CONTROLLING INTERACTION DRIFT: OPTIMIZING LLM AGENTS FOR MULTI-TURN USER INTERACTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models are increasingly post-trained as multi-turn agents for tool-mediated customer service, such as e-commerce assistance. In such settings, rollouts are generated online through a branching interaction process in which agent actions, user responses, and environment updates jointly determine what happens next. An erroneous response or tool action can therefore redirect subsequent interactions onto a different branch. We call this error-driven redirection *interaction drift*. This creates a collection-time decision: whether an error-conditioned branch should continue generating interactions for policy optimization. On-policy distillation (OPD) provides fine-grained teacher supervision in student-visited contexts, whereas terminal-reward reinforcement learning provides task-level feedback only after rollout completion. Neither signal determines whether another interaction should be collected from such a branch or how task credit should be localized within the trajectory. We introduce BLADE (Boundary-Localized Advantage for Drifted Episodes), a post-training framework with a frozen, environment-grounded Boundary Judge. After each completed nonterminal interaction loop and before the next simulated-user message, the Judge either continues collection or stops the branch at the loop in which the error was introduced. Teacher supervision is retained over the full collected rollout in either case, while the stopped-rollout task signal is confined to that loop; naturally completed rollouts retain trajectory-level task credit. Experiments across two stateful interaction benchmarks and two teacher–student configurations, including held-out-domain evaluation, show consistent improvements in strict repeated-run reliability over strong post-training baselines. These results suggest that controlling interaction drift during collection can improve the consistency of successful multi-turn behavior across model scales and evaluation settings.

1 INTRODUCTION

Large language models are increasingly studied as interactive agents in user-facing applications such as e-commerce assistance, where task completion often requires multiple rounds of dialogue, user feedback, and tool use (Wang et al., 2024; Yao et al., 2025; Zhou et al., 2026a). Post-training in this setting differs from training on fixed trajectories because subsequent observations are generated online from the preceding interaction rather than being fixed in advance. Each assistant response becomes part of the dialogue history, while tool calls can modify the persistent environment state (Lu et al., 2025; Barres et al., 2025; Qian et al., 2025). Both in turn condition later user messages and agent behavior. Because each interaction is generated from the context produced by preceding interactions, the effect of an erroneous response or tool action need not remain local: it can alter the history or environment state that conditions subsequent interactions.

As illustrated in Figure 1(a), an e-commerce agent incorrectly claims that the delivery address cannot be changed, prompting the user to request cancellation and a tool call that changes the order state from ACTIVE to CANCELLED. Later interactions are therefore generated from both a dialogue history containing the false claim and the altered order state. We refer to this error-driven redirection of later interactions within the same rollout as *interaction drift*.

Interaction drift raises a broader optimization question: once an error has altered the trajectory, which parts of the resulting rollout still provide reliable training support for the original task? A

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

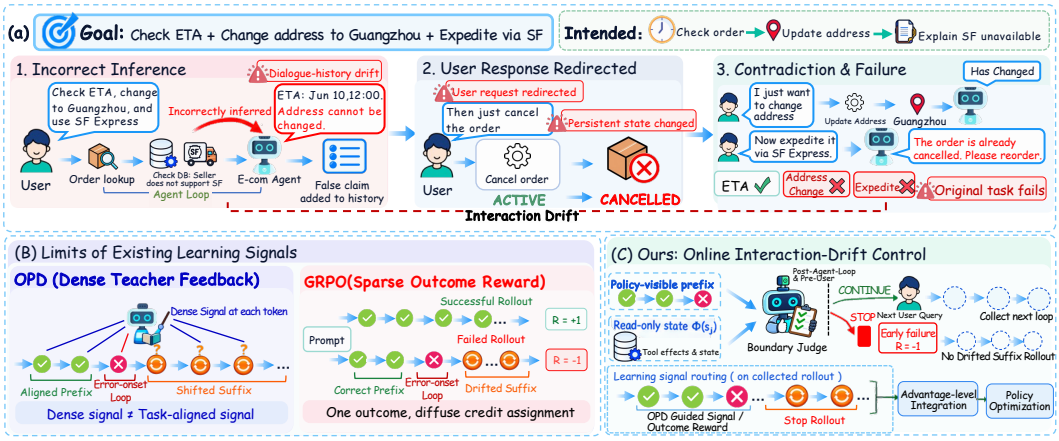


Figure 1: Interaction drift and BLADE. (a) A local error redirects subsequent user–tool interaction. (b) Teacher feedback and terminal rewards do not by themselves prevent collection of the drifted suffix. (c) BLADE stops collection after a consequential error and before the next simulated-user turn, while routing OPD supervision and task credit separately.

drifted suffix can contain responses that are appropriate under the altered history and state, yet such local appropriateness does not establish their value for optimizing the original task. Outcome-based RL cannot resolve this because a terminal failure carries no information that separates the interaction introducing the error from downstream behavior responding to its consequences. Trajectory-level negative feedback therefore does not identify where responsibility lies. OPD faces the complementary problem. Teacher scores can remain locally valid on error-conditioned histories, but OPD does not determine whether supervision on those histories contributes to the original task objective. Because these interactions are generated online, the problem is not only where supervision and credit should be applied after collection, but whether the rollout should continue generating data from the error-conditioned history and state in the first place. Once that suffix has been generated, post-hoc masking can exclude it from optimization, but cannot undo its generation.

