arXiv:submit/8206534 [cs.CL] 9 Oct 2026

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

Xingjian Diao[1*], Tianyu Yang[2*], Wenjun Huang[3], Xiangchi Yuan[4], Chunhui Zhang[1]
Xingchen Zhao[5], Weiyi Wu[1], Haotian Xu[6], Soroush Vosoughi[1],
Xiangliang Zhang[2†], Jiang Gui[1†]
[1]Dartmouth College [2]University of Notre Dame [3]University of California, Irvine
[4]Georgia Institute of Technology [5]Northeastern University [6]Independent 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., 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.

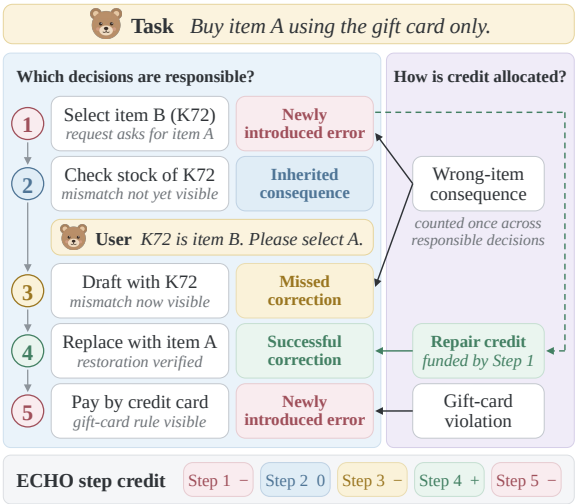

Figure 1: ECHO on an error chain. *Left:* Responsibility depends on decision-time visibility: Step 2 inherits the error before the mismatch is visible, while Step 3 misses a correction after disclosure. *Right:* Credit allocation counts the wrong-item consequence once across Steps 1 and 3 and funds Step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

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** (**E**rror-**CH**ain-Aware **O**ptimization), 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 error 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 5 is charged separately, Step 2 receives none, and Step 4's repair credit is funded by Step 1's penalty.

Algorithm 1 One ECHO update

Require: policy π_θ, frozen judge $\mathcal{J}$, batch of tasks
Sample G trajectories per task; record R_i and $h_{i,t}$
for each trajectory i **do**
  Propose events, links, and restorations from the full trajectory
  for each selected action t ($\leq K$) **do**
    Assess type z from decision-time context only
  end for
  Retain events passing structural checks and g_e
  Allocate and weight events — Eqs. (3)–(6)
  Normalize into budgets b_e^s — Eq. (7)
  Cap self-induced repair credit — Eqs. (8)–(9)
  Aggregate corrections $\delta_{i,t}$ — Eq. (10)
end for
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 "payment = gift card" in Figure 1. The judge outputs responsibility events

$$e = \langle t, f, d, z \rangle, \tag{1}$$

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 and additionally requires 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. Of the $K = 8$ slots, six are assigned to proposed actions and two to non-proposed sentinel actions (§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 strata whose semantic error rate, estimated from

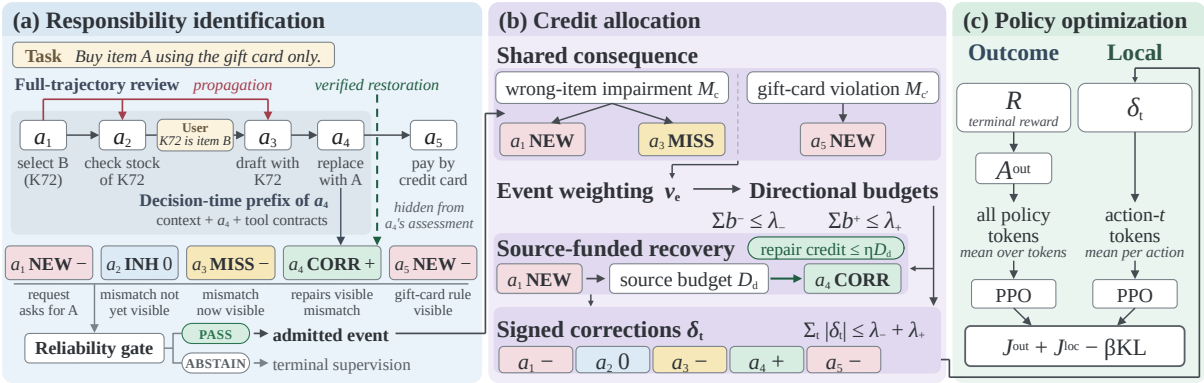

Figure 2: Overview of ECHO. (a) Full-trajectory review proposes candidate events, propagation links, and restoration evidence, while decision-time assessment classifies each selected action as NEW, INH, MISS, or CORR. (b) Shared consequences are counted once across responsible actions, then weighted, 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.

