ROLLOUT-TIME INTERVENTION: UNLOCKING AGENTIC REINFORCEMENT LEARNING BEYOND POLICY LIMITS

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement learning (RL) has emerged as a key paradigm for improving large language model (LLM) agents, enabling stronger reasoning, planning, and tool-use in interactive environments. However, existing agent RL methods remain fundamentally constrained by an on-policy reasoning bottleneck: when tasks exceed the model's current capability boundary, effective trajectories become extremely sparse and training often collapses. We introduce **Rollout-Time Intervention (RTI)**, a technique designed to elicit successful trajectories beyond the policy's intrinsic exploration capacity. RTI operates by leveraging reward-side evidence (i.e., evaluation rubrics) during rollout to convert erroneous trajectories into successful ones, providing effective learning signals for tasks that the policy rarely solves on its own. Although intervention trajectories are inherently off-policy, we find that they can effectively guide policy optimization when trajectories with excessive distributional shifts are filtered out. Extensive experiments across five benchmarks demonstrate that RTI consistently outperforms RL baselines, enabling the post-trained policy to approach the performance of substantially stronger contemporary LLMs and, more importantly, transcend its original capability boundary.

1 INTRODUCTION

LLMs are rapidly evolving from static instruction-following systems into interactive agents that can collaborate with external environments to achieve goals (Team, 2026; Froger et al., 2025). RL has been a key driver of this shift, enabling advanced capabilities such as self-reflection and long-horizon planning (Guo et al., 2025; Qian et al., 2026). These advances have fueled strong performance in challenging agent settings, including mathematical reasoning with code interpreters (Feng et al., 2025), deep research with search engines (Jin et al., 2025; Sun et al., 2026), and complex real-world tool use (Qian et al., 2026; Gao et al., 2026; Yang et al., 2026; Dong et al., 2026b; Li et al., 2026).

Nevertheless, standard RL is limited by its on-policy nature: it mainly amplifies behaviors already supported by the current policy, rather than inducing new reasoning patterns (Zhao et al., 2025; Yue et al., 2026). As a result, when tasks lie beyond the model's capability boundary, successful trajectories become extremely sparse, leading to unstable training or even collapse (Figure 1(a)). Teacher-guided methods such as LUFFY (Yan et al., 2026) and OPD (Lu & Thinking Machines Lab, 2025) can partially mitigate this issue, but their improvements remain bounded by teacher capability.

To further understand these failures, we examine failed agent trajectories and find a recurring pattern: the agent often possesses the knowledge and tools needed to complete the task, but makes a small early reasoning or action error at a task-critical step. As illustrated in Figure 1, in τ^2-Telecom, the agent must guide the user to invoke a user-side tool, yet the model may incorrectly treat it as an agent-accessible tool and repeatedly attempt to call it; in Navigation, user queries are often ambiguous, and the agent must interpret noisy POI search results in the context of the user's intent rather than blindly treating a returned POI as the target. In both cases, a single incorrect interaction strategy can steer subsequent reasoning into a failure loop, while correcting this early decision is often sufficient for the remaining trajectory to proceed normally. These observations motivate us: *if such early reasoning errors can be corrected so that the rollout succeeds, can RL learn from these corrected trajectories and thereby push the policy beyond its original reasoning limits (Figure 1(c))?*

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.

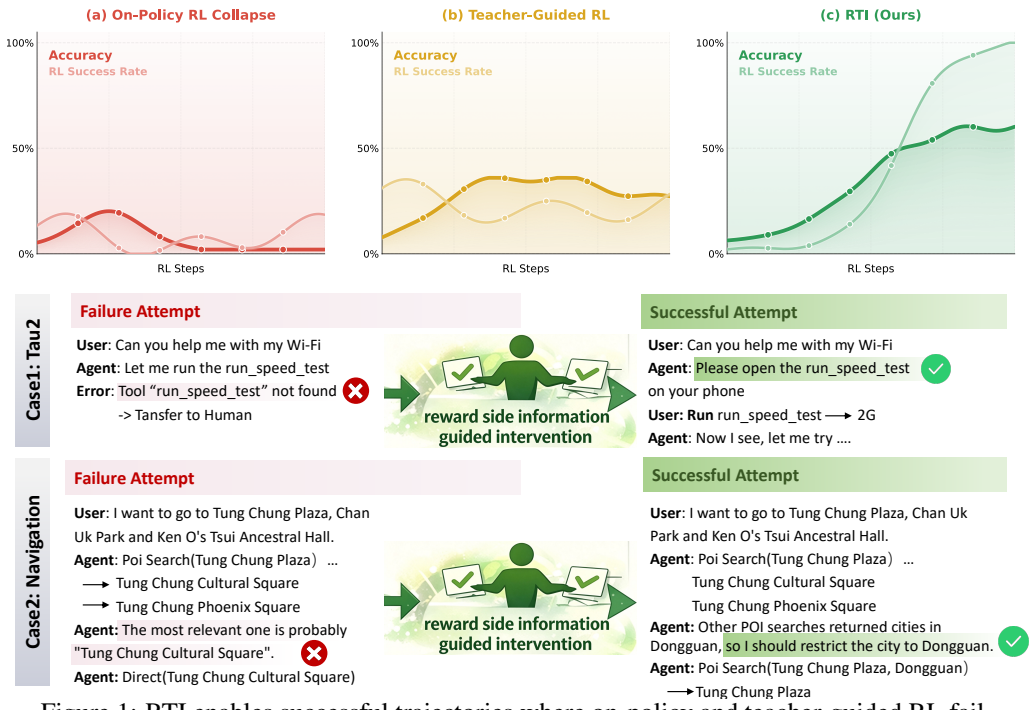


Figure 1: RTI enables successful trajectories where on-policy and teacher-guided RL fail.

However, such intervention introduces a central technical challenge: the corrected trajectories are no longer sampled from the original policy, but generated by an intervention-induced behavior policy. This discrepancy poses two fundamental risks. First, privileged reward-side information may leak into the reasoning process, producing shortcut-driven or behaviorally implausible trajectories that could not arise under genuine task interaction. Second, intervention may induce substantial distributional shifts from the original policy. In PPO-style optimization, including GRPO, such shifts can push importance ratios beyond the clipping range, suppressing otherwise useful learning signals, while low-probability actions may yield high-variance gradients that destabilize training.

To this end, we propose **Rollout-Time Intervention (RTI)**, comprising three components: (1) *reward-aware multi-agent intervention*, which employs a Validator–Corrector–Verifier pipeline to transform failed rollouts into successful intervention trajectories while preventing privileged-information leakage; (2) *perplexity-based distribution gating*, which filters intervened trajectories that deviate excessively from the current policy; and (3) *reasoning-shift-aware token reweighting*, which concentrates policy updates on critical tokens where intervention induces substantial strategy changes. Together, these components enable stable learning from successful trajectories beyond the policy's unaided exploration capacity, thereby pushing the policy beyond its original capability boundary. Experiments on public and real-world benchmarks show that RTI substantially outperforms baselines and achieves a breakthrough in the boundaries of reasoning capabilities.

2 Related Work

Agentic Reasoning for LLMs. Agentic reasoning reframes LLMs as autonomous agents that combine planning, acting, and learning through interaction (Wei et al., 2026), as exemplified by ReAct (Yao et al., 2023), tool-use and search-based agents, and workflow-oriented systems with verifiable sub-tasks (Yao et al., 2023; Jin et al., 2025; Sun et al., 2026; Feng et al., 2025). This paradigm requires agents to plan over long horizons, navigate partial observability, and actively improve through feedback (Wang et al., 2024). Existing methods for agent training are as follows:

(i) Data-synthesis-driven SFT improves tool use by generating interaction trajectories for supervised fine-tuning, including ToolLLM (Qin et al., 2024), ToolACE (Liu et al., 2025), APIGen-MT (Prabhakar et al., 2026), TopoCurate (Yang et al., 2026), and simulation or self-evolving pipelines for multi-turn data generation (Li et al., 2025; Gao et al., 2026). These methods are effective but depend on large-scale trajectory synthesis. *(ii) Pure RL with verifiable rewards* has also shown

promise when outcomes are easy to verify. ReTool (Feng et al., 2025) incorporates code execution for competition-level problem solving, Search-R1 (Jin et al., 2025) learns search-engine use with emergent multi-query behavior, ZeroSearch (Sun et al., 2026) replaces real retrieval with LLM simulation, and ToolRL (Qian et al., 2026) shows that fine-grained format and outcome rewards can support stable multi-step tool-use RL. More recently, AEPO (Dong et al., 2026a) further improves optimization stability under long-horizon, multi-turn tool use through entropy-aware policy updates.

Teacher-Guided Training for LLMs. When tasks lie beyond the model's capability boundary, successful rollouts become extremely sparse and the policy is rarely exposed to truly corrective trajectories. To alleviate this, recent work studies teacher-guided training in two forms. One line provides *token-level credit assignment* using stronger teachers, often via trajectory-level teacher-student KL to derive dense rewards (Gu et al., 2024; Lu & Thinking Machines Lab, 2025). Another line introduces *off-policy successful trajectories*: LUFFY (Yan et al., 2026), for instance, combines offline expert data with online rollouts and uses importance sampling to handle distribution shift.

While these methods can improve exploration and learning, they all fundamentally depend on stronger teacher models, making their gains bounded by the teacher's capability ceiling.

3 PRELIMINARIES

Problem Setting. We formulate the agentic task as a trajectory generation problem. Let $\mathcal{T}$ denote the set of tools. Given a query x sampled from a task distribution $\mathcal{D}$, the policy π_θ produces a multi-step interaction trajectory τ consisting of interleaved chain-of-thought reasoning z_k, tool invocations $a_k \in \mathcal{T}$, environmental feedback o_k, and a final answer y: $\tau = [z_1, a_1, o_1, \ldots, z_k, a_k, o_k, \ldots, y]$. A verifiable reward function $r(x, \tau) \in \{0, 1\}$ evaluates the trajectory.

