

Teach the Magnitude, Not the Direction: Verifier-Anchored Credit Assignment for Multi-Turn Multi-Step LLM Agents

Zechuan Wang^{1,3*}, Siyuan Lu^{1,2,3,4*}, Hongxuan Zhang^{3,5}, Linjian Mo³,
Chenyi Zhuang³, Leilei Gan^{1†},

¹Zhejiang University, ²Shanghai Innovation Institute, ³AWorld Team, Ant Group,
⁴Westlake University, ⁵Nanjing University

Abstract

Reinforcement learning with verifiable rewards (RLVR) directly optimizes verifier-defined outcomes for multi-turn tool-use agents, yet its trajectory-level credit assignment conflates heterogeneous per-turn outcomes into a single reward signal. On-policy distillation provides dense per-token supervision but uses a teacher-defined token-level objective and can be prone to gradient concentration collapse. We introduce CREST, a hierarchical credit assignment framework that anchors each sampled token’s scalar advantage sign to verified rewards while incorporating dense magnitude signals from a privileged self-teacher. CREST resolves credit at two levels: turn-segmented verified advantages address inter-turn dilution, while entropy-gated self-teacher modulation refines intra-turn token contributions. Experiments on BFCL V3 and WildToolBench show that CREST consistently outperforms both RL and distillation baselines across two model scales, with the largest gains on long-trajectory and strict session-level metrics. Our work demonstrates that the teacher’s role in policy optimization can be reduced from determining token-level reinforcement or suppression to modulating update magnitudes, enabling dense credit assignment while keeping verifier rewards as the training objective.

1 Introduction

Large language models (Hurst et al., 2024; Comanici et al., 2025; Qwen et al., 2025; Liu et al., 2024) are evolving from pure question-answering systems into autonomous agents capable of reasoning, acting, and interacting with diverse environments (Merrill et al., 2026; Jimenez et al., 2024; Xie et al., 2024; Liu et al., 2026; Froger et al., 2025; Towers et al., 2025). These scenarios share a

common structure: agents must handle **multi-turn** sessions (sequences of user requests) where each turn requires **multi-step** execution (chains of actions and observations) (Shridhar et al., 2021; Wei et al., 2025a; Xie et al., 2024; OpenClaw, 2026).

Reinforcement learning (Schulman et al., 2017; Ahmadian et al., 2024), particularly RL with verifiable rewards (RLVR) (Shao et al., 2024; Qian et al., 2025), is now the dominant post-training paradigm for tool-use agents (Liu et al., 2025; Zhang et al., 2025a; Yu et al., 2025), directly optimizing verifier-defined outcome rewards. Yet standard objectives broadcast a single trajectory-level reward across all tokens. Within a single turn, where all actions serve one query, this is merely noisy (Lu et al., 2026a). In **multi-turn** sessions (Li et al., 2023; Patil et al., 2025; Yu et al., 2026), however, turns carry independent outcomes yet share one reward signal, making credit assignment ill-posed: on WildToolBench, no model exceeds 15% session accuracy, and performance degrades sharply with turn count (Laban et al., 2025).

On-policy distillation (OPD) (Lu and Thinking Machines, 2025; Li et al., 2026; Song and Zheng, 2026) provides a natural credit assignment mechanism, where the teacher inherently scores each token in the student’s trajectory and distinguishes which tokens are preferable from the teacher’s perspective. However, this requires a same-family, tokenizer-matched teacher (Fu et al., 2026), which is a rare constraint satisfied in practice. On-policy self-distillation (OPSD) (Zhao et al., 2026; Hübotter et al., 2026; Shenfeld et al., 2026a; Penalzoa et al., 2026) removes this dependency by using the student itself as self-teacher, conditioned on privileged information such as ground truth or successful demonstrations from rollout group peers. Yet OPSD is brittle and restricts exploration. First, Zhao et al. (2026); Brown (2026) found that without per-token KL clipping, OPSD collapses within ~100 training steps due to gradient concentra-

*Equal contributions. Work was done during Zechuan and Siyuan’s internship in Ant Group.

†Correspondence: leileigan@zju.edu.cn

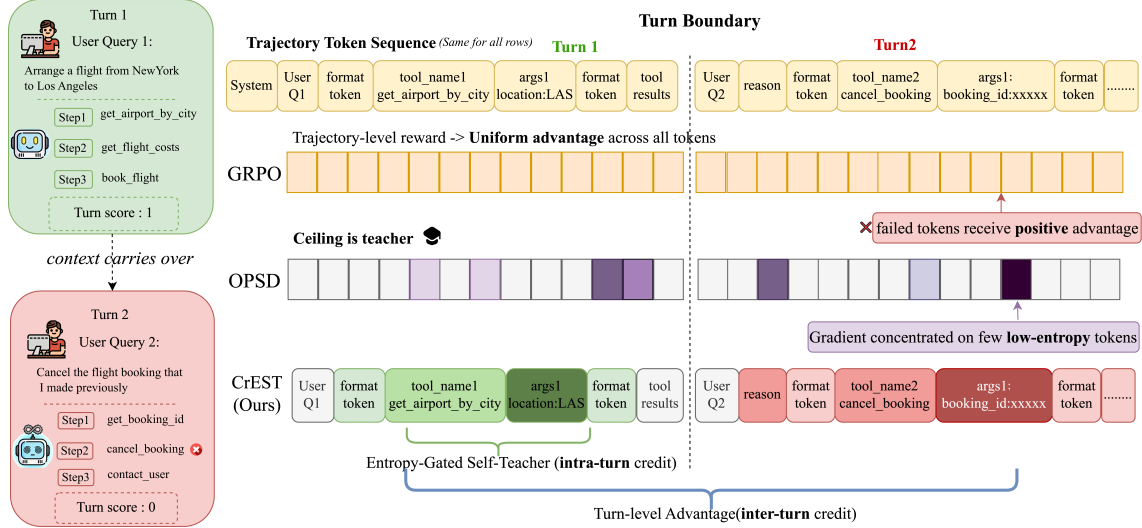


Figure 1: Comparison of credit assignment strategies on a 2-turn trajectory (Turn 1 succeeds, Turn 2 fails). **GRPO** broadcasts a single trajectory-level reward uniformly, erroneously reinforcing failed-turn tokens. **OPSD** provides a dense but teacher-defined token-level signal and concentrates gradient on low-entropy format tokens. **CREST (Ours)** combines turn-level advantages (inter-turn credit) with entropy-gated self-teacher modulation (intra-turn credit), correctly assigning negative advantage to Turn 2 while focusing gradient mass on high-uncertainty content tokens in Turn 1.

tion on a few tokens. Second, OPSD optimizes a teacher-defined token-level target (Kim et al., 2026), making its training behavior dependent on the privileged teacher distribution.

This raises a central question: *Can we design a training approach that retains verified rewards as the optimization target while incorporating dense token-level signals to address hierarchical credit assignment in multi-turn multi-step LLM agent training?*

We propose CREST (Hierarchical Credit Assignment via Entropy-Gated Self-Teacher), which addresses both levels through a unified policy gradient with structured per-token advantages (Figure 1). At the turn level, CREST uses turn-segmented verifier rewards to prevent rewards from being diluted across long multi-turn trajectories. At the token level, to prevent gradient concentration on a few tokens, CREST uses an entropy-gated self-teacher to modulate token-level contributions within each turn. Additionally, CREST uses verified rewards to determine the sign of each sampled token’s scalar advantage while reducing the self-teacher’s role to magnitude modulation, enabling dense credit assignment while keeping verifier rewards as the training objective and avoiding the need for an external same-family teacher. The two levels of credit assignment are complementary—neither alone recovers the full performance.

Our main contributions are:

1. We identify and formalize the two-level credit assignment problem in multi-turn multi-step agentic RL, and provide a gradient-geometric analysis explaining why existing dense-signal methods fail in this setting (Section A).
2. We propose CREST, a hierarchical credit assignment framework combining turn-segmented advantage with entropy-gated self-teacher token reweighting. CREST is verifier-anchored at the token level by construction and tunes only one teacher-influence hyperparameter (λ) beyond standard GRPO (Section 3).
3. We report that CREST achieves 52.0% average accuracy on BFCL V3 (Qwen3-4B) and 9.38% session accuracy on WildToolBench (Qwen3-8B), obtaining the highest observed aggregate scores among the compared methods (Section 4).

