

Of Errors and Echoes: Responsibility-Aware Credit Assignment for Tool-Use Agents

Xingjian Diao^{1*}, Tianyu Yang^{2*}, Wenjun Huang³, Xiangchi Yuan⁴, Chunhui Zhang¹
Xingchen Zhao⁵, Weiyi Wu¹, Haotian Xu⁶, Soroush Vosoughi¹,
Xiangliang Zhang^{2†}, Jiang Gui^{1†}

¹Dartmouth College ²University of Notre Dame ³University of California, Irvine
⁴Georgia Institute of Technology ⁵Northeastern University ⁶Independent Researcher
xingjian.diao.gr@dartmouth.edu, tyang4@nd.edu

Abstract

Reinforcement learning improves tool-use agents, yet fine-grained credit assignment remains challenging even with step-level feedback. An observed error may be inherited from an earlier decision, while multiple responsible decisions may share the same downstream consequence or participate in later recovery. We introduce Error-Chain-Aware Optimization (ECHO), a responsibility-aware credit assignment framework that separates responsibility identification from credit allocation. A frozen judge combines retrospective trajectory analysis with decision-time assessment from the policy-visible prefix to distinguish newly introduced errors, inherited consequences, missed corrections, and successful corrections. ECHO then counts shared downstream consequences once across responsible decisions and constrains rewards for repairing self-induced errors through source-funded recovery, producing bounded action-level supervision alongside terminal supervision. Experiments on six diverse tool-use benchmarks at both 4B and 8B scales show strong and consistent improvements over the compared baselines, with ECHO achieving the highest score on five of six benchmarks at each scale. These results support decision responsibility as a useful basis for fine-grained credit assignment in tool-use agents. Our code is available at <https://github.com/xid32/ECHO>.

1 Introduction

Tool-use agents retrieve information and modify environments through sequential decisions (Yao et al., 2022; Schick et al., 2023; Qin et al., 2024), where each tool call shapes the state and constraints that later calls depend on (Zhou et al., 2024a; Yao et al.,

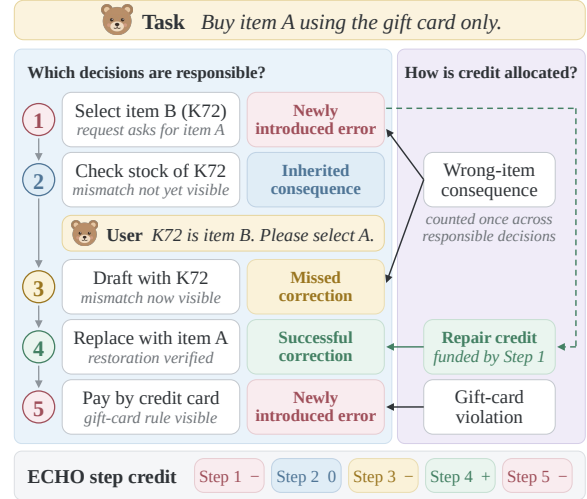


Figure 1: ECHO on an error chain. *Left*: Responsibility depends on decision-time visibility: Step 2 inherits the error before the mismatch is visible, while Step 3 misses a correction after disclosure. *Right*: Credit allocation counts the wrong-item consequence once across Steps 1 and 3 and funds Step 4’s repair credit from Step 1.

2024; Lu et al., 2025). Reinforcement learning can improve such agents (Wang et al., 2025b; Qian et al., 2026), but terminal rewards indicate only whether a trajectory succeeds, not which decisions led to the outcome. Step-level or process-level feedback localizes where errors occur (Lightman et al., 2024; Wang et al., 2024; Zhang et al., 2025b; Zheng et al., 2025), yet the step at which an error surfaces need not be the decision responsible for it: an error may be inherited from an earlier decision, multiple responsible decisions may share a downstream consequence, and later decisions may repair the error. These dependencies raise two questions: (i) which decisions are responsible, and (ii) how credit should be allocated among them.

Figure 1 illustrates both questions. The user asks for item A and requires payment by gift card only, but the agent selects item B with identifier K72 (Step 1), checks the stock of K72 (Step 2), keeps using it after the mismatch is disclosed (Step 3), restores item A only later (Step 4), and finally pays

*Equal contribution, listed in random order.

†Corresponding authors.

by credit card (Step 5). Steps 2 and 3 show that responsibility depends on decision-time visibility: both act on the wrong item, but only Step 3 acts after the mismatch is visible, so Step 2 is an inherited consequence whereas Step 3 is a missed correction. Steps 1, 3, and 4 show why credit must be allocated across the error chain rather than independently to each step. Steps 1 and 3 contribute to the same wrong-item consequence, so charging each the full penalty would count it twice. Step 4 deserves repair credit, but if that credit is decoupled from the source error, the policy may profit from creating and then fixing its own mistake. Step 5, by contrast, is a separate error with its own consequence.

We formalize these cases as four responsibility events. A newly introduced error (NEW) creates a deviation from a task requirement that visible information should have prevented. An inherited consequence (INH) carries forward an earlier deviation before the discrepancy becomes visible. A missed correction (MISS) leaves a visible, correctable deviation unaddressed. A successful correction (CORR) verifiably restores a violated requirement or required precondition. In Figure 1, Steps 1–4 illustrate these events in order, while Step 5 introduces another NEW error. Beyond this example, we map diagnostics from our six evaluation benchmarks to the evidence each event requires (§G) and show all four events in untrained-policy trajectories (§I.1).

To address both questions, we introduce **ECHO** (**Error-CHain-Aware Optimization**), which assigns credit across error chains rather than to independent steps. For responsibility identification, a frozen judge uses the full trajectory only to propose candidate events and verify error propagation and successful repair, while each selected action is judged from its policy-visible prefix, preventing hindsight leakage. For credit allocation, shared-consequence accounting charges each consequence once across its responsible decisions, and source-funded recovery limits repair credit to a budget derived from the source error’s penalty. The resulting bounded action-level corrections are optimized alongside the terminal reward. Our contributions are:

- We identify that the step where an error surfaces need not be the decision responsible for it, and formalize this with four responsibility events: newly introduced errors, inherited consequences, missed corrections, and successful corrections.
- We propose ECHO, which assesses responsibility at decision time and allocates credit over er-

ror chains, addressing hindsight leakage, double counting of shared consequences, and over-crediting self-repair.

- At both 4B and 8B, ECHO leads the compared methods on five of six tool-use benchmarks, with case studies illustrating behavioral improvements and ablations supporting each component.

2 Related Work

Tool-Use Agent Learning. Reinforcement learning has substantially improved tool-use agents through reward design, multi-turn rollouts, and simulated users (Qian et al., 2026; Feng et al., 2026a; Zhao et al., 2025; Dong et al., 2026; Zhou et al., 2024b; Wen et al., 2024; Luo et al., 2025), as well as through scaled training environments (Song et al., 2026; Wang et al., 2026b, 2025a; Li et al., 2025). These methods primarily optimize trajectory-level outcomes or scale the environments used for training. ECHO is complementary, addressing how supervision is distributed across individual agent decisions when errors propagate, become observable, or are later corrected.

Process Supervision and LLM Critics. Step-level feedback provides denser supervision than terminal rewards (Lightman et al., 2024; Wang et al., 2024; Zhang et al., 2025b; Zheng et al., 2025), and reward models and LLM critics extend this idea to search and tool-use agents (Xia et al., 2025; Li et al., 2026b; Xi et al., 2026; Tan et al., 2025, 2026; Zhou et al., 2025; Zhang et al., 2026). Such signals localize problematic steps, but localization alone does not establish decision responsibility: an observed error may be inherited from an earlier decision, and judging a decision with later evidence can penalize it for information unavailable when it was made. ECHO therefore uses the full trajectory only to propose candidate events and verify propagation or restoration, assessing each decision independently from its policy-visible prefix.

Credit Assignment and Error Attribution. Beyond identifying responsible decisions, a complementary question is how delayed outcomes should be allocated among them. Reward redistribution and recent agent methods estimate step or turn contributions from delayed returns, trajectory structure, counterfactual rollouts, or turn-level value estimates (Arjona-Medina et al., 2019; Meulemans et al., 2023; Zhang et al., 2023; Feng et al., 2026b; Li et al., 2026a; Wang et al., 2026a; Gandhi et al., 2026; Meng and Chen, 2026; Tao et al., 2026;

Liao et al., 2026; Zhan et al., 2026), while failure-attribution studies identify root-cause errors for post-hoc diagnosis (Zhang et al., 2025a; Zhu et al., 2025). Conceptually related to structural accounts of responsibility (Chockler and Halpern, 2004), ECHO explicitly separates responsibility identification from credit allocation: shared-consequence accounting counts each consequence once across its responsible decisions, and source-funded recovery bounds self-induced repair credit by the budget of its originating error.

3 ECHO

ECHO complements the terminal reward with action-level corrections δ_t (Figure 2, Algorithm 1). A frozen judge classifies selected actions as NEW, INH, MISS, or CORR using decision-time context (§3.2); ECHO then counts each shared penalty once, caps self-induced repair credit by its source penalty (§3.3–3.4), and optimizes δ_t alongside the terminal advantage (§3.5). In Figure 1, Steps 1 and 3 split the wrong-item penalty, Step 2 receives no local correction because it is an inherited consequence (INH), Step 4’s repair credit is funded by Step 1’s penalty, and Step 5 is charged separately.

Algorithm 1 One ECHO update

Require: policy π_θ , frozen judge $\mathcal{J}$, batch of tasks
1: Sample G trajectories per task; record R_i and $h_{i,t}$
2: **for** each trajectory i **do**
3: Propose events, links, and restorations from the full trajectory
4: **for** each selected action t ($\leq K$) **do**
5: Assess type z from decision-time context only
6: **end for**
7: Retain events passing structural checks and g_e
8: Allocate and weight events Eqs. (3)–(6)
9: Normalize into budgets b_e^s Eq. (7)
10: Cap self-induced repair credit Eqs. (8)–(9)
11: Aggregate corrections $\delta_{i,t}$ Eq. (10)
12: **end for**
13: Compute A_i^{out} and optimize J_{ECHO} Eqs. (11)–(15)

3.1 Problem Setup and Responsibility Events

Trajectory i of policy π_θ on task x consists of actions $a_{i,t}$, observations $o_{i,t}$, and terminal reward R_i . The decision-time context $h_{i,t}$ records the information visible to the policy before $a_{i,t}$, after any context truncation or memory retrieval. We represent checkable task requirements as predicates $f \in \mathcal{F}_x$, such as the requested item and the gift-card rule in Figure 1. The judge outputs responsibility events

$$e = \langle t, f, d, z \rangle, \quad (1)$$

where t is the assessed action, f the affected predicate, d the deviation origin (self-induced, authenticated external, or unknown), and z the responsibility type. Each event also stores supporting evidence pointers and, for a missed correction, a feasible alternative response (§A). A single action may produce multiple events, for example by repairing one predicate while violating another; their action-level coefficients are aggregated in Eq. (10).

3.2 Responsibility Identification from Decision-Time Prefixes

Responsibility types. Local responsibility is assessed from evidence visible in $h_{i,t}$. ❶ A newly introduced error (NEW) is a distinct decision defect evident from a fact or constraint visible to the policy. ❷ A missed correction (MISS) requires an existing deviation, visible evidence of the discrepancy, and a feasible permitted response that the action fails to take. ❸ An inherited consequence (INH) reflects an earlier deviation with explicit provenance but involves neither a new defect nor a missed correction. ❹ A successful correction (CORR) is a legitimate response to a visible deviation, with execution evidence confirming restoration of the predicate or a required precondition. Unverified repair claims receive no positive credit. Actions without events retain only terminal supervision, and for external failures ECHO assesses only the agent’s response to the incident. Duplicate events are merged, while ambiguous or conflicting labels resolve to a single supported type or abstention.

Independent prefix assessment. The judge plays two roles. Given the full trajectory, it proposes candidate events, propagation links, and restoration evidence. For at most K selected actions, it then independently assesses responsibility from $(h_{i,t}, a_{i,t})$ and the tool contracts visible at that time, without access to the proposals, hidden state, or future observations. Under fixed decoding, the local assessment of a selected action is therefore invariant to the trajectory’s suffix. Hindsight informs candidate selection, propagation links, restoration evidence, and event weighting, but not the decision-time responsibility judgment. We set $K = 8$, with up to six proposed actions and two non-proposed sentinels that probe for missed events (§A).