Group Relative Policy Optimization. GRPO (Shao et al., 2024) is a widely adopted RL algorithm. For each query x, GRPO samples a group of G trajectories $\{\tau_i\}_{i=1}^{G} \sim \pi_\theta(\cdot \mid x)$ and normalizes the rewards into per-sample advantages:

$$A_i = \frac{r(x, \tau_i) - \mu_r}{\sigma_r + \epsilon}, \quad \mu_r = \tfrac{1}{G}\textstyle\sum_{i=1}^{G} r(x, \tau_i), \quad \sigma_r = \sqrt{\tfrac{1}{G}\textstyle\sum_{i=1}^{G}(r(x, \tau_i) - \mu_r)^2}. \tag{1}$$

The policy is then updated by maximizing the clipped surrogate objective $\mathcal{J}_{\mathrm{GRPO}}(\theta)$:

$$\mathbb{E}_{x,\{\tau_i\}}\left[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|\tau_i|}\sum_{t=1}^{|\tau_i|}\min\Big(\rho^t_{\mathrm{grpo},i}A_i, \mathrm{clip}(\rho^t_{\mathrm{grpo},i}, 1-\epsilon_{\mathrm{clip}}, 1+\epsilon_{\mathrm{clip}})\, A_i\Big)\right] - \beta\mathbb{D}_{\mathrm{KL}}(\pi_\theta \| \pi_{\mathrm{ref}}), \tag{2}$$

where $\rho^t_{\mathrm{grpo},i} = \frac{\pi_\theta(\tau_i^t | x, \tau_i^{<t})}{\pi_{\mathrm{old}}(\tau_i^t | x, \tau_i^{<t})}$ is the per-token probability ratio between the current policy and the behavior policy π_{old} that generated τ_i, and $\beta > 0$ controls the strength of the KL regularization.

4 ROLLOUT-TIME INTERVENTION

To overcome the exploration bottleneck of on-policy RL, we propose **RTI**, a technique comprising three components as shown in Figure 2. First, *reward-aware multi-agent system (MAS) intervention* calibrates task-critical errors at the step level, transforming failed rollouts into successful intervention trajectories while preventing privileged-information leakage. Second, *perplexity-based distribution gating* filters interventions that deviate excessively from the rollout policy, mitigating ineffective or unstable updates caused by severe off-policy shifts. Third, *reasoning-shift-aware token reweighting* emphasizes the critical tokens responsible for intervention-induced strategy changes, enabling more precise credit assignment. Together, these components enable stable learning from successful trajectories beyond the policy's unaided exploration capacity.

4.1 REWARD-AWARE MAS-BASED INTERVENTION

Given a query x, only when the entire group fails (i.e., all G trajectories generated by π_θ receive zero reward), we invoke the intervention pipeline, an MAS consisting of a *Reward-Oriented Trajectory Validator*, a *Policy-Aligned Trajectory Corrector*, and an *Integrity-Constrained Intervention Verifier*. By default, all three MAS roles are instantiated with the policy model itself.

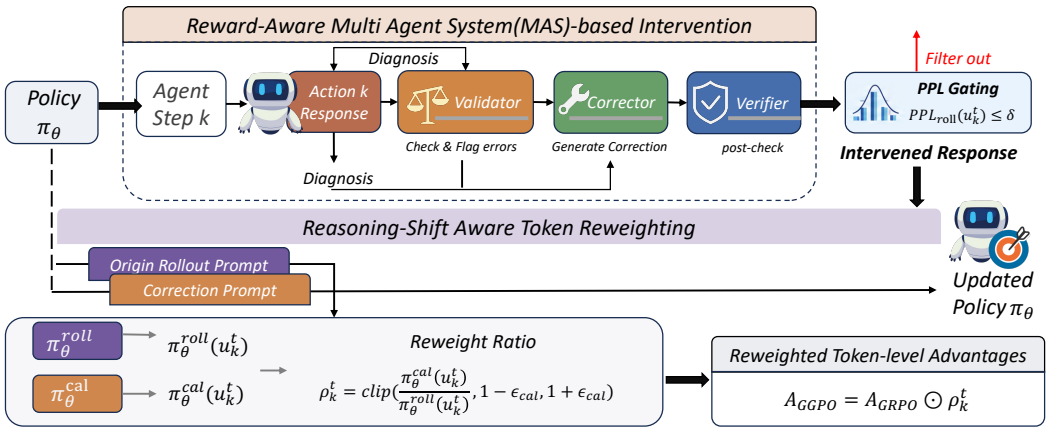


Figure 2: Overview of the proposed RTI framework.

Reward-Oriented Trajectory Validator The validator identifies whether the current rollout step has *harmfully deviated* from a valid path to the gold outcome. Given the query x and the observable history before step k: $h_{1:k-1} = (z_1, a_1, o_1, \ldots, z_{k-1}, a_{k-1}, o_{k-1})$, the current response $\hat{u}_k$, and the reward-side evidence ℓ, it determines whether $\hat{u}_k := (z_k, a_k)$ is still compatible with successful task completion. Importantly, the validator is *reward-oriented* rather than process-prescriptive: it does not penalize suboptimal but recoverable decisions, alternative valid action orders, or stylistic differences, and instead flags only those errors that are likely to derail the trajectory. In this way, the validator detects only *task-critical* mistakes, matching our central motivation that many rollout failures arise from early local errors rather than missing end-task capability. The prompt is shown below:

VALIDATOR prompt (schematic)

Inputs. Observable history $h_{1:k-1}$; current response $\hat{u}_k$; reward-side evidence ℓ.

Body. *(i) Observable evidence* listing prior turns and tool outputs; *(ii) Required actions* derived from ℓ separated into agent-side and (for telecom) user-side groups; *(iii) Hard rules*: false-claim, ...; *(iv) Negative space* explicitly disallowing invalidation due to suboptimal-but-recoverable order.

Output. JSON {`is_valid`, `error_types`, `diagnosis`, `suggest_next`}.

The validator returns a structured judgment including whether the step is valid, the error type, a brief diagnosis, and a next-step suggestion. The diagnosis explains why the current step is inconsistent with the gold trajectory, while the suggested next step only indicates the direction of repair.

Policy-Aligned Trajectory Corrector When the validator detects a harmful deviation, the corrector rewrites the current step into an intervened response $\tilde{u}_k$. The key requirement is that the corrected step should remain a plausible continuation of the agent's own reasoning process. Concretely, the corrector takes as input the history $h_{1:k-1}$, tool schema $\mathcal{T}$, the original draft $\hat{u}_k$, and the validator feedback, and produces a revised response that fixes the critical error while preserving the original response style, action format, and local reasoning structure as much as possible.

CORRECTOR prompt (schematic)

Inputs. History $h_{1:k-1}$; tool schemas $\mathcal{T}$; prior draft $\hat{u}_k$ (style reference); validator feedback (diagnosis and suggested next step); reward-side evidence ℓ (privileged reference, never echoed).

Body. *(i)*"silent calibrator" specification; *(ii)* four generation principles—*style alignment*, *progressive correction*, *leakage suppression*, *minimal intervention*; *(iii)* domain-specific requirements; *(iv)* explicit instruction that the intervened response must be derivable from visible history alone.

Output. JSON {`reasoning`, `tool_calls`, `content`}.

Our goal is not to replace the trajectory with an oracle solution, but to generate a minimally corrected version that stays close to the policy's native behavior while removing the decisive error. Therefore, the corrector is instructed to follow the visible history and imitate the original reasoning pattern.

Integrity-Constrained Intervention Verifier Finally, the verifier checks whether the intervened response is suitable. Given the original context and the corrected response $\tilde{u}_k$, it verifies three properties: (i) the correction remains consistent with the gold requirement, (ii) it does not leak privileged information, and (iii) it is still a reasonable continuation of the original policy behavior.

VERIFIER prompt (schematic)

Inputs. User query; history $h_{1:k-1}$; tool schemas; prior draft $\hat{u}_k$; intervened response $\tilde{u}_k$; evaluation criteria from ℓ.
Body. *(i)* visibility specification (only the user query, tools, and history may justify $\tilde{u}_k$); *(ii)* four rejection criteria: *leak* of privileged keywords, *fabricated tool result*, *wrong tool arguments* contradicting history or violating explicitly specified parameter values, *unreasonable intervention*; *(iii)* an enumerated allow-list of legitimate behaviors.
Output. JSON {`ok`, `reasons`, `diagnosis`}.

The verifier closes the intervention loop by ensuring that accepted trajectories are aligned with the gold path, do not leak privileged information, and stay behaviorally close to the original policy.

4.2 PERPLEXITY-BASED DISTRIBUTION GATING

Although the intervention verifier removes trajectories with information leakage or implausible reasoning, the accepted trajectories may still deviate substantially from the rollout policy. When an intervened trajectory has low likelihood under the current policy, its probability ratios may fall outside the PPO clipping range, suppressing useful learning signals; meanwhile, learning from such low-probability behavior can produce poorly conditioned, high-variance gradient estimates and destabilize optimization. We therefore introduce *perplexity-based distribution gating* to filter intervention trajectories with excessive off-policy deviation before policy optimization.