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

$$M_c = \omega_{f(c)} s_c, \qquad \sum_c M_c \le 1, \tag{2}$$

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:

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

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 slots are not reassigned. Equal division is an accounting convention: in Figure 1, the source error at Step 1 and the missed correction at Step 3 each receive half, not all, of the wrong-item consequence.

Event weights. For a negative event,

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

where $\iota_e \in \{0, 1\}$ indicates a certified distinct decision defect and $0 < \kappa \le 1$. 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 $\sum_r z_{rc} \le 1$, and define

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

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,

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

where $\widehat{\ell}_e$ denotes the corresponding negative-event or correction base weight and $0 \le \alpha \le 1$. Since $p_e \le 1$, 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. Event weights are normalized into directional budgets

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

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

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

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

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

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

$$\delta_{i,t} = \sum_{r \in \mathcal{E}_{i,t}^+} \widetilde{b}_r^+ - \sum_{e \in \mathcal{E}_{i,t}^-} b_e^-, \tag{10}$$

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)

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

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)

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

ECHO maximizes

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

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

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

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

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

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>	<u>48.78</u>	<u>33.05</u>	62.25	**63.33**
	Arctic-AWM (Wang et al., 2026b)	<u>38.89</u>	49.55	26.58	22.57	<u>68.24</u>	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 <u>underlined</u> 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

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

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	53.51	36.00	15.79	35.10
	Arctic-AWM (Wang et al., 2026b)	39.88	35.40	37.64	41.23	38.50	23.47	33.45
	Klear-AgentForge (Wang et al., 2025a)	56.70	41.50	49.10	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	40.16
	ECHO (ours)	**58.17**	**43.00**	**50.59**	**54.63**	42.10	**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	38.50	28.50	35.50	37.25
	Arctic-AWM (Wang et al., 2026b)	48.55	36.50	**34.22**	**36.30**	38.89
	ECHO (ours)	**50.50**	**46.58**	32.59	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	44.00	41.38
	Arctic-AWM (Wang et al., 2026b)	53.40	45.00	32.40	44.00	43.70
	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)	54.50	36.50	37.54	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 responsibility 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-

(a) Preserve the requested movie genre

NEW → CLEAN

Task (BFCL-v4): Find Action movies directed by Sujeeth Reddy and featuring Supreet Reddy.

Qwen3-8B The user requires Action movies, whereas dontcare omits the genre criterion. Qwen3 submits Movies_3_FindMovies with genre=dontcare while retaining the director and cast. Dropping the explicit genre filter introduces NEW.

ECHO-8B submits Movies_3_FindMovies with genre=Action, directed_by=Sujeeth Reddy and cast=Supreet Reddy. The query preserves every requested filter and is CLEAN for the genre predicate.

(b) Use the accepted archive destination

MISS → CLEAN

Task (BFCL-v3): Locate analysis_report.csv and move it into the archive directory beside the source file.

Qwen3-4B Inside project, Qwen3 receives an error forbidding destination paths, then repeats mv with destination=../archive/analysis_report.csv. The same rejection establishes MISS despite an available destination=archive alternative. Later attempts fail, and the final filesystem leaves the report outside the archive.

ECHO-4B locates the report, enters project and calls mv with source=analysis_report.csv and destination=archive. The tool confirms the move, and final state preserves its content inside project/archive. This first successful move is CLEAN, with no preceding move failure.

(c) Retry sending after enabling service

MISS → CORR

Task (ToolSandbox): Text Fredrik Thordendal with the album question supplied after clarification.

Qwen3-8B The lookup and user reply supply the phone number and album question. Sending fails because cellular service is disabled. Qwen3 enables service, then claims delivery without retrying, a separate NEW report. Its complete request sequence contains no later send: MISS for the pending text.

ECHO-8B also resolves the contact, asks for content and encounters the cellular-off rejection. After enabling service, it retries send_message_with_phone_number to +12453344098 with the supplied album question. The matching call returns a new message UUID: CORR for tool-level sending.

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

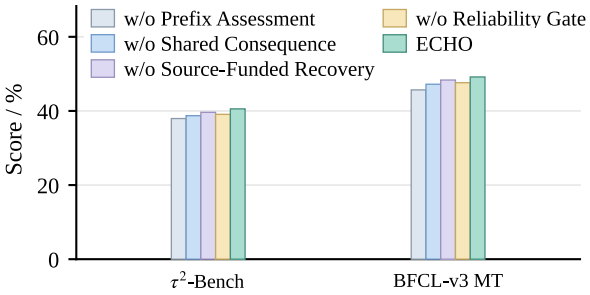

Figure 4: Ablation of ECHO components on τ^2-Bench and BFCL-v3 Multi-Turn at the 8B scale.