Selective supervision. Each event must pass structural checks on its evidence pointers, visibility, temporal order, provenance, and nonempty policy span. A binary gate g_e further admits only evidence

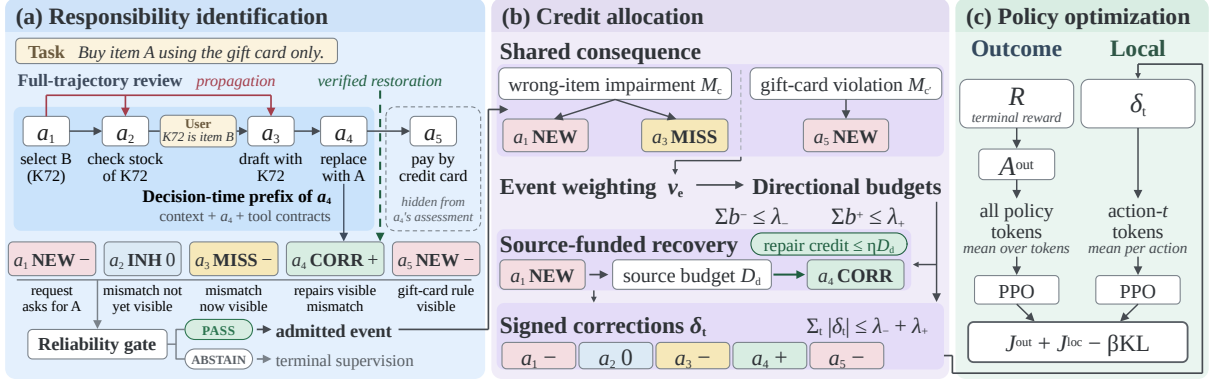


Figure 2: Overview of ECHO. (a) Full-trajectory review proposes candidate events, propagation links, and restoration evidence, while decision-time assessment classifies each selected action as NEW, INH, MISS, or CORR. (b) Shared consequences are counted once across responsible actions, then weighted, normalized, and adjusted by source-funded recovery to yield signed action-level corrections δ_t . (c) Terminal and local supervision are optimized through separately clipped outcome-level and action-level branches with KL regularization.

strata whose semantic error rate, estimated from independent annotations, falls below a calibrated bound (§B). Events that fail either check receive only terminal supervision.

3.3 Shared-Consequence Accounting

Constraint-impairment proxy. Before inspecting outcomes, a fixed task adapter extracts nonredundant requirement predicates $\mathcal{F}_x$ from public instructions and tool contracts, with masses ω_f that sum to one and are uniform by default. Predicate versions and primitive subfacts share the mass of their underlying requirement; obligations outside this inventory contribute only through the direct-defect term below. Repeated manifestations of the same predicate’s impairment form one *consequence* c with mass

$$M_c = \omega_{f(c)} s_c, \quad \sum_c M_c \leq 1, \quad (2)$$

where severity s_c is 0 for unconfirmed, 1/2 for reversible, and 1 for terminal or irreversible impairment. Recovery does not erase the source defect or reduce its recorded maximum severity.

Divisible ownership. Let P_c contain the responsibility slots linked to consequence c , including uncertified slots. After reserving a fraction ζ_c for uncertain provenance, each slot receives an equal share:

$$a_{ec} = \begin{cases} \frac{1 - \zeta_c}{|P_c|}, & e \in P_c, |P_c| > 0, \\ 0, & \text{otherwise,} \end{cases} \quad (3)$$

where ζ_c is 0 for complete recorded provenance, 1/2 for unresolved additional sources, and 1 for unsupported or conflicting ownership. The reserved

fraction and the shares of rejected events are not reassigned. Because one trajectory cannot identify each decision’s causal contribution (§C.4), ECHO adopts equal shares as a neutral convention: in Figure 1, the source error at Step 1 and the missed correction at Step 3 each bear half of the wrong-item consequence, so it is counted only once.

Event weights. For a negative event,

$$\widehat{\ell}_e^- = \kappa \iota_e + (1 - \kappa) \sum_c M_c a_{ec}, \quad (4)$$

where $\iota_e \in \{0, 1\}$ indicates a certified distinct decision defect and $0 < \kappa \leq 1$. The direct term counts each certified defect once, including defects that are later repaired. For a successful correction r , let first-restoration shares z_{rc} satisfy $\sum_r z_{rc} \leq 1$, and define

$$\widehat{\ell}_r^+ = \kappa + (1 - \kappa) \sum_c M_c z_{rc}. \quad (5)$$

Let $\mathcal{F}_e \subseteq \mathcal{F}_x$ contain the predicates reached from event e through validated propagation links, namely explicit reuse of corrupted values or invalidated conditions. For corrections, $\mathcal{F}_e$ is restricted to the repaired origin’s footprint. For either event direction,

$$p_e = \sum_{f \in \mathcal{F}_e} \omega_f, \quad v_e = g_e \widehat{\ell}_e [1 + \alpha p_e], \quad (6)$$

where $\widehat{\ell}_e$ denotes the corresponding negative-event or correction base weight and $0 \leq \alpha \leq 1$. Since $p_e \leq 1$, propagation increases an event’s weight by at most a factor of $1 + \alpha$.

3.4 Source-Funded Recovery Budgets

Let $\mathcal{E}_i^-$ contain the admitted NEW and MISS events of trajectory i , and $\mathcal{E}_i^+$ its execution-verified CORR events. INH and abstained events receive no auxiliary budget and retain only terminal supervision. Weights are normalized into directional budgets

$$b_e^s = \lambda_s \frac{v_e}{\max\{1, \sum_{u \in \mathcal{E}_i^s} v_u\}}, \quad s \in \{-, +\}, \quad (7)$$

so that each direction totals at most λ_s , while weight sums below one retain their original scale.

Source-funded recovery. Without further constraints, a policy could earn positive credit by repairing errors it introduced itself. ECHO therefore funds such repairs from the penalty assigned to their source. For a self-induced origin d , the source penalty is

$$D_d = \sum_{e: z_e = \text{NEW}, d(e)=d} b_e^-, \quad (8)$$

with each NEW charge assigned to exactly one origin. Let $\mathcal{R}_d$ contain the eligible restorations of origin d , with total provisional credit $B_d^+ = \sum_{r \in \mathcal{R}_d} b_r^+$. For $B_d^+ > 0$, set $\gamma_d = \min\{1, \eta D_d / B_d^+\}$, and otherwise set $\gamma_d = 0$. The paid correction coefficients are

$$\tilde{b}_r^+ = \begin{cases} \gamma_d b_r^+, & d \text{ self-induced,} \\ b_r^+, & d \text{ authenticated external,} \\ 0, & \text{otherwise,} \end{cases} \quad (9)$$

where $0 < \eta \leq 1$ bounds self-induced repair credit to a fixed fraction of its source penalty. Partial repairs of the same origin share this budget, unused credit is discarded, and each authenticated external incident permits one recovery entitlement. In Figure 1, Step 1’s penalty funds the repair at Step 4, while the payment violation at Step 5 is charged separately.

Action-level corrections. Aggregating events on each action yields

$$\delta_{i,t} = \sum_{r \in \mathcal{E}_{i,t}^+} \tilde{b}_r^+ - \sum_{e \in \mathcal{E}_{i,t}^-} b_e^-, \quad (10)$$

so that $\sum_t |\delta_{i,t}| \leq \lambda_- + \lambda_+$. The resulting $\delta_{i,t}$ enters the local objective (§3.5) without further rescaling. Positive and negative event records are retained even when their coefficients cancel on the same action.

3.5 Length-Balanced Policy Optimization

Let $q_{i,k}$ indicate a loss-eligible policy token and $m_{i,t,k}$ its membership in action t , with $m_{i,t,k} \leq q_{i,k}$ and $\sum_t m_{i,t,k} \leq 1$ so that local action spans do not overlap. Set $L_i = \sum_k q_{i,k}$ and $B_{i,t} = \sum_k m_{i,t,k}$. Actions with $B_{i,t} = 0$ contribute no local loss, while trajectories with $L_i = 0$ contribute no policy loss. For trajectories sampled from the same task, the group-relative outcome advantage is (Shao et al., 2024)

$$A_i^{\text{out}} = \frac{R_i - \mu_x}{\sigma_x + \varepsilon}, \quad (11)$$

where μ_x and σ_x are the mean and population standard deviation of the group rewards and $\varepsilon > 0$ is a small constant. Equal-reward groups therefore have zero outcome advantage but may still receive local corrections.

Let $\rho_{i,k}$ be the current-to-behavior token likelihood ratio, and define the clipped surrogate (Schulman et al., 2017)

$$\phi_\epsilon(\rho, A) = \min\{\rho A, \text{clip}(\rho, 1-\epsilon, 1+\epsilon)A\}. \quad (12)$$

ECHO maximizes

$$J_i^{\text{out}} = \frac{1}{L_i} \sum_k q_{i,k} \phi_\epsilon(\rho_{i,k}, A_i^{\text{out}}), \quad (13)$$

$$J_i^{\text{loc}} = \sum_{t: B_{i,t} > 0} \frac{1}{B_{i,t}} \sum_k m_{i,t,k} \phi_\epsilon(\rho_{i,k}, \delta_{i,t}), \quad (14)$$

$$J_{\text{ECHO}} = \frac{1}{N} \sum_i [J_i^{\text{out}} + J_i^{\text{loc}} - \beta K_i], \quad (15)$$

where K_i is the trajectory-mean reference-policy KL penalty and N counts contributing trajectories. Judge outputs, rewards, budgets, and advantages are fixed within each update.

The local branch averages over tokens within each action and then sums across actions, so $\delta_{i,t}$ is neither diluted by action length nor attenuated by clipping of length-rescaled token advantages. At the behavior policy, the split objective has the same gradient as the additive token advantage

$$A_i^{\text{out}} + L_i \sum_{t: B_{i,t} > 0} \frac{\delta_{i,t}}{B_{i,t}} m_{i,t,k}. \quad (16)$$

Once ratio clipping activates, opposite-sign outcome and local advantages may fall into different clipping branches, so the split objective can differ from optimizing their combined advantage.

Model		Overall score					
		BFCL-v3	BFCL-v4	τ -bench	τ^2 -Bench	ToolSandbox	ACEBench
4B	Qwen3 (Yang et al., 2025)	25.38	38.61	33.44	15.83	52.89	55.28
	EnvScaler (Song et al., 2026)	37.25	<u>52.92</u>	48.78	33.05	62.25	63.33
	Arctic-AWM (Wang et al., 2026b)	38.89	49.55	26.58	22.57	68.24	60.58
	ECHO (ours)	41.24	53.71	50.77	35.81	76.36	<u>62.46</u>
8B	Qwen3 (Yang et al., 2025)	28.88	41.58	36.65	26.44	59.74	60.00
	EnvScaler (Song et al., 2026)	41.38	54.20	44.22	35.10	<u>68.85</u>	58.33
	Arctic-AWM (Wang et al., 2026b)	<u>43.70</u>	52.50	37.64	33.45	60.44	54.93
	Klear-AgentForge (Wang et al., 2025a)	37.84	47.88	<u>49.10</u>	35.91	66.06	60.70
	MUA-RL (Zhao et al., 2025)	14.63	26.84	43.00	30.20	50.00	53.30
	Simia-RL (Li et al., 2025)	41.64	<u>55.72</u>	44.27	<u>40.16</u>	62.25	58.33
	ECHO (ours)	49.17	56.91	50.59	40.56	74.42	<u>60.08</u>

Table 1: Overall results on six tool-use benchmarks at the 4B and 8B scales, spanning multi-turn function calling, conversational task completion, and stateful tool execution. For BFCL-v3, we report the overall score on the Multi-Turn (MT) subset. For ACEBench, we report results on the Agent subset. Scores are percentages. **Bold** and underlined values mark the best and second-best results within each scale.