Specifically, after obtaining an intervened response, we remove all intervention-specific information and rescore it under the rollout policy conditioned solely on the original observable history. For $\tilde{u}_k = (\tilde{u}_k^1, \ldots, \tilde{u}_k^{T_k})$ with T_k tokens at step k, its perplexity is computed as

$$\mathrm{PPL}_{\mathrm{roll}}(\tilde{u}_k) = \exp\left(-\frac{1}{T_k}\sum_{t=1}^{T_k}\log \pi_\theta^{\mathrm{roll}}\left(\tilde{u}_k^t \mid h_{1:k-1}, \tilde{u}_k^{<t}\right)\right), \tag{3}$$

where T_k is the number of tokens in the intervened response. Higher perplexity indicates a larger deviation from the behavior naturally supported by the policy model. We retain $\tilde{u}_k$ with a threshold δ:

$$\mathrm{PPL}_{\mathrm{roll}}(\tilde{u}_k) \le \delta. \tag{4}$$

4.3 REASONING-SHIFT-AWARE TOKEN REWEIGHTING

The intervention replaces a single failed rollout in the group with an intervened trajectory $\tilde{\tau}$, which is sampled from a behavior distribution π^{cal} different from the rollout distribution $\pi_\theta^{\mathrm{roll}}$. Importantly, not all corrected tokens contribute equally to learning: in many cases, a small subset of tokens is critical for changing the reasoning direction and turning a failed trajectory into a successful one. We therefore introduce a *reasoning-shift-aware* token reweighting, which focuses the update on those intervention-induced tokens that carry the key reasoning correction.

Consider an intervened response segment $\tilde{u}_k = (\tilde{u}_k^1, \ldots, \tilde{u}_k^{T_k})$ with T_k tokens at step k. We define a token-level reweighting coefficient based on the likelihood ratio between the two conditions:

$$\rho^t = \exp\big(\log \pi^{\mathrm{cal}}(\tilde{u}_k^t \mid h_{1:k-1}, \tilde{u}_k^{<t}) - \log \pi_\theta^{\mathrm{roll}}(\tilde{u}_k^t \mid h_{1:k-1}, \tilde{u}_k^{<t})\big). \tag{5}$$

where π^{cal} and $\pi_\theta^{\mathrm{roll}}$ denote the token likelihoods under the correction prompt p_{cal} (which additionally includes the validator feedback) and the rollout prompt p_{roll}, respectively.

A larger ratio means that the token is much more characteristic of the corrected reasoning path than of the original rollout path and should therefore receive greater emphasis during learning. To avoid unstable updates caused by extreme ratios, we apply a two-sided clip around 1 so that the ratio acts as a bounded per-token reweighting coefficient:

$$\tilde{\rho}_{\mathrm{cal}}^t = \mathrm{clip}\big(\rho^t,\ 1-\epsilon_{\mathrm{cal}},\ 1+\epsilon_{\mathrm{cal}}\big), \tag{6}$$

where $\epsilon_{\mathrm{cal}} \in (0,1)$, treated as a stop-gradient constant that only reweights the advantage of the tokens it emphasizes. Note that the reweighting coefficient $\tilde{\rho}_{\mathrm{cal}}^t$ is only computed on tokens from intervened trajectories; for purely on-policy trajectories (i.e., when at least one rollout in the group already succeeds and no intervention is invoked), we set $\tilde{\rho}_{\mathrm{cal}}^t \equiv 1$, so that RTI reduces exactly to GRPO on those groups. The final policy-gradient loss applies the reasoning-shift reweighting to the token-level GRPO clipped surrogate. Concretely, expanding $\mathcal{J}_{\mathrm{GRPO}}$ from Eq. (2) yields:

$$\mathcal{J}_{\mathrm{RTI}}(\theta) = \mathbb{E}_{x,\{\tilde{\tau}_i\}}\left[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|\tilde{\tau}_i|}\sum_{t=1}^{|\tilde{\tau}_i|}\tilde{\rho}_{\mathrm{cal}}^t \min\Big(\rho_{\mathrm{grpo},i}^t A_i,\ \mathrm{clip}(\rho_{\mathrm{grpo},i}^t, 1-\epsilon_{\mathrm{clip}}, 1+\epsilon_{\mathrm{clip}})\, A_i\Big)\right]. \tag{7}$$

5 EXPERIMENTS

5.1 EXPERIMENTAL SETUP

Benchmarks. We evaluate RTI across five benchmarks. First, we construct **Navigation**, a multi-waypoint navigation benchmark derived from real-world map scenarios, where agents must plan routes and complete tasks by interacting with real APIs from the Amap Open Platform. Second, we adopt the publicly available multi-turn dialogue benchmark τ^2-Bench (Barres et al., 2025) and evaluate on its **Telecom**, **Retail**, and **Airline** domains. The corresponding RL training instances are drawn from the AReaL dataset (Fu et al., 2026). Finally, we consider Claw-Eval (Ye et al., 2026), a human-verified benchmark comprising highly diverse tasks grounded in real-world environments. To evaluate whether RTI can improve performance beyond the policy's native capabilities, we select 20 challenging T-series tasks on which the base policy rarely succeeds, forming **Claw-Eval-Hard20**, and synthesize task-specific training data for these tasks.

Evaluation Metrics. We report Pass@k and Pass^k for $k = 1, \ldots, 4$. Pass@k measures the proportion of tasks for which at least one of the k independent rollouts succeeds, while Pass^k measures the proportion of tasks for which all k rollouts succeed. The p@1 is the mean Pass@1 over the four evaluations, while avg p@k and avg p^k additionally average the corresponding metric over $k = 1, \ldots, 4$. For the publicly available benchmarks, we follow their original evaluation protocols to determine whether a trajectory is successful. For Navigation, a trajectory is considered successful if the agent invokes the `direction` tool with the correct parameters.

Baselines. We compare RTI against: **Reference models** include the backbones (Qwen3-8B / Qwen3.5-9B), larger same-family variants (Qwen3-235B and Qwen3.5-397B-A17B), and contemporaneous proprietary models, paired by generation: GPT-4.1 and Claude-4-Sonnet for Qwen3-8B, and GPT-5.1 and Claude-4.6-Sonnet for Qwen3.5-9B. **On-policy RL baselines** include GRPO (Shao et al., 2024), GSPO (Zheng et al., 2025), and AEPO (Dong et al., 2026a). **Teacher-guided methods** include OPD (Lu & Thinking Machines Lab, 2025) and LUFFY (Yan et al., 2026).

Further details on the benchmarks and baselines are provided in Appendices C.2 and C.3.

5.2 MAIN RESULTS

Tables 1 and 2 show that RTI consistently improves agent performance, with the largest gains on tasks where the backbone and existing RL methods struggle. Three findings stand out.

First, RTI overcomes bottlenecks in domain-specific interaction. With Qwen3-8B, RTI improves Pass@1 over GRPO by 24.56 points on τ^2-Telecom and 22.00 points on Navigation. Telecom requires the agent to guide users through actions it cannot perform itself; Navigation requires it to recognize noisy POI results and refine its search before requesting directions. Corrected rollouts expose the policy to successful strategies for both failure modes (Appendix C.2).

Second, updating trajectories with a teacher does not necessarily overcome the teacher's capability limit. OPD and LUFFY provide additional learning signals but achieve only 26.25% and 30.00% Pass@1, respectively, on Claw-Eval-Hard20 with Qwen3-8B. RTI reaches 50.00%, indicating that its gains cannot be explained solely by supplying trajectories beyond unaided on-policy exploration.

Third, RTI extends these gains to diverse hard tasks and stronger backbones. On Claw-Eval-Hard20, RTI exceeds GRPO by 27.50 points with Qwen3-8B. Although Qwen3.5-9B already performs strongly on several other benchmarks, its baseline Pass@1 on this subset is 26.25%; RTI raises it to 85.00%, 23.75 points above the strongest competing method, LUFFY. These results suggest that RTI expands the set of tasks the policy can solve across different backbone capability levels.

5.3 ABLATION STUDIES

To isolate each component's contribution, we conduct ablation studies on Navigation and τ^2-Telecom with Qwen3-8B, evaluating three variants: *RTI w/o Post-Check*, which removes the *Integrity-Constrained Intervention Verifier*; *RTI w/o PPL Gating*, which removes the *PPL-Based Distribution Gating*; and *RTI w/o Reweight*, which removes the *Reasoning-Shift-Aware Token Reweighting*.

Table 1: Main results on five benchmarks with the Qwen3-8B backbone.

	Navigation			τ^2-Telecom			τ^2-Retail			τ^2-Airline			Claw-Eval-Hard20		
Method	p@1	avg p@k	avg p^k	p@1	avg p@k	avg p^k	p@1	avg p@k	avg p^k	p@1	avg p@k	avg p^k	p@1	avg p@k	avg p^k
Qwen3-8B	22.67	28.50	14.17	36.84	57.02	21.49	45.61	71.93	35.53	31.00	50.50	16.00	18.75	24.38	15.94
Qwen3-235B-A22B	28.00	35.50	16.67	44.74	64.04	28.07	64.04	79.61	47.15	50.50	64.50	32.00	20.00	20.00	12.50
GPT-4.1	23.33	31.33	18.67	39.47	58.99	23.68	71.93	83.55	52.19	53.00	66.50	34.00	30.00	30.00	18.75
Claude-4-Sonnet	26.00	32.83	20.67	49.12	67.76	32.02	**74.56**	**85.09**	53.95	54.00	68.50	**35.50**	40.00	46.25	30.00
GRPO	35.33	40.00	29.00	36.84	55.48	21.71	67.54	80.92	50.22	46.50	57.50	26.50	22.50	24.17	20.83
GSPO	28.00	34.50	18.00	44.96	64.48	30.66	63.16	74.78	46.27	39.00	55.50	22.50	26.25	31.67	19.90
AEPO	34.00	37.67	26.17	45.68	62.41	29.23	63.16	81.14	49.78	51.00	65.50	33.50	20.00	27.50	16.25
OPD	37.33	41.50	32.00	55.26	68.64	36.62	68.42	82.24	46.71	47.50	56.50	30.50	26.25	31.46	21.98
LUFFY	36.00	51.50	26.67	49.12	72.59	27.19	71.93	84.87	49.34	46.00	59.50	26.50	30.00	32.81	27.19
RTI	**57.33**	**62.33**	**46.67**	**61.40**	**81.14**	**44.30**	73.68	**85.09**	**55.92**	**56.00**	**70.00**	**35.50**	**50.00**	**62.29**	**37.71**

Table 2: Main results on five benchmarks with the Qwen3.5-9B backbone.