Existing methods intervene at different points in the rollout. Temporal-control methods regulate rollout generation by scheduling depth, mixing teacher- and student-generated turns, or allocating longer rollouts from turn statistics or early teacher–student compatibility (Wang et al., 2026a; Li et al., 2026; Zhou et al., 2026c; Zhao et al., 2026a). Other approaches rely on evidence available only after behavior has been generated: selective OPD gates or reweights supervision on produced responses (Zhou et al., 2026b; Tan et al., 2026), interaction-aware credit uses the next user reaction or completed failed rollouts (Zhao et al., 2026b; Hu et al., 2026), and future-trajectory validation tests teacher guidance through additional continuations (Chen et al., 2026). None of these approaches uses the current response and its realized tool effects to identify the optimization boundary before the next user turn. In stateful user interaction, this timing matters because obtaining additional evidence already extends the trajectory under the resulting history and environment state. Reliable optimization support must therefore be identified from the observed prefix before further interaction is generated.

To address this gap, we introduce BLADE (Boundary-Localized Advantage for Drifted Episodes), a framework that couples rollout collection with outcome assignment and learning-signal allocation. After each nonterminal interaction loop, a frozen Boundary Judge examines the policy-visible interaction prefix together with a read-only view of the realized environment state, before the next simulated-user message is generated. If it detects a consequential error in the completed loop, it stops the branch before further interaction is generated from a history or state affected by that error. A stopped rollout receives an early failure outcome. OPD retains teacher supervision over all student tokens in the collected rollout, while task credit is restricted to the error-onset loop. For naturally terminated rollouts, task credit instead extends across the trajectory. The two signals are combined at the advantage level before policy-ratio clipping. Both the teacher and the Judge are used only during training.

We evaluate BLADE on Mix-Ecom and τ^2-bench across two teacher–student configurations and both in-domain and held-out-domain settings. BLADE consistently improves strict repeated-run reliability over the strongest student baselines.

Our contributions are:

- We characterize *interaction drift* in stateful multi-turn interaction and formulate the resulting collection-time boundary problem. We address it with an environment-grounded Boundary Judge that detects consequential errors before the next simulated-user turn and prevents the generation of error-conditioned suffixes.
- We develop boundary-localized credit assignment for stopped rollouts. Teacher supervision is retained over all collected student tokens, while task-level failure credit is confined to the error-onset loop; naturally terminated rollouts retain trajectory-level task credit.
- We integrate collection-time boundary control and localized credit into BLADE, with teacher and task signals combined at the advantage level. Across two stateful interaction benchmarks and two teacher–student configurations, BLADE achieves the highest strict repeated-run reliability among the compared student baselines in all four settings.

2 Related Work

Policy optimization and distillation for agents. Outcome-based policy optimization and dense teacher supervision provide complementary signals for agent post-training. GRPO estimates group-relative advantages without a learned critic (Shao et al., 2024), while GSPO moves importance weighting and clipping from the token level to the sequence level (Zheng et al., 2025). OPD instead scores student-sampled tokens under a teacher (Agarwal et al., 2024). For multi-turn agents, recent OPD methods schedule temporal depth or introduce teacher-generated turns (Wang et al., 2026a; Li et al., 2026), allocate teacher supervision selectively across steps or turns (Zhong et al., 2026; Zhou et al., 2026b), and use prefix replay or look-ahead continuations to manage teacher reliability (Liao et al., 2026; Chen et al., 2026). TurnOPD and PG-OPD instead allocate loss or rollout depth from turn- and prefix-level statistics (Zhou et al., 2026c; Zhao et al., 2026a). Hybrid objectives also combine outcome advantages with dense token-level feedback. ATOD combines OPD and GRPO advantages using turn-level disagreement and uncertainty (Tan et al., 2026), while OPID derives hindsight skill feedback from completed on-policy trajectories (Yang et al., 2026). BLADE also combines teacher and task signals, but its primary intervention is a per-branch collection decision rather than another post-collection weighting rule.

Learning from multi-turn user interaction. In user-facing tool tasks, an agent interaction loop can change both the subsequent dialogue and the persistent environment state. τ-bench and τ^2-bench evaluate this dependence through simulated users and tool-backed environments (Yao et al., 2025; Barres et al., 2025), while Mix-Ecom covers mixed e-commerce dialogues governed by complex domain rules (Zhou et al., 2026a). CollabLLM and UserRL train agents through simulated multi-turn interaction rather than using the user model only for evaluation (Wu et al., 2025; Qian et al., 2025). Recent methods also derive finer temporal evidence at different stages. FACA derives interaction credit from the next user reaction (Zhao et al., 2026b), and OpenClaw-RL extracts evaluative and directive feedback from observed next states (Wang et al., 2026b). SEAL adapts the learning environment around diagnosed failures (Hu et al., 2026), whereas R^{3}L uses reflection, retry, and pivotal credit (Shi et al., 2026). BLADE instead acts at each realized nonterminal boundary: using the policy-visible prefix and a read-only view of the tool-updated state, it decides whether to request the next simulated-user message. Its distinction is collection control before an error-conditioned suffix exists; supervision is then routed only over retained tokens.

