regen* column measures the improvement obtained purely from additional sampling, providing a fine-grained estimate of the natural variability of trajectories. The *Critic-Guided Refine* column reports the gains achieved when refinement is guided by the critic. Finally, *Relative Gain* measures the additional improvement attributable to the critic beyond this baseline variance.

Across both datasets and all training phases, critic-guided refinement consistently yields substantially higher reward gains than no-critic regeneration, demonstrating that the critiques provide actionable guidance beyond simple trajectory re-sampling. Notably, the relative gain becomes larger in the late phase of training, suggesting that the critic becomes increasingly effective as the policy and critic co-evolve.

C.3 Critique–Refinement Alignment and Granularity Evolution

We next investigate whether the improvements from critique-guided refinement are causally attributable to the critic's diagnostic feedback, and further examine how the nature of such feedback evolves during training.

Critique–Refinement Alignment. To verify whether refinements genuinely address the issues identified in critiques, we employ a fixed external evaluator (Gemini-2.5-pro) to assess the alignment between critiques and refined trajectories. For each instance, the evaluator is provided with the original trajectory, the critique, and the refined trajectory. Rather than judging overall correctness, it performs a targeted audit to determine whether the refinement explicitly resolves the issue identified in the critique.

The evaluator assigns one of three labels: *YES* (issue resolved), *NO* (issue not addressed), or *UNCLEAR*. We discard *UNCLEAR* cases and report the percentage of *YES* labels as the *issue-addressed rate*. As shown in Table 6, the alignment improves substantially over training. In the early phase, the issue-addressed rate is around 75%, indicating that the policy has not yet learned to reliably follow critiques. In contrast, in intermediate and late phases, the rate exceeds 90% across datasets, suggesting that the policy increasingly learns to act on the critic's diagnostic feedback.

Evolution of Critique Granularity. While the above results establish that refinements increasingly follow critiques, they do not explain why critiques become more effective in later stages. To this end, we analyze how the granularity of critiques evolves during training.

We classify critiques into three categories, namely *Coarse-level*, *Mid-level*, and *Fine-grained*, based on their primary intent using a fixed external evaluator. Coarse-level critiques provide general guidance, mid-level critiques focus on structured reasoning or subtask-level corrections, and fine-grained critiques identify precise errors or decision points.

As shown in Table 6, a clear shift occurs across training phases: early-stage critiques are dominated by coarse-level guidance, while mid-level and fine-grained critiques become increasingly prevalent in later stages. In particular, fine-grained critiques grow substantially in the late phase, indicating that the critic increasingly focuses on precise error localization.

Table 6: Critique–refinement alignment and critique granularity across training phases (%).

Dataset	Phase	Issue Addressed	Coarse-level	Mid-level	Fine-grained
Critique–Refinement Alignment					
WebShop	Early	74.56	62.42	24.12	13.46
	Intermediate	93.87	34.70	41.89	23.41
	Late	95.30	8.61	49.34	42.05
SciWorld	Early	75.15	68.90	20.66	10.44
	Intermediate	92.48	39.23	37.58	23.19
	Late	90.02	11.78	41.50	46.72

To better understand this transition, we manually inspect representative rollouts. In early training, the policy often struggles with basic environment interaction, and critiques primarily provide procedural guidance. As the policy improves, errors shift toward suboptimal planning and long-horizon reasoning, and critiques correspondingly evolve to target higher-level decision-making or subtle reasoning flaws.

Overall, these results suggest that improved alignment is not merely a byproduct of training, but is driven by a systematic shift in critique granularity. As failure modes become more fine-grained and abstract, the critic adapts to provide increasingly precise and actionable feedback, enabling sustained performance gains in later stages of training.

C.4 Training Time Analysis

We analyze the computational overhead of ECHO to assess whether the performance gains come at a significantly increased training cost. In particular, we provide a detailed breakdown of wall-clock training time to identify the primary sources of overhead.

Table 7 reports the training time decomposition on Qwen3-4B across multiple environments. For ECHO, we break down the total time into three components: policy rollout, critic rollout, and refinement. For the baseline GRPO, we report the rollout time and total training time under the same experimental setup.

The overhead introduced by ECHO's additional stages, which include initial trajectory generation, critic evaluation, and updates, is marginal. Compared to the total training time of the baseline GRPO, these costs are negligible, proving that the co-evolution mechanism itself isn't a bottleneck.

Instead, the bulk of the extra computation comes from the refinement stage. This is a logical trade-off: because refinement processes longer contexts, decoding naturally takes more time. Even with this refinement step, the impact on total wall-clock time remains manageable. We observed an average increase of roughly 15% over GRPO, representing a modest trade-off given the substantial performance gains reported in our main experiments. Note that in more demanding environments like ALFWorld and DeepSearch, training times do climb for both ECHO and GRPO. This is driven by the complexity of the tasks, not by the co-evolution logic itself.

Ultimately, these results show that ECHO's success isn't just the product of a massive compute budget. By delivering significantly stronger performance for a well-defined, moderate increase in time, ECHO offers a highly favorable efficiency trade-off.

D Pseudo-code for ECHO

Algorithm 1 provides the complete training procedure of ECHO. It summarizes the cascaded rollout pipeline (on-policy proposal τ_o, multi-view critiques $\{c_o^{(j)}\}$, and critique-conditioned refinements $\{\tau_r^{(j)}\}$), the saturation-aware critic reward computation in Eq. (6), and the synchronized dual-track GRPO updates

Table 7: Training wall-clock time breakdown (seconds). We report the decomposition of training time for ECHO and compare it with the baseline GRPO under the same setup.