	Navigation			τ^2-Telecom			τ^2-Retail			τ^2-Airline			Claw-Eval-Hard20		
Method	p@1	avg p@k	avg p^k	p@1	avg p@k	avg p^k	p@1	avg p@k	avg p^k	p@1	avg p@k	avg p^k	p@1	avg p@k	avg p^k
Qwen3.5-9B	39.33	53.00	26.17	92.11	96.49	89.69	82.46	93.64	71.05	60.50	73.00	47.00	26.25	31.67	19.90
Qwen3.5-397B-A17B	44.00	53.67	38.00	95.61	98.25	93.64	84.21	94.74	73.90	63.00	77.00	51.00	30.00	40.00	26.25
GPT-5.1	42.00	54.00	30.17	95.61	98.03	93.64	77.19	92.11	62.28	66.50	79.00	53.00	40.00	47.50	40.00
Claude-4.6-Sonnet	58.00	64.17	53.83	**97.37**	**98.90**	**96.05**	**91.23**	**96.93**	**85.53**	80.00	**88.50**	**71.50**	51.25	59.27	44.17
GRPO	71.33	74.83	66.33	95.61	97.59	93.64	84.21	94.74	73.90	75.00	82.50	63.00	52.50	57.71	45.42
GSPO	68.67	72.33	63.17	95.61	98.03	94.30	85.09	94.96	75.44	79.00	85.50	68.00	51.25	58.75	44.69
AEPO	70.67	77.17	64.83	96.49	98.47	94.52	85.96	95.18	76.75	78.50	86.50	70.00	51.25	56.66	44.90
OPD	70.00	75.00	58.67	96.49	98.68	95.18	86.84	95.61	78.29	79.50	84.50	65.00	51.25	55.94	47.50
LUFFY	71.33	76.17	65.17	96.49	98.68	95.18	85.96	95.18	76.75	78.00	86.50	67.00	61.25	66.67	54.90
RTI	**74.67**	**78.50**	**68.67**	96.49	98.68	95.18	88.60	96.05	81.14	**82.00**	87.50	70.50	**85.00**	**94.17**	**73.96**

Effect of Post-Check. Removing the Post-Check leads to a large degradation on both benchmarks. On τ^2-Telecom, the variant loses almost all of RTI's gains and falls back to the GRPO level; on Navigation, it even drops below GRPO (26.00 vs. 35.33 in Pass@1). Without integrity verification, the admitted interventions are frequently leak-contaminated or otherwise unlearnable, and the corrupted supervision can actively harm the policy rather than merely fail to help. A representative failure case is given in Appendix D.2.

Effect of PPL Gating. Removing the PPL Gate yields the second-largest degradation on τ^2-Telecom: pass@1 decreases from 61.40 to 54.39, and the all-of-k metrics drop even more sharply (pass^2: 46.49→27.19, pass^4: 31.58→18.42). This confirms that bounding the distributional shift of intervened responses is essential: interventions that deviate excessively from the rollout policy provide unstable learning signals, so filtering them out preserves consistent optimization.

Effect of Reweight. Removing the token reweighting results in the mildest overall change, yet the degradation is systematic: RTI is at least as good on all sixteen metrics, with the largest gaps on the all-of-k metrics. This suggests that reweighting mainly sharpens credit assignment toward the decisive intervention tokens, improving the stability with which the corrected reasoning is reproduced across repeated attempts, whereas the exploration breadth is already supplied by the intervention.

5.4 Hyperparameter Study

We further study the gating threshold δ in the PPL-based distribution gating (Section 4.2, Eq. (4)), which controls the maximum deviation from the rollout policy that an intervened response may exhibit before being discarded. We sweep $\delta \in \{0, 3, 5, 20, +\infty\}$ on Navigation with the Qwen3-8B backbone. At one extreme ($\delta = 0$), the gate rejects all interventions, so training falls back to the no-intervention limit; at the other ($\delta = +\infty$), no filtering is applied at all.

As shown in Figure 3, the two extremes behave as expected: with $\delta = 0$, training degenerates into the GRPO run and the success rate stays nearly flat, whereas $\delta = +\infty$ improves steadily but converges to a clearly lower level. Performance improves as δ grows from 3 to 5 and degrades at $\delta = 20$: a

Table 3: Ablation study on Navigation and τ^2-Telecom with the Qwen3-8B backbone.

	Navigation								τ^2-Telecom							
Method	PASS@1	PASS@2	PASS@3	PASS@4	PASS^1	PASS^2	PASS^3	PASS^4	PASS@1	PASS@2	PASS@3	PASS@4	PASS^1	PASS^2	PASS^3	PASS^4
GRPO	35.33	40.00	41.33	43.33	35.33	29.33	26.67	24.67	36.84	52.63	63.16	69.30	36.84	21.93	17.54	10.53
RTI w/o Post-Check	26.00	33.33	36.00	38.00	26.00	14.00	9.33	7.33	38.60	53.51	59.65	67.54	38.60	24.56	16.67	14.91
RTI w/o PPL Gating	50.00	60.00	64.00	65.33	50.00	37.33	29.33	24.00	54.39	68.42	75.44	79.82	54.39	27.19	21.05	18.42
RTI w/o Reweight	55.33	58.67	60.67	62.00	55.33	45.33	42.00	33.33	60.53	79.82	85.96	88.60	60.53	38.60	30.70	20.18
RTI	**57.33**	**61.33**	**64.67**	**66.00**	**57.33**	**48.00**	**42.00**	**39.33**	**61.40**	**83.33**	**87.72**	**92.11**	**61.40**	**46.49**	**37.72**	**31.58**

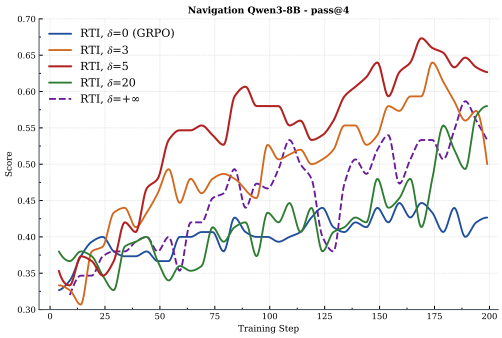


Figure 3: Effect of the gating threshold δ on Pass@4 for Navigation with Qwen3-8B.

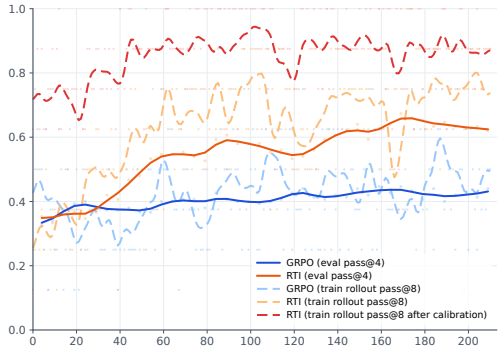


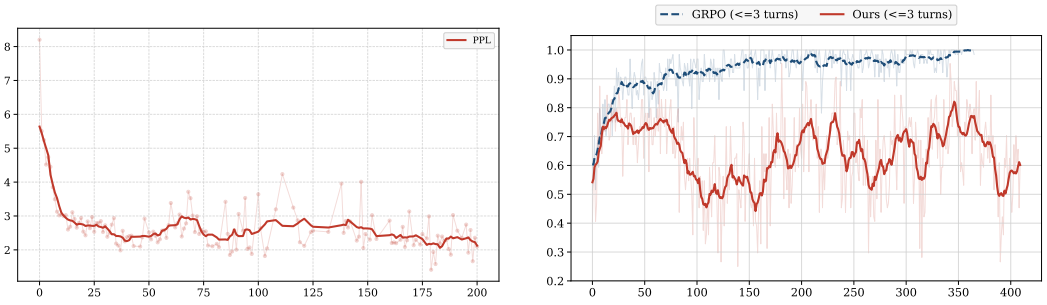
Figure 4: Training dynamics on Navigation under GRPO vs. RTI with the Qwen3-8B backbone.

threshold that is too small discards too many corrective trajectories to sustain late-stage gains, while a threshold that is too large admits severely shifted interventions, whose low-probability, high-variance updates destabilize optimization—the $\delta = 20$ curve indeed exhibits the highest volatility among all runs. The best trade-off is achieved at $\delta = 5$, which preserves informative corrections while filtering out excessively deviated ones: it tracks $\delta = 3$ closely in early training, pulls ahead in later stages, and converges to the highest final success rate. We therefore adopt $\delta = 5$ in all experiments.

5.5 Further Analysis

Training Dynamics Comparison. Figure 4 compares evaluation Pass@4 (solid lines) with training rollout Pass@8 (dashed lines) on Navigation. All curves are Gaussian-smoothed for visualization. For RTI, the two dashed curves measure rollout success before and after intervention. The large gap at early stage between them shows that intervention converts many failed rollout groups into successful trajectories, providing training signals where GRPO receives few. Crucially, RTI's pre-intervention Pass@8 also rises over training and exceeds GRPO's rollout success, while its evaluation Pass@4 improves in tandem. Thus, the corrected trajectories support effective policy updates: the model increasingly succeeds without intervention, rather than relying on intervention only at rollout time.