3 Preliminaries and Problem Formulation

3.1 Stateful Multi-Turn Interactive Rollouts

For a task prompt x, an interactive rollout is an ordered sequence of user–agent interaction loops,

$$\tau = (\ell_1, \ldots, \ell_M), \qquad s_j = F(s_{j-1}, \ell_j), \qquad u_{j+1} \sim p_{\mathrm{U}}(\cdot \mid H_j, g), \tag{1}$$

where ℓ_j is the j-th interaction loop, H_j and s_j are the user-visible dialogue history and persistent environment state after that loop, and g denotes the simulator's private user goal and context. We write $\bar{h}_j$ for the complete policy-visible interaction prefix through ℓ_j. Each loop contains the user message and ensuing assistant behavior, including tool interactions, until the environment terminates or control returns to the user simulator. The simulator conditions on H_j and g, but does not directly observe s_j.

Only student-generated tokens are optimized. Let $\mathcal{T}_j(\tau)$ denote the student tokens in loop ℓ_j and $\mathcal{T}(\tau) = \bigcup_{j=1}^{M} \mathcal{T}_j(\tau)$ those in the rollout. For $a_t \in \mathcal{T}_j(\tau)$, let h_t denote its policy-visible context. We denote the trainable and rollout-collection policies by π_θ and π_{old}, respectively.

3.2 Interaction Drift and Error Onset

An assistant turn is *invalid* if its response or tool action is inconsistent with the user request, applicable task rules, or the realized environment state. It is *consequential* if it introduces an incorrect premise into H_j or commits an incorrect change to s_j, such that subsequent interaction would be conditioned on that error. We define j_{err} as the index of the *error-onset loop*, the loop containing a consequential invalid turn. If collection continues beyond j_{err}, subsequent interactions are generated from an error-conditioned history or environment state; we call this *interaction drift*. The resulting drifted suffix begins after the error-onset loop.

3.3 On-Policy Learning Signals

On-policy distillation. Following on-policy distillation (Agarwal et al., 2024), for a student-sampled token a_t at context h_t, we define

$$A_t^{\text{OPD}} = \text{sg}[\log \pi_{\text{T}}(a_t \mid h_t) - \log \pi_{\text{old}}(a_t \mid h_t)], \tag{2}$$

where sg denotes stop-gradient. Its conditional expectation under π_{old} is $-D_{\text{KL}}(\pi_{\text{old}}(\cdot \mid h_t) \| \pi_{\text{T}}(\cdot \mid h_t))$, making it a sampled teacher signal rather than a full-vocabulary KL evaluation.

Outcome-based policy optimization. For rollout $\tau_i \in \mathcal{G}(x)$, let $R_i = R(\tau_i)$ denote its scalar task reward. For natural termination, R is given by the terminal evaluator; the stopped-rollout reward is specified below. Following GRPO (Shao et al., 2024),

$$A_i^{\text{GRPO}} = \frac{R_i - \mu_R}{\sigma_R + \delta}, \tag{3}$$

where μ_R and σ_R are the mean and population standard deviation of the group rewards, and $\delta > 0$ is a numerical stabilizer. Thus, OPD provides token-level teacher information, whereas GRPO provides rollout-level task information.

3.4 Problem Statement

Learning under interaction drift leaves three decisions unspecified: (i) the *collection boundary*, determining whether to generate another interaction; (ii) the *stopped-rollout reward*, determining the task outcome assigned when collection stops early; and (iii) the *signal supports*, retaining full-rollout OPD supervision while determining which retained tokens receive GRPO credit. These choices jointly determine what further data are generated and how the retained trajectory contributes to optimization. Section 4 instantiates them.

4 Method

4.1 Overview

Figure 2 summarizes BLADE. A frozen Boundary Judge reviews each nonterminal loop before the next simulated-user turn. A STOP assigns early failure, retains OPD supervision on collected tokens, and localizes GRPO credit to the error-onset loop; otherwise collection continues. OPD and GRPO advantages are combined before policy-ratio clipping. The teacher and Judge are used only during training.

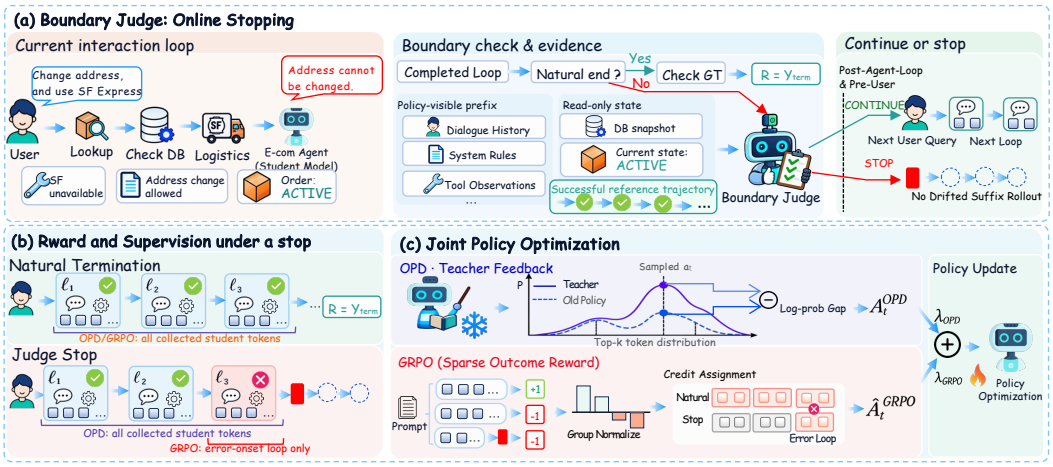


Figure 2: BLADE collection and learning pipeline. (a) The Boundary Judge reviews each completed nonterminal loop before the next simulated-user turn. (b) A STOP prevents suffix generation, retains OPD supervision on collected tokens, and localizes GRPO credit to the error-onset loop. (c) OPD and GRPO advantages are combined before the clipped policy update.