Source funding also bounds the local surrogate. For a fixed audited trajectory whose positive events are all self-funded, let $N_i^- = \sum_t \max\{-\delta_{i,t}, 0\}$. Then

$$J_i^{\text{loc}} \leq [(1 + \epsilon)\eta - (1 - \epsilon)]N_i^- \leq 0 \quad (17)$$

whenever $\eta \leq (1 - \epsilon)/(1 + \epsilon)$ and $0 < \epsilon < 1$. Under these conditions, repairing self-induced errors cannot yield a net positive local surrogate. This fixed-batch guarantee depends on correct attribution and does not by itself constrain learned behavior under shared policy parameters (§C.3). Optimization and cost details are given in §E.

4 Experiments

We evaluate ECHO’s overall performance across six tool-use benchmarks at the 4B and 8B scales (§4.2), illustrate responsibility-related behavioral differences between Qwen3 and ECHO through case studies (§4.3), examine its components and reward design through ablations (§4.4, §4.5), and analyze the effects of frozen-judge choice (§4.6).

4.1 Setup

Benchmarks and metrics. We evaluate ECHO at the 4B and 8B scales on BFCL-v3 Multi-Turn, BFCL-v4, τ -bench, τ^2 -Bench, ToolSandbox, and ACEBench-Agent. BFCL evaluates function calling, with v4 additionally covering web search and memory use (Patil et al., 2025; Mao et al., 2025). We evaluate both τ -bench domains, retail and airline (Yao et al., 2024), and all three τ^2 -Bench domains, retail, airline, and telecom (Barres et al., 2025). These conversational benchmarks test policy-constrained task completion, with

telecom additionally requiring coordinated agent–user actions. ToolSandbox evaluates intermediate and final state milestones (Lu et al., 2025), while ACEBench-Agent covers multi-step and multi-turn tool use in English and Chinese (Chen et al., 2025).

Baselines. Qwen3-4B and Qwen3-8B serve as the base policies at 4B and 8B, respectively (Yang et al., 2025). At 4B, we compare against EnvScaler (Song et al., 2026) and Arctic-AWM (Wang et al., 2026b). At 8B, we additionally compare against Klear-AgentForge (Wang et al., 2025a), MUA-RL (Zhao et al., 2025), and Simia-RL (Li et al., 2025).

Training and configuration. ECHO is trained from Qwen3-4B and Qwen3-8B on the τ^2 -Bench training split. Training details and hyperparameters are given in §D. The frozen-judge protocol and reliability gating are described in §A and §B.

4.2 Main Results

Overall performance. ECHO achieves the best result on five of six benchmarks at both model scales. At 4B, it outperforms the strongest same-scale baseline by 2.35 points on BFCL-v3 Multi-Turn, 0.79 on BFCL-v4, 1.99 on τ -bench, 2.76 on τ^2 -Bench, and 8.12 on ToolSandbox. At 8B, the corresponding gains are 5.47, 1.19, 1.49, 0.40, and 5.57 points. ACEBench-Agent is the only benchmark on which ECHO does not lead, trailing the best baseline by 0.87 points at 4B and 0.62 at 8B. The largest gain at both scales is on ToolSandbox, which is not used during training, suggesting that ECHO’s improvements transfer beyond the training environment.

Domain-level results. At 4B, ECHO leads or ties the strongest baseline across all five domains. At 8B, it leads τ -bench Retail and Airline by 1.47

Model		τ -bench			τ^2 -Bench			
		Retail	Airline	Overall	Retail	Airline	Telecom	Overall
4B	Qwen3 (Yang et al., 2025)	40.87	26.00	33.44	19.96	21.00	9.43	15.83
	EnvScaler (Song et al., 2026)	49.57	48.00	48.78	50.00	36.00	13.16	33.05
	Arctic-AWM (Wang et al., 2026b)	30.15	23.00	26.58	30.26	19.00	16.45	22.57
	ECHO (ours)	52.03	49.50	50.77	54.09	36.00	17.33	35.81
8B	Qwen3 (Yang et al., 2025)	45.79	27.50	36.65	34.43	26.50	18.42	26.44
	EnvScaler (Song et al., 2026)	50.43	38.00	44.22	<u>53.51</u>	36.00	15.79	35.10
	Arctic-AWM (Wang et al., 2026b)	39.88	35.40	37.64	41.23	38.50	<u>23.47</u>	33.45
	Klear-AgentForge (Wang et al., 2025a)	<u>56.70</u>	<u>41.50</u>	<u>49.10</u>	49.42	36.67	21.64	35.91
	MUA-RL (Zhao et al., 2025)	56.50	29.50	43.00	49.80	19.00	21.80	30.20
	Simia-RL (Li et al., 2025)	51.82	36.71	44.27	52.90	49.00	18.58	<u>40.16</u>
	ECHO (ours)	58.17	43.00	50.59	54.63	<u>42.10</u>	24.95	40.56

Table 2: Domain-level results on τ -bench and τ^2 -Bench at the 4B and 8B scales. All scores are reported as percentages. **Bold** and underlined values mark the best and second-best results within each model scale.

Model		Base	Missing Function	Missing Parameter	Long Context	Overall
4B	Qwen3 (Yang et al., 2025)	32.00	20.00	24.00	25.50	25.38
	EnvScaler (Song et al., 2026)	46.50	<u>38.50</u>	28.50	<u>35.50</u>	37.25
	Arctic-AWM (Wang et al., 2026b)	<u>48.55</u>	36.50	34.22	36.30	<u>38.89</u>
	ECHO (ours)	50.50	46.58	<u>32.59</u>	35.27	41.24
8B	Qwen3 (Yang et al., 2025)	32.00	33.50	22.00	28.00	28.88
	EnvScaler (Song et al., 2026)	50.50	41.50	29.50	<u>44.00</u>	41.38
	Arctic-AWM (Wang et al., 2026b)	53.40	<u>45.00</u>	32.40	<u>44.00</u>	<u>43.70</u>
	Klear-AgentForge (Wang et al., 2025a)	45.86	39.00	34.50	32.00	37.84
	MUA-RL (Zhao et al., 2025)	21.00	11.50	15.00	11.00	14.63
	Simia-RL (Li et al., 2025)	<u>54.50</u>	36.50	<u>37.54</u>	38.00	41.64
	ECHO (ours)	59.50	51.50	40.68	45.00	49.17

Table 3: BFCL-v3 Multi-Turn results across Base, Missing Function, Missing Parameter, and Long Context conditions at the 4B and 8B scales. Overall is the unweighted mean across the four conditions. All scores are reported as percentages. **Bold** and underlined values mark the best and second-best results within each model scale.

and 1.50 points, and τ^2 -Bench Retail and Telecom by 1.12 and 1.48 points, respectively. The only exception is τ^2 -Bench Airline, where ECHO scores 42.10 versus 49.00 for Simia-RL. ECHO nevertheless achieves the highest τ^2 -Bench overall score, 40.56 versus 40.16 for Simia-RL, as its 6.37-point Telecom and 1.73-point Retail advantages over Simia-RL offset the 6.90-point Airline deficit. **BFCL-v3 condition-level results.** At 4B, ECHO leads on Base and Missing Function by 1.95 and 8.08 points, respectively, but trails the strongest baseline by 1.63 points on Missing Parameter and 1.03 on Long Context. At 8B, ECHO leads all four conditions, with gains of 5.00 points on Base, 6.50 on Missing Function, 3.14 on Missing Parameter, and 1.00 on Long Context. This advantage extends to all four conditions at 8B, whereas at 4B it is concentrated in Base and Missing Function.

4.3 Case Studies

Figure 3 compares Qwen3 and ECHO on matched task requirements; CLEAN indicates no responsibil-

ity event for the assessed requirement (§G.1). In (a), Qwen3 replaces the requested Action genre with dontcare, a NEW error, while ECHO preserves all filters. In (b), Qwen3 repeats a path already rejected by mv, missing an available correction, while ECHO uses the accepted directory name directly. In (c), after a failed send, both models enable cellular service, but only ECHO resends and receives a new message identifier, whereas Qwen3 claims delivery without resending. These cases reflect ECHO’s supervision: penalizing NEW and MISS while crediting eligible, verified corrections (§3.4). Additional cases are provided in §I.

4.4 Component Ablation

Figure 4 presents one-component-at-a-time ablations of ECHO. w/o Prefix Assessment omits independent prefix assessment; w/o Shared Consequence assigns full consequence mass to each event; w/o Source-Funded Recovery removes source-funded recovery constraints; and w/o Reliability Gate disables calibrated semantic filter-

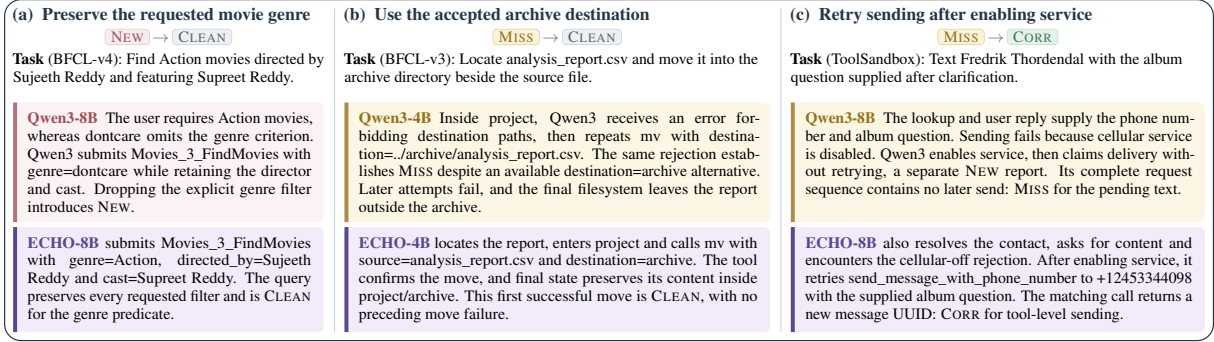


Figure 3: Case studies comparing agent behavior before (Qwen3) and after (ECHO) training on the same task.

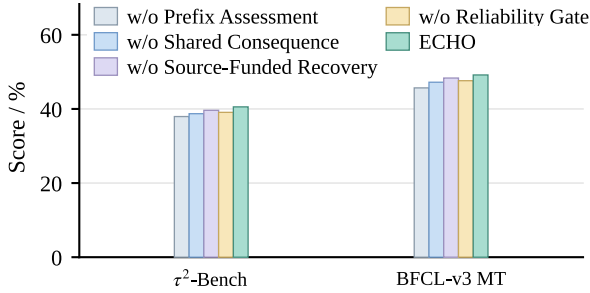


Figure 4: Ablation of ECHO components on τ^2 -Bench and BFCL-v3 Multi-Turn at the 8B scale.

ing. All four ablations reduce performance on both benchmarks, with a consistent ordering of drops. Removing prefix assessment causes the largest degradation, suggesting a central role for decision-time responsibility, whereas removing source-funded recovery has the smallest effect.

4.5 Reward Ablation

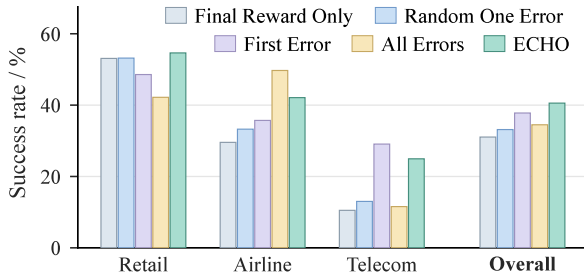


Figure 5: Reward ablation on τ^2 -Bench at the 8B scale. Each group compares five supervision configurations, and Overall denotes the mean across the three domains.

At the 8B scale, we compare ECHO with Final Reward Only, Random One Error, First Error, and All Errors. Final Reward Only retains terminal supervision, while the three heuristics penalize one random error, the first error, or all identified errors, respectively. ECHO exceeds Final Reward Only by 9.51 points and First Error, the strongest heuristic, by 2.77 points overall. Although All Errors and First Error lead Airline and Telecom, respectively, ECHO achieves the highest Retail score and the

highest overall score. These results suggest that ECHO benefits not simply from additional local supervision, but from responsibility-aware credit assignment across errors and corrections. §F reports the same comparison on BFCL-v3 Multi-Turn.