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

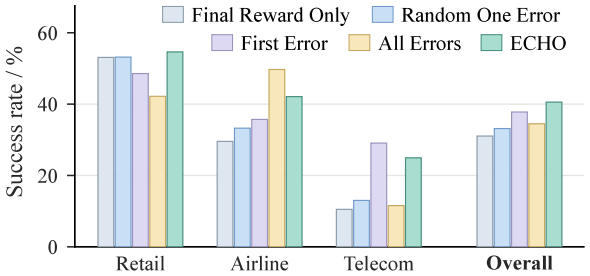

Figure 5: Reward ablation on τ^2-Bench at the 8B scale. Each group compares five supervision configurations, and Overall denotes the mean across the three domains.

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

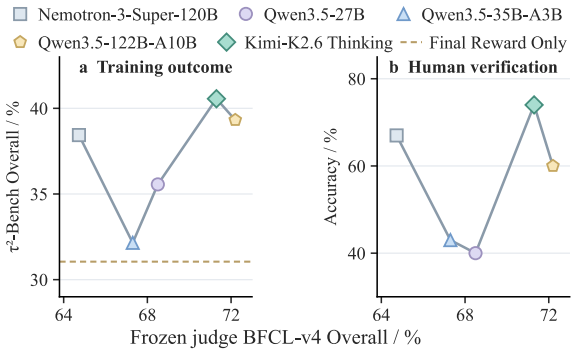

Figure 6: Frozen-judge comparison at the 8B scale. Left: downstream ECHO performance on τ^2-Bench. Right: responsibility identification accuracy on 100 manually verified samples before reliability gating.

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 Poovendran, 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. Toolsandbox: 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. Envscaler: 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, Kefan 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, Ziyan 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 Case Studies **19**
- I.1 Responsibility Events in Base-Model Trajectories 19
- I.2 Paired Comparisons Before and After ECHO Training 20

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 marks omitted records. Each hypothesis identifies its action, policy span, versioned predicate, deviation origin, evidence pointers, and proposed responsibility type. Impact records 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 chronologically; two non-proposed sentinel actions follow a fixed-seed hash ordering. This ordering also fills unused proposal slots until the budget or available actions are exhausted. Each request returns at most four predicate-level records, with omitted records assigned abstention.

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

fixed prompts and decoding. Prefix assessment returns supported defects, legitimate repair intentions, defect-free assessments, or abstention. The assembler combines an accepted repair intention with execution confirmation to establish a successful correction.

Sentinel-discovered defects contribute through the direct term when their prefix evidence passes validation. Unsupported impact and propagation terms receive zero weight, and sentinel repairs require authenticated restoration evidence for positive supervision.

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 support 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 rollouts 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 $p_e \leq 1$,

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

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

$$\sum_t |\delta_{i,t}| \leq \sum_e b_e^- + \sum_r \widetilde{b}_r^+ \leq \lambda_- + \lambda_+. \tag{21}$$

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

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

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

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

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 $P_0 = \sum_r \widetilde{b}_r^+$ 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

$$P_0 \leq \eta \sum_d D_d \leq \eta N_0. \tag{24}$$

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

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

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

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

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

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

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

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

$$g_{\text{loc}}[0] = -\frac{D}{4} + \frac{\eta D H}{8}. \tag{29}$$

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

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

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

$$\begin{aligned} \chi_{i,t,k} &= \mathbf{1}\{\delta_{i,t} > 0, \rho_{i,k} \leq 1+\epsilon\} \\ &\quad + \mathbf{1}\{\delta_{i,t} < 0, \rho_{i,k} \geq 1-\epsilon\}, \end{aligned} \tag{31}$$

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

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

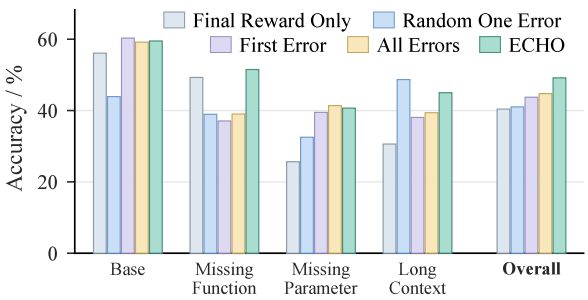

Figure 7: Reward ablation on BFCL-v3 Multi-Turn at 8B. Overall is the mean across Base, Missing Function, Missing Parameter, and Long Context.