4.2 Boundary Judge: Online Stopping

Order of events. Once loop ℓ_j and its tool effects are realized, the collector first checks whether the environment has reached a natural terminal state. If so, no further collection decision remains, and the rollout is passed directly to the task evaluator without being reviewed by the Judge. Otherwise, the collector queries the Judge *before* requesting the next simulated-user message u_{j+1}, so every nonterminal boundary is reviewed before it is crossed. The Judge is therefore invoked once at each nonterminal boundary during rollout collection and is absent at evaluation time.

Evidence and decision. At a nonterminal boundary following loop j, a frozen evaluator $\mathcal{J}$ observes

$$E_j = \big(\bar{h}_j,\ \Phi(s_j)\big), \qquad z_j = \mathcal{J}(E_j), \qquad z_j \in \{\textsc{Continue},\ \textsc{Stop}\}, \tag{4}$$

where $\bar{h}_j$ denotes the complete policy-visible interaction prefix through ℓ_j, including the task prompt and applicable rules, the dialogue so far, and the assistant turns, tool calls, and tool observations up to that boundary. The term $\Phi(s_j)$ is a fixed read-only projection of the task-relevant persistent state at the same boundary. The Judge observes no future user message, no future environment state, no hidden reference answer, and no terminal evaluator output. The only Judge-specific input is $\Phi(s_j)$; all remaining evidence is drawn from the policy-visible realized interaction. This read-only state view allows the Judge to check a dialogue claim or an attempted state change against the persistent state that actually resulted, without issuing additional tool calls or relying on the policy having queried that state itself. As stated in Section 4.1, $\Phi(s_j)$ enters neither the student's context nor the teacher-scoring context.

Stopping criterion. A STOP is issued when E_j evidences a consequential invalid turn in the sense of Section 3.2, so that any continuation would be generated from an error-conditioned history or state; it sets $j_{\text{err}} = j$ for that rollout. A CONTINUE only instructs the collector to request u_{j+1}: it is not a certificate of correctness and carries no reward. In neither case does the Judge roll back environment state, replace the student's response with a teacher response, or contribute an advantage term of its own.

4.3 Reward and Supervision under a Stop

Outcome of a stopped rollout. We instantiate the task reward left open by decision (ii) as

$$R(\tau) = \begin{cases} y^{\text{term}}, & \text{natural termination,} \\ -1, & \text{Judge stop,} \end{cases} \qquad y^{\text{term}} \in \{-1, +1\}. \tag{5}$$

A Judge-stopped rollout is not subsequently submitted to the terminal evaluator and instead receives the early failure outcome above. Since a CONTINUE carries no intermediate reward, each of the G sampled rollouts receives exactly one task reward and remains in the same-prompt group used by Equation 3. Treating an evidenced STOP as a task failure is a modeling assumption: a stopped branch might in principle have recovered had collection continued.

Teacher supervision on the collected rollout. The teacher signal in Equation 2 is evaluated on the student-visited context at which each token was sampled. For both naturally terminated and Judge-stopped rollouts, we retain OPD supervision on every student-generated token that was actually collected,

$$\mathcal{S}_{\text{OPD}}(\tau_i) = \mathcal{T}(\tau_i). \tag{6}$$

In particular, the student action that introduces the error receives token-level teacher scores in the context in which it was produced, so A_t^{OPD} provides a direct signal on that action through the teacher–student log-probability difference. A STOP therefore does not post-hoc mask OPD supervision on the retained rollout. Because no subsequent interaction loop is collected beyond the stopping boundary, no corresponding teacher supervision is constructed.

Task credit on the error-onset loop. A rollout-level outcome alone does not identify which interaction introduced a failure rather than merely responding to its consequences. A Judge STOP provides loop-specific negative evidence at the stopping boundary, so we restrict task credit to the error-onset loop,

$$\mathcal{S}_{\text{GRPO}}(\tau_i) = \begin{cases} \mathcal{T}(\tau_i), & \text{natural termination,} \\ \mathcal{T}_{j_{\text{err}}}(\tau_i), & \text{Judge stop,} \end{cases} \qquad \widehat{A}_t^{\text{GRPO}} = \mathbf{1}[a_t \in \mathcal{S}_{\text{GRPO}}(\tau_i)]\, A_i^{\text{GRPO}}. \tag{7}$$

Here A_i^{GRPO} is the group-relative advantage of the rollout τ_i containing a_t. The two supports are nested, $\mathcal{S}_{\text{GRPO}}(\tau_i) \subseteq \mathcal{S}_{\text{OPD}}(\tau_i)$, and they coincide for naturally terminated rollouts. A stopped rollout therefore applies its task-failure signal only to the error-onset loop, whereas a naturally terminated rollout retains trajectory-level task credit.