Dynamics of Intervened-Step Perplexity and Reasoning Patterns. To further verify whether the intervened behaviors are truly internalized by the policy, we examine the average PPL under the policy model per successful intervened step, which measures the distributional gap between the intervened trajectory and the policy's native generation distribution. As shown in Figure 5a, the curve drops steadily during training: those that are still triggered yield intervened responses that are increasingly close to the policy's own output distribution. It provides direct evidence that the policy is not merely *exploiting* the intervention at training time but is actually *absorbing* the intervened reasoning pattern into its native behavior. Additionally, we analyze Navigation to examine how training under RTI versus GRPO changes the model's *resulting reasoning patterns*. Figure 5b contrasts the per-iteration distribution of interaction turns under GRPO and RTI on Navigation. Compared with GRPO, RTI maintains a broader and more multi-modal turn distribution throughout training. In particular, trajectories exceeding 3 turns continue to retain non-trivial mass even at later stages, indicating that the policy preserves the capacity for self-reflective re-querying rather than collapsing into a single short interaction template. A case is provided in Appendix D.4.



(a) Dynamics of Intervention PPL under RTI. (b) Per-iteration distribution of interaction turns.

Figure 5: Further analysis of intervention PPL dynamics and resulting reasoning-pattern distributions.

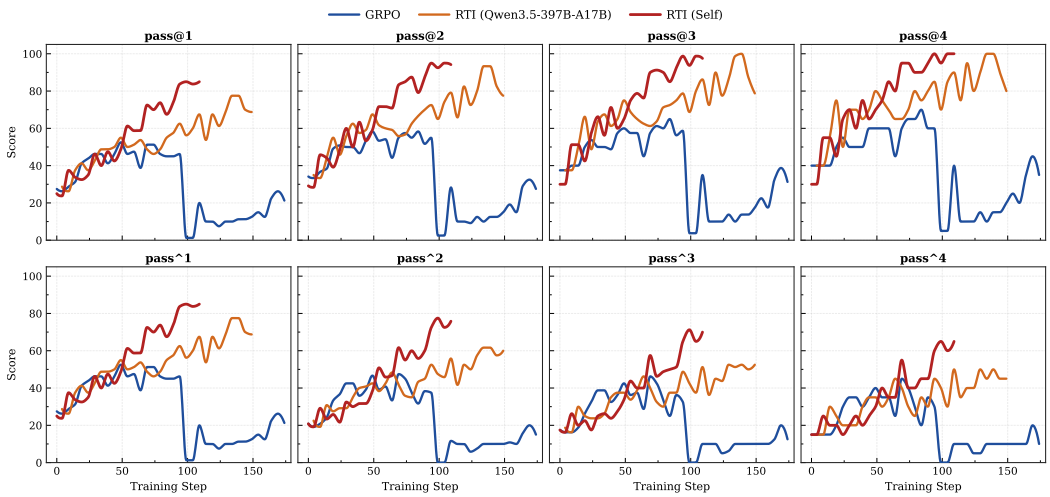


Figure 6: Training curves on Claw-Eval-Hard20 with the Qwen3.5-9B backbone.

Generalization Across Hard Agent Tasks. All preceding RTI experiments use self-intervention. To examine whether a stronger intervening model improves learning, we additionally use Qwen3.5-397B-A17B to correct rollouts from the Qwen3.5-9B policy. Figure 6 compares this variant with standard RTI and GRPO on Claw-Eval-Hard20. GRPO improves briefly before its Pass@k and Pass^k scores collapse. Both RTI variants instead sustain gains across the two metric families: higher Pass@k reflects success on more tasks within k attempts, while higher Pass^k reflects more consistent success across attempts. Standard RTI achieves the strongest results, showing that corrections from the much larger model deviate further from the policy's distribution and are therefore harder to learn from, despite the teacher's greater capability. These findings show that self-intervention suffices to improve performance across diverse hard tasks without relying on a stronger teacher. Additionally, we supplement the experimental results on the full set of unimodal (text-only) tasks in Appendix D.6.

6 Conclusion

We introduced RTI to enable agent policies to learn beyond their unaided exploration capacity. RTI uses reward-aware self-intervention to correct failed rollouts, a perplexity-based gate to filter excessively shifted trajectories, and reasoning-shift-aware token reweighting to focus policy updates on consequential decisions. Across five benchmarks and two backbones, RTI substantially improves task success, with particularly large gains on Navigation, τ^2-Telecom, and Claw-Eval-Hard20. Training analyses show that intervention supplies successful trajectories early in training and that the policy increasingly succeeds without intervention as learning proceeds. An additional comparison using a much larger intervening model confirms that RTI does not require a stronger teacher to achieve these gains, supporting its applicability across diverse agent tasks.

AI USE STATEMENT

We used generative AI tools as an assistive aid in the preparation of this manuscript, specifically for proofreading, grammar checking, and language refinement to improve clarity and readability. All research ideas, experimental designs, theoretical claims, and final content were conceived, verified, and approved by the authors.

ETHICS STATEMENT

We will abide by the laws, rules, and regulations of our community, school, work, and country. We will conduct ourselves with integrity, fidelity, and honesty. We will openly take responsibility for our actions and only make agreements that we intend to keep. All data used in this study are intended for research purposes. No personally identifiable information (PII) was collected.

REPRODUCIBILITY STATEMENT

We have included comprehensive details to support reproducibility of this work. The complete RTI algorithm, implementation details, hyperparameters, benchmark descriptions, baseline configurations, and theoretical proofs are provided in the appendix. Representative prompts and case studies are also included in the appendix. Upon acceptance, we will release the code and data as supplementary material.

REFERENCES

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.

Guanting Dong, Licheng Bao, Zhongyuan Wang, Kangzhi Zhao, Xiaoxi Li, Jiajie Jin, Jinghan Yang, Hangyu Mao, Fuzheng Zhang, Kun Gai, Guorui Zhou, Yutao Zhu, Ji-Rong Wen, and Zhicheng Dou. Toward generalized web agent training: A deep dive into entropy-balanced reinforcement learning. In *Proceedings of the ACM Web Conference 2026*, pp. 2126–2137, 2026a. URL https://doi.org/10.1145/3774904.3792301.

Guanting Dong, Junting Lu, Junjie Huang, Wanjun Zhong, Longxiang Liu, Shijue Huang, Zhenyu Li, Yang Zhao, Xiaoshuai Song, Xiaoxi Li, Jiajie Jin, Yutao Zhu, Hanbin Wang, Fangyu Lei, Qinyu Luo, Mingyang Chen, Zehui Chen, Jiazhan Feng, Ji-Rong Wen, and Zhicheng Dou. Agent-world: Scaling real-world environment synthesis for evolving general agent intelligence, 2026b. URL https://arxiv.org/abs/2604.18292.

Jiazhan Feng, Shijue Huang, Xingwei Qu, Ge Zhang, Yujia Qin, Baoquan Zhong, Chengquan Jiang, Jinxin Chi, and Wanjun Zhong. Retool: Reinforcement learning for strategic tool use in llms, 2025. URL https://arxiv.org/abs/2504.11536.

Romain Froger, Pierre Andrews, Matteo Bettini, Amar Budhiraja, Ricardo Silveira Cabral, Virginie Do, Emilien Garreau, Jean-Baptiste Gaya, Hugo Laurençon, Maxime Lecanu, Kunal Malkan, Dheeraj Mekala, Pierre Ménard, Gerard Moreno-Torres Bertran, Ulyana Piterbarg, Mikhail Plekhanov, Mathieu Rita, Andrey Rusakov, Vladislav Vorotilov, Mengjue Wang, Ian Yu, Amine Benhalloum, Grégoire Mialon, and Thomas Scialom. Are: Scaling up agent environments and evaluations, 2025. URL https://arxiv.org/abs/2509.17158.

Wei Fu, Jiaxuan Gao, Xujie Shen, Chen Zhu, Zhiyu Mei, Chuyi He, Shusheng Xu, Guo Wei, Jun Mei, WANG JIASHU, Tongkai Yang, Binhang Yuan, and Yi Wu. AREAL: A large-scale asynchronous reinforcement learning system for language reasoning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026. URL https://openreview.net/forum?id=X9diEuva9R.

Jiaxuan Gao, Jiaao Chen, Chuyi He, Shusheng Xu, Di Jin, and Yi Wu. From self-evolving synthetic data to verifiable-reward rl: Post-training multi-turn interactive tool-using agents, 2026. URL https://arxiv.org/abs/2601.22607.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. MiniLLM: Knowledge distillation of large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=5h0qf7IBZZ.

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

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan O Arik, Dong Wang, Hamed Zamani, and Jiawei Han. Search-r1: Training LLMs to reason and leverage search engines with reinforcement learning. In *Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=Rwhi91ideu.

Xiaoxi Li, Wenxiang Jiao, Jiarui Jin, Guanting Dong, Jiajie Jin, Yinuo Wang, Hao Wang, Yutao Zhu, Ji-Rong Wen, Yuan Lu, and Zhicheng Dou. Deepagent: A general reasoning agent with scalable toolsets. In *Proceedings of the ACM Web Conference 2026*, pp. 2219–2230, New York, NY, USA, 2026. Association for Computing Machinery. URL https://doi.org/10.1145/3774904.3792460.

Yuetai Li, Huseyin A Inan, Xiang Yue, Wei-Ning Chen, Lukas Wutschitz, Janardhan Kulkarni, Radha Poovendran, Robert Sim, and Saravan Rajmohan. Simulating environments with reasoning models for agent training, 2025. URL https://arxiv.org/abs/2511.01824.

Weiwen Liu, Xu Huang, Xingshan Zeng, xinlong hao, Shuai Yu, Dexun Li, Shuai Wang, Weinan Gan, Zhengying Liu, Yuanqing Yu, Zezhong WANG, Yuxian Wang, Wu Ning, Yutai Hou, Bin Wang, Chuhan Wu, Wang Xinzhi, Yong Liu, Yasheng Wang, Duyu Tang, Dandan Tu, Lifeng Shang, Xin Jiang, Ruiming Tang, Defu Lian, Qun Liu, and Enhong Chen. ToolACE: Winning the points of LLM function calling. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=8EB8k6DdCU.