2 Related Work

Tool use for LLM-based agents. LLM tool-use research spans a complexity spectrum, from reasoning before function-calling (Zhang et al., 2025a; Qian et al., 2025) to tool-integrated reasoning (TIR) for fulfilling user’s query (Li et al., 2025; Jin et al., 2025; Feng et al., 2025; Zheng et al., 2025). Most relevant to our work, multi-turn multi-step benchmarks (Zhang et al., 2025b; Yu et al., 2026; Patil

et al., 2025) evaluate agents across sessions of multiple queries, each requiring its own execution chain. These settings expose two challenges absent in single-turn work: cross-turn state consistency and error propagation across turns. Empirically, agent performance degrades substantially with turn count; Yu et al. (2026) report session accuracy below 15% across 57 LLMs, motivating training-time methods that respect the turn structure of multi-turn sessions.

Agentic Reinforcement Learning. RL-based post-training has become the standard approach for aligning LLM agents with task objectives (Shao et al., 2024; Yu et al., 2025; Wang et al., 2025). Some methods focus on reward design: Lu et al. (2026a) decompose trajectory rewards into fine-grained per-step process scores combining state correctness and execution accuracy, while Qian et al. (2025) and Zhang et al. (2025a) ground rewards directly in environment feedback from tool execution. Despite these advances, most existing methods compute a single advantage per trajectory (or per-step within a single turn), leaving the inter-turn credit assignment problem unaddressed—when turns within the same session have heterogeneous outcomes, trajectory-level or step-level averaging still mislabels tokens in failed turns. MT-GRPO (Wei et al., 2025b) attempts per-agent-step advantage computation, yet still lacks intra-step token-level credit assignment, treating all tokens within each step identically.

On-policy distillation. On-policy distillation (OPD) (Lu and Thinking Machines, 2025) replaces RL’s sparse reward with dense per-token reverse KL against a same-family teacher, providing token-level credit assignment throughout the trajectory (Li et al., 2026). OPD converges faster than RL at moderate compute budgets, but it optimizes a teacher-defined token-level target and requires tokenizer-matched, recipe-matched teachers (Brown, 2026). When such teachers are unavailable, self-distillation methods use the student itself conditioned on privileged information like expert demonstrations (Shenfeld et al., 2026b), feedback (Hübotter et al., 2026) and ground-truth answers (Zhao et al., 2026). More recently, hybrid methods attempt to combine verifier-derived scalar advantage signs with distillation’s dense signal: SDAR (Lu et al., 2026b) gates self-distillation as an auxiliary loss alongside RL, and RLSD (Yang et al., 2026) uses self-distillation for update magni-

Table 1: Design space of hierarchical credit assignment. Methods differ in how they allocate credit across turns ($A_{[t]}^{\text{turn}}$) and tokens (ϕ_t). MT-GRPO computes per-turn advantages but propagates discounted future-outcome credit to earlier turns (*cumulative*), whereas CREST computes each turn’s advantage from that turn’s reward only (*independent*). OPD and OPSD share the same slot ($\phi_t \propto \Delta_t$ with no verified-reward base, hence teacher-bounded); they differ only in teacher source (external same-family vs. self with privileged context).

Method	$A_{[t]}^{\text{turn}}$	ϕ_t
GRPO	trajectory-level	1
MT-GRPO	per-turn (cumulative)	1
OPD / OPSD	—	$\propto \Delta_t$
CREST (Ours)	per-turn (independent)	$1 + \lambda_t^{\text{eff}}(w_t - 1)$

tudes while relying on verifiable rewards for token-level reinforcement or suppression. However, these hybrids either lack principled control over gradient concentration (Zhao et al., 2026; Brown, 2026) or do not address the inter-turn credit assignment structure unique to multi-turn settings.

3 Method

Multi-turn multi-step training faces a *hierarchical* credit assignment problem: at the coarse level, trajectories mix successes and failures across turns; at the fine level, individual turns conflate high-value decisions with low-entropy formatting. CREST addresses both levels through a unified policy gradient framework with structured per-token advantages (Equation (1)):

- **Inter-turn (coarse):** Turn-segmented verified-reward advantages (Section 3.2) isolate per-turn outcomes, preventing dilution from trajectory-level averaging.
- **Intra-turn (fine):** Entropy-gated self-teacher signals (Section 3.3) refine per-token contributions, concentrating gradient mass on high-uncertainty content tokens.

The two levels are *complementary*: without inter-turn segmentation, no amount of intra-turn refinement can rescue tokens in failed turns; without intra-turn reweighting, pivot decisions in successful turns remain indistinguishable from formatting tokens. Critically, verified reward determines the sign of each sampled token’s scalar advantage while the self-teacher only modulates its magnitude, yielding verifier-anchored token reweighting (Section A).

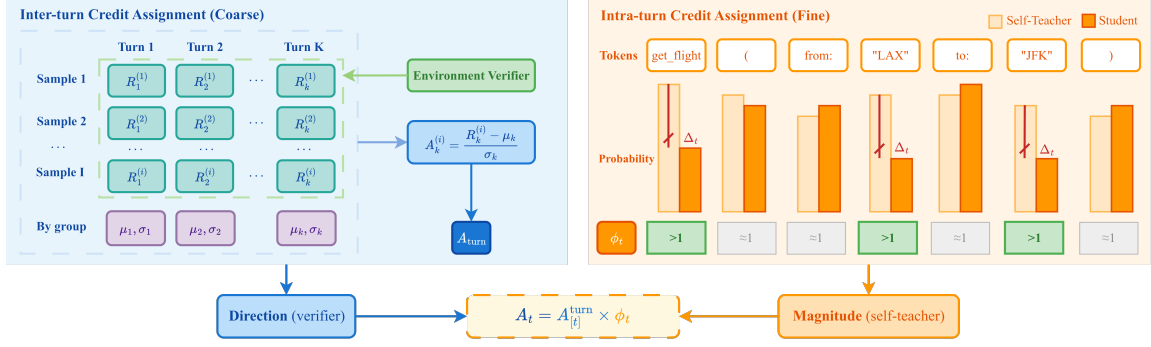


Figure 2: Overview of CREST. **Left (Inter-turn)**: A group of G rollouts is scored by the environment verifier per turn; group-relative advantages A_{turn} are computed independently for each turn, providing the verifier-derived scalar sign. **Right (Intra-turn)**: Within each turn, a privileged self-teacher (conditioned on ground truth) provides per-token divergence Δ_t ; an entropy gate concentrates modulation on high-uncertainty content tokens (e.g., `get_flight`, `"JFK"`) while suppressing format tokens, yielding the *magnitude* factor ϕ_t . **Bottom**: The final per-token advantage $\mathcal{A}_t = A_{[t]}^{\text{turn}} \times \phi_t$ combines a verifier-derived scalar sign with teacher-modulated magnitude.

3.1 Unified Objective

CREST is a standard on-policy policy gradient with a *structured per-token advantage*. Given a prompt x and an on-policy rollout $y \sim \pi_\theta(\cdot | x)$, the objective is:

$$J(\theta) = \mathbb{E} \left[\sum_t \mathcal{A}_t \cdot \log \pi_\theta(y_t | y_{<t}, x) \right] \quad (1)$$

where the per-token advantage $\mathcal{A}_t$ exposes the hierarchical structure:

$$\mathcal{A}_t = \underbrace{A_{[t]}^{\text{turn}}}_{\text{inter-turn: which turn?}} \cdot \underbrace{\phi_t}_{\text{intra-turn: which token?}} \quad (2)$$

$A_{[t]}^{\text{turn}}$ is a turn-segmented verified-reward advantage (Section 3.2) that resolves *which turn* contributed to success or failure. $\phi_t \in [1, 1+\lambda\epsilon]$ is a magnitude modulation factor (Section 3.3) that refines *which tokens within that turn* deserve stronger updates, derived from a privileged self-teacher gated by student uncertainty.

The factorization in Equation (2) reveals why both levels are necessary: $A_{[t]}^{\text{turn}}$ provides the correct *sign* (reinforce or suppress) for each turn’s tokens, but treats all tokens within a turn uniformly; ϕ_t refines the *magnitude* of each token’s contribution, but cannot change sign (proof in Section A). Together, they anchor each sampled token’s scalar advantage sign to verified rewards while allowing token-level magnitudes to benefit from dense teacher signals. Table 1 shows how existing methods occupy restricted subspaces of this design.

3.2 Turn-Level Credit Assignment

Instead of computing a single trajectory-level reward averaged across turns, we compute group-relative advantages *within each turn independently* (Figure 2, left). For a trajectory containing turns $\{1, \dots, K\}$, each turn k receives its own verified reward R_k from the environment. Within a GRPO group of G rollouts, the advantage for turn k in rollout i is:

$$A_k^{(i)} = \frac{R_k^{(i)} - \text{mean}(\{R_k^{(j)}\}_{j=1}^G)}{\text{std}(\{R_k^{(j)}\}_{j=1}^G) + \epsilon} \quad (3)$$

All tokens within turn k of rollout i share the advantage $A_k^{(i)}$. This ensures that a failed turn receives negative advantage regardless of whether other turns in the same trajectory succeeded.