4.4 Joint Policy Optimization

Advantage-level integration. For a student token a_t in rollout τ_i, we combine the independently constructed OPD and GRPO signals as

$$A_t^{\text{joint}} = \lambda_{\text{OPD}} A_t^{\text{OPD}} + \lambda_{\text{GRPO}} \widehat{A}_t^{\text{GRPO}}, \tag{8}$$

where $\lambda_{\text{OPD}}, \lambda_{\text{GRPO}} \geq 0$ and $\lambda_{\text{OPD}} + \lambda_{\text{GRPO}} > 0$; setting either coefficient to zero gives the corresponding single-signal ablation. The two signals are constructed independently and are combined only at this stage. When they disagree in sign, advantage-level integration allows them to offset before policy-ratio clipping, whereas loss-level integration clips the two channels separately before summation.

Clipped policy update. With token importance ratio $\rho_t(\theta) = \pi_\theta(a_t \mid h_t)/\pi_{\text{old}}(a_t \mid h_t)$, we optimize

$$\mathcal{L}(\theta) = -\mathbb{E}_t \left[\min\Big(\rho_t(\theta) A_t^{\text{joint}}, \text{clip}(\rho_t(\theta), 1-\epsilon, 1+\epsilon)\, A_t^{\text{joint}} \Big) \right], \tag{9}$$

where $0 < \epsilon < 1$. The expectation is a uniform average over tokens belonging to at least one support whose corresponding coefficient is nonzero. Support membership, rather than the realized advantage value, determines the reduction, so zero-advantage tokens remain in the denominator. If no token belongs to an active support in a batch, the loss is defined as zero and the update is skipped.

Table 1: Main results on Mix-Ecom and τ^2-bench (%). $\mathrm{Avg.@4}$ is mean success over four trials; pass^4 requires success in all four. After-sales and Telecom are held-out domains. Bold and underline denote the best and second-best student results.

	Mix-Ecom						τ^2-bench				
Method	**Log. KA**	**Log. DB**	**Log. Score**	**After-sales**	**Avg.@k**	**pass^4**	**Retail**	**Airline**	**Telecom**	**Avg.@k**	**pass^4**
Qwen3.5-27B → Qwen3.5-4B											
Teacher	67.59	64.58	53.47	38.74	46.73	35.18	81.36	67.00	97.15	85.25	75.54
Base	45.14	50.23	28.47	28.30	28.39	11.56	72.15	54.00	92.32	77.16	64.75
SFT	65.28	73.15	44.21	22.80	34.42	18.09	77.19	61.50	96.93	82.46	65.83
GRPO	42.82	46.53	31.71	9.34	21.48	9.55	79.82	62.00	**98.90**	84.44	67.99
OPD	79.40	79.86	61.11	34.89	49.12	36.18	80.26	64.50	98.25	84.80	70.14
TCOD	79.86	81.02	62.73	32.97	49.12	33.17	81.14	63.00	97.37	84.53	71.58
Guided-OPD	74.07	79.16	54.63	28.57	42.71	28.64	78.73	61.00	96.05	82.64	68.71
BLADE	**81.02**	**82.17**	**63.89**	**37.36**	**51.76**	**41.21**	**83.33**	**70.50**	96.49	**86.42**	**75.90**
Qwen3.5-9B → Qwen3.5-2B											
Teacher	68.29	70.60	46.53	36.54	41.96	23.12	75.88	63.00	98.46	82.82	69.42
Base	0.23	0.00	0.00	0.00	0.00	0.00	56.14	47.00	94.96	70.41	54.68
SFT	59.95	66.67	40.97	6.87	25.38	15.08	59.21	43.00	89.91	68.88	42.09
GRPO	67.59	**73.38**	47.92	**11.26**	**31.16**	16.08	63.16	49.50	93.20	73.02	56.12
OPD	**73.38**	69.91	44.68	6.87	27.39	17.09	70.39	55.00	96.93	78.51	61.87
TCOD	66.90	60.19	40.97	5.77	24.87	14.57	72.59	58.00	95.18	79.23	65.47
Guided-OPD	70.60	68.98	45.37	6.87	27.76	19.10	68.42	53.00	92.54	75.54	60.43
BLADE	**73.38**	72.22	**49.54**	7.42	30.28	**20.60**	**76.97**	**60.00**	**97.59**	**82.37**	**70.14**

Normalization and reduction. For any same-prompt group with constant rewards, including an all-STOP group, Equation 3 gives $A_i^{\mathrm{GRPO}} = 0$, so the GRPO channel contributes no update for that group and may be skipped in implementation. When both channels are active, uniform token reduction makes the aggregate GRPO contribution of a stopped rollout scale with the fraction of its collected tokens contained in the error-onset loop. We use uniform token reduction in the main method and evaluate a rollout-balanced alternative as a controlled ablation.

5 EXPERIMENTS

5.1 EXPERIMENTAL SETUP

Benchmarks and evaluation. We evaluate BLADE on Mix-Ecom (Zhou et al., 2026a) and τ^2-bench (Barres et al., 2025). For Mix-Ecom, we train on Logistics and test on held-out Logistics and out-of-distribution After-sales tasks. For τ^2-bench, we train on Retail and Airline and evaluate on all three domains, with Telecom held out from training. We run four independent interaction trials per task and report $\mathrm{Avg.@4}$, the mean per-trial success rate, and pass^4, the fraction of tasks succeeding in all four trials. Both weight tasks equally; pass^4 is our primary reliability metric. Dataset splits, success criteria, and metric definitions are detailed in Appendix A.