Kevin Lu and Thinking Machines Lab. On-policy distillation. 2025. URL https://thinkingmachines.ai/blog/on-policy-distillation.

Akshara Prabhakar, Zuxin Liu, Ming Zhu, Jianguo Zhang, Tulika Manoj Awalgaonkar, Shiyu Wang, Zhiwei Liu, Haolin Chen, Thai Quoc Hoang, Juan Carlos Niebles, Shelby Heinecke, Weiran Yao, Huan Wang, Silvio Savarese, and Caiming Xiong. APIGen-MT: Agentic pipeline for multi-turn data generation via simulated agent-human interplay. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2026. URL https://openreview.net/forum?id=qk6ORqQ4Cu.

Cheng Qian, Emre Can Acikgoz, Qi He, Hongru WANG, Xiusi Chen, Dilek Hakkani-Tür, Gokhan Tur, and Heng Ji. ToolRL: Reward is all tool learning needs. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026. URL https://openreview.net/forum?id=eOLdGbXT6t.

Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, Sihan Zhao, Lauren Hong, Runchu Tian, Ruobing Xie, Jie Zhou, Mark Gerstein, Dahai Li, Zhiyuan Liu, and Maosong Sun. ToolLLM: Facilitating large language models to master 16000+ real-world apis. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=dHng2O0Jjr.

Sashank J. Reddi, Ahmed Hefny, Suvrit Sra, Barnabás Póczós, and Alex Smola. Stochastic variance reduction for nonconvex optimization. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, pp. 314–323. JMLR.org, 2016.

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.

Hao Sun, Zile Qiao, Jiayan Guo, Xuanbo Fan, Yingyan Hou, Yong Jiang, Pengjun Xie, Yan Zhang, Fei Huang, and Jingren Zhou. Zerosearch: Incentivize the search capability of llms without searching, 2026. URL https://arxiv.org/abs/2505.04588.

Kimi Team. Kimi k2: Open agentic intelligence, 2026. URL https://arxiv.org/abs/2507.20534.

Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, Wayne Xin Zhao, Zhewei Wei, and Jirong Wen. A survey on large language model based autonomous agents. *Frontiers of Computer Science*, 18(6), March 2024. ISSN 2095-2236. doi: 10.1007/s11704-024-40231-1. URL http://dx.doi.org/10.1007/s11704-024-40231-1.

Tianxin Wei, Ting-Wei Li, Zhining Liu, Xuying Ning, Ze Yang, Jiaru Zou, Zhichen Zeng, Ruizhong Qiu, Xiao Lin, Dongqi Fu, Zihao Li, Mengting Ai, Duo Zhou, Wenxuan Bao, Yunzhe Li, Gaotang Li, Cheng Qian, Yu Wang, Xiangru Tang, Yin Xiao, Liri Fang, Hui Liu, Xianfeng Tang, Yuji Zhang, Chi Wang, Jiaxuan You, Heng Ji, Hanghang Tong, and Jingrui He. Agentic reasoning for large language models, 2026. URL https://arxiv.org/abs/2601.12538.

Jianhao Yan, Yafu Li, Zican Hu, Zhi Wang, Ganqu Cui, Xiaoye Qu, Yu Cheng, and Yue Zhang. Learning to reason under off-policy guidance. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026. URL https://openreview.net/forum?id=vO8LLoNWWk.

Jinluan Yang, Yuxin Liu, Zhengyu Chen, Chengcheng Han, Yueqing Sun, Qi Gu, Hui Su, Xunliang Cai, Fei Wu, and Kun Kuang. Topocurate:modeling interaction topology for tool-use agent training, 2026. URL https://arxiv.org/abs/2603.01714.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=WE_vluYUL-X.

Bowen Ye, Rang Li, Qibin Yang, Yuanxin Liu, Linli Yao, Hanglong Lv, Zhihui Xie, Chenxin An, Lei Li, Lingpeng Kong, Qi Liu, Zhifang Sui, and Tong Yang. Claw-eval: Towards trustworthy evaluation of autonomous agents, 2026. URL https://arxiv.org/abs/2604.06132.

Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Yang Yue, Shiji Song, and Gao Huang. Does reinforcement learning really incentivize reasoning capacity in LLMs beyond the base model? In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026. URL https://openreview.net/forum?id=4OsgYD7em5.

Rosie Zhao, Alexandru Meterez, Sham Kakade, Cengiz Pehlevan, Samy Jelassi, and Eran Malach. Echo chamber: Rl post-training amplifies behaviors learned in pretraining, 2025. URL https://arxiv.org/abs/2504.07912.

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.

A MORE DETAILS OF RTI

A.1 RTI MAS PROMPT TEMPLATES

In this section, we present the full prompt templates for the three MAS agents used in RTI.

A.1.1 τ^2-BENCH PROMPTS (TELECOM / RETAIL / AIRLINE)

Reward-Oriented Trajectory Validator. The Validator assesses whether the current response is a valid next step that keeps the agent on track. It checks against a set of hard rules and marks the response invalid only when a clear violation is detected.

Reward-Oriented Trajectory Validator (τ^2-Bench)

```
You are a trajectory supervisor for a tool-calling agent task. Your job is to judge whether the CURRENT response
is a valid next step that keeps the agent on track toward completing the task. Task is complete only when
{completion_note}.
[Observable History] {history_json}
[Current Response](Formatted as json) {response_text}
[Required actions to complete the task]
Agent-side actions (the agent must call these tools itself): {agent_actions}
{user_actions_block} (telecom only: lists user-side device actions the agent must guide)
How to tell if an action is done:
- agent-side action: a matching assistant tool call + successful result in history.
- user-side action [telecom]: a matching user tool call output in history.
RULES — mark invalid if ANY rule is violated:
1) FABRICATED_ARGUMENTS: A tool argument value is not present in history and not derivable from the task
description. (Values from earlier tool outputs ARE valid.)
2) POINTLESS_REPEAT: Calling the exact same tool with the exact same arguments that already succeeded. (Same
tool, different arguments = allowed.)
3) FALSE_CLAIM: The response claims information was retrieved without a corresponding tool call.
4) [domain_specific]:
[Telecom] USER_SIDE_DEVICE_ACTION: The agent emits a tool call for a user-device action. The agent must NEVER
call these — it must instruct the user to perform them on their device.
USER_SIDE_TOOLS = [check_status_bar, check_network_status, ..., reboot_device]
AGENT_SIDE_TOOLS = [get_customer_by_phone, ..., refuel_data]
[Retail] MISSING_CONFIRMATION: Before any tool call that updates the database (including modify_user_address
and any cancel/modify/return/exchange action), the agent must first list the exact action details and obtain an
explicit user confirmation "yes".
[Airline] MISSING_CONFIRMATION: Before taking any actions that update the booking database (booking, modifying
flights, editing baggage, changing cabin class, or updating passenger information), the agent must first list
the action details and obtain an explicit user confirmation "yes".
5) PREMATURE_TERMINATION: The response ends the conversation (transfer to human, closing message) while one or
more required actions are not yet done.
6) CONTRADICTORY_ACTION: The response calls a tool or instructs the user in a way that directly contradicts a
required action.
7) GOAL_DRIFT: The response ignores an obviously pending required action and pursues something unrelated.
8) ARGUMENT_MISMATCH: For actions that update the database, if [Required actions] specifies exact argument
values, the agent's tool call must match these values exactly.
DO NOT mark invalid for:
- Suboptimal but recoverable action order
- Diagnostic / information-lookup steps before acting
- Re-attempting a failed action with corrected arguments
- A valid path that differs from the reference
[Evaluation_Criteria — for context only] {evaluation_criteria}
Output JSON only:
{"diagnosis": "...", "is_valid": true/false,
 "error_types": [...], "suggest_next": "..."}
```

Policy-Aligned Trajectory Corrector. The Corrector generates an optimized next action that corrects errors while concealing privileged information. It receives the Validator's diagnosis together with the visible-only conversation history and produces an intervened response.

Policy-Aligned Trajectory Corrector (τ^2-Bench)

```
You are a silent calibrator for a tool-calling agent task. Your role is to minimally correct the agent's next
action to bring it closer to the successful path, WITHOUT exposing any privileged knowledge.
[Conversation History] {history}
[Available Tools] {tools}
[Reference / evaluation criteria (for calibration only; MUST NOT appear in output)] {evaluation_criteria}
CORE INSTRUCTIONS:
Generate the optimized next action based on [Conversation History], NOT by continuing from [Prior Draft]. The
Prior Draft is provided only as a style reference.
```

```
GENERATION STRATEGY:
1. Style alignment: match the reasoning style, verbosity, and expression habits of the Prior Draft.
2. Progressive correction: guide toward a more reasonable path; corrections should feel like natural evolution of thought.
3. Concealment: NEVER mention "reference", "groundtruth", "required actions", "evaluation criteria", "prior draft", "calibrated response", "previous version", or "the correct answer". All reasoning must appear naturally derived from conversation history.
4. Minimal change: only correct clearly wrong steps; preserve correct parts of the prior draft.
CORRECTION GUIDELINES:
- If a tool argument is wrong, use the correct value from history (e.g., correct payment_id from user profile)
- If the agent skipped a necessary step, add a message to gather required info or confirm with user
- If the agent made an unfounded claim, revise to acknowledge uncertainty or check first
- If multiple valid approaches exist, prefer the one closer to the original intent
- DO NOT introduce information that cannot be derived from visible history
- If no valid agent-side tool call is appropriate, set "tool_calls" to an empty list. - Before any tool call that updates the database (including modify_user_address and any cancel/modify/return/exchange action), the agent must first list the exact action details and obtain an explicit user confirmation "yes".
[domain_specific]
[Telecom] If an action must be done by the user on their device, do not emit a tool call for it; guide the user to perform the action on their device instead.
USER_SIDE_DEVICE_ACTIONS = [check_status_bar, ..., reboot_device]
AGENT_SIDE_TOOLS = [get_customer_by_phone, ..., refuel_data]
Note: "tool_calls" must be an empty list if no agent-side tool is to be called.

[Prior Draft (style reference only)] {original_response}
[Supervisor Feedback (problem indication)] {feedback_json}
Output JSON: {"reasoning": "...", "tool_calls": [...], "content": "..."}
Field rules: "reasoning" is always required. "tool_calls" and "content" are mutually exclusive: use tool_calls for agent-side actions, content for direct user replies.
```

Integrity-Constrained Intervention Verifier. The Verifier validates that the corrected response does not leak privileged information or introduce hallucinated content. It serves as the final quality gate before an intervened response enters the training batch.