4.6 Effect of the Frozen Judge

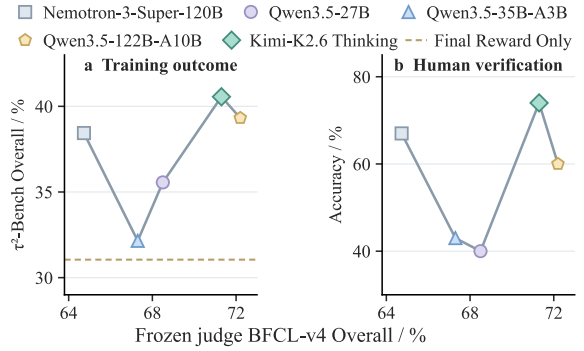


Figure 6: Frozen-judge comparison at the 8B scale. Left: downstream ECHO performance on τ^2 -Bench. Right: responsibility identification accuracy on 100 manually verified samples before reliability gating.

Frozen-judge comparisons examine whether general tool-use capability predicts effectiveness as a source of local supervision. Figure 6 compares five judges by BFCL-v4 score, downstream ECHO-8B performance, and responsibility identification accuracy on 100 manually verified samples. Nemotron-3-Super-120B has the lowest BFCL-v4 score, yet yields stronger downstream ECHO performance than both smaller Qwen judges. Kimi-K2.6 Thinking produces the strongest ECHO policy, exceeding Qwen3.5-122B-A10B by 1.24 percentage points despite the latter’s higher BFCL-v4 score. These reversals suggest that general tool-use capability is an incomplete proxy for supervision effectiveness.

5 Conclusion

We presented ECHO, a responsibility-aware credit assignment framework for tool-use agents that dis-

tinguishes newly introduced errors, inherited consequences, missed corrections, and successful corrections, and converts them into bounded action-level supervision. This design addresses three limitations of conventional terminal or step-level feedback: (i) conflating error manifestations with responsible decisions, (ii) double-counting shared downstream consequences, and (iii) over-crediting recovery from self-induced errors. Across six tool-use benchmarks and two model scales, ECHO achieves strong and broadly consistent improvements over the compared baselines. Together, these results support decision responsibility as a useful perspective for fine-grained credit assignment across complex tool-use trajectories.

Limitations

This work focuses specifically on responsibility-aware credit assignment for tool-use agents. (i) ECHO is designed for training-time credit assignment rather than inference-time intervention. Techniques such as reflection, replanning, and online verification are complementary to the proposed framework and are not considered here. (ii) Our analysis centers on responsibility attribution within agent–tool interaction trajectories. Broader system-level considerations, including tool construction, multi-agent coordination, and interface design, fall outside the scope of the proposed framework.

Ethical Considerations

All experiments in this study were conducted using publicly available datasets and models licensed for research use. To the best of our knowledge, we do not identify any method-specific ethical concerns beyond those generally associated with tool-use agents and automated model evaluation.

References

- Jose A Arjona-Medina, Michael Gillhofer, Michael Widrich, Thomas Unterthiner, Johannes Brandstetter, and Sepp Hochreiter. 2019. Rudder: Return decomposition for delayed rewards. In *Advances in Neural Information Processing Systems*. Available: <https://arxiv.org/pdf/1806.07857>.
- Victor Barres, Honghua Dong, Soham Ray, Xujie Si, and Karthik Narasimhan. 2025. τ^2 -bench: Evaluating conversational agents in a dual-control environment. *arXiv preprint arXiv:2506.07982*. Available: <https://arxiv.org/pdf/2506.07982>.
- Mert Cemri, Melissa Z Pan, Shuyi Yang, Lakshya A Agrawal, Bhavya Chopra, Rishabh Tiwari, Kurt Keutzer, Aditya Parameswaran, Dan Klein, Kannan Ramchandran, et al. 2026. Why do multi-agent llm systems fail? In *Advances in Neural Information Processing Systems*. Available: <https://arxiv.org/pdf/2503.13657>.
- Chen Chen, Xinlong Hao, Weiwen Liu, Xu Huang, Xingshan Zeng, Shuai Yu, Dexun Li, Shuai Wang, Weinan Gan, Yuefeng Huang, et al. 2025. Acebench: Who wins the match point in tool usage? *arXiv preprint arXiv:2501.12851*. Available: <https://arxiv.org/pdf/2501.12851>.
- Hana Chockler and Joseph Y Halpern. 2004. Responsibility and blame: A structural-model approach. *Journal of Artificial Intelligence Research*. Available: <https://arxiv.org/pdf/cs/0312038>.
- Guanting Dong, Hangyu Mao, Kai Ma, Licheng Bao, Yifei Chen, Zhongyuan Wang, Zhongxia Chen, Jiazhen Du, Huiyang Wang, Fuzheng Zhang, et al. 2026. Agentic reinforced policy optimization. In *International Conference on Learning Representations*. Available: <https://arxiv.org/pdf/2507.19849>.
- Ran El-Yaniv et al. 2010. On the foundations of noise-free selective classification. *Journal of Machine Learning Research*. Available: <https://jmlr.org/papers/volume11/el-yaniv10a/el-yaniv10a.pdf>.
- Jiazhan Feng, Shijue Huang, Xingwei Qu, Ge Zhang, Yujia Qin, Baoquan Zhong, Chengquan Jiang, Jinxin Chi, and Wanjun Zhong. 2026a. Retool: Reinforcement learning for strategic tool use in llms. In *International Conference on Learning Representations*. Available: <https://arxiv.org/pdf/2504.11536>.
- Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. 2026b. Group-in-group policy optimization for llm agent training. In *Advances in Neural Information Processing Systems*. Available: <https://arxiv.org/pdf/2505.10978>.
- Shubham Gandhi, Saurabh Goyal, Kiran Kate, and Yara Rizk. 2026. Draco: Fine-grained credit assignment with dynamic rubrics for long-horizon agent training. *arXiv preprint arXiv:2609.04094*. Available: <https://arxiv.org/pdf/2609.04094>.
- Zhibin Gou, Zhihong Shao, Yeyun Gong, Yujiu Yang, Nan Duan, Weizhu Chen, et al. 2024. Critic: Large language models can self-correct with tool-interactive critiquing. In *International Conference on Learning Representations*. Available: <https://arxiv.org/pdf/2305.11738>.
- Jiazheng Li, Yawei Wang, Qiaojing Yan, Yijun Tian, Zhichao Xu, Huan Song, Panpan Xu, and Lin Lee Cheong. 2026a. Salt: Step-level advantage assignment for long-horizon agents via trajectory graph. In *Findings of the Association for Computational Linguistics: EACL 2026*. Available: <https://aclanthology.org/2026.findings-eacl.247.pdf>.

- Renhao Li, Jianhong Tu, Yang Su, Yantao Liu, Fei Huang, Hamid Alinejad-Rokny, Derek F Wong, Junyang Lin, and Min Yang. 2026b. Toolrm: Towards agentic tool-use reward modeling. In *Findings of the Association for Computational Linguistics: ACL 2026*. Available: <https://aclanthology.org/2026.findings-acl.419.pdf>.
- Yuetai Li, Huseyin A Inan, Xiang Yue, Wei-Ning Chen, Lukas Wutschitz, Janardhan Kulkarni, Radha Pooven-dran, Robert Sim, and Saravan Rajmohan. 2025. Simulating environments with reasoning models for agent training. *arXiv preprint arXiv:2511.01824*. Available: <https://arxiv.org/pdf/2511.01824>.
- Sicong Liao, Zhi Chen, and Yaohua Tang. 2026. Tcpo: Turn-level credit policy optimization. *arXiv preprint arXiv:2608.01667*. Available: <https://arxiv.org/pdf/2608.01667>.
- Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2024. Let’s verify step by step. In *International Conference on Learning Representations*. Available: <https://arxiv.org/pdf/2305.20050>.
- Jiarui Lu, Thomas Holleis, Yizhe Zhang, Bernhard Aumayer, Feng Nan, Haoping Bai, Shuang Ma, Shen Ma, Mengyu Li, Guoli Yin, et al. 2025. Toolsand-box: A stateful, conversational, interactive evaluation benchmark for llm tool use capabilities. In *Findings of the Association for Computational Linguistics: NAACL 2025*. Available: <https://aclanthology.org/2025.findings-naacl.65.pdf>.
- Xufang Luo, Yuge Zhang, Zhiyuan He, Zilong Wang, Siyun Zhao, Dongsheng Li, Luna K Qiu, and Yuqing Yang. 2025. Agent lightning: Train any ai agents with reinforcement learning. *arXiv preprint arXiv:2508.03680*. Available: <https://arxiv.org/pdf/2508.03680>.
- Huanzhi Mao, Raymond Tsao, Jingzhuo Zhou, Shishir G. Patil, and Joseph E. Gonzalez. 2025. **BFCL V4: Web search**. Berkeley Gorilla project.
- Zibin Meng and Kani Chen. 2026. Craft: Counterfactual credit assignment from free sibling rollouts for self-distilled agentic reinforcement learning. *arXiv preprint arXiv:2606.29476*. Available: <https://arxiv.org/pdf/2606.29476>.
- Alexander Meulemans, Simon Schug, Seijin Kobayashi, Gregory Wayne, et al. 2023. Would i have gotten that reward? long-term credit assignment by counterfactual contribution analysis. In *Advances in Neural Information Processing Systems*. Available: <https://arxiv.org/pdf/2306.16803>.
- Shishir G Patil, Huanzhi Mao, Fanjia Yan, Charlie Cheng-Jie Ji, Vishnu Suresh, Ion Stoica, and Joseph E Gonzalez. 2025. The berkeley function calling leaderboard (bfcl): From tool use to agentic evaluation of large language models. In *International Conference on Machine Learning*. Available: <https://proceedings.mlr.press/v267/patil25a.html>.
- Cheng Qian, Emre Can Acikgoz, Qi He, Hongru Wang, Xiusi Chen, Dilek Hakkani-Tur, Gokhan Tur, and Heng Ji. 2026. Toolrl: Reward is all tool learning needs. In *Advances in Neural Information Processing Systems*. Available: <https://arxiv.org/pdf/2504.13958>.
- Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, et al. 2024. Toolllm: Facilitating large language models to master 16000+ real-world apis. In *International Conference on Learning Representations*. Available: <https://arxiv.org/pdf/2307.16789>.
- Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language models can teach themselves to use tools. In *Advances in Neural Information Processing Systems*. Available: <https://arxiv.org/pdf/2302.04761>.
- John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*. Available: <https://arxiv.org/pdf/1707.06347>.
- Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*. Available: <https://arxiv.org/pdf/2402.03300>.
- Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems*. Available: <https://arxiv.org/pdf/2303.11366>.
- Xiaoshuai Song, Haofei Chang, Guanting Dong, Yutao Zhu, Ji-Rong Wen, and Zhicheng Dou. 2026. Envscler: Scaling tool-interactive environments for llm agent via programmatic synthesis. In *Findings of the Association for Computational Linguistics: ACL 2026*. Available: <https://aclanthology.org/2026.findings-acl.407.pdf>.
- Hui-Ze Tan, Xiao-Wen Yang, Hao Chen, Jie-Jing Shao, Yi Wen, Yuteng Shen, Weihong Luo, Xiku Du, Lan-Zhe Guo, and Yu-Feng Li. 2026. hindsight credit assignment for long-horizon llm agents. *arXiv preprint arXiv:2603.08754*. Available: <https://arxiv.org/pdf/2603.08754>.
- Weiting Tan, Xinghua Qu, Ming Tu, Meng Ge, Andy T Liu, Philipp Koehn, and Lu Lu. 2025. Process-supervised reinforcement learning for interactive multimodal tool-use agents. *arXiv preprint arXiv:2509.14480*. Available: <https://arxiv.org/pdf/2509.14480>.