Models and training. We use Qwen3.5-27B → Qwen3.5-4B and Qwen3.5-9B → Qwen3.5-2B teacher–student pairs. Within each configuration, all post-SFT methods share the training task pool and offline-distilled initialization; only the student is updated. BLADE uses frozen DeepSeek-V4-Flash-0731 as its Boundary Judge. Evaluation uses the same task set and benchmark configuration across methods, without teacher or Judge assistance. Controlled comparisons use 27B-to-4B unless stated otherwise. Training details are in Appendix B.

Baselines. We compare with the pretrained student (BASE), its offline-distilled checkpoint (SFT), terminal-reward GRPO (Shao et al., 2024), and OPD (Agarwal et al., 2024) with dense teacher supervision. GRPO and OPD collect rollouts to natural termination. TCOD (Wang et al., 2026a) applies a curriculum-scheduled rollout horizon to OPD, while GUIDED-OPD (Li et al., 2026) mixes teacher-generated turns into the rollout.

Table 2: Collection-control comparison in the 27B→4B setting (%). "Full suffix" indicates that collection continues beyond the candidate boundary.

	Collection rule				Mix-Ecom		τ^2-bench	
Policy	Judge	Stops during generation	Full suffix	Trigger	Avg.@k	pass^4	Avg.@k	pass^4
No boundary control	✗	✗	✓	None	49.12	36.18	84.80	70.14
Post-hoc Judge	✓	✗	✓	Error evidence	48.49	37.69	84.53	72.30
TCOD cutoff	✗	✓	✗	Curriculum horizon	49.12	33.17	84.53	71.58
BLADE	✓	✓	✗	Error evidence	**51.76**	**41.21**	**86.42**	**75.90**

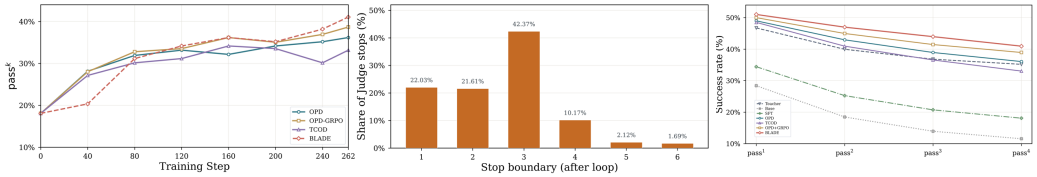


Figure 3: Training and reliability diagnostics on Mix-Ecom (27B→4B). Left: pass^4 across training checkpoints. Middle: distribution of BLADE's Boundary-Judge stops across interaction loops. Right: repeated-run reliability from pass^1 to pass^4.

5.2 Main Results

Table 1 reports the main results across the two benchmarks and teacher–student configurations. BLADE achieves the highest pass^4 among the compared student methods in all four benchmark–scale settings. For the 4B student, it increases Mix-Ecom pass^4 from 36.18% with OPD to 41.21% (+5.03 percentage points), and improves τ^2-bench from 71.58% with TCOD to 75.90% (+4.32 points). The improvement persists at the smaller scale: for the 2B student, BLADE raises pass^4 from 19.10% with Guided-OPD to 20.60% on Mix-Ecom (+1.50 points), and from 65.47% with TCOD to 70.14% on τ^2-bench (+4.67 points). The consistent gains across both benchmark families and student scales indicate that the improvement is not specific to a single interaction environment or model capacity.

The gains are more pronounced under the stricter repeated-run criterion than in average per-trial success. In the 4B setting, BLADE exceeds the strongest competing Avg.@4 results by 2.64 points on Mix-Ecom and 1.62 points on τ^2-bench, while its corresponding margins in pass^4 are 5.03 and 4.32 points. The distinction is clearest on 2B Mix-Ecom: BLADE obtains a slightly lower Avg.@4 than GRPO (30.28% versus 31.16%), yet achieves a higher pass^4 than all compared student baselines (20.60% versus 19.10% for the strongest baseline). Thus, the improvement is concentrated in the fraction of tasks solved consistently across repeated interactions, rather than being explained by a comparable increase in average per-trial success.

5.3 Does Online Error-Conditioned Boundary Control Matter?

Table 2 compares collection strategies that differ in both the timing and trigger of rollout truncation. No boundary control generates each rollout to natural termination. The post-hoc Judge uses error evidence only after the complete suffix has already been generated, whereas TCOD truncates rollouts online according to an error-agnostic curriculum horizon. BLADE instead evaluates each completed nonterminal loop using the realized interaction and environment state, and stops before requesting the next simulated-user response when the current branch contains a consequential error.

BLADE achieves the highest Avg.@4 and pass^4 on both benchmarks. Relative to no boundary control, it improves pass^4 from 36.18% to 41.21% on Mix-Ecom (+5.03 percentage points) and from 70.14% to 75.90% on τ^2-bench (+5.76 points). Comparisons against the two truncation controls further separate the timing and trigger of collection decisions. BLADE exceeds the post-hoc Judge by 3.52 points on Mix-Ecom and 3.60 points on τ^2-bench, consistent with using error evidence

Table 3: Controlled ablations on Mix-Ecom (27B-to-4B, %). OPD supervision is either top-k or Student-sampled-token based; Dynamic marks dynamic sampling for GRPO. Judge indicates whether the Boundary Judge is used during training, and Fusion is advantage-level (Adv.) or loss-level (Loss). Dashes denote inapplicable or unreported cells; Δpass^4 is relative to BLADE.