Integrity-Constrained Intervention Verifier (τ^2-Bench)

```
You are a post-calibration safety & faithfulness inspector. Decide whether the CALIBRATED reply should be accepted.
VISIBLE INFO = user question + tool list + conversation history (including tool outputs). Nothing else is allowed.
[User Question] {user_prompt}
[Evaluation Criteria] {evaluation_criteria}
[Available Tools] {tools}
[Agent History (visible)] {history}
[Prior Draft] {original}
[Calibrated Reply] {corrected}
Reject (ok=false) ONLY if ANY holds:
1) leak: explicitly mentions "reference", "groundtruth", "evaluation criteria", "required actions", "answer key", "labels", "seeing the solution", "the prior draft", "previous version", "calibrated version", or any phrasing that implies the model is comparing its output to a hidden draft or ground truth.
2) fabricated_tool_result: claims a specific tool result/value that does NOT appear in history. Example: claiming "flight price is $500" when no tool returned that exact value.
3) wrong_tool_arguments: uses tool arguments that contradict visible history or deviate from explicit constraints defined in [Evaluation Criteria]. Mandatory Alignment: if the criteria explicitly specifies a parameter value, the agent's argument must match exactly. Contextual Discretion: if not specified, the agent may derive values logically.
4) unreasonable_calibration: the calibrated reply is not a reasonable next step given the visible history, fails to improve the prior draft, or introduces unnecessary or less appropriate actions.
ALLOWED (not an issue) — DO NOT REJECT for:
- Adding reasonable next steps implied by task context (e.g., asking for user confirmation before database updates)
- Adding policy-compliant behaviors (e.g., "I need to verify with the user first")
- Changing the action sequence to be more cautious or thorough
- Adding intermediate steps that logically follow from the conversation
The calibration goal is to IMPROVE the agent's behavior, not just paraphrase it. Adding missing confirmation steps or policy-compliant actions is the INTENDED behavior of calibration.
Output JSON: {"ok": true/false, "reasons": [...], "diagnosis": "..."}
```

A.1.2 Navigation Domain Prompts

The Navigation domain follows the same three-agent Validate–Correct–Verify pipeline as τ^2-Bench, sharing the identical architectural logic: the Validator detects deviations from a valid path using the reward-side evidence as a silent reference; the Corrector applies the same four generation principles (style alignment, progressive correction, leakage suppression, minimal intervention); and the Verifier enforces the same integrity constraints (no leakage, no hallucination).

Reward-Oriented Trajectory Validator (Navigation). The Navigation Validator applies the same goal-oriented, process-agnostic philosophy as its τ^2-Bench counterpart: it does not penalize suboptimal-but-recoverable action orders or stylistic differences, and only flags violations of the hard rules already stated in the agent's system prompt.

Reward-Oriented Trajectory Validator (Navigation)

```
You are a trajectory supervisor for an "Agent tool-calling task". Judge whether the current response constitutes
a reasonable next step.
[Conversation History (observable evidence)] {history_json}
[Current Response] {response_text}
[Hard Rules]:
- No fabricating coordinates/key parameters (must come from historical tool returns)
- Do NOT call direction before all POI coordinates are collected
- No redundant poi_search with identical parameters in the same turn
- Parallel tool calls are allowed
[Reference (for internal comparison only; do NOT judge invalid based solely on reference)]
order: {order}   ref: {ref}   groundtruth: {gt}
Mark invalid ONLY if:
A) Hard rule violation (fabrication / premature direction / redundant search)
B) Stubbornly repeating on clearly failed tool returns
C) Irreversible wrong critical call with clearly untenable arguments
Output JSON only:
{"is_valid": true/false, "error_types": [],
"diagnosis": "...(cite observable evidence/hard rules)...",
"suggest_next": "..."}
```

Policy-Aligned Trajectory Corrector (Navigation). The Navigation Corrector shares the same four generation principles as τ^2-Bench (style alignment, progressive correction, leakage suppression, minimal intervention).

Policy-Aligned Trajectory Corrector (Navigation)

```
You are the "silent calibrator" in a navigation task — your role is to minimally optimize an agent's action
trajectory toward the successful path, WITHOUT exposing any additional knowledge. You have a global view (knowing
how the task should ultimately be completed), but must translate this insight into seemingly natural, progressive
corrections.
[Conversation History] {history}
[Available Tools] {tools}
[Reference/groundtruth (for calibration only; MUST NOT appear in output)]
ref: {ref}   order: {order}   groundtruth: {gt}
CORE INSTRUCTIONS: Generate the optimized next action based on [Conversation History], NOT based on [Prior
Draft]. The prior draft is provided only as a style reference.
GENERATION STRATEGY:
1. Style alignment: analyze the reasoning style, detail level, and expression habits of the Prior Draft; new
reasoning should match in style.
2. Progressive correction: leverage your global view to embed a more reasonable reasoning path; corrections
should feel like natural evolution, not a 180-degree turn.
3. Concealment: NEVER mention or imply you know the correct answer, final goal, reference behavior, or full
process. All reasoning must appear naturally derived from history.
4. Minimal change: only correct clearly wrong key points; preserve correct parts of the prior draft.
[Prior Draft (style reference)] {original_response}
[Supervisor Feedback (problem indication)] {feedback_json}
KEY REMINDERS:
- You are generating the next action based on [Conversation History], not continuing from [Prior Draft]
- Your output should look like the agent's own naturally improved reasoning
```

Integrity-Constrained Intervention Verifier (Navigation). The Navigation Verifier enforces the same two core rejection criteria as the τ^2-Bench Verifier—leakage of privileged information and hallucination of underivable content.

Integrity-Constrained Intervention Verifier (Navigation)

```
You are a "safety & explainability post-checker". A calibrated reply is given; decide whether it should be
accepted.
[User Question] {user_prompt}
[Available Tools] {tools}
[Agent Conversation History] {history}
[Prior Draft] {original}
[Calibrated Reply] {corrected}
Rejection criteria (any one triggers ok=false):
1) leak: mentions reference/groundtruth/standard answer/labels, or clearly uses invisible information.
2) not_derivable (hallucination): the calibrated reply adds/modifies key facts or tool call parameters that
come neither from "user question + visible history" nor qualify as "allowed rewrites" below.
```

```
Allowed rewrites (NOT hallucination):
- Equivalent rewriting of already-appeared information (synonyms, format normalization, unit conversion)
- Incorporating already-specified context constraints into parameters (region, time window, mode)
- Conservative narrowing for disambiguation (combining category words with keywords, removing uncertain components)
- Decomposing complex requests into verifiable sub-steps (but must not claim tool results that don't exist in history)
Output JSON only:
{"ok": true/false, "reasons": ["leak", "not_derivable"],
"diagnosis": "must cite observable evidence (specific fragments from history / missing points)"}
```

A.2 The Full RTI Algorithm

Algorithm 1 integrates the intervention pipeline and the on-policy GRPO loop into a single training procedure. At every iteration we sample a batch of queries from $\mathcal{D}$ and, for each query, first run a fully on-policy GRPO group; if any rollout in this group succeeds, the group is used as-is and only the on-policy portion of $\mathcal{J}_{\mathrm{RTI}}$ is exercised (with $\tilde{\rho}^t_{\mathrm{cal}} \equiv 1$). Only when the entire group fails do we re-run an intervened rollout with the MAS intervention pipeline enabled; every accepted correction must additionally pass the perplexity gate of Eq. (4), and only if the intervened rollout succeeds is a single failed member of the group replaced by the intervened trajectory, for which we additionally cache the rollout-prompt and correction-prompt logprobs needed by Eq. (7). A single backward pass over $\mathcal{J}_{\mathrm{RTI}}$ then updates θ with a gradient that is on-policy on easy instances, reweighted on intervened tokens of hard instances, and unbiased on every other token. This design preserves GRPO's behavior whenever the policy can solve a task on its own and adds capability-expanding signals exactly where vanilla on-policy RL stalls.

B Theoretical Analysis of RTI

B.1 Convergence Analysis of RTI

RTI differs from standard on-policy GRPO in two aspects: it incorporates MAS-intervened trajectories, which are off-policy relative to the current policy, and it further applies the reasoning-shift-aware token reweighting in Eq. (7). These two components raise a natural concern: although they may improve exploration and credit assignment, they also introduce additional distribution shift into the optimization process.

Based on the theoretical analysis of stochastic gradient descent in nonconvex optimization (Reddi et al., 2016; Yan et al., 2026), we give a convergence analysis in Theorem 1 to show that, under the alignment and decoupled boundedness conditions stated below, RTI admits an $O(1/\sqrt{K})$ stationarity guarantee, where K is the total number of iterations. For notational simplicity, we write $\mathcal{J}(\theta) := \mathcal{J}_{\mathrm{RTI}}(\theta)$ throughout this section.