The BFCL-v3 Multi-Turn ablation tests supervision under distinct information constraints, as shown in Figure 7. ECHO exceeded Final Reward Only by 8.76 percentage points and All Errors, the strongest ablation, by 4.42 points overall. Its advantage over All Errors was concentrated in Missing Function, where all three error-selection heuristics fell below terminal supervision. ECHO improved this subset while retaining gains over Final Reward Only in Missing Parameter and Long Context.

Subset leaders varied: First Error led Base, All Errors led Missing Parameter, and Random One Error led Long Context. The Long Context advantage of Random One Error accompanied a substantial Base deficit, illustrating the uneven effects of unstructured selection. The strongest heuristic also changed from First Error on τ^2-Bench to All Errors on BFCL-v3 Multi-Turn. ECHO's aggregate lead across both benchmarks supports its balance across decision conditions, alongside the observed advantages of simpler rules on individual subsets.

G Responsibility Event Taxonomy and Benchmark Coverage

This section expands the responsibility types of §3.1–3.2. §G.1 gives the rule that assigns each candidate event at most one type, §G.2 separates responsibility from error content, and §G.3 maps the diagnostics of the six evaluation benchmarks to the evidence each type requires.

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:

$$z = \begin{cases} \textsc{New}, & \mathsf{Def}_e, \\ \textsc{Corr}, & \neg\mathsf{Def}_e \wedge \mathsf{Rep}_e, \\ \textsc{Miss}, & \neg\mathsf{Def}_e \wedge \neg\mathsf{Rep}_e \wedge \mathsf{Opp}_e, \\ \textsc{Inh}, & \neg\mathsf{Def}_e \wedge \neg\mathsf{Rep}_e \wedge \neg\mathsf{Opp}_e \wedge \mathsf{Prov}_e. \end{cases} \tag{33}$$

Because each case negates the indicators above it, the four types are mutually exclusive. Precedence resolves cases in which several indicators are supported; when the evidence for an indicator is insufficient or contradictory, the event abstains (§3.2). Abstention also covers sufficient evidence that fits no type: absent Def_e and Rep_e, a visible discrepancy with no feasible response satisfies neither Opp_e nor Prov_e, and its visibility rules out CLEAN. Visibility means discrepancy evidence actually present in $h_{i,t}$ (§A); the mere absence of information the policy has no reason to expect does not make a deviation visible. An action that touches several predicates yields one event per predicate, so a single transaction can restore one field (CORR) while violating another (NEW); Eq. (10) aggregates their coefficients. A new defect that reopens a repaired predicate starts a separate self-induced origin (§A).

Clean pairs and abstention. Two outcomes carry no responsibility event. 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

Semantic family	Observed diagnostics	Evidence used to assign responsibility
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.	A visible disclosure or clarification obligation and an available response.
Forbidden outcomes and guardrails	Forbidden-state or minefield matches.	The applicable public constraint 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 error.	The incident origin and the agent's response, with execution evidence of recovery.
Aggregate outcome only	Benchmark-reported aggregate similarity deficit without a localized check.	The task's 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; §H) 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 diagnostic occurrences are unlocalized or unmapped and fall outside these families (§H).

this label. An event abstains when the evidence needed to decide an indicator is missing or conflicting, or when a discrepancy on f is visible but no type applies, following selective classification (El-Yaniv et al., 2010); §B calibrates which evidence strata are admitted. Both outcomes retain only terminal supervision, but they record different findings: CLEAN is a supported judgment that the decision raises no responsibility event for f, whereas abstention records unresolved evidence.

Relation to training signals. Under §3.4, admitted NEW and MISS events receive negative budgets. Admitted, execution-verified CORR events receive positive budgets, paid as source-funded credit for self-induced origins, in full for authenticated external origins, and not at all for unknown origins. INH events receive no auxiliary budget and thus no local correction. Post-training differences on INH cases therefore arise from the terminal reward or from local supervision of other events, such as fewer upstream NEW events; §I.2 should be read accordingly.

G.2 Error Content versus Responsibility

An incorrect argument, a policy violation, and a corrupted memory describe the *content* of a failure. Responsibility instead 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 contradiction becomes observable, ignoring it when a feasible response exists constitutes MISS, whereas an execution-confirmed repair constitutes CORR. The same semantic failure can thus receive different types at different stages, and the four types apply across tools, domains, and interaction stages without enumerating every surface error.