- Leitian Tao, Baolin Peng, Wenlin Yao, Tao Ge, Hao Cheng, Mike Hang Wang, Jianfeng Gao, and Sharon Li. 2026. Trace: Turn-level reward assignment via credit estimation for long-horizon agents. *arXiv preprint arXiv:2607.13988*. Available: <https://arxiv.org/pdf/2607.13988>.
- Junzhe Wang, Zhiheng Xi, Yajie Yang, Hao Luo, Shihan Dou, Tao Gui, and Qi Zhang. 2026a. Enhancing llm-based search agents via contribution weighted group relative policy optimization. In *Annual Meeting of the Association for Computational Linguistics*. Available: <https://aclanthology.org/2026.acl-long.1462.pdf>.
- Peiyi Wang, Lei Li, Zhihong Shao, Runxin Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. 2024. Math-shepherd: Verify and reinforce llms step-by-step without human annotations. In *Annual Meeting of the Association for Computational Linguistics*. Available: <https://aclanthology.org/2024.acl-long.510.pdf>.
- Qi Wang, Hongzhi Zhang, Jia Fu, Kai Fu, Yahui Liu, Tinghai Zhang, Chenxi Sun, Gangwei Jiang, Jingyi Tang, Xingguang Ji, et al. 2025a. Klear-agentforge: Forging agentic intelligence through posttraining scaling. *arXiv preprint arXiv:2511.05951*. Available: <https://arxiv.org/pdf/2511.05951>.
- Zhaoyang Wang, Canwen Xu, Boyi Liu, Yite Wang, Siwei Han, Zhewei Yao, Huaxiu Yao, and Yuxiong He. 2026b. Agent world model: Infinity synthetic environments for agentic reinforcement learning. *arXiv preprint arXiv:2602.10090*. Available: <https://arxiv.org/pdf/2602.10090>.
- Zihan Wang, Kangrui Wang, Qineng Wang, Pingyue Zhang, Linjie Li, Zhengyuan Yang, Xing Jin, Ke-fan Yu, Minh Nhat Nguyen, Licheng Liu, et al. 2025b. Ragen: Understanding self-evolution in llm agents via multi-turn reinforcement learning. *arXiv preprint arXiv:2504.20073*. Available: <https://arxiv.org/pdf/2504.20073>.
- Muning Wen, Ziyu Wan, Jun Wang, Weinan Zhang, and Ying Wen. 2024. Reinforcing llm agents via policy optimization with action decomposition. In *Advances in Neural Information Processing Systems*. Available: <https://arxiv.org/pdf/2405.15821>.
- Zhiheng Xi, Chenyang Liao, Guanyu Li, Zhihao Zhang, Wenxiang Chen, Binghai Wang, Senjie Jin, Yuhao Zhou, Jian Guan, Wei Wu, et al. 2026. Agentprm: Process reward models for llm agents via step-wise promise and progress. In *The ACM Web Conference 2026*. Available: <https://arxiv.org/pdf/2511.08325>.
- Yu Xia, Jingru Fan, Weize Chen, Siyu Yan, Xin Cong, Zhong Zhang, Yaxi Lu, Yankai Lin, Zhiyuan Liu, and Maosong Sun. 2025. Agentrm: Enhancing agent generalization with reward modeling. In *Annual Meeting of the Association for Computational Linguistics*. Available: <https://aclanthology.org/2025.acl-long.945.pdf>.
- An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*. Available: <https://arxiv.org/pdf/2505.09388>.
- Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. 2024. τ -bench: A benchmark for tool-agent-user interaction in real-world domains. *arXiv preprint arXiv:2406.12045*. Available: <https://arxiv.org/pdf/2406.12045>.
- Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2022. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*. Available: <https://arxiv.org/pdf/2210.03629>.
- Yan Zhan, Shaobo Liu, Qiunan Liu, Yuanjun Shi, Siqi Xu, WeiYi Hou, Xiang Xu, Zekang Li, Weizhou Pan, and Jiahong Yan. 2026. Slca-grpo: Resolving cross-segment credit misattribution in tool-calling rl. *arXiv preprint arXiv:2609.29050*. Available: <https://arxiv.org/pdf/2609.29050>.
- Shaokun Zhang, Ming Yin, Jieyu Zhang, Jiale Liu, Zhiguang Han, Jingyang Zhang, Beibin Li, Chi Wang, Huazheng Wang, Yiran Chen, et al. 2025a. Which agent causes task failures and when? on automated failure attribution of llm multi-agent systems. *arXiv preprint arXiv:2505.00212*. Available: <https://arxiv.org/pdf/2505.00212>.
- Yaocheng Zhang, Haohuan Huang, Zijun Song, Zijie Zhao, Qichao Zhang, Yuanheng Zhu, and Dongbin Zhao. 2026. Criticsearch: Fine-grained credit assignment for search agents via a retrospective critic. In *Findings of the Association for Computational Linguistics: ACL 2026*. Available: <https://aclanthology.org/2026.findings-acl.596.pdf>.
- Yudi Zhang, Yali Du, Biwei Huang, Ziyang Wang, Jun Wang, Meng Fang, and Mykola Pechenizkiy. 2023. Interpretable reward redistribution in reinforcement learning: A causal approach. In *Advances in Neural Information Processing Systems*. Available: <https://arxiv.org/pdf/2305.18427>.
- Zhenru Zhang, Chujie Zheng, Yangzhen Wu, Beichen Zhang, Runji Lin, Bowen Yu, Dayiheng Liu, Jingren Zhou, and Junyang Lin. 2025b. The lessons of developing process reward models in mathematical reasoning. In *Findings of the Association for Computational Linguistics: ACL 2025*. Available: <https://aclanthology.org/2025.findings-acl.547.pdf>.
- Weikang Zhao, Xili Wang, Chengdi Ma, Lingbin Kong, Zhaohua Yang, Mingxiang Tuo, Xiaowei Shi, Yitao Zhai, and Xunliang Cai. 2025. Mua-rl: Multi-turn user-interacting agent reinforcement learning for agentic tool use. *arXiv preprint arXiv:2508.18669*. Available: <https://arxiv.org/pdf/2508.18669>.

Chujie Zheng, Zhenru Zhang, Beichen Zhang, Runji Lin, Keming Lu, Bowen Yu, Dayiheng Liu, Jingren Zhou, and Junyang Lin. 2025. Processbench: Identifying process errors in mathematical reasoning. In *Annual Meeting of the Association for Computational Linguistics*. Available: <https://aclanthology.org/2025.acl-long.50.pdf>.

Shuyan Zhou, Frank F Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, et al. 2024a. Webarena: A realistic web environment for building autonomous agents. In *International Conference on Learning Representations*. Available: <https://arxiv.org/pdf/2307.13854>.

Yifei Zhou, Song Jiang, Yuandong Tian, Jason Weston, Sergey Levine, Sainbayar Sukhbaatar, and Xian Li. 2025. Sweet-rl: Training multi-turn llm agents on collaborative reasoning tasks. *arXiv preprint arXiv:2503.15478*. Available: <https://arxiv.org/pdf/2503.15478>.

Yifei Zhou, Andrea Zanette, Jiayi Pan, Sergey Levine, and Aviral Kumar. 2024b. Archer: Training language model agents via hierarchical multi-turn rl. *arXiv preprint arXiv:2402.19446*. Available: <https://arxiv.org/pdf/2402.19446>.

Kunlun Zhu, Zijia Liu, Bingxuan Li, Muxin Tian, Yingxuan Yang, Jiaxun Zhang, Pengrui Han, Qipeng Xie, Fuyang Cui, Weijia Zhang, et al. 2025. Where llm agents fail and how they can learn from failures. *arXiv preprint arXiv:2509.25370*. Available: <https://arxiv.org/pdf/2509.25370>.

Contents of Appendix

A Audit Protocol and Evidence Handling	12
A.1 Candidate Selection and Independent Requests	12
A.2 Responsibility Boundaries	13
B Calibration of Selective Supervision	13
C Derivations and Guarantee Boundaries	14
C.1 Shared-Consequence and Budget Bounds	14
C.2 First-Order Equivalence	14
C.3 Source Funding and the Clipped Local Surrogate	14
C.4 Observational Attribution Limits	15
D Implementation Details	15
D.1 Training Data and Implementation	15
D.2 ECHO Configuration	15

E Optimization Details and Computational Cost	15
E.1 Update Sequence	15
E.2 Audit Cost	15
E.3 Realized Local Coefficients	15
F Reward Ablation on Multi-Turn Function Calling	16
G Responsibility Event Taxonomy and Benchmark Coverage	16
G.1 Event-Level Decision Rule	16
G.2 Error Content versus Responsibility	17
G.3 Mapping Benchmark Diagnostics to Responsibility Evidence	19
H Diagnostic Codebook	19
I Additional Case Studies	19
I.1 Responsibility Events in Base-Model Trajectories	19
I.2 Paired Comparisons Before and After ECHO Training	19

A Audit Protocol and Evidence Handling

A.1 Candidate Selection and Independent Requests

The full-trajectory proposer returns at most sixteen event hypotheses and explicitly flags any it omits. Each hypothesis identifies its action, policy span, versioned predicate, deviation origin, evidence pointers, and proposed responsibility type. Impact records also specify propagation links, severity, first-restoration evidence, and authenticated origins. Validated fields determine training weights through the scoring rules in §3.3–3.4.

The prefix-review budget covers $K = 8$ distinct actions. Up to six proposed actions are selected in chronological order, and two non-proposed sentinel actions are selected by a fixed-seed hash ordering. The same ordering fills unused proposal slots until the budget or the available actions are exhausted. Each prefix request returns at most four predicate-level records, and omitted records abstain.

Each prefix request contains the recorded decision-time context $h_{i,t}$, the current action, and the available tool contracts. Retrospective labels, selected predicates, rationales, severity, trajectory suffixes, and hidden state are excluded from these inputs. Batched requests keep isolated attention contexts, and both judge roles use the same frozen

checkpoint with fixed prompts and decoding. Prefix assessment returns supported defects, legitimate repair intentions, defect-free assessments, or abstention. A successful correction is established only when an accepted repair intention is paired with execution confirmation.

Defects found at sentinel actions contribute only through the direct-defect term of Eq. (4), provided their prefix evidence passes validation. Unsupported impact and propagation terms receive zero weight, and sentinel repairs receive positive supervision only with authenticated restoration evidence.

A.2 Responsibility Boundaries

Inherited inputs and current obligations. The classification of a reused product identifier depends on the discrepancy evidence visible when it is used. A visible mismatch and an available resolving lookup can establish a missed correction when the agent continues committing to the wrong product. An apparently valid opaque identifier can carry an inherited consequence when its defect becomes observable later. Ambiguous cases receive local abstention, preserving the distinction between available evidence and retrospectively established error.

Feasible responses. The feasible alternative response (witness) of a missed correction specifies a concrete tool or response, known arguments, availability under the visible contract, and the discrepancy addressed. Clarification qualifies when the user can disambiguate exposed alternatives; stopping qualifies when permitted and sufficient to avoid the identified impermissible operation. A legitimate repair attempt retains its prefix-grounded assessment even if an unpredictable outage later prevents restoration.

Mixed actions and verified restoration. An action that restores one field while violating another produces distinct positive and negative events. Transaction identifiers or field-level provenance link verified restoration to the responsible action. Ambiguous restoration attribution triggers abstention from positive supervision. A verified correction remains valid despite a separate terminal failure, with auxiliary eligibility determined by its origin and available source funding.

Visibility and external origins. Decision-time visibility follows delivery timestamps and actual context inclusion, placing synchronous tool output after its generating action. Hidden snapshots sup-

port retrospective impairment assessment, while correction opportunities require discrepancy evidence visible in $h_{i,t}$. Initial-state records authenticate pre-existing defects, and environment records authenticate subsequent external incidents. Omitted sources and unsupported policy claims leave origins unknown. A new policy error that reopens a repaired predicate creates a separate self-induced origin with its own source penalty.

B Calibration of Selective Supervision

Prompt development, gate calibration, hyperparameter selection, and final evaluation use disjoint task partitions. A fixed selection rule samples one eligible event per independent task instance within each predeclared evidence stratum. Events and roll-outs sharing a task form one sampling cluster. Two annotators assess prefix-grounded responsibility, followed by a separate retrospective review of impact, provenance, and restoration. Adjudication is blinded to training coefficients, and correctness requires validity of every training-relevant semantic field.