3.3 Token-Level Credit: Self-Teacher Signal

The modulation factor ϕ_t uses a privileged self-teacher (the same model conditioned on ground-truth context) to refine per-token gradient magnitude within each turn (Figure 2, right):

$$\phi_t = 1 + \lambda_t^{\text{eff}} \cdot (w_t - 1) \quad (4)$$

When $\lambda_t^{\text{eff}} = 0$, $\phi_t = 1$ (no teacher influence); when $\lambda_t^{\text{eff}} > 0$ and $w_t > 1$, $\phi_t > 1$ (teacher amplifies the gradient). We now define the two components: the raw teacher signal ($\Delta_t \rightarrow w_t$) and the gating mechanisms that control it (Section 3.4).

Teacher–student divergence. For each response token t , let $\pi_T(\cdot | h_t^T)$ denote the self-teacher—the same model conditioned on ground truth as

privileged context, following Zhao et al. (2026)—and $\pi_\theta(\cdot | h_t)$ the student:

$$\Delta_t = \frac{\log \pi_T(y_t | h_t^T) - \log \pi_\theta(y_t | h_t)}{\tau} \quad (5)$$

where τ is a temperature that smooths the divergence signal.

Token weight. The raw teacher preference is converted into a bounded multiplicative weight:

$$w_t = \text{clip}\left(\exp(\text{sign}(A_{[t]}^{\text{turn}}) \cdot \Delta_t), 1-\epsilon, 1+\epsilon\right) \quad (6)$$

The $\text{sign}(A_{[t]}^{\text{turn}}) \cdot \Delta_t$ construction aligns teacher preference with the verifier-derived scalar sign: for positive-advantage tokens, teacher agreement increases w_t ; for negative-advantage tokens, teacher agreement with the “this was bad” signal also increases w_t . This guarantees **token-wise sign preservation** (P1): $\text{sign}(\mathcal{A}_t) = \text{sign}(A_{[t]}^{\text{turn}})$ for all t (proof in Section A).

3.4 Stabilization via Selective Gating

The self-teacher provides dense per-token signals, but unconstrained use leads to concentration collapse (Brown, 2026): gradient mass flows to a small set of tokens where teacher–student divergence is largest, often low-entropy formatting rather than high-value decisions. We stabilize the intra-turn reweighting through a composed gating function λ_t^{eff} (Equation (10)) that controls *where* (agreement with verifier) and *how much* (student uncertainty) the teacher signal is applied.

Direction gate.

$$g_t^{\text{dir}} = 1 \left[\text{sign}(A_{[t]}^{\text{turn}}) \cdot \Delta_t > 0 \right] \quad (7)$$

When teacher and verifier disagree on the token-level reinforcement or suppression sign, $g_t^{\text{dir}} = 0$ and the teacher signal is fully suppressed. This prevents the teacher signal from flipping the verifier-derived scalar advantage sign of any sampled token.

Entropy gate. To prevent gradient concentration on low-entropy format tokens, we modulate the reweighting strength by the student’s token-level uncertainty, using surprisal $u_t = -\log \pi_\theta(y_t | h_t)$ as a lightweight proxy:

$$g_t^{\text{ent}} = \sigma\left(\frac{u_t - \mathbb{E}[u]}{\text{std}(u) + \epsilon}\right) \quad (8)$$

$$m_t^{\text{ent}} = 1 + \rho(2g_t^{\text{ent}} - 1) \quad (9)$$

where σ is the sigmoid function and statistics are computed over response tokens. The Z-score normalization centers surprisal around the mean and scales by standard deviation, mapping to $[0, 1]$ via sigmoid. For low-uncertainty format tokens, $m_t^{\text{ent}} < 1$ reduces modulation strength; for high-uncertainty content tokens, $m_t^{\text{ent}} > 1$ increases it. The range $[1-\rho, 1+\rho]$ adjusts the strength of selective amplification without suppressing any token’s base verifier-derived contribution.

Composed gating. The direction and entropy gates compose into a single effective gating function:

$$\lambda_t^{\text{eff}} = \text{clip}\left(\lambda \cdot \underbrace{g_t^{\text{dir}} \cdot m_t^{\text{ent}}}_{\text{safety \& focus}}, 0, \lambda\right) \quad (10)$$

This yields $\phi_t = 1 + \lambda_t^{\text{eff}}(w_t - 1) \in [1, 1+\lambda\epsilon]$, ensuring **three formal properties** (Section A): (P1) direction preservation ($\text{sign}(\mathcal{A}_t) = \text{sign}(A_{[t]}^{\text{turn}})$ always); (P2) the absolute perturbation bound $\|\sum_t (\phi_t - 1)g_t\| \leq \lambda\epsilon \sum_t \|g_t\|$, where $g_t = A_{[t]}^{\text{turn}} \nabla_\theta \log \pi_\theta(y_t | h_t)$; and (P3) sign-consistent amplification ($\phi_t \geq 1$ when active). With $\lambda=0.3$ and $\epsilon=0.28$, the 8.4% value bounds only the change to each token’s advantage coefficient, not the relative bias of the aggregate gradient.

λ is the **only tunable hyperparameter** controlling teacher influence strength. The remaining values— $\tau=2.0$ (divergence temperature), $\epsilon=0.28$ (clip bound, inherited from OPSD stabilization), $\rho=0.5$ (entropy redistribution range)—are fixed constants requiring no tuning (ablation in Section 4).

The complete training procedure, consolidating all equations into executable pseudocode, is provided in Section B.

4 Experiments

4.1 Experimental Setup

Benchmarks. We conduct experiments on two multi-turn multi-step tool-use benchmarks: (1) **Berkeley Function-Calling Leaderboard (BFCL) V3** (Patil et al., 2025). Under our processed protocol, we use 100 fixed IDs from the multi-turn *Base* split for training and 400 non-overlapping examples for evaluation: 100 examples each from *Base*, *Missing Functions*, *Missing Parameters*, and *Long-Context*. (2) **WildTool-Bench** (Yu et al., 2026) contains 256 multi-turn

Table 2: Main results on BFCL V3 multi-turn and WildToolBench benchmarks. Best results per model in **bold**, second-best underlined. Methods are grouped by training paradigm: RL-based (top), distillation-based (middle), and our method (bottom).

Method	BFCL V3 Multi-Turn (%)					WildToolBench (%)	
	Average	Base	Miss Func	Miss Param	Long Context	Task Acc.	Session Acc.
<i>Qwen3-4B-Instruct</i>	22.12	26.5	21.0	15.5	25.5	37.89	3.13
+ GRPO	43.63	53.5	42.5	31.5	47.0	40.23	4.69
+ MT-GRPO	<u>49.25</u>	<u>63.0</u>	46.0	35.0	<u>53.0</u>	43.36	<u>6.25</u>
+ EnvTuning	47.25	60.0	<u>47.0</u>	32.0	50.0	<u>44.92</u>	4.69
+ OPD	44.50	52.0	43.0	<u>38.0</u>	45.0	42.58	<u>6.25</u>
+ OPSD	38.75	46.0	41.0	26.0	42.0	38.88	4.69
+ CREST	52.00 (+29.88)	67.0 (+40.5)	48.0 (+27.0)	38.0 (+22.5)	60.0 (+34.5)	48.44 (+10.55)	7.03 (+3.90)
<i>Qwen3-8B</i>	33.38	41.5	38.5	27.0	26.5	43.75	4.69
+ GRPO	43.25	53.0	46.0	36.0	38.0	45.43	5.47
+ MT-GRPO	44.00	<u>57.0</u>	45.0	36.0	38.0	49.61	7.03
+ EnvTuning	<u>46.00</u>	54.0	45.0	<u>40.0</u>	<u>45.0</u>	49.02	<u>7.81</u>
+ OPD	44.75	50.0	52.0	39.0	38.0	45.25	3.91
+ OPSD	41.75	48.0	43.0	39.0	37.0	43.95	3.91
+ CREST	50.00 (+16.62)	60.0 (+18.5)	<u>51.0 (+12.5)</u>	42.0 (+15.0)	47.0 (+20.5)	52.34 (+8.59)	9.38 (+4.69)

sessions designed to capture real-world interaction challenges, including compositional tasks, implicit intent through pronoun reference or ellipsis, and task switching between goal-directed queries and casual chat. The benchmark evaluates both action correctness and parameter accuracy. We allocate 128 sessions for training and reserve the remaining ones for evaluation.