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 within the error-and-recovery chain and uses it for training-time credit. ProcessBench locates the earliest erroneous step because later steps can be locally valid while building on an incorrect premise (Zheng et al., 2025), which motivates separating 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.,

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 semantic family and occurrence count.

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 semantic family and occurrence count.

2023; Gou et al., 2024); ECHO instead evaluates such responses as training events, separating an unaddressed visible discrepancy (MISS) from an execution-confirmed repair (CORR). As in step-level process supervision (Lightman et al., 2024), the unit of assessment is an individual decision, but ECHO judges local responsibility from the policy-visible prefix and uses hindsight only to verify propagation and restoration.

G.3 Mapping Benchmark Diagnostics to Responsibility Evidence

The six benchmarks expose 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 provides state dependencies and intermediate milestones (Lu et al., 2025), τ-bench adds domain policies and database-state evaluation (Yao et al., 2024), τ^2-Bench adds user-controlled state changes (Barres et al., 2025), and ACEBench-Agent adds multi-step service interactions (Chen et al., 2025).

Table 4 groups the checker diagnostics recorded on Qwen3 base-model trajectories into eleven semantic families and lists the evidence required to assign a responsibility type within each family. Checker diagnostics describe *where* a trajectory fails a check, whereas responsibility additionally requires the decision-time, provenance, feasibility, and restoration evidence in the third column. Diagnostics that report only an aggregate score cannot be localized; responsibility is then assessed on the task's requirement predicates, which §3.3 extracts from public instructions and tool contracts independently of any outcome. The mapping analyzes the evaluation benchmarks and is not an input 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. Not every benchmark emits its own codes; failures without a benchmark-specific code fall under the generic signals in Table 6. Counts are diagnostic occurrences and are not additive: several diagnostics can occur in one trajectory, and some derived signals are computed from the same underlying record, such as the database match and the database reward component.

Family names follow Table 4. Aggregate similarity deficits reported by the benchmark itself form the aggregate-outcome family, whereas score deficits for which the benchmark reports no diagnostic, together with unmapped terminations, are marked as unmapped or unlocalized and fall outside the eleven families. Semantic families describe diagnostic content only; assigning a responsibility type additionally requires the evidence specified in §G.1.

I Case Studies

This section presents trajectory-level instances of the four responsibility events from evaluation runs of Qwen3 and ECHO at 4B and 8B on the benchmarks in §4.1. Labels apply the decision rule in §G.1 to the displayed evidence, separating information available at decision time 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; all execution claims rely on the displayed tool results or state changes. Because BFCL-v4 also contains the BFCL-v3 multi-turn categories, each panel's benchmark label names the run from which the trajectory was taken. The panels are qualitative examples and do not estimate how often each pattern occurs.

I.1 Responsibility Events in Base-Model Trajectories

Figures 8–11 illustrate each event type in Qwen3 base-model trajectories. NEW events contradict a visible user constraint or tool contract. INH events arise when a fact deleted in an earlier session is sought in a context that contains neither the fact nor its deletion. MISS events repeat an unchanged call after a tool exposes a discrepancy that admits a feasible response. CORR events link a corrective call to a tool result or state snapshot that confirms restoration. Because events are assessed per action and predicate, one trajectory can contain both a missed correction on one requirement and a successful correction 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 task, 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. Where Qwen3 introduces a NEW error, ECHO's corresponding action is CLEAN. Where Qwen3's answer inherits information lost in an earlier session, ECHO retains the fact upstream, even though INH events receive no local correction (§3.4). Where Qwen3 misses a correction, often after completing a prerequisite repair such as logging in or freeing inbox capacity, ECHO either avoids the discrepancy or completes the pending operation, with a tool result or final state confirming restoration.

R1 **An unsupported status filter** NEW

BFCL-v4 `simple_python_177` Qwen3-4B

Visible context User request: "Find all Patent lawsuit cases of Facebook in 2018." Tool default: "defaults to 'all' "

Action `"status": "open"`

Observed evidence Static checker: "false" (single-turn item; no execution is recorded).

> **Responsibility.** Selecting open narrows the requested status scope despite the visible default of all. This creates a new filtering error in the first response.

R2 **Changing an explicit entity name** NEW

BFCL-v4 `simple_python_316` Qwen3-4B

Visible context "Serena Williams"

Action `"player_name": "Serrena Williams"`

Observed evidence Static checker: "false" (single-turn item; no execution is recorded).

> **Responsibility.** The submitted name differs from the user's explicit reference to Serena Williams, introducing a new entity-name error.