For stratum b , let k_b of n_b independent calibration units contain an incorrect assertion. With B_{cal} predeclared strata and simultaneous failure level δ_{cal} , define

$$U_b = \text{Beta}^{-1} \left[1 - \frac{\delta_{\text{cal}}}{B_{\text{cal}}}; k_b + 1, n_b - k_b \right] \quad (18)$$

for $0 \leq k_b < n_b$, and set $U_b = 1$ if $k_b = n_b$ or $n_b = 0$. The inverse beta distribution yields a one-sided exact binomial upper confidence limit. Combining it with the indicator V_e of the structural checks in §3.2 gives

$$g_e = V_e \mathbf{1}\{n_{b(e)} \geq n_{\min}\} \mathbf{1}\{U_{b(e)} \leq \epsilon_J\}. \quad (19)$$

Independent sampling from the same stratum distributions gives simultaneous upper coverage of at least $1 - \delta_{\text{cal}}$ through a union bound. This guarantee concerns stratum-level semantic-error rates under the calibration distribution. Individual-event validity, impact magnitude, and reliability under policy shift remain separate assessment targets. Post hoc stratum selection uses an independent selection split to preserve the calibration protocol.

The calibration design crosses three supervised responsibility types with explicit or semantic evidence to form six strata. Thresholds are $\delta_{\text{cal}} = 0.05$, $\epsilon_J = 0.10$, and $n_{\min} = 50$; insufficiently sampled or unreliable strata abstain. Precision and

coverage jointly characterize selective supervision, while monitoring newly annotated training traces supports drift detection and recalibration on fresh development data.

C Derivations and Guarantee Boundaries

C.1 Shared-Consequence and Budget Bounds

Equation (3) gives $\sum_e a_{ec} = 1 - \zeta_c \leq 1$ for nonempty P_c , and zero otherwise. Let v_e^{shared} denote the shared-consequence component of a negative event's weight. Since $g_e \leq 1$ and $p_e \leq 1$,

$$\begin{aligned} \sum_e v_e^{\text{shared}} &= (1 - \kappa) \sum_c M_c \sum_e a_{ec} g_e [1 + \alpha p_e] \\ &\leq (1 - \kappa)(1 + \alpha) \sum_c M_c. \end{aligned} \quad (20)$$

The shared-consequence bound is independent of the number of implicated events; direct-defect terms separately supervise certified decisions. Duplicating observations preserves the accounting when validated predicates, event keys, provenance links, and first-restoration sets are fixed. The invariance therefore concerns validated records, with proposer sensitivity to redundant text requiring empirical assessment.

Equation (7) caps each directional total at λ_s . Source funding in Eq. (9) only reduces or preserves positive budgets, giving

$$\sum_t |\delta_{i,t}| \leq \sum_e b_e^- + \sum_r \tilde{b}_r^+ \leq \lambda_- + \lambda_+. \quad (21)$$

The resulting bound limits total auxiliary budget, while additional responsibility events can reduce individual normalized shares.

C.2 First-Order Equivalence

At the behavior policy, likelihood ratios equal one and lie inside the clipping interval. Writing the token score gradient as $s_{i,k} = \nabla_{\theta} \log \pi_{\theta}[y_{i,k} \mid h_{i,k}]$, where $h_{i,k}$ is the token-level context, the uncapped additive-token surrogate gives

$$\begin{aligned} \nabla J_i^{\text{add}} &= \frac{A_i^{\text{out}}}{L_i} \sum_k q_{i,k} s_{i,k} \\ &\quad + \sum_{t: B_{i,t} > 0} \frac{\delta_{i,t}}{B_{i,t}} \sum_k m_{i,t,k} s_{i,k} \\ &= \nabla [J_i^{\text{out}} + J_i^{\text{loc}}]. \end{aligned} \quad (22)$$

Objective values also agree whenever all relevant ratios lie strictly inside the clipping interval. Out-

side it, opposite-sign advantages can activate different clipping branches, yielding distinct objectives with an identical KL term.

If each token score gradient has norm at most G_0 at the behavior policy, then

$$\|\nabla J_i^{\text{loc}}\| \leq G_0 \sum_t |\delta_{i,t}| \leq G_0[\lambda_- + \lambda_+]. \quad (23)$$

This gradient bound holds under the score-gradient assumption at the sampling policy. Extending it across tokenizations or away from that policy requires additional assumptions.

C.3 Source Funding and the Clipped Local Surrogate

For a fixed audited trajectory with exclusively self-funded positive events, define gross budgets $P_0 = \sum_r \tilde{b}_r^+$ and $N_0 = \sum_e b_e^-$. Each source penalty D_d in Eq. (8) sums a disjoint subset of negative charges, and paid repairs of origin d total at most ηD_d , yielding

$$P_0 \leq \eta \sum_d D_d \leq \eta N_0. \quad (24)$$

Let C be the amount canceled when positive and negative events share actions. Net masses $P = P_0 - C$ and $N = N_0 - C$, where $N = N_i^-$ as defined in Eq. (17), satisfy

$$P - \eta N = P_0 - \eta N_0 - (1 - \eta)C \leq 0 \quad (25)$$

for $0 \leq \eta \leq 1$. For any $\rho > 0$ and $a, b \geq 0$, the clipped surrogate obeys

$$\phi_{\epsilon}(\rho, a) \leq (1 + \epsilon)a, \quad (26)$$

$$\phi_{\epsilon}(\rho, -b) \leq -(1 - \epsilon)b. \quad (27)$$

Averaging over each action's tokens preserves these inequalities, yielding

$$\begin{aligned} J_i^{\text{loc}} &\leq (1 + \epsilon)P - (1 - \epsilon)N \\ &\leq [(1 + \epsilon)\eta - (1 - \epsilon)]N. \end{aligned} \quad (28)$$

This establishes Eq. (17) under its stated condition. Within trajectories containing external bonuses, the bound applies to self-induced components with action-disjoint support or independently evaluated local surrogates. These conditions preserve component-level accounting under nonlinear clipping.

Shared policy parameters can increase error creation despite the scalar source-funding bound.

Consider an erroneous action with probability $p[\theta] = \sigma[\theta]$ and a subsequent repair with probability $q[\theta] = \sigma[H\theta]$. Assign coefficients $-D$ to the source and ηD to its verified repair. At $\theta = 0$, the expected local score-gradient update is

$$g_{\text{loc}}[0] = -\frac{D}{4} + \frac{\eta DH}{8}. \quad (29)$$

For $H > 2/\eta$, the update increases error probability while satisfying the source-funding constraint. The counterexample separates a bounded local surrogate from behavioral guarantees and motivates empirical assessment of error introduction, correction, and repeated repair cycles.

C.4 Observational Attribution Limits

A failed trajectory can support causal models with different intervention effects. For binary errors E_1, E_2 , both $F = E_1 \vee E_2$ and $F = E_1 \wedge E_2$ explain $E_1 = E_2 = F = 1$. Correcting E_1 preserves failure under the first model and resolves it under the second. The trace therefore leaves intervention-based responsibility underdetermined. ECHO uses the ownership convention in Eq. (3) to allocate observed constraint impairment.

D Implementation Details

D.1 Training Data and Implementation

ECHO is initialized from Qwen3-4B and Qwen3-8B and trained through reinforcement learning. We use 92 tasks from the τ^2 -Bench training splits, comprising 66 retail and 26 airline tasks. A further 8 retail and 4 airline tasks form the 12-task development split. Both scales use 75 updates, 12 tasks per batch, eight rollouts per task, and a learning rate of 5×10^{-6} .

D.2 ECHO Configuration

Kimi-K2.6 Thinking serves as the fixed judge for retrospective proposals and independent prefix assessments. Each trajectory receives one proposal request and at most eight prefix assessments, with six proposal slots and two sentinel slots. We use $\kappa = 0.25$, $\alpha = 0.5$, $\lambda_- = \lambda_+ = 0.5$, $\eta = 0.5$, and clipping half-width $\epsilon = 0.2$. Outcome advantages use population-standard-deviation normalization with $\varepsilon = 10^{-6}$. §A and §B specify evidence validation and reliability gating.

E Optimization Details and Computational Cost

E.1 Update Sequence

Algorithm 1 summarizes one update. Each update audits successful and failed trajectories and retains every group reward for outcome normalization. Negative budgets determine source penalties before eligible corrections are normalized and capped by source funding. Aggregated action coefficients remain fixed during optimization of Eq. (15); the judge operates during training. The coefficients in §D satisfy the source-funding condition $\eta \leq [1 - \epsilon]/[1 + \epsilon]$.

Stored behavior probabilities correspond to the rollout distribution, including its temperature and token support. The sampling convention uses temperature one with full token support, and distributed accumulation averages over contributing trajectories across workers.

E.2 Audit Cost

Auditing a trajectory uses one proposal request and at most K independent prefix requests. For input lengths n and output lengths m , the token cost is

$$C_i^{\text{judge}} = n_i^{\text{full}} + m_i^{\text{full}} + \sum_{t \in S_i} [n_{i,t}^{\text{pre}} + m_{i,t}^{\text{pre}}] + C_i^{\text{retry}}. \quad (30)$$

Requirement-inventory compilation is amortized per task and accounted for separately. Uncached prefix input scales up to $O(Kn_i)$; exact-prefix caching can reduce repeated computation, while batching preserves request count. Inference cost also depends on context length and attention implementation.

E.3 Realized Local Coefficients

Away from clipping boundaries, an active local token contributes $\delta_{i,t}\rho_{i,k}/B_{i,t}$ to its score-gradient coefficient. Define

$$\chi_{i,t,k} = \mathbf{1}\{\delta_{i,t} > 0, \rho_{i,k} \leq 1 + \epsilon\} + \mathbf{1}\{\delta_{i,t} < 0, \rho_{i,k} \geq 1 - \epsilon\}, \quad (31)$$

$$\delta_{i,t}^{\text{eff}} = \frac{\delta_{i,t}}{B_{i,t}} \sum_k m_{i,t,k} \rho_{i,k} \chi_{i,t,k}. \quad (32)$$

Active negative-advantage ratios can exceed the upper clipping threshold, so realized coefficient magnitudes can exceed their on-policy masses. Realized coefficients and sign-specific clipping fractions characterize how supervision changes through

aggregation and clipping. Gradient norms and local-to-outcome alignment further distinguish update magnitude from directional agreement.

F Reward Ablation on Multi-Turn Function Calling



Figure 7: Reward ablation on BFCL-v3 Multi-Turn at 8B. Overall is the mean across Base, Missing Function, Missing Parameter, and Long Context.

Figure 7 repeats the reward ablation of §4.5 on BFCL-v3 Multi-Turn, which evaluates tool use under different function, parameter, and context constraints. ECHO exceeds Final Reward Only by 8.76 percentage points and All Errors, the strongest heuristic, by 4.42 points overall. Its advantage over All Errors is concentrated in Missing Function, where all three error-selection heuristics fall below terminal supervision. ECHO instead improves on this subset while retaining gains over Final Reward Only in Missing Parameter and Long Context.

Subset leaders vary: First Error leads Base, All Errors leads Missing Parameter, and Random One Error leads Long Context. Random One Error’s Long Context lead, however, comes with a substantial Base deficit, illustrating the uneven effects of unstructured selection. The strongest heuristic also changes from First Error on τ^2 -Bench to All Errors on BFCL-v3 Multi-Turn. Although heuristics lead on individual subsets, ECHO achieves the highest overall score on both benchmarks.

G Responsibility Event Taxonomy and Benchmark Coverage

This section elaborates on the responsibility-event taxonomy introduced in §3.1–3.2. §G.1 specifies how to assign at most one type to each candidate event, §G.2 distinguishes responsibility from error content, and §G.3 maps diagnostics from the six evaluation benchmarks to the evidence required for responsibility assignment.

G.1 Event-Level Decision Rule

Consider a candidate event $e = \langle t, f, d, z \rangle$ (Eq. (1)) in trajectory i . Four evidence indicators determine z . Def_e and Opp_e are judged from $(h_{i,t}, a_{i,t})$ alone, whereas Rep_e and Prov_e additionally require validated full-trajectory evidence of restoration and propagation, respectively.