The turn index is semantically aligned within each rollout group: all rollouts use the same fixed session and the same ordered list of user turns, so turn k denotes the same benchmark request across the group. This assumption is specific to these benchmarks and does not directly extend to variable turn structures, terminal-only rewards, or strongly coupled turns without local verification. BFCL rewards compare post-execution environment states with annotated per-turn target executions, including result-coverage checks for read-only tools; WildToolBench rewards path-match calls against human-verified valid paths and score action correctness, parameter correctness, and path completion.

Base models. We evaluate on two model scales from the Qwen3 family (Yang et al., 2025): *Qwen3-4B-Instruct*,¹ an instruction-tuned model, and *Qwen3-8B*, a thinking model. Both have native tool-calling support and represent practical deployment sizes where post-training efficiency matters most.

Baselines. We compare against four training methods spanning RL-based and distillation-based

paradigms:

GRPO (Shao et al., 2024) applies standard Group Relative Policy Optimization with binary trajectory-level reward (1 if all turns succeed, 0 otherwise). Advantages are averaged across all tokens in the trajectory regardless of per-turn outcomes.

MT-GRPO (Wei et al., 2025b) computes advantages at the per-agent-step granularity, providing finer-grained credit assignment than trajectory-level methods. However, within each step, all tokens still share the same advantage, lacking token-level differentiation.

EnvTuning (Lu et al., 2026a) uses fine-grained process rewards that combine per-turn state correctness and execution accuracy, aggregating per-turn scores into a trajectory-level reward via weighted averaging. Despite richer reward signals, it still computes a single trajectory-level advantage.

OPD (Lu and Thinking Machines, 2025) performs on-policy distillation with a same-family, same-variant teacher, providing dense per-token supervision via reverse KL. To satisfy the tokenizer-matched constraint, we pair each student with a teacher of the same model variant: *Qwen3-4B-Instruct* uses *Qwen3-235B-A22B-Instruct*² as teacher, while *Qwen3-8B* uses *Qwen3-32B*. OPD therefore optimizes a teacher-defined token-level target.

OPSD (Zhao et al., 2026) uses on-policy self-distillation where the student itself, conditioned on ground-truth context, serves as the privileged teacher. This removes the external teacher dependency but risks training instability from gradient

¹<https://huggingface.co/Qwen/Qwen3-4B-Instruct-2507>

²<https://huggingface.co/Qwen/Qwen3-235B-A22B-Instruct-2507>

concentration.

Implementation details. All methods are trained with the same on-policy rollout infrastructure, group size $G=16$, and learning rate schedule. For CREST, we set $\lambda=0.3$, $\epsilon=0.28$, $\tau=2.0$, and $\rho=0.5$; only λ is tuned, while the other values are fixed design choices across all experiments. Each method uses a single training seed. We evaluate each trained policy with three decodes at temperature 10^{-6} to measure decoding consistency. Full training configuration and OPSD stabilization details are provided in Section C, with training cost and throughput reported in Table 6.

4.2 Main Results

Table 2 reports BFCL V3 and WildToolBench performance for both model scales across all baselines. CREST obtains higher point estimates than all baselines on average and achieves the best or tied-best result on nearly all individual splits across both benchmarks and model scales.

RL outperforms distillation Two patterns stand out in Table 2. First, RL-based methods (GRPO, MT-GRPO, EnvTuning) systematically outperform distillation-based methods (OPD, OPSD) at both model scales. The clearest signal is OPSD: on Qwen3-4B BFCL Avg it scores 38.75%, falling *below* GRPO (43.63%) despite using a privileged ground-truth-conditioned teacher, illustrating an empirical limitation of the teacher-defined objective in this setting. Second, within the RL family, finer-grained credit assignment helps: on Qwen3-4B BFCL Avg, MT-GRPO (49.25%) and EnvTuning (47.25%) both improve clearly over trajectory-level GRPO (43.63%), yet every token within a step still shares a single advantage. The remaining headroom for dense per-token supervision, on top of verifier-determined directions, is exactly the room CREST is designed to fill.

CREST closes both gaps simultaneously. CREST reaches 52.00% on Qwen3-4B-Instruct and 50.00% on Qwen3-8B, surpassing the strongest RL baseline at each scale and achieving the best or tied-best result in 13 of 14 cells. OPD is best on Qwen3-8B *Missing Functions*, while CrEST ties OPD on Qwen3-4B *Missing Parameters*. The margins are largest on the splits that stress hierarchical credit assignment: on BFCL *Long Context* (the longest-trajectory split), CREST improves over the strongest baseline by +7.0 points at

4B and +2.0 points at 8B; on WildToolBench Session Accuracy (the strictest end-to-end multi-turn metric), by +0.78 points at 4B and +1.57 points at 8B. These are exactly the regimes where credit dilution is worst, validating that turn-segmented advantages and entropy-gated modulation address complementary failure modes.

4.3 Training Dynamics Analysis

Figure 3a compares training rollout accuracy on BFCL V3 (Qwen3-4B). CREST reaches 0.60 accuracy within ~ 20 steps and converges to ~ 0.70 , while GRPO rises slowly to ~ 0.57 by step 160. OPSD plateaus at ~ 0.49 and declines thereafter, consistent with the observed limitations of its teacher-defined optimization target discussed by Lu and Thinking Machines (2025) and Brown (2026). CREST exceeds this observed plateau by step 20 and continues improving, showing that the direction gate can combine teacher-informed magnitude modulation with verifier-derived token-wise signs in this setting.

Figure 3b provides a mechanistic explanation for the performance gap. We measure the fraction of total gradient magnitude captured by the top- $p\%$ of tokens (ranked by per-token advantage magnitude) for each method. OPSD exhibits extreme concentration: the top-1% of tokens account for $\sim 42\%$ of the total gradient signal, and the top-5% account for $\sim 77\%$. This matches the gradient-geometric analysis of Brown (2026): a handful of pivot tokens dominate the update, causing instability. GRPO, by contrast, is far more diffuse (top-10% $\approx 31\%$), but at the cost of treating all tokens equally. CREST occupies the desirable middle ground: more concentrated than GRPO (top-10% $\approx 57\%$), confirming that the self-teacher signal provides meaningful token differentiation, but far less extreme than OPSD, confirming that the entropy gate prevents concentration collapse.

4.4 Ablation Studies

Table 3: Hierarchical ablation on BFCL V3 (Qwen3-4B-Instruct). Neither inter-turn nor intra-turn credit assignment alone recovers the full CREST performance.

Method	Avg	Base	M.Func	M.Param	L.Ctxt
GRPO (baseline)	43.63	53.5	42.5	31.5	47.0
+ Inter-turn only	47.88	62.0	44.0	35.5	50.0
+ Intra-turn only	48.75	61.0	50.0	32.0	52.0
+ Both (CREST)	52.00	67.0	48.0	38.0	60.0

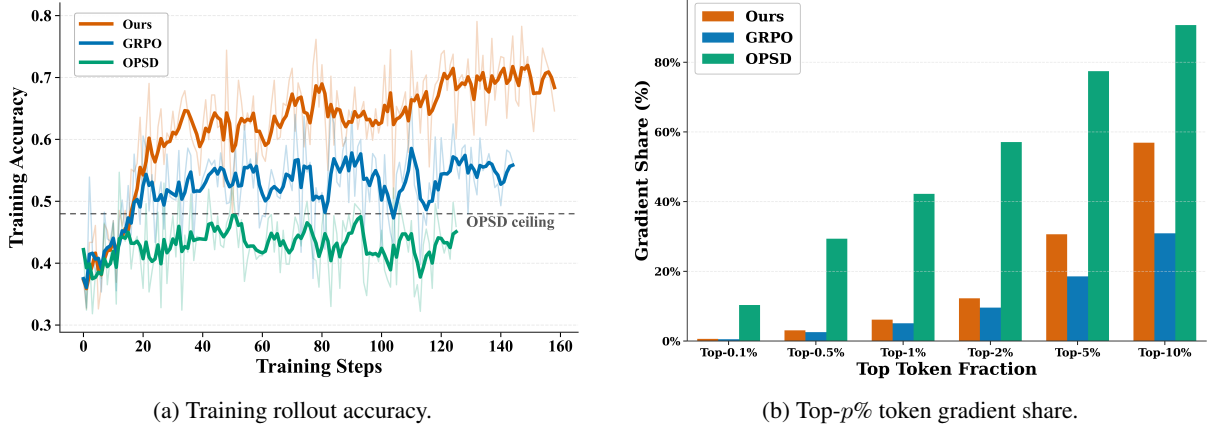


Figure 3: Training dynamics on BFCL V3 (Qwen3-4B). (a) CREST converges faster and reaches higher final rollout accuracy than GRPO, exceeding the observed OPSD plateau (dashed) by step 20 while OPSD subsequently declines. (b) Top- $p\%$ token gradient share: OPSD concentrates $>77\%$ of gradient on the top-5% tokens, so its signal is dominated by a few tokens; CREST’s entropy gate selectively amplifies high-uncertainty content tokens while limiting amplification on low-uncertainty tokens.