R3 **Confusing package weight with item count** NEW

BFCL-v4 `simple_python_370` Qwen3-4B

Visible context User request: "a five pound bag of rice". Quantity definition: "Quantity of each item in the order list."

Action `"items": ["olive oil", "rice"], "quantity": [3, 5]`

Observed evidence Static checker: "false" (single-turn item; no execution is recorded).

> **Responsibility.** The request specifies one bag weighing five pounds, whereas the quantity field counts items. Submitting five therefore introduces a new quantity error.

R4 **Shifting the requested centuries** NEW

BFCL-v4 `parallel_multiple_109` Qwen3-4B

Visible context User request: "France during the 17th century". Tool convention: "for the 1700s, input '18' "

Action `France: "country": "France", "century": 16. England: "country": "England", "century": 17. Italy: "country": "Italy", "century": 18`

Observed evidence Static checker: "false" (single-turn item; no execution is recorded).

> **Responsibility.** For France, the requested 17th century is submitted as 16, contradicting the visible integer convention and introducing a new argument error.

R5 **Dropping part of the tool name** NEW

BFCL-v4 `live_multiple_1035-263-0` Qwen3-4B

Visible context "api_token_api.APITokenApi.get_api_tokens"

Action `"name": "api_token_api.get_api_tokens"`

Observed evidence Static checker: "false" (single-turn item; no execution is recorded).

> **Responsibility.** The submitted tool name omits the class component exposed in the interface. This introduces a new tool-selection error.

Figure 8: Base-model NEW events.

R6 **Searching after a career-history key was erased** INH

BFCL-v4 `memory_kv_60-finance-5` Qwen3-4B

Visible context "Where did I start my career in 2001?" No backup is visible in the current context, and the available memory lookup is a reasonable diagnostic step.

Action `{"name": "archival_memory_key_search", "arguments": {"query": "career start 2001", "k": 5}}`

Observed evidence Earlier stored fact: "2001: Goldman Sachs analyst". Deletion in an earlier session: "{"name": "archival_memory_remove", "arguments": {"key": "career_trajectory"}}". Deletion result: "{"status": "Key removed."}". Current inspection: "{"ranked_results": [[0.0, "landmark_deals"], [0.0, "strategic_insights"], [0.0, "current_transactions"], [0.0, "operational_routine"], [0.0, "risk_management_focus"]]}".

> **Responsibility.** The earlier deletion removes the stored career fact, and the current search returns no matching key. The permitted search inherits this information loss without introducing another defect.

R7 **Searching after a deadline key was erased** INH

BFCL-v4 `memory_kv_142-notetaker-12` Qwen3-8B

Visible context "I may be forgetting, what are the documents I need to urgently submit before Friday?" No backup is visible in the current context, and the available memory lookup is a reasonable diagnostic step.

Action `{"name": "core_memory_list_keys", "arguments": {}}`

Observed evidence Earlier stored fact: "Submit tax documents before Friday". Deletion in an earlier session: "{"name": "core_memory_remove", "arguments": {"key": "tax_documents_deadline"}}". Deletion result: "{"status": "Key removed."}". Current inspection: "{"keys": []}".

> **Responsibility.** The earlier deletion removes the stored deadline, and the current result shows an empty core memory. The permitted inspection inherits this information loss without introducing another defect.

Figure 9: Base-model INH events.

R8 **Retrying cruise activation with the engine stopped** MISS

BFCL-v3 `multi_turn_base_79` Qwen3-4B

Visible context "{"error": "Start the engine before activating the cruise control."}" A feasible response was to address the engine prerequisites or explain the blockage and pause cruise activation.

Action `{"name": "setCruiseControl", "arguments": {"speed": 65, "activate": true, "distanceToNextVehicle": 100}}`

Observed evidence "{"error": "Start the engine before activating the cruise control."}"

> **Responsibility.** The repeated cruise call leaves the exposed engine prerequisite unaddressed despite an available response.

R9 **Repeating a write to a missing file** MISS

BFCL-v3 `multi_turn_base_35` Qwen3-4B

Visible context "{"error": "echo: cannot write to 'diff.txt': No such file"}" The available touch tool permits creating diff.txt before retrying the write.

Action `Function: "name": "echo". Target argument: "file_name": "diff.txt"`

Observed evidence "{"error": "echo: cannot write to 'diff.txt': No such file"}"

> **Responsibility.** The agent repeats the rejected write without first creating the missing file, leaving an actionable prerequisite unresolved.

R10 **Retrying an undefined market-status function** MISS

BFCL-v3 `multi_turn_base_112` Qwen3-4B

Visible context "Error during execution: name 'update_market_status' is not defined" The agent could explain that the requested operation is unavailable and stop the unchanged retries.