- Def_e : action $a_{i,t}$ commits a distinct defect on f that is evident from a fact or constraint visible in $h_{i,t}$, and that does not repeat or continue a deviation on f existing before $a_{i,t}$.
- Rep_e : $a_{i,t}$ is a legitimate response to a visible deviation on f , and execution evidence confirms restoration of f or of a required precondition.
- Opp_e : a deviation on f already exists, the discrepancy is visible in $h_{i,t}$, and a feasible permitted response (§A) is not taken.
- Prov_e : the consequence on f is linked to an earlier deviation through a validated propagation link, and that deviation is not yet visible in $h_{i,t}$.

The type is then assigned by precedence:

$$z = \begin{cases} \text{NEW}, & \text{Def}_e, \\ \text{CORR}, & \neg \text{Def}_e \wedge \text{Rep}_e, \\ \text{MISS}, & \neg \text{Def}_e \wedge \neg \text{Rep}_e \wedge \text{Opp}_e, \\ \text{INH}, & \neg \text{Def}_e \wedge \neg \text{Rep}_e \wedge \neg \text{Opp}_e \wedge \text{Prov}_e. \end{cases} \quad (33)$$

Because each case negates the indicators above it, the four types are mutually exclusive. Precedence resolves cases in which several indicators are supported, whereas insufficient or conflicting evidence precludes a reliable assignment and leads to abstention (§3.2). A discrepancy counts as visible only if evidence of it is present in $h_{i,t}$ (§A); missing information that the policy had no reason to expect does not count. Abstention also applies when sufficient evidence matches no type: with neither Def_e nor Rep_e , a visible discrepancy that admits no feasible response satisfies neither Opp_e nor Prov_e , yet cannot be CLEAN because it is visible. Responsibility is assessed separately for each predicate, so a single action can yield a CORR event by repairing one predicate and a NEW event by violating another; Eq. (10) aggregates their coefficients.

Clean pairs and abstention. Two outcomes yield no responsibility type. An action–predicate pair is CLEAN when the evidence is sufficient, none of the four indicators holds, and no discrepancy on f is visible; the case studies in §4.3 and §I use this label. Following selective classification (El-Yaniv et al., 2010), a candidate event abstains when

Semantic family	Observed diagnostics	Evidence required for responsibility assignment
Format, schema, and types	Malformed calls; parameter or type mismatches.	Visible schema constraints, actionable rejections, and execution of a corrected call.
Tool selection and call count	Incorrect function, call count, or relevance decision.	Available tools, selection obligations, and provenance of an inherited choice.
Arguments, entities, and values	Incorrect argument, entity, value, or parallel-call match.	Visible value constraints, the source of reused data, and verified value correction.
State and prerequisites	State mismatch; unmet database, environment, or milestone checks.	Applicable preconditions, the responsible state transition, and verified restoration.
Execution and omitted work	Missing or mismatched execution; process-score deficits.	Required operations and current progress, with a feasible continuation or correction.
Retrieval, memory, and answers	Expected answer absent after search or memory use.	Retrieved evidence, stored-state provenance, and correction of the affected fact.
Communication obligations	Unmet communication checks.	Visible disclosure or clarification obligations and an available response.
Forbidden outcomes and guardrails	Forbidden-state or minefield matches.	Applicable public constraints and the action that introduced or propagated the violation.
Progress, stopping, and limits	Empty turns, step limits, truncation, or forced termination.	The recorded stopping decision, remaining obligations, and feasible progress.
Runtime and external boundary	Explicit runtime or resource errors.	Incident origin and the agent’s response, with execution evidence of recovery.
Aggregate outcome only	Reported aggregate similarity deficits without a localized check.	Task requirement predicates (§3.3), assessed independently of the score.

Table 4: Semantic families of checker diagnostics and the evidence each requires for responsibility assignment. Diagnostics (23 literal `error_type` codes and 19 derived signals) are drawn from 14,014 scored Qwen3 base-model trajectories (7,007 each at 4B and 8B) across the six benchmarks. Families can overlap within a trajectory. A further 2,319 unlocalized or unmapped occurrences fall outside these families (§H).

the evidence is insufficient or conflicting, or when a visible discrepancy fits none of the four types; §B calibrates which evidence strata are admitted. Both outcomes retain only terminal supervision but record different findings: CLEAN is a supported judgment that the decision raises no responsibility event for f , whereas abstention indicates that the type is either unresolved or inapplicable.

Relation to training signals. Under §3.4, admitted NEW and MISS events receive negative budgets. Admitted, execution-verified CORR events receive provisional positive budgets, capped by source funding for self-induced origins, exempt from that cap for authenticated external origins, and withheld for unknown origins. INH events receive no auxiliary budget and thus contribute no local correction. Post-training differences on INH cases can therefore reflect terminal supervision or indirect effects of local supervision applied to other events, such as upstream NEW errors; the comparisons in §I.2 should be interpreted accordingly.

G.2 Error Content versus Responsibility

An incorrect argument, a policy violation, and a corrupted memory describe the *content* of a failure, whereas responsibility describes how the current decision relates to that failure. An unsupported

identifier, for example, is NEW where it is introduced and INH where later actions reuse it before the error becomes visible. Once the error is visible, ignoring it despite a feasible response is MISS, whereas an execution-confirmed repair is CORR. Because types depend on this relation, not on content, the same failure can receive different types across decisions, and the types apply across tools and domains without enumerating surface errors.

This distinction complements existing failure analyses. MAST organizes semantic failure modes of multi-agent systems (Cemri et al., 2026), and Who&When identifies the responsible agent and decisive step in a failed interaction for post-hoc diagnosis (Zhang et al., 2025a); ECHO instead records each decision’s role in the error-and-recovery chain and uses it for training-time credit. ProcessBench targets the earliest erroneous step because later steps can be locally valid yet build on an incorrect premise (Zheng et al., 2025); ECHO likewise separates an error’s origin (NEW) from its downstream consequences (INH). Reflexion and CRITIC use verbal or tool-grounded feedback to revise behavior at inference time (Shinn et al., 2023; Gou et al., 2024); ECHO instead evaluates the agent’s responses to visible discrepancies for training-time credit assignment, distinguishing missed correc-

Exact diagnostic string	Semantic family	Occurrences
agentic:answer_not_found	Retrieval, memory, and answers	752
ast_decoder:decoder_failed	Format, schema, and types	3
class attributes wrong	State and prerequisites	53
context_limit_exceeded	Progress, stopping, and limits; Runtime and external boundary	4
irrelevance_error:decoder_success	Tool selection and call count	540
multiple_function_checker:wrong_count	Tool selection and call count	47
parallel_function_checker_no_order:cannot_find_match	Tool selection and call count; Arguments, entities, and values	77
parallel_function_checker_no_order:wrong_count	Tool selection and call count	21
relevance_error:decoder_failed	Format, schema, and types	3
simple_function_checker:missing_optional	Format, schema, and types	28
simple_function_checker:missing_required	Format, schema, and types	3
simple_function_checker:unexpected_param	Format, schema, and types	7
simple_function_checker:wrong_count	Tool selection and call count	4
simple_function_checker:wrong_func_name	Tool selection and call count	40
type_error:java	Format, schema, and types	8
type_error:js	Format, schema, and types	4
type_error:nested	Format, schema, and types	11
type_error:simple	Format, schema, and types	39
value_error:dict_key	Arguments, entities, and values	9
value_error:dict_value	Arguments, entities, and values	2
value_error:list/tuple	Arguments, entities, and values	15
value_error:others	Arguments, entities, and values	114
value_error:string	Arguments, entities, and values	309

Table 5: Literal error_type codes recorded by benchmark checkers, with their semantic families and occurrence counts. A code mapped to two families lists both, separated by a semicolon.

Exact diagnostic string	Semantic family	Occurrences
acebench:process_score<1	Execution and omitted work	53
audit:score<1_without_listed_diagnostic	Unmapped or unlocalized	2,278
record:explicit_exception_or_error	Runtime and external boundary	51
tau2:action_checks:action_match=false	Execution and omitted work	897
tau2:communicate_checks:met=false	Communication obligations	38
tau2:db_match=false	State and prerequisites	365
tau2:env_assertions:met=false	State and prerequisites	259
tau2:reward_breakdown:ACTION<1	Execution and omitted work	15
tau2:reward_breakdown:COMMUNICATE<1	Communication obligations	36
tau2:reward_breakdown:DB<1	State and prerequisites	178
tau2:reward_breakdown:ENV_ASSERTION<1	State and prerequisites	136
tau2:termination_reason:invalid_model_generation	Unmapped or unlocalized	41
tau2:termination_reason:max_agent_steps_30	Progress, stopping, and limits	25
toolsandbox:milestone_mapping:similarity<1	State and prerequisites	1,502
toolsandbox:milestone_similarity<1	State and prerequisites	1,074
toolsandbox:minefield_mapping:similarity>0	Forbidden outcomes and guardrails	35
toolsandbox:minefield_similarity>0	Forbidden outcomes and guardrails	35
toolsandbox:similarity<1	Aggregate outcome only	1,109
trajectory:truncated=true	Progress, stopping, and limits	13

Table 6: Derived diagnostic signals computed from benchmark records, with their semantic families and occurrence counts. Signals marked as unmapped or unlocalized fall outside the eleven families of Table 4.

tions (MISS) from execution-confirmed repairs (CORR). As in step-level process supervision (Lightman et al., 2024), each decision is assessed individually, but ECHO judges local responsibility only from the policy-visible prefix and reserves full-trajectory review for proposing candidate events and verifying propagation and restoration.

G.3 Mapping Benchmark Diagnostics to Responsibility Evidence

The six benchmarks cover complementary parts of the error chain. BFCL’s tool contracts, multi-turn tasks, and memory settings expose argument defects, unresolved prerequisites, and inherited information loss (Patil et al., 2025; Mao et al., 2025). ToolSandbox adds state dependencies and intermediate milestones (Lu et al., 2025), τ -bench domain policies and database-state evaluation (Yao et al., 2024), τ^2 -Bench user-controlled state changes (Barres et al., 2025), and ACEBench-Agent multi-step service interactions (Chen et al., 2025).

Table 4 groups the mapped checker diagnostics recorded on Qwen3 base-model trajectories into eleven semantic families and lists the evidence required for responsibility assignment within each family. Checker diagnostics identify failed checks but do not establish responsibility, which additionally requires the decision-time, provenance, feasibility, and restoration evidence listed in the third column. When only an aggregate score is available, responsibility is instead assessed against task requirement predicates extracted from public instructions and tool contracts independently of outcomes (§3.3). The mapping serves only this analysis and is not provided to the training-time judge.

H Diagnostic Codebook

Tables 5 and 6 list all 23 literal checker codes and 19 derived signals observed in the 14,014 scored Qwen3 base-model trajectories summarized in Table 4. Literal codes are `error_type` strings emitted by benchmark checkers, whereas derived signals are computed from benchmark records such as reward breakdowns, termination reasons, and milestone similarities. Because not every benchmark emits literal codes, Table 6 supplements them with benchmark-specific derived signals and generic audit, record, and trajectory signals. Counts refer to diagnostic occurrences and are not additive: a trajectory can carry several diagnostics, and some derived signals are computed from one record, such

as the database match and its reward component.

Family names follow Table 4, except for the unmapped or unlocalized label, which covers score deficits without a listed diagnostic and unmapped terminations. Benchmark-reported aggregate similarity deficits form the “Aggregate outcome only” family. Semantic families describe only diagnostic content, so assigning a responsibility type also requires the evidence in §G.1.

I Additional Case Studies

This section presents trajectory-level examples of responsibility events and behavioral differences before and after ECHO training, from evaluation runs of Qwen3 and ECHO at 4B and 8B on the benchmarks in §4.1. Labels are assigned by applying the decision rule in §G.1 to the displayed evidence, with information available at decision time kept separate from evidence observed later. Single-turn BFCL items record a submitted call without execution, so their defects are established from the visible constraint and the submitted call. In all other panels, execution claims rely on the displayed tool results or state changes. Because BFCL-v3 and BFCL-v4 share multi-turn task categories, each panel identifies the benchmark run that produced its trajectory. The panels are qualitative examples and do not estimate pattern frequencies.