Effect of hierarchical credit decomposition. Table 3 isolates the contribution of each credit assignment level. The *Intra-turn only* variant retains the trajectory-level GRPO group-relative advantage A_i^{traj} for every token and applies only the teacher modulation, $\mathcal{A}_t = A_i^{\text{traj}} \phi_t$; it uses neither turn segmentation nor process rewards. Inter-turn segmentation alone improves average accuracy over GRPO (43.63 $\rightarrow$ 47.88), with the largest gain on *Base* (+8.5) where cross-turn reward dilution is most severe. Intra-turn modulation alone yields +5.12 points, with particular strength on *Miss Func* (+7.5) where distinguishing content tokens from format tokens matters most. Combining both levels achieves +8.37 points (43.63 $\rightarrow$ 52.00), with *Long Context* showing the strongest synergy. Neither component alone approaches the full model, confirming that the two levels address complementary failure modes: turn segmentation provides correct per-turn reward direction, while entropy gating selectively amplifies high-value tokens within each turn.

Table 4: Gating ablation on BFCL V3 (Qwen3-4B-Instruct). Removing the direction gate caps performance at the teacher-bounded ceiling; removing the entropy gate causes gradient concentration on format tokens; removing both leads to training instability.

Variant	Avg	Base	M.Func	M.Param	L.Ctxt
CREST	52.00	67.0	48.0	38.0	60.0
+ w/o Dir. gate	46.75	60.0	47.0	28.0	52.0
+ w/o Ent. gate	46.25	59.0	50.0	27.0	49.0
+ w/o Both gates	43.50	53.0	43.0	27.0	51.0

Both gates are necessary. Table 4 ablates the direction gate and entropy gate individually and jointly. Removing the direction gate drops average accuracy from 52.00% to 46.75% (−5.25), with *Miss Param* suffering the largest decline (38.0 $\rightarrow$ 28.0), consistent with the teacher overriding verifier signals on ambiguous parameter tokens. Removing the entropy gate yields a similar degradation (52.00 $\rightarrow$ 46.25), with *Long Context* dropping from 60.0 to 49.0, where long sequences amplify gradient concentration on format tokens. Removing both gates (43.50%) falls to GRPO-level performance, confirming that uncontrolled self-distillation provides no benefit over standard RL. A sensitivity analysis of λ is provided in Section C.3.

5 Conclusion

We presented CREST, a hierarchical credit assignment framework for multi-turn multi-step agent training. By decomposing per-token advantages into turn-segmented verified rewards and entropy-gated self-teacher modulation, CREST resolves the two-level credit assignment problem that afflicts standard RL in multi-turn settings: inter-turn reward dilution and intra-turn token uniformity. Experiments on BFCL V3 and WildToolBench show that CREST achieves the strongest aggregate BFCL results and higher WildToolBench scores in the evaluated runs, while ablations indicate that the two levels address complementary failure modes. Our central finding is that a self-teacher need not determine token-level reinforcement or suppression to be useful—restricting it to magnitude modula-

tion, gated by student uncertainty, provides dense credit while keeping each sampled token’s scalar advantage sign anchored to verified rewards.

Several directions for future work are worth exploring. The magnitude-only modulation principle demonstrated here could be applied to other structured generation tasks beyond tool use, such as multi-hop retrieval or collaborative dialogue, where hierarchical outcome structure is present. Additionally, the design space between our fixed self-teacher and fully online co-training remains largely unexplored; adaptive scheduling of teacher influence strength λ across training stages may yield further gains. Finally, combining CREST with complementary advances in reward design (e.g., progress reward from Lu et al. (2026a)) could compound the benefits of both finer reward signals and better credit allocation.

Limitations

Our evaluation is conducted on two multi-turn tool-use benchmarks at the 4B and 8B scale. While these represent distinct interaction paradigms (structured function-calling and naturalistic agentic sessions), broader validation across additional benchmarks and larger model scales would further strengthen the generality claims. Additionally, the entropy gate uses per-token surprisal as a proxy for token importance, which is effective for separating format tokens from content tokens in tool-use trajectories but remains a heuristic that may not optimally capture token informativeness in all generation contexts.

Ethics Statement

Our work improves the training efficiency of LLM-based tool-calling agents, which carries dual-use risks common to agent capability research. More capable agents could be misused for automated social engineering, unauthorized system access, or large-scale data extraction via tool APIs. We note that CREST does not introduce new capabilities beyond what existing models can already attempt—it improves the *reliability* of tool-use within sanctioned benchmarks. Nevertheless, we encourage practitioners to pair improved agent training with robust safety measures, including action sandboxing, human-in-the-loop approval for sensitive tool calls, and rate limiting on API access. All experiments in this work are conducted on public benchmarks with synthetic tool environments that do not

interact with real-world systems.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (No. 62602580), the Ningbo Youth Science and Technology Innovation Leading Talent Program (No. 2025QL059), the Ant Group Research Fund, and the “Pioneer and Leading Goose” R&D Program of Zhejiang (No. 2025C02037).

We acknowledge the use of AI assistants (Claude, Anthropic) for writing polish and L^AT_EX formatting during the preparation of this manuscript. All scientific content, experimental design, and technical contributions are solely the work of the authors.

References

- Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. 2024. Back to basics: Revisiting reinforce-style optimization for learning from human feedback in llms. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12248–12267.
- Will Brown. 2026. On SFT, RL, and on-policy distillation. <https://x.com/willccbb/status/2050038277454143918>. Blog post.
- Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*.
- Jiazhan Feng, Shijue Huang, Xingwei Qu, Ge Zhang, Yujia Qin, Baoquan Zhong, Chengquan Jiang, Jinxin Chi, and Wanjun Zhong. 2025. Retool: Reinforcement learning for strategic tool use in llms. *arXiv preprint arXiv:2504.11536*.
- Romain Froger, Pierre Andrews, Matteo Bettini, Amar Budhiraja, Ricardo Silveira Cabral, Virginie Do, Emilien Garreau, Jean-Baptiste Gaya, Hugo Laurençon, Maxime Lecanu, et al. 2025. Are: Scaling up agent environments and evaluations. *arXiv preprint arXiv:2509.17158*.
- Yuqian Fu, Haohuan Huang, Kaiwen Jiang, Jiakai Liu, Zhuo Jiang, Yuanheng Zhu, and Dongbin Zhao. 2026. Revisiting on-policy distillation: Empirical failure modes and simple fixes. *arXiv preprint arXiv:2603.25562*.