Action `{"name": "update_market_status", "arguments": {"current_time_str": "10:30 AM"}}`

Observed evidence "Error during execution: name 'update_market_status' is not defined"

> **Responsibility.** The unchanged retry follows an explicit undefined-function error and leaves the execution limitation unaddressed.

R11 **Repeating an undefined withdrawal operation** MISS

BFCL-v3 `multi_turn_base_121` Qwen3-4B

Visible context "Error during execution: name 'make_transaction' is not defined" The agent could report the execution failure and pause the withdrawal instead of repeating the same call.

Action `{"name": "make_transaction", "arguments": {"account_id": 12345, "xact_type": "withdrawal", "amount": 500}}`

Observed evidence "Error during execution: name 'make_transaction' is not defined"

> **Responsibility.** The agent repeats the unavailable withdrawal operation despite the visible failure and a feasible response to the blockage.

Figure 10: Base-model MISS events.

R12 **Restoring a rejected copy operation** CORR

BFCL-v3 `multi_turn_base_22` Qwen3-4B

Visible context "cp: path not allowed in destination. Provide only a file or directory name."

Action `{"name": "cp", "arguments": {"source": "project_analysis.txt", "destination": "project_archive"}}`

Observed evidence "'project_analysis.txt' copied to 'project_archive/project_analysis.txt' "

> **Responsibility.** Using a local destination name resolves the rejected path argument, and the tool confirms the copy.

R13 **Creating the missing guide file** CORR

BFCL-v3 `multi_turn_base_23` Qwen3-4B

Visible context "echo: cannot write to 'Project_Guide_1.md': No such file"

Action `{"name": "touch", "arguments": {"file_name": "Project_Guide_1.md"}}`

Observed evidence Later state snapshot: "«File: Project_Guide_1.md, Content: Comprehensive guide for the new initiative.»"

> **Responsibility.** The corrected touch call creates the missing file. The later snapshot verifies file existence, while the displayed content results from a subsequent write.

R14 **Restoring directory access** CORR

BFCL-v3 `multi_turn_base_35` Qwen3-4B

Visible context "Only one folder level at a time is supported." After this rejection, the agent first enters projects with a one-level cd.

Action `{"name": "cd", "arguments": {"folder": "deep_folder"}}`

Observed evidence Subsequent read: {"name": "tail", "arguments": {"file_name": "config.py", "lines": 1}}, returning "last_lines": "Initialization of the system Error in module Setup complete Initialization successful Error detected"

> **Responsibility.** Entering deep_folder one level at a time follows the restriction, and the line returned by the subsequent read confirms that access was restored.

R15 **Restoring the brake prerequisite** CORR

BFCL-v3 `multi_turn_base_52` Qwen3-8B

Visible context "Brake pedal needs to be pressed when starting the engine."

Action `{"name": "pressBrakePedal", "arguments": {"pedalPosition": 1}}`

Observed evidence "brakePedalStatus": "pressed"

> **Responsibility.** The brake action restores the prerequisite identified by the failed startup attempt. The receipt confirms a pressed brake state.

R16 **Restoring ticket authentication** CORR

BFCL-v4 `multi_turn_base_152` Qwen3-4B

Visible context "User not authenticated. Please log in to create a ticket."

Action `{"name": "ticket_login", "arguments": {"username": "mthompson", "password": "securePass123"}}`

Observed evidence "success": true

> **Responsibility.** After the authentication refusal, the login call succeeds and restores the authenticated state.

Figure 11: Base-model CORR events.

P1 **Ask for the missing email recipient**

NEW → CLEAN

BFCL-v4 `live_irrelevance_498-144-0`

Qwen3-8B → ECHO-8B

Task. Email Mom a proposal to go out and play if Friday works, using her actual address.

Qwen3-8B before training

The user names Mom without supplying an email address, and no contact lookup result is visible. Qwen3 submits python_send_mail with address=mom@example.com, introducing NEW through an unsupported recipient.

ECHO-8B after training

ECHO asks for Mom's email address without making a tool call. The clarification is CLEAN because it seeks the information required to identify the recipient. The recorded response ends with this question, so delivery remains pending.

Reading. ECHO requests the missing address before submitting an email.

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).

P7 **Prepare the archive before searching** MISS → CLEAN

BFCL-v4 `multi_turn_base_36` Qwen3-4B → ECHO-4B

Task. Create project_summary.txt, copy it into archive as summary_2024.txt, and report lines containing Progress.