Component	WebShop	ALFWorld	SciWorld	DeepSearch
Policy Rollout (ECHO)	10	14	10	17
Critic Rollout	10	8	9	10
Refinement	272	541	192	581
Total ECHO	327	592	231	649
GRPO Rollout	259	502	168	552
Total GRPO	273	519	184	581
Overhead (%)	+19	+14	+25	+11

for the policy and the critic performed on the same on-policy batch.

Algorithm 1: ECHO: Evolving Critic for Hindsight-Guided Optimization

Input : Dataset $\mathcal{D}$; reward model R; policy P_θ; critic C_ψ; group size N; GRPO hyperparams (ϵ, β); smoothing $\eta > 0$.

Output : Updated parameters (θ, ψ).

foreach *training step* **do**

Sample a batch of queries $q \sim \mathcal{D}$

`// Stage 0: On-policy proposal and baseline score`

Sample initial trajectory $\tau_o \sim P_\theta(\cdot \mid q)$

Compute baseline score $s_o \leftarrow R(q, \tau_o)$

`// Stage 1: Multi-view diagnosis (critic group)`

for $j \leftarrow 1$ **to** N **do**

Sample critique $c_o^{(j)} \sim C_\psi(\cdot \mid q, \tau_o, s_o)$

$\mathcal{G}_C \leftarrow \{c_o^{(j)}\}_{j=1}^N$

`// Stage 2: Conditional refinement (policy group)`

for $j \leftarrow 1$ **to** N **do**

Form augmented input $\tilde{q}^{(j)} \leftarrow (q, c_o^{(j)})$

Sample refinement $\tau_r^{(j)} \sim P_\theta(\cdot \mid \tilde{q}^{(j)})$

Evaluate post-correction score $s_r^{(j)} \leftarrow R(q, \tau_r^{(j)})$

$\mathcal{G}_P(q) \leftarrow \{\tau_r^{(j)}\}_{j=1}^N$

`// Saturation-aware critic reward for each critique`

for $j \leftarrow 1$ **to** N **do**

$r_c^{(j)} \leftarrow \ln\left(\frac{1-s_o+\eta}{1-s_r^{(j)}+\eta}\right)$ `// Eq. 6`

`// Dual-track group-relative advantage estimation`

Compute policy advantages $\{A_P^{(j)}\}_{j=1}^N$ by group-relative normalization of $\{s_r^{(j)}\}_{j=1}^N$

Compute critic advantages $\{A_C^{(j)}\}_{j=1}^N$ by group-relative normalization of $\{r_c^{(j)}\}_{j=1}^N$

`// Synchronized GRPO updates (same batch, two tracks)`

Update θ by maximizing the GRPO surrogate objective $\mathcal{J}(\theta)$ using sequences $\{\tau_r^{(j)}\}$ with advantages $\{A_P^{(j)}\}$

Update ψ by maximizing the GRPO surrogate objective $\mathcal{J}(\psi)$ using sequences $\{c_o^{(j)}\}$ with advantages $\{A_C^{(j)}\}$

E Prompt for Critic Model

We provide the exact prompting template used to elicit critiques from the critic model in Box A.1. The prompt constrains the critic to ground its feedback in the official scoring information and to output at most 1–2 high-level, actionable suggestions in a fixed format, which stabilizes training and keeps critiques consistent across rollouts.

Box A.1 : The prompt for critique generation.

You are a **Critic model**, used to provide guidance on a model's overall performance on a given task.

The system will provide you with:
- The user's task description;
- Several rounds of the model's interactions with the environment as it attempts to complete the task;
- The official final scoring.

Your task: **Strictly based on the official scoring information**, output clear improvement suggestions and behavioral guidance inside the <critic> tag. Only point out issues and directions for improvement. Do not mention strengths, and do not give praise or encouragement.

Input Description
Each round of the model's actions is provided within '<model_response>...</model_response>'. Environment feedback is provided within '<env_feedback>...</env_feedback>'.

Official Scoring Criteria (for your understanding; do not modify or question)

{Detail scoring criteria on specific task.}

You must:
- Treat the official scoring result as absolutely correct and final.
- Not reinterpret, question, or adjust the official score.

Original Question
{original_prompt}

Response to be Evaluated
{initial_response}

Official Scoring Information
{score_info_text}

Output Format
You must output **only** the following two tags and their contents. Do not add or remove tags, and do not output anything outside these tags:
```
<reason>
Your detailed reasoning process:
- Based on the official scoring information, analyze the model's performance on each scoring dimension;
- Strictly follow the official scoring information; do not question, revise, or supplement it;
```

- Whenever you have any doubts or subjective judgments about any evaluation dimension, always take the official scoring conclusion as the final basis.
</reason>
<critic>
Your final guidance:
- If the model's score is a full score (1 point), directly return "none".
- Give **at most 1–2** brief, high-level suggestions; only mention the most critical issues.
- Do **not** refer to specific reply text or dialogue details; only describe how model should change its general behavior or strategy.
- Do not give any praise or encouragement; only point out problems and how You should improve.
- Always address the model as **"You"** (second person), not "the model", "it", or similar.
</critic>
```Based on the above standards, provide guidance on the given model's behavior, and output in the specified format.

## Output
Now, begin your reasoning!
```