- Jonas Hübner, Frederike Lübeck, Lejs Behric, Anton Baumann, Marco Bagatella, Daniel Marta, Ido Hakimi, Idan Shenfeld, Thomas Kleine Buening, Carlos Guestrin, et al. 2026. Reinforcement learning via self-distillation. *arXiv preprint arXiv:2601.20802*.
- Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.
- Carlos E Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. 2024. Swe-bench: Can language models resolve real-world github issues? In *International Conference on Learning Representations*, volume 2024, pages 54107–54157.
- Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. 2025. Search-rl: Training llms to reason and leverage search engines with reinforcement learning. *arXiv preprint arXiv:2503.09516*.
- Jeonghye Kim, Xufang Luo, Minbeom Kim, Sangmook Lee, Dohyung Kim, Jiwon Jeon, Dongsheng Li, and Yuqing Yang. 2026. Why does self-distillation (sometimes) degrade the reasoning capability of llms? *arXiv preprint arXiv:2603.24472*.
- Philippe Laban, Hiroaki Hayashi, Yingbo Zhou, and Jennifer Neville. 2025. Llms get lost in multi-turn conversation. *arXiv preprint arXiv:2505.06120*.
- Minghao Li, Yingxiu Zhao, Bowen Yu, Feifan Song, Hangyu Li, Haiyang Yu, Zhoujun Li, Fei Huang, and Yongbin Li. 2023. Api-bank: A comprehensive benchmark for tool-augmented llms. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pages 3102–3116.
- Xuefeng Li, Haoyang Zou, and Pengfei Liu. 2025. Torl: Scaling tool-integrated rl. *arXiv preprint arXiv:2503.23383*.
- Yaxuan Li et al. 2026. Rethinking on-policy distillation of large language models: Phenomenology, mechanism, and recipe. *arXiv preprint arXiv:2604.13016*.
- Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. 2024. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.
- Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. 2025. Understanding rl-zero-like training: A critical perspective. *arXiv preprint arXiv:2503.20783*.
- Zichen Liu, Anya Sims, Keyu Duan, Changyu Chen, Simon Yu, Xiangxin Zhou, Haotian Xu, Shaopan Xiong, Bo Liu, Chenmian Tan, Weixun Wang, Hao Zhu, Weiyan Shi, Diyi Yang, Michael Qizhe Shieh, Yee Whye Teh, Wee Sun Lee, and Min Lin. 2026. GEM: A gym for generalist LLMs. In *The Fourteenth International Conference on Learning Representations*.
- Kevin Lu and Thinking Machines. 2025. On-policy distillation of language models. <https://thinkingmachines.ai/blog/on-policy-distillation/>. Blog post.
- Siyuan Lu, Zechuan Wang, Hongxuan Zhang, Qintong Wu, Leilei Gan, Chenyi Zhuang, Jinjie Gu, and Tao Lin. 2026a. Don’t just fine-tune the agent, tune the environment. In *The Fourteenth International Conference on Learning Representations*.
- Zhengxi Lu, Zhiyuan Yao, Zhuowen Han, Zi-Han Wang, Jinyang Wu, Qi Gu, Xunliang Cai, Weiming Lu, Jun Xiao, Yueting Zhuang, et al. 2026b. Self-distilled agentic reinforcement learning. *arXiv preprint arXiv:2605.15155*.
- Mike A Merrill, Alexander G Shaw, Nicholas Carlini, Boxuan Li, Harsh Raj, Ivan Bercovich, Lin Shi, Jeong Yeon Shin, Thomas Walshe, E Kelly Buchanan, et al. 2026. Terminal-bench: Benchmarking agents on hard, realistic tasks in command line interfaces. *arXiv preprint arXiv:2601.11868*.
- OpenClaw. 2026. Openclaw. <https://github.com/openclaw/openclaw>. Open-source personal AI assistant, version 2026.3.8, accessed 2026-03-09.
- 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 *Forty-second International Conference on Machine Learning*.
- Emiliano Penalosa, Dheeraj Vattikonda, Nicolas Gontier, Alexandre Lacoste, Laurent Charlin, and Massimo Caccia. 2026. Privileged information distillation for language models. *arXiv preprint arXiv:2602.04942*.
- Cheng Qian, Emre Can Acikgoz, Qi He, Hongru Wang, Xiushi Chen, Dilek Hakkani-Tur, Gokhan Tur, and Heng Ji. 2025. Toolrl: Reward is all tool learning needs. *Advances in Neural Information Processing Systems*, 38:105523–105553.
- Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, and 25 others. 2025. Qwen2.5 technical report.
- John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv: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*.
- Idan Shenfeld, Mehul Damani, Jonas Hübner, and Pulkit Agrawal. 2026a. Self-distillation enables continual learning. *arXiv preprint arXiv:2601.19897*.
- Idan Shenfeld et al. 2026b. Self-distillation enables continual learning. *arXiv preprint arXiv:2601.19897*.
- Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Cote, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. 2021. [{ALFW}orld: Aligning text and embodied environments for interactive learning](#). In *International Conference on Learning Representations*.
- Mingyang Song and Mao Zheng. 2026. A survey of on-policy distillation for large language models. *arXiv preprint arXiv:2604.00626*.
- Mark Towers, Ariel Kwiatkowski, John Balis, Gianluca De Cola, Tristan Deleu, Manuel Goulão, Kallinteris Andreas, Markus Krimmel, Arjun Kg, Rodrigo Perez-Vicente, et al. 2025. Gymnasium: A standard interface for reinforcement learning environments. *Advances in Neural Information Processing Systems*, 38.
- Zihan Wang, Kangrui Wang, Qineng Wang, Pingyue Zhang, Linjie Li, Zhengyuan Yang, Xing Jin, Kefan Yu, Minh Nhat Nguyen, Licheng Liu, et al. 2025. Ragen: Understanding self-evolution in llm agents via multi-turn reinforcement learning. *arXiv preprint arXiv:2504.20073*.
- Jason Wei, Zhiqing Sun, Spencer Papay, Scott McKinney, Jeffrey Han, Isa Fulford, Hyung Won Chung, Alex Tachard Passos, William Fedus, and Amelia Glaese. 2025a. Browsecomp: A simple yet challenging benchmark for browsing agents. *arXiv preprint arXiv:2504.12516*.
- Quan Wei, Siliang Zeng, Chenliang Li, William Brown, Oana Frunza, Wei Deng, Yuriy Nevmyvaka, Yang Katie Zhao, Alfredo Garcia, and Mingyi Hong. 2025b. [Reinforcing multi-turn reasoning in LLM agents via turn-level reward design and credit assignment](#). In *First Workshop on Multi-Turn Interactions in Large Language Models*.
- Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Toh J Hua, Zhoujun Cheng, Dongchan Shin, Fangyu Lei, et al. 2024. Oworld: Benchmarking multimodal agents for open-ended tasks in real computer environments. *Advances in Neural Information Processing Systems*, 37:52040–52094.
- 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*.
- Chenxu Yang et al. 2026. Self-distilled RLVR. *arXiv preprint arXiv:2604.03128*.
- Peijie Yu, Wei Liu, Yifan Yang, Jinjian Li, Zelong Zhang, Xiao Feng, and feng zhang. 2026. [Benchmarking LLM tool-use in the wild](#). In *The Fourteenth International Conference on Learning Representations*.
- Qiyang Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. 2025. Dapo: An open-source llm reinforcement learning system at scale. *Advances in Neural Information Processing Systems*, 38:113222–113244.
- Shaokun Zhang, Yi Dong, Jieyu Zhang, Jan Kautz, Bryan Catanzaro, Andrew Tao, Qingyun Wu, Zhiding Yu, and Guilin Liu. 2025a. Nemotron-research-tool-n1: Tool-using language models with reinforced reasoning. *arXiv preprint arXiv:2505.00024*, 2(10).
- Yiran Zhang, Mo Wang, Xiaoyang Li, Kaixuan Ren, Chencheng Zhu, and Usman Naseem. 2025b. Turnbench-ms: A benchmark for evaluating multi-turn, multi-step reasoning in large language models. *Findings of the Association for Computational Linguistics: EMNLP*, pages 19892–19924.
- Siyan Zhao, Zhihui Xie, Mengchen Liu, Jing Huang, Guan Pang, Feiyu Chen, and Aditya Grover. 2026. Self-distilled reasoner: On-policy self-distillation for large language models. *arXiv preprint arXiv:2601.18734*.
- Yuxiang Zheng, Dayuan Fu, Xiangkun Hu, Xiaojie Cai, Lyumanshan Ye, Pengrui Lu, and Pengfei Liu. 2025. Deepresearcher: Scaling deep research via reinforcement learning in real-world environments. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 414–431.

A Gradient Analysis

We prove the three properties stated in §3. Recall the structured per-token advantage:

$$\begin{aligned}\mathcal{A}_t &= A_{[t]}^{\text{turn}} \cdot \phi_t, \\ \phi_t &= 1 + \lambda_t^{\text{eff}} \cdot (w_t - 1)\end{aligned}\quad (11)$$

and the resulting policy gradient:

$$\nabla_{\theta} J(\theta) = \mathbb{E} \left[\sum_t \mathcal{A}_t \nabla_{\theta} \log \pi_{\theta}(y_t) \right] \quad (12)$$

A.1 (P1) Direction Preservation

Proposition 1 (Direction Preservation). *For all tokens t , $\text{sign}(\mathcal{A}_t) = \text{sign}(A_{[t]}^{\text{turn}})$.*

Proof. We show $\phi_t > 0$ always holds, so multiplication by ϕ_t preserves the sign of $A_{[t]}^{\text{turn}}$.

Case 1: $g_t^{\text{dir}} = 0$ (direction gate off). Then $\lambda_t^{\text{eff}} = 0$ by Eq. 10, so $\phi_t = 1 > 0$.