Qwen3-4B before training

After missing-file and navigation errors, Qwen3 repeats cd, cp, cd and grep without resolving its location: MISS despite available pwd or ls. Later attempts fail, no archived summary appears, and its final answer invents matching lines, a separate NEW report.

ECHO-4B after training

ECHO creates the file, copies it into archive, enters that directory and renames the copy summary_2024.txt. The grep search for Progress returns no matches, which ECHO reports accurately. Both empty files remain in the final state. The search is CLEAN because the required file was prepared before retrieval.

Reading. ECHO prepares the requested archive file and reports the observed search result.

P8 **Place the order after login** MISS → CORR

ACEBench-Agent `agent_multi_step_3` Qwen3-8B → ECHO-8B

Task. Check Jiutian BBQ's menu and place one order Jack can afford within 120 yuan.

Qwen3-8B before training

The menu offers Grilled Pork Belly for 78 yuan, but ordering fails because Jack is logged out. After successful login, Qwen3 declares completion without retrying. Final orders remain empty: MISS for the pending purchase, with a separate NEW completion claim.

ECHO-8B after training

After the rejected order and successful login, ECHO retries add_food_delivery_order for Jack at Jiutian BBQ with one Grilled Pork Belly. The receipt confirms 78 yuan, and the final state contains one order and a 42-yuan balance. These observations verify CORR.

Reading. ECHO retries the affordable order after authentication and obtains a purchase receipt.

P9 **Send the reply after freeing capacity** MISS → CORR

ACEBench-Agent `agent_multi_step_12` Qwen3-4B → ECHO-4B

Task. Buy Grace's requested cheaper Heytea milk tea and reply, using the user's authorization to delete the latest message if needed.

Qwen3-4B before training

Qwen3 buys the milk tea, but its reply is rejected because the inbox is full. It retrieves latest message ID 4 and deletes it, then stops without resending. The final inbox lacks the confirmation: MISS for the pending reply.

ECHO-4B after training

ECHO also buys the milk tea and encounters the reply rejection. After deleting the authorized latest message, it retries send_message from Helen to Grace with message=Already bought. The tool result and final inbox confirm that exact reply: CORR.

Reading. ECHO resends the rejected purchase reply after the authorized deletion.

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

P10 **Deliver the purchase-price notification** MISS → CORR

ACEBench-Agent `agent_multi_turn_18` Qwen3-8B → ECHO-8B

Task. Order two Hema Fresh Gift Packs for Eve and notify Frank of the total to one decimal place.

Qwen3-8B before training

The 198.0-yuan order succeeds, but the notification fails because the inbox is full. Qwen3 obtains consent to delete message ID 1 and does so, then claims sending without retrying, a separate NEW report. Final inbox evidence shows MISS for the pending notification.

ECHO-8B after training

After the successful purchase and capacity refusal, ECHO asks which message to delete. The user selects the oldest, ID 3. ECHO deletes it and resends Eve's 198.0-yuan notification to Frank. The send result and final inbox confirm CORR.

Reading. ECHO retries the pending notification after an authorized deletion and confirms delivery.

P11 **Write the diff after creating the file** MISS → CORR

BFCL-v3 `multi_turn_base_20` Qwen3-4B → ECHO-4B

Task. Write the observed difference between the first two documents into a new file named file5.txt.

Qwen3-4B before training

The write fails because file5.txt does not exist. Qwen3 corrects an invalid touch argument and creates it, then claims the diff was written without retrying. Final state shows an empty file: MISS for content, plus a separate NEW completion assertion.

ECHO-4B after training

After the missing-file rejection, ECHO creates file5.txt and retries echo with file_name=file5.txt and the observed diff as content. The write succeeds, and the final state contains that exact diff. This confirms CORR for document content beyond creation of an empty destination.

Reading. ECHO retries the failed write after creating its destination and stores the requested diff.

P12 **Recover retrieval after failed search** MISS → CORR

BFCL-v3 `multi_turn_base_30` Qwen3-4B → ECHO-4B

Task. Find the Apollo test-result JSON in the project directory and report its contents.

Qwen3-4B before training

A guessed filename yields no matches. Qwen3 enters project successfully but repeatedly attempts invalid child-directory navigation, then stops without listing or reading the file. The visible listing tool remains usable: MISS for retrieval in an already accessible directory.

ECHO-4B after training

ECHO encounters an unsuccessful search and an invalid ls argument. It then enters project, lists the directory and reads the returned test_results.json. The tool returns the Apollo test JSON, confirming CORR after its own failed search and listing.

Reading. ECHO lists the accessible directory, obtains the filename and retrieves the requested JSON.

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).