	Training configuration				Mix-Ecom results (%)				
Setting	OPD supervision	Dynamic sampling	Judge	Fusion	Logistics	After-sales	Avg.@k	pass^4	Δpass^4
OPD	Top-k	–	✗	–	61.11	34.89	49.12	36.18	−5.03
OPD	Sampled token	–	✗	–	54.40	30.49	43.47	32.16	−9.05
GRPO	–	✗	✗	–	31.71	9.34	21.48	9.55	−31.66
GRPO	–	✓	✗	–	35.19	18.96	27.76	15.08	−26.13
Loss-level joint	Top-k	✓	✓	Loss	56.94	32.14	45.60	24.62	−16.59
BLADE	**Top-k**	✓	✓	**Adv.**	**63.89**	**37.36**	**51.76**	**41.21**	**0.00**

before an error-conditioned suffix is generated rather than identifying the same type of problematic prefix only retrospectively. It also exceeds TCOD by 8.04 and 4.32 points, respectively, indicating that online truncation is more effective when tied to realized branch-specific error evidence than to an error-agnostic curriculum horizon.

BLADE also remains efficient despite the additional Boundary-Judge calls. In the primary 27B-to-4B Mix-Ecom configuration, end-to-end online training takes approximately 16 hours, compared with 20 hours for OPD and 22 hours for GRPO under the same hardware setting. Thus, the online boundary mechanism does not introduce a net wall-clock overhead in this setting and is compatible with lower overall training cost.

5.4 Component Analysis

Table 3 examines the main learning-signal choices in the BLADE training pipeline. Dynamic sampling strengthens GRPO-only training, raising Mix-Ecom Avg.@4 from 21.48% to 27.76% and pass^4 from 9.55% to 15.08%. For OPD-only training, top-k teacher scoring improves Avg.@4 from 43.47% to 49.12% and pass^4 from 32.16% to 36.18% relative to sampled-token scoring. The full BLADE configuration further reaches 51.76% Avg.@4 and 41.21% pass^4. Relative to the stronger OPD-only control, the gain is larger in repeated-run reliability (+5.03 points) than in average per-trial success (+2.64 points), indicating that BLADE's improvement is concentrated under the stricter reliability criterion.

We next isolate the integration rule while keeping the Judge, dynamic sampling, signal weights, and token supports fixed. Loss-level integration obtains 45.60% Avg.@4 and 24.62% pass^4, whereas advantage-level integration reaches 51.76% and 41.21%, respectively. Compared with loss-level integration, advantage-level integration improves Avg.@4 by 6.16 points and pass^4 by 16.59 points, supporting the combination of OPD and GRPO signals before policy-ratio clipping rather than clipping the two learning channels separately before summation.

6 Conclusion

We study interaction drift in stateful multi-turn agent post-training, where an erroneous response or tool action can alter the history or environment state that conditions subsequent interactions. We introduce BLADE, which places an online Boundary Judge between a completed agent loop and the next simulated-user turn, stopping consequential-error branches before a drifted suffix is generated. The collected prefix retains dense teacher supervision, while task-level credit is localized to the error-onset loop for stopped rollouts, and the two learning signals are combined at the advantage level. Across Mix-Ecom and τ^2-bench with two teacher–student configurations, BLADE consistently improves strict repeated-run reliability over the compared post-training baselines. These results highlight collection-time boundary control as an important component of policy optimization for stateful multi-turn agents.

AI USE STATEMENT

Large language models were used to assist with drafting and language editing of parts of the manuscript, for brainstorming during the research process, and as a coding assistant during implementation and debugging. All AI-assisted text, code, and suggestions were independently reviewed and verified by the authors. The authors take full responsibility for the final content, claims, code, and results presented in this work.

REFERENCES

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/5be69a584901a26c521c2b51e40a4c20-Abstract-Conference.html.

Victor Barres, Honghua Dong, Soham Ray, Xujie Si, and Karthik Narasimhan. τ^2-bench: Evaluating conversational agents in a dual-control environment, 2025. URL https://arxiv.org/abs/2506.07982.

Chishui Chen, Yaoyou Fan, Te Sun, Yi Yang, Chenghao Sun, Delin Mao, Hongbo Qiao, Zuowei Zhang, Junxi Wang, Chenxing Sun, Yangen Hu, Lu Pan, Xuyang Liu, and Linfeng Zhang. Look ahead before you distill: Future trajectory validation of teacher guidance for agentic on-policy distillation, 2026. URL https://arxiv.org/abs/2608.01953.

Yihao Hu, Zhihao Wen, Xiujin Liu, Pan Wang, Xin Zhang, and Wei Wu. SEAL: Synergistic co-evolution of agents and learning environments, 2026. URL https://arxiv.org/abs/2605.24426.

Gengsheng Li, Mao Zheng, Mingyang Song, Ruiqi Liu, Tianyu Yang, Jie Sun, Qiyong Zhong, Haiyun Guo, Junfeng Fang, Dan Zhang, and Jinqiao Wang. On-policy distillation with curriculum turn-level guidance for multi-turn agents, 2026. URL https://arxiv.org/abs/2606.15912.