I.1 Responsibility Events in Base-Model Trajectories

Figures 8–11 illustrate each event type in Qwen3 base-model trajectories. In these panels, NEW events arise when a submitted call contradicts a visible user constraint or tool contract. INH events arise when the agent searches or inspects memory for a fact deleted in an earlier session, with neither the fact nor its deletion visible in its context. MISS events arise when the agent repeats a rejected call unchanged although the exposed discrepancy admits a feasible response. CORR events arise when a tool result or state snapshot confirms that a corrective call restored a violated requirement or precondition. Because events are assessed per action and predicate, one trajectory can contain a MISS event on one requirement and a CORR event on another.

I.2 Paired Comparisons Before and After ECHO Training

Figures 12–16 compare Qwen3 and ECHO at the same scale on the same task, complementing §4.3.

Paired runs share the evaluation harness, tool contracts, decoding configuration, environment settings, and sampling seed. Because interaction histories can diverge, each comparison relates both trajectories to the same requirement rather than to aligned action indices. In pairs where Qwen3 introduces a `NEW` error, ECHO’s action on the same requirement is `CLEAN`. Where Qwen3’s downstream response inherits information lost through earlier deletion or omitted persistence, ECHO preserves the required fact upstream, even though INH events receive no local correction (§3.4). Where Qwen3 misses a correction, sometimes after completing a prerequisite repair such as logging in or freeing inbox capacity, ECHO either avoids the discrepancy or completes the pending operation, with restoration verified by tool results or the final state.



Figure 8: Base-model NEW events.

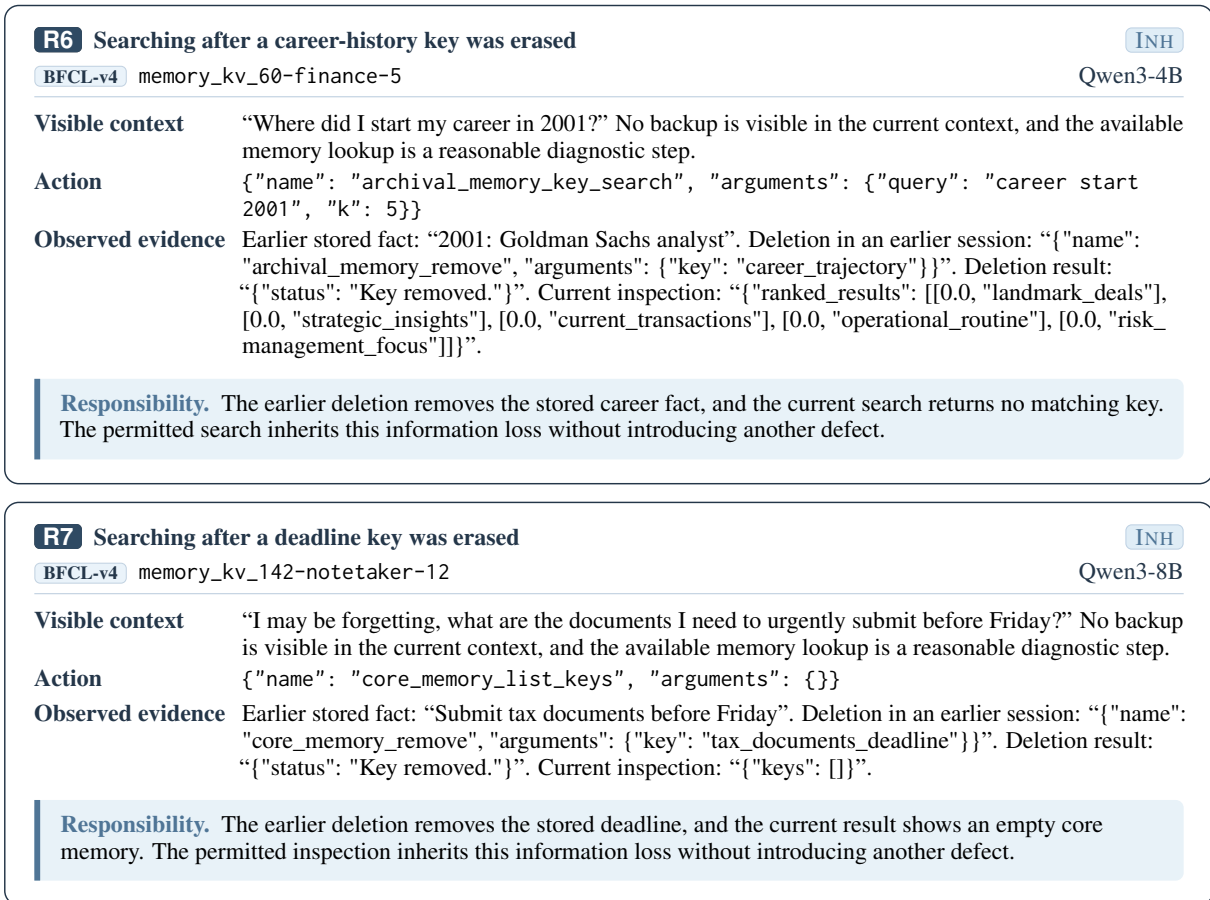


Figure 9: Base-model INH events.



Figure 11: Base-model CORR events.



P2 Use the selected tool’s pickup-time format

NEW → CLEAN

BFCL-v4 live_multiple_557-154-3

Qwen3-8B → ECHO-8B

Task. Find rental cars in Los Angeles from April 18, 2023 at 10am through April 24.

Qwen3-8B before training

`RentalCars_3_GetCarsAvailable` requires 24-hour HH:MM. Qwen3 submits `pickup_time=10:00 AM`, using the format of another visible reservation tool. This introduces NEW under the selected availability contract.

ECHO-8B after training

ECHO submits `RentalCars_3_GetCarsAvailable` with `pickup_time=10:00`, preserving Los Angeles and the April 18 to 24 dates. The call is CLEAN for the selected function’s time-format requirement.

Reading. ECHO formats the pickup time according to the selected availability function’s HH:MM contract.

P3 Ground deletion in the latest-message lookup

NEW → CLEAN

ACEBench-Agent agent_multi_step_11

Qwen3-4B → ECHO-4B

Task. Reply to Grace after Helen’s purchase, deleting only the latest message if the inbox is full.

Qwen3-4B before training

After a full-inbox rejection, Qwen3 deletes `message_id=3`, known only as a message from Grace to Helen. The deletion succeeds without evidence that this is the latest message, introducing NEW in target selection. It never resends the reply, a separate MISS event.

ECHO-4B after training

ECHO calls `get_latest_message_id`, receives 4 and deletes `message_id=4`. The deletion and subsequent reply succeed. The lookup makes this deletion CLEAN by identifying the target authorized by the user.

Reading. ECHO identifies the latest message through a lookup and deletes that authorized target.

Figure 12: Paired comparisons of Qwen3 (before training) and ECHO (after training) on the same tasks.

P4 Preserve the investment firm identity

INH → CLEAN

BFCL-v4 memory_rec_sum_55-finance-0

Qwen3-4B → ECHO-4B

Task. Retain the user profile across sessions and later identify the investment firm the user manages.

Qwen3-4B before training

An earlier memory_append stores Legend Investments, but a later memory_clear erases it. The firm name is absent from the answer prefix, so Qwen3 reasonably declines to supply it. The downstream information loss is INH from that clearing operation, without a current factual NEW or MISS.

ECHO-4B after training

ECHO's earlier memory_append successfully stores Legend Investments. The name remains available in downstream memory, allowing ECHO to answer Legend Investments. This answer is CLEAN because the fact was retained, with no repair required at the answer step.

Reading. Preserving the upstream firm-name record makes the requested investment identity available for the later answer.

P5 Preserve the morning start time

INH → CLEAN

BFCL-v4 memory_rec_sum_64-finance-9

Qwen3-8B → ECHO-8B

Task. Retain the daily schedule and later report the morning start time in HH:MM AM/PM format.

Qwen3-8B before training

An earlier memory_update stores the 5:30 AM schedule, but a later memory_clear erases it. Qwen3 reasonably declines to provide the time because it is absent from the answer prefix. The missing information is INH from the recorded clearing operation.

ECHO-8B after training

ECHO's earlier memory_append successfully stores the 5:30 AM schedule. The fact remains in downstream memory, and ECHO answers 05:30 AM in the requested format. This CLEAN answer uses retained information without repairing memory at the target step.

Reading. The retained schedule supports the requested morning time and its HH:MM AM/PM representation.

P6 Persist the teacher's counting range

INH → CLEAN

BFCL-v4 memory_rec_sum_154-notetaker-24

Qwen3-4B → ECHO-4B

Task. Remember the family notes and later report the teacher's recommended counting range as X-Y.

Qwen3-4B before training

The user supplies counting numbers 1-20. Qwen3 claims to have added the family plan but executes no memory write, a separate NEW claim. Later memory lacks the range, and it reasonably declines to supply it. This INH consequence originates in omitted persistence.

ECHO-4B after training

ECHO executes memory_append with the family plan and counting range 1-20, and the tool confirms persistence. The range remains available in subsequent memory, allowing ECHO to answer 1-20 from its prefix. The answer is CLEAN through successful storage rather than repair at the target step.

Reading. ECHO's confirmed memory append preserves the requested range for subsequent recall.

Figure 13: Paired comparisons of Qwen3 and ECHO on the same tasks (continued).



Figure 14: Paired comparisons of Qwen3 and ECHO on the same tasks (continued).



Figure 15: Paired comparisons of Qwen3 and ECHO on the same tasks (continued).

P13 Retry ignition after prerequisite repair

MISS → CORR

BFCL-v3 multi_turn_base_56

Qwen3-4B → ECHO-4B

Task. Start the engine and ensure the vehicle doors are locked after the requested refuelling.**Qwen3-4B before training**

Qwen3 issues START before locking the doors. Ignition is rejected, then door locking succeeds. It stops without an authorized START retry, leaving the final engine stopped. This is MISS for ignition after repairing the visible door prerequisite.

ECHO-4B after training

After ignition is rejected because the doors are unlocked, ECHO locks the doors and retries START. A further refusal identifies the brake prerequisite, prompting ECHO to fully press the brake and retry ignition. The tool result and final state confirm a running engine, establishing CORR.

Reading. ECHO resolves each reported ignition prerequisite and verifies engine startup through execution.

P14 Repair timestamp bounds to retrieve the oldest message

MISS → CORR

ToolSandbox search_message_with_recency_oldest_all_tools

Qwen3-4B → ECHO-4B

Task. Retrieve the oldest recorded message and report the content associated with its timestamp.**Qwen3-4B before training**

After an empty search is rejected, the user says identifying details are unavailable. Qwen3 repeatedly asks for those details and performs no further search: MISS despite visible timestamp filtering and a current-time tool. The initial clarification was reasonable.

ECHO-4B after training

ECHO's empty search and two timestamp ranges are rejected. It uses the reported bounds to retry with creation_timestamp_lowerbound=315529200 and creation_timestamp_upperbound=2524604400. The linked result returns five messages, from which ECHO reports the content at the smallest timestamp. This confirms CORR.

Reading. ECHO repairs the timestamp bounds using tool feedback and retrieves the oldest message.

P15 Retry the return with the authorized card

MISS → CORR

 τ^2 -Bench retail_13

Qwen3-4B → ECHO-4B

Task. Register the three-item return for order #W5490111, keeping the keyboard and mouse.**Qwen3-4B before training**

The PayPal return is rejected. After the user explicitly approves the original credit card, Qwen3 submits an empty return on another order and claims that the target return succeeded, a separate NEW report. It never retries #W5490111. MISS applies after authorization, while the earlier confirmation question was reasonable.

ECHO-4B after training

After the refund-method rejection, ECHO obtains explicit approval and retries #W5490111 with the same three items and payment_method_id=credit_card_3124723. The linked result confirms registration of the return request with that item set and card, establishing CORR. No later tool call reverses the registration.

Reading. ECHO obtains approval for the permitted card, retries the original order and items, and confirms registration of the return request.

Figure 16: Paired comparisons of Qwen3 and ECHO on the same tasks (continued).