Case 2: $g_t^{\text{dir}} = 1$ (direction gate on). This requires $\text{sign}(A_{[t]}^{\text{turn}}) \cdot \Delta_t > 0$ by Eq. 7, which implies $\exp(\text{sign}(A_{[t]}^{\text{turn}}) \cdot \Delta_t) > 1$. After clipping, $w_t \in [1, 1 + \epsilon]$, so $w_t - 1 \geq 0$. Since $\lambda_t^{\text{eff}} \geq 0$:

$$\phi_t = 1 + \underbrace{\lambda_t^{\text{eff}}}_{\geq 0} \underbrace{(w_t - 1)}_{\geq 0} \geq 1 > 0 \quad (13)$$

In both cases $\phi_t > 0$, so $\text{sign}(\mathcal{A}_t) = \text{sign}(A_{[t]}^{\text{turn}} \cdot \phi_t) = \text{sign}(A_{[t]}^{\text{turn}})$. $\square$

Interpretation. The scalar coefficient of each token’s policy-gradient contribution has the same sign as the inter-turn-only coefficient (i.e., CREST with $\phi_t=1$). The teacher signal modulates the *magnitude* of each token’s contribution but cannot flip its scalar sign, yielding a local, token-wise anchoring property.

A.2 (P2) Absolute Perturbation Bound

Proposition 2 (Absolute Perturbation Bound). *For a sampled trajectory, let $g_t = A_{[t]}^{\text{turn}} \nabla_{\theta} \log \pi_{\theta}(y_t | h_t)$ denote the per-token inter-turn gradient contribution. Let $\widehat{\nabla_{\theta} J}$ and $\widehat{\nabla_{\theta} J}_{\text{TG}}$ denote the reweighted and inter-turn-only single-trajectory gradients, respectively. The teacher-induced perturbation satisfies:*

$$\left\| \sum_t (\phi_t - 1) g_t \right\| \leq \lambda \epsilon \sum_t \|g_t\|. \quad (14)$$

The bound is absolute; it is not relative to $\|\sum_t g_t\|$.

Proof. The difference between the reweighted and inter-turn-only gradients on one sampled trajectory is:

$$\widehat{\nabla_{\theta} J} - \widehat{\nabla_{\theta} J}_{\text{TG}} = \sum_t (\phi_t - 1) g_t. \quad (15)$$

If $g_t^{\text{dir}} = 0$, then $\phi_t = 1$. If $g_t^{\text{dir}} = 1$, then $\phi_t - 1 = \lambda_t^{\text{eff}}(w_t - 1)$, with $\lambda_t^{\text{eff}} \leq \lambda$ and $w_t - 1 \leq \epsilon$ by Eqs. 10 and 6. Thus, for every token, $|\phi_t - 1| \leq \lambda \epsilon$. Applying the triangle inequality gives:

$$\begin{aligned}\left\| \sum_t (\phi_t - 1) g_t \right\| &\leq \sum_t |\phi_t - 1| \|g_t\| \\ &\leq \lambda \epsilon \sum_t \|g_t\|.\end{aligned}\quad (16)$$

$\square$

Interpretation. The teacher-induced perturbation is controlled in absolute magnitude by the sum of per-token inter-turn gradient norms. With the default settings ($\lambda = 0.3$, $\epsilon = 0.28$), each token’s advantage coefficient can be amplified by at most 8.4%; this is not a relative bound on the aggregate gradient, which may be small because token contributions cancel. Reducing λ still makes the per-token modulation arbitrarily weak, smoothly approaching the inter-turn-only ablation.

A.3 (P3) Sign-Consistent Amplification

Proposition 3 (Sign-Consistent Amplification). *When $g_t^{\text{dir}} = 1$, $\phi_t \geq 1$. That is, the teacher can only amplify the gradient magnitude along the verifier-approved direction, never suppress it.*

Proof. This was established in the proof of (P1), Case 2: $g_t^{\text{dir}} = 1 \implies w_t \geq 1 \implies \phi_t = 1 + \lambda_t^{\text{eff}}(w_t - 1) \geq 1$. $\square$

Interpretation. Combined with (P1), this means: for every token, either the teacher has no effect ($\phi_t = 1$, when the direction gate is off) or the teacher *increases* the magnitude of the update in the direction already approved by the verified reward ($\phi_t > 1$). The teacher signal is therefore a selective token-level amplifier.

A.4 Discussion: Relationship to Adaptive Preconditioning

The modulation factor ϕ_t can be interpreted as a form of token-level adaptive preconditioning on the policy gradient. Standard GRPO applies a uniform scale to all tokens within a trajectory (or turn). Our method replaces this with a per-token scale that is:

Table 5: Comparison with hybrid RL and self-distillation methods. “Token direction” denotes whether a sampled token is reinforced or suppressed by its scalar advantage, not the direction of the aggregate parameter gradient.

Method	Objective signal	Token direction	Magnitude	Supervision
MT-GRPO (Wei et al., 2025b)	$A_t = A_k^{\text{sum}}$	turn verifier	uniform per turn	intermediate + outcome rewards
EnvTuning (Lu et al., 2026a)	$A_t = A_{\text{traj}}^V$	process verifier	uniform per trajectory	process rewards
RLSD (Yang et al., 2026)	$A_t = A_{\text{traj}}^V m_t$	trajectory verifier	self-teacher	terminal reward + reference
SDAR (Lu et al., 2026b)	$\mathcal{L}_{\text{RL}} + \lambda \mathcal{L}_{\text{SD}}$	verifier + teacher branch	gated distillation	reward + retrieved skills
OPD (Lu and Thinking Machines, 2025)	$A_t = \Delta_t^{\text{ext}}$	external teacher	external teacher	same-family 235B teacher
OPSD (Zhao et al., 2026)	$A_t = \Delta_t^{\text{self}}$	self-teacher	self-teacher	privileged context
CREST (Ours)	$A_t = A_k^V \phi_t$	independent turn verifier	gated self-teacher	per-turn reward + ground truth

- Larger on tokens where the privileged teacher agrees with the verifier direction *and* the student is uncertain (high surprisal).
- Equal to 1 on tokens where the teacher disagrees, or the student is already confident.

This is analogous to how natural gradient methods or Adam apply per-parameter adaptive learning rates. In our case, the teacher–student divergence gated by student uncertainty serves as the “curvature proxy,” yielding adaptive scalar reweighting at the token level rather than the parameter level.

The key difference from OPSD is that OPSD uses the teacher signal as the advantage itself (both direction and magnitude), whereas our method uses it only as a magnitude modulator on top of verified reward. Thus, OPSD’s token-level optimization target is teacher-defined, while CREST keeps each token’s scalar update sign anchored to the verifier.

Relationship to hybrid self-distillation. As summarized in Table 5, RLSD is the closest predecessor: it combines verifier-derived signs with self-teacher magnitudes, but operates at the sequence level without uncertainty gating and is evaluated only in single-step settings. CREST adds independent per-turn credit and uncertainty gating. SDAR targets multi-turn agents with gated token supervision, but adds a separate distillation loss to trajectory-level RL, whereas CREST uses the single multiplicative advantage $A_k^V \phi_t$ without an auxiliary teacher objective.

B Complete Algorithm

Algorithm 1 provides the complete pseudocode for a single CREST training step, consolidating all equations from Section 3 into an executable procedure.

C Experimental Details

C.1 Training Configuration

All experiments use the following shared configuration:

- **Hardware:** Single-node $8 \times$ H200 GPUs. For methods requiring a teacher (OPSD, CREST), 4 GPUs serve the teacher via SGLang and 4 GPUs handle student training and rollout generation, with tensor parallelism = 2 and sequence parallelism enabled. For OPD with Qwen3-4B-Instruct, the external teacher (Qwen3-235B-A22B-Instruct) requires an additional 8 GPUs for deployment, resulting in a 16-GPU setup (8 for teacher, 8 for student).
- **Rollout:** On-policy sampling with group size $G=16$, temperature 1.0, max generation length 10,000 tokens per rollout.
- **Batch size:** 32 prompts per step $\times$ 16 samples per prompt = 512 rollouts per step.
- **Optimizer:** Adam with $\beta_1=0.9$, $\beta_2=0.999$, weight decay 0.01, gradient clipping 1.0.
- **Learning rate:** 1×10^{-6} , constant schedule.
- **Training steps:** The main Qwen3-4B BFCL V3 runs use method-specific schedules: 160 steps for GRPO, 150 for MT-GRPO, 160 for EnvTuning, 100 for OPD, 89 for OPSD, and 160 for CREST. We do not assume a single universal step count across methods.
- **PPO clipping:** $\epsilon_{\text{low}}=0.2$, $\epsilon_{\text{high}}=0.28$.
- **KL penalty:** $\beta_{\text{KL}}=0.0$ (no explicit KL regularization; clipping provides implicit constraint).