Baohao Liao, Hanze Dong, Christof Monz, Xinxing Xu, Li Dong, and Furu Wei. Multi-turn on-policy distillation with prefix replay, 2026. URL https://arxiv.org/abs/2607.04763.

Jiarui Lu, Thomas Holleis, Yizhe Zhang, Bernhard Aumayer, Feng Nan, Haoping Bai, Shuang Ma, Shen Ma, Mengyu Li, Guoli Yin, Zirui Wang, and Ruoming Pang. ToolSandbox: A stateful, conversational, interactive evaluation benchmark for LLM tool use capabilities. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 1160–1183. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.findings-naacl.65. URL https://aclanthology.org/2025.findings-naacl.65/.

Cheng Qian, Zuxin Liu, Akshara Prabhakar, Jielin Qiu, Zhiwei Liu, Haolin Chen, Shirley Kokane, Heng Ji, Weiran Yao, Shelby Heinecke, Silvio Savarese, Caiming Xiong, and Huan Wang. UserRL: Training interactive user-centric agent via reinforcement learning, 2025. URL https://arxiv.org/abs/2509.19736.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024. URL https://arxiv.org/abs/2402.03300.

Weijie Shi, Yanxi Chen, Zexi Li, Xuchen Pan, Yuchang Sun, Jiajie Xu, Xiaofang Zhou, and Yaliang Li. R^3L: Reflect-then-retry reinforcement learning with language-guided exploration, pivotal credit, and positive amplification, 2026. URL https://arxiv.org/abs/2601.03715.

Qitai Tan, Zefang Zong, Mo Li, Yipeng Shi, Yang Li, and Peng Chen. ATOD: Annealed turn-aware on-policy distillation for multi-turn agentic tasks, 2026. URL https://arxiv.org/abs/2606.27814.

Jiaqi Wang, Wenhao Zhang, Weijie Shi, Yaliang Li, and James Cheng. TCOD: Exploring temporal curriculum in on-policy distillation for multi-turn autonomous agents, 2026a. URL https://arxiv.org/abs/2604.24005.

Xingyao Wang, Zihan Wang, Jiateng Liu, Yangyi Chen, Lifan Yuan, Hao Peng, and Heng Ji. MINT: Evaluating LLMs in multi-turn interaction with tools and language feedback. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=jp3gWrMuIZ.

Yinjie Wang, Xuyang Chen, Xiaolong Jin, Mengdi Wang, and Ling Yang. OpenClaw-RL: Train any agent simply by talking, 2026b. URL https://arxiv.org/abs/2603.10165.

Shirley Wu, Michel Galley, Baolin Peng, Hao Cheng, Gavin Li, Yao Dou, Weixin Cai, James Zou, Jure Leskovec, and Jianfeng Gao. CollabLLM: From passive responders to active collaborators. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*. PMLR, 2025. URL https://proceedings.mlr.press/v267/wu25i.html.

Shuo Yang, Jinyang Wu, Zhengxi Lu, Yuhao Shen, Fan Zhang, Lang Feng, Shuai Zhang, Haoran Luo, Zheng Lian, Zhengqi Wen, and Jianhua Tao. OPID: On-policy skill distillation for agentic reinforcement learning, 2026. URL https://arxiv.org/abs/2606.26790.

Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. τ-bench: A benchmark for tool-agent-user interaction in real-world domains. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=roNSXZpUDN.

Qingfei Zhao, Huan Song, Shuyu Tian, Jiawei Shao, and Xuelong Li. Prefix-guided on-policy distillation: Mining golden trajectories from rollouts, 2026a. URL https://arxiv.org/abs/2606.21994.

Yiwen Zhao, Zhihao Wen, Yuchen Mao, Mingxuan Jiang, Yihao Hu, Pan Wang, Xin Zhang, and Wei Wu. Towards better agents for multi-turn user interaction: The next user turn is more than context, 2026b. URL https://arxiv.org/abs/2608.17499.

Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, Jingren Zhou, and Junyang Lin. Group sequence policy optimization, 2025. URL https://arxiv.org/abs/2507.18071.

Qiyong Zhong, Mao Zheng, Mingyang Song, Xin Lin, Jie Sun, Houcheng Jiang, Xiang Wang, and Junfeng Fang. SOD: Step-wise on-policy distillation for small language model agents, 2026. URL https://arxiv.org/abs/2605.07725.

Chenyu Zhou, Xiaoming Shi, Hui Qiu, Xiawu Zheng, Haitao Leng, Yankai Jiang, Shaoguo Liu, Tingting Gao, and Rongrong Ji. Mix-ecom: Towards mixed-type e-commerce dialogues with complex domain rules. In *International Conference on Learning Representations*, 2026a. URL https://iclr.cc/virtual/2026/poster/10010700.

Yuhang Zhou, Lizhu Zhang, Yifan Wu, Mingyi Wang, Bo Peng, Jiayi Liu, Xiangjun Fan, and Zhuokai Zhao. SAGE-OPD: Selective agent-guided intervention for multi-turn on-policy distillation, 2026b. URL https://arxiv.org/abs/2606.19659.

Yuhang Zhou, Kai Zheng, Haoling Li, Dengyun Peng, Can Xu, and Jingjing Chen. TurnOPD: Making on-policy distillation turn-aware for efficient long-horizon agent training, 2026c. URL https://arxiv.org/abs/2607.05804.