Accuracy–compute trade-off. Table 6 shows that CREST has the same eight-GPU peak footprint as GRPO, split between four student GPUs and four self-teacher GPUs. OPD instead requires a second node and a same-family 235B external teacher, so its lower total GPU-hours are not directly comparable to the eight-GPU methods. Relative to GRPO, CREST increases average step time from 29.1 to 39.4 minutes (+35.4%), total train-

Algorithm 1 CREST Training Step

Require: Prompt batch $\mathcal{D}$, policy π_θ , group size G , hyperparams $\lambda, \tau, \epsilon, \rho$

```
1: for each prompt  $x \in \mathcal{D}$  do
2:   Sample  $G$  rollouts  $\{y^{(i)}\}_{i=1}^G \sim \pi_\theta(\cdot|x)$ 
3:   Obtain per-turn verified rewards  $\{R_k^{(i)}\}$ 
4:   // Inter-turn: turn-segmented advantage
5:   for each turn  $k$  do
6:      $A_k^{(i)} \leftarrow (R_k^{(i)} - \mu_k) / (\sigma_k + \epsilon)$  (Equation (3))
7:   end for
8:   // Intra-turn: entropy-gated self-teacher
9:   Compute privileged teacher logprobs  $\log \pi_T(y_t | h_t^T)$ 
10:  for each token  $t$  in rollout  $i$  do
11:     $\Delta_t \leftarrow (\log \pi_T - \log \pi_\theta) / \tau$  (Equation (5))
12:     $w_t \leftarrow \text{clip}(\exp(\text{sign}(A_{[t]}^{\text{turn}}) \cdot \Delta_t), 1-\epsilon, 1+\epsilon)$  (Equation (6))
13:     $g_t^{\text{dir}} \leftarrow \mathbf{1}[\text{sign}(A_{[t]}^{\text{turn}}) \cdot \Delta_t > 0]$  (Equation (7))
14:     $g_t^{\text{ent}} \leftarrow \sigma((u_t - \mathbb{E}[u]) / (\text{std}(u) + \epsilon))$  (Equation (8))
15:     $\lambda_t^{\text{eff}} \leftarrow \text{clip}(\lambda \cdot g_t^{\text{dir}} \cdot (1 + \rho(2g_t^{\text{ent}} - 1)), 0, \lambda)$  (Equation (10))
16:     $\mathcal{A}_t \leftarrow A_{[t]}^{\text{turn}} \cdot (1 + \lambda_t^{\text{eff}} \cdot (w_t - 1))$ 
17:  end for
18: end for
19: Update  $\theta$  via policy gradient with per-token advantages  $\{\mathcal{A}_t\}$ 
```

Table 6: Training cost on BFCL V3 with Qwen3-4B-Instruct. All runs use H200 GPUs. GRPO wall-clock and GPU-hours are normalized to the reported 29.1 minutes per step over the unified 160-step schedule.

Method	Peak GPUs	Allocation	Extra teacher	Steps	Wall-clock (h)	min/step	GPU-h	BFCL Avg.
GRPO	8	8 student	no	160	77.6	29.1	621	43.63
MT-GRPO	8	8 student	no	150	81.2	32.5	650	49.25
EnvTuning	8	8 student	no	160	88.3	33.1	707	47.25
OPD	16	8 student + 8 teacher	235B external	100	22.4	13.5	359	44.50
OPSD	8	4 student + 4 teacher	self	89	31.0	20.9	248	38.75
CREST	8	4 student + 4 teacher	self	160	105.0	39.4	840	52.00

ing time from 77.6 to 105.0 hours (+35.3%), and compute from 621 to 840 GPU-hours (+35.3%). Teacher scoring is pipelined with student rollout, so the overhead is not a fully serialized teacher forward pass; it also reflects the reduction from eight to four student-side GPUs. This additional compute yields a +4.12 BFCL improvement over the inter-turn-only variant (47.88 to 52.00) and +8.37 over GRPO (43.63 to 52.00), representing an explicit accuracy–compute trade-off.

For CREST-specific hyperparameters: $\lambda=0.3$ (modulation strength), $\epsilon=0.28$ (clip bound), $\tau=2.0$ (temperature for Δ_t), and $\rho=0.5$ (entropy gate range). Only λ is tuned; the remaining values are fixed design choices across all benchmarks and model scales.

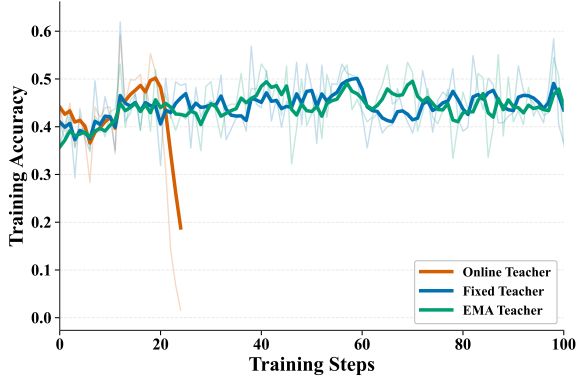
The self-teacher is constructed by conditioning the same model on ground-truth tool-call results for preceding turns, providing privileged context

without requiring an external teacher model or tokenizer alignment.

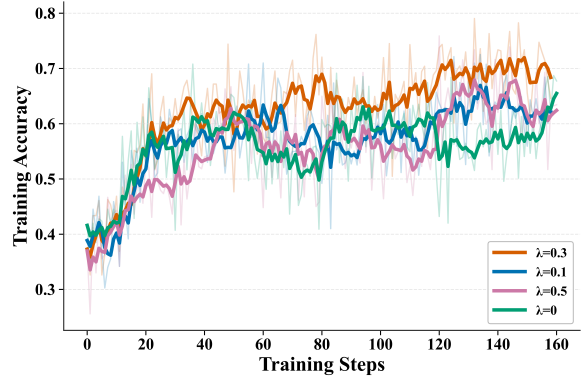
C.2 OPSD Stabilization and Teacher Update Strategies

Directly applying OPSD (Zhao et al., 2026) to multi-turn tool-use trajectories results in training collapse within ~ 20 steps, consistent with findings reported by the original authors. To establish a fair OPSD baseline, we investigated three teacher update strategies (Figure 4a):

- **Online teacher** (updated with policy): Catastrophic collapse by step 20—training accuracy drops from ~ 0.45 to near zero, caused by pivot-token concentration shifting direction each update.
- **Fixed teacher** (frozen at initialization): The most stable strategy, maintaining training accuracy around 0.44–0.48 throughout. However,



(a) OPSD teacher update strategies.



(b) Modulation strength λ sensitivity.

Figure 4: Analysis of OPSD teacher update strategies and modulation strength on BFCL V3 (Qwen3-4B). (a) Comparison of OPSD teacher update strategies: the online teacher collapses catastrophically within ~ 20 steps, while fixed and EMA teachers are both stable, with the fixed teacher achieving marginally higher and more consistent accuracy; we adopt the fixed teacher with per-token KL clipping (threshold 0.2) for the OPSD baseline in all experiments. (b) Effect of the modulation strength λ on training accuracy: $\lambda=0.3$ provides the best trade-off; higher values over-concentrate gradients.

the static teacher distribution defines the token-level optimization target and produces an observed plateau.

- **EMA teacher** (exponential moving average, $\alpha=0.995$): Comparable stability to the fixed teacher but no improvement—the teacher converges to a lagged copy of the student, erasing the privileged information advantage.

Based on these results, our OPSD baseline adopts the fixed teacher with per-token KL clipping (clip threshold 0.2) as recommended by Zhao et al. (2026). This prevents collapse but leaves the student at the observed plateau associated with a teacher-defined optimization target shown in Figure 4a. This empirical observation directly motivates CREST’s design: rather than using the teacher signal as the optimization target, we use it solely as a gradient magnitude modulator anchored to the verified token sign.

C.3 Modulation Strength Sensitivity

Figure 4b varies the modulation strength λ , which controls how aggressively the entropy gate selectively amplifies token updates. At $\lambda=0$ (inter-turn-only CREST, $\phi_t=1$), final accuracy is ~ 0.63 . At $\lambda=0.3$, entropy-gated modulation raises final accuracy to ~ 0.69 (+6%) by emphasizing high-uncertainty content tokens and reducing amplification on low-entropy format tokens. At $\lambda=0.5$, over-concentration causes early gains but stagnation below $\lambda=0.1$ by step 80, ending at ~ 0.61 . The sweet spot at $\lambda=0.3$ confirms that *calibrated* selec-

tive amplification, not maximal concentration, best resolves intra-turn credit ambiguity.