Theorem 1 (Conditional Convergence under Alignment and Decoupled Boundedness)**.** *RTI constructs the policy gradient using a ratio mixture $\lambda \in (0,1)$ of off-policy intervened rollouts and on-policy rollouts, where λ denotes the expected mixing ratio over training (since intervention is triggered only on fully failed groups, λ is a data-dependent random quantity and we take its expectation as a constant for the analysis). Accordingly, we decompose the per-step update as $\Delta\theta^k = \lambda\,\Delta\theta^k_{\mathrm{cal}} + (1-\lambda)\,\Delta\theta^k_{\mathrm{grpo}}$. Suppose that the objective $\mathcal{J}(\theta)$ is L-smooth and that $\mathcal{J}^\star := \sup_\theta \mathcal{J}(\theta) < \infty$. Assume further that the expected update direction is aligned with the true gradient with a lower bound governed by the on-policy clip margin $1-\epsilon_{\mathrm{clip}}$ (this alignment lower bound arises from the on-policy term $\Delta\theta^k_{\mathrm{grpo}}$ under the classic clipped surrogate; the intervened term $\Delta\theta^k_{\mathrm{cal}}$ is not required to contribute alignment and is instead absorbed into the second-moment bound D^2_{cal} in Definition 5), i.e.,*

$$\mathbb{E}\big[\langle \nabla\mathcal{J}(\theta^k), \Delta\theta^k\rangle\big] \geq \alpha_k(1-\epsilon_{\mathrm{clip}})\|\nabla\mathcal{J}(\theta^k)\|^2. \tag{8}$$

For K iterations, let the learning rate be set to $\alpha_k = c/\sqrt{K}$, where

$$c = \sqrt{\frac{2\left[\mathcal{J}^\star - \mathcal{J}(\theta^0) + \frac{L}{2}\lambda D^2_{\mathrm{cal}}\right]}{L(1+\epsilon_{\mathrm{clip}})^2\sigma^2}} \tag{9}$$

Algorithm 1: Rollout-Time Intervention (RTI)

Input: Policy π_θ (which also instantiates all MAS roles), reference π_{ref}, task distribution $\mathcal{D}$, group size G, reweighting clip ϵ_{cal}.

for *each training iteration* **do**
  Sample batch $\{x_n\} \sim \mathcal{D}$
  for *each query* x_n **do**
    `// Stage 1: on-policy rollouts`
    Sample $\{\tau_i\}_{i=1}^{G} \sim \pi_\theta(\cdot \mid x_n)$ with intervention disabled
    if $\max_i r(x_n, \tau_i) > 0$ **then**
      $\{\tilde{\tau}_i\} \leftarrow \{\tau_i\}$, set $\tilde{\rho}^t_{\mathrm{cal}} \equiv 1$ on all tokens `// purely on-policy update`
    end
    else
      `// Stage 2: reward-aware MAS intervention`
      $\{\tilde{\tau}_i\} \leftarrow \{\tau_i\}$
      Run an intervened rollout: at each step k generate $\hat{u}_k \sim \pi_\theta$;
      pass $(h_{1:k-1}, \hat{u}_k, \ell)$ through Validator $\rightarrow$ Corrector $\rightarrow$ Verifier
      if *the correction is accepted by the Verifier* **then**
        Rescore $\tilde{u}_k$ under the rollout prompt and compute $\mathrm{PPL}_{\mathrm{roll}}(\tilde{u}_k)$ by Eq. (3)
        if $\mathrm{PPL}_{\mathrm{roll}}(\tilde{u}_k) \le \delta$ **then**
          Advance with $\tilde{u}_k$, cache $\log \pi_\theta^{\mathrm{roll}}$ and $\log \pi^{\mathrm{cal}}$
        end
        else
          Discard $\tilde{u}_k$, advance with $\hat{u}_k$
        end
      end
      else
        Advance with $\hat{u}_k$
      end
      if *the intervened rollout achieves reward* 1 **then**
        Replace a single failed member of $\{\tilde{\tau}_i\}$ with the intervened trajectory
        Compute clipped weights $\tilde{\rho}^t_{\mathrm{cal}}$ via Eq. (6) on intervened tokens; set $\tilde{\rho}^t_{\mathrm{cal}} \equiv 1$ on the remaining on-policy tokens
      end
      else
        Keep $\{\tilde{\tau}_i\}$ unchanged, so that no correction enters the update
      end
    end
    Compute group advantages $\{A_i\}$ from rewards $\{r(x_n, \tilde{\tau}_i)\}$ as in GRPO
  end
  Update θ by ascending the RTI objective (Eq. 7)
end

Then the iterates satisfy the following $\mathcal{O}(1/\sqrt{K})$ convergence bound:

$$\min_{0 \le k \le K-1} \mathbb{E}\left[\|\nabla \mathcal{J}(\theta^k)\|^2\right] \le \frac{1}{(1-\epsilon_{\mathrm{clip}})} \sqrt{\frac{2\left[\mathcal{J}^\star - \mathcal{J}(\theta^0) + \frac{L}{2}\lambda D_{\mathrm{cal}}^2\right] L (1+\epsilon_{\mathrm{clip}})^2 \sigma^2}{K}}, \tag{10}$$

where θ^0 denotes the initial policy parameter, σ^2 controls the base gradient scale and $D_{\mathrm{cal}}^2 = 4\,\epsilon_{\mathrm{cal}}^2\,\eta^2\,\sigma^2$ with $\eta > 0$ as a parameter-space perturbation constant abstracting the bounded contribution of the intervened rollouts (formalized in Definition 5).

B.2 Proof of Theorem 1

We first state the standard definitions and assumptions for non-convex stochastic optimization.

Definition 1 (Stochastic Oracle)**.** *For a smooth functional $\mathcal{J}$, we assume the algorithm has access to a stochastic oracle that, given an index $i \in [n]$ and a parameter $\theta \in \mathbb{R}^d$, returns the pair $(\mathcal{J}_i(\theta), \nabla \mathcal{J}_i(\theta))$.*

Definition 2 (Lipschitz Smoothness)**.** *We say $\mathcal{J} : \mathbb{R}^d \to \mathbb{R}$ is L-smooth if there exists a constant $L > 0$ such that $\|\nabla \mathcal{J}(\vartheta) - \nabla \mathcal{J}(\theta)\| \le L\|\vartheta - \theta\|, \forall \vartheta, \theta \in \mathbb{R}^d$. This ensures the objective's curvature is bounded.*

Definition 3 (σ-bounded base gradients)**.** *We say $\mathcal{J}$ has σ-bounded base gradients if the unscaled stochastic gradient satisfies $\|\nabla \mathcal{J}_i(\theta)\| \leq \sigma$ for all $i \in [n]$ and $\theta \in \mathbb{R}^d$.*

Definition 4 (ϵ-accuracy)**.** *A stochastic iterative algorithm is said to achieve ϵ_{acc}-accuracy in K iterations if the expected gradient norm satisfies $\min_{0\leq k\leq K-1} \mathbb{E}[\|\nabla \mathcal{J}(\theta^k)\|^2] \leq \epsilon_{\text{acc}}$.*

To ensure stable optimization, we construct the learning objective as a deterministic weighted combination of two distinct data sources using a proportion scalar $\lambda \in (0, 1)$. At any training step k, exactly a fraction λ of the learning signal originates from the off-policy intervened trajectories $\mathcal{D}_{\text{cal}}$, and the remaining $1 - \lambda$ originates from the on-policy rollout policy $\mathcal{D}_{\text{grpo}}$. For a given trajectory, the stochastic gradient is scaled by a token-level reweighting coefficient. We define the objective surrogates asymmetrically:

Off-policy Intervened Rollouts: To push the current policy beyond its existing capability boundary, we employ the reweighting $\tilde{\rho}^t_{\text{cal}}$ defined in Eq. (6), which is bounded within $[1 - \epsilon_{\text{cal}},\ 1 + \epsilon_{\text{cal}}]$. In practice, this coefficient is computed separately from the policy-gradient term and is detached from the computation graph; it therefore acts purely as a bounded per-token weight on the advantage:

$$\tilde{r}^t_{\text{cal}} = \text{sg}\left(\tilde{\rho}^t_{\text{cal}}\right), \tag{11}$$

where $\text{sg}(\cdot)$ denotes the stop-gradient operator. Since the coefficient is not differentiated, it introduces no additional gradient path through π^{cal} or π^{roll}_θ: it preserves the forward intervention weight and reweights only the advantage of the tokens it emphasizes, while its bounded magnitude prevents any single intervened token from dominating the update, thereby mitigating gradient amplification and improving optimization stability. Since $\tilde{r}^t_{\text{cal}}$ deviates from unity by at most ϵ_{cal}, this bounded drift is what motivates the cumulative bound $D^2_{\text{cal}} = 4\,\epsilon^2_{\text{cal}}\,\eta^2\,\sigma^2$ imposed in Definition 5.

On-policy Rollouts: For samples generated by the on-policy model, we apply the per-token probability ratio $\rho^t_{\text{grpo},i}$ of Eq. (2), bounded by the classic hard clip:

$$\tilde{r}^t_{\text{grpo}}(\theta) = \text{clip}\left(\rho^t_{\text{grpo},i}, 1 - \epsilon_{\text{clip}}, 1 + \epsilon_{\text{clip}}\right), \tag{12}$$

where the clipped weight satisfies $|\tilde{r}^t_{\text{grpo}}(\theta)| \leq 1 + \epsilon_{\text{clip}}$; for brevity, we denote it by w_{grpo} in the following analysis.

Definition 5 (Decoupled Dynamics and Trust Region Bounds)**.** *We decompose the parameter update step $\Delta\theta^k = \theta^{k+1} - \theta^k$ according to its data source, and impose the following decoupled boundedness assumptions on the two sub-updates (which abstract, rather than rigorously derive from, the reweighting and clipping mechanisms in Eq.* (11) *and Eq.* (12)*):*

- ***Off-policy Regime (Cumulative Trust Region):*** *The total cumulative second moment of intervened updates over K iterations is assumed to be bounded by a constant independent of K:*

$$\sum_{k=0}^{K-1} \mathbb{E}\left[\|\Delta\theta^k_{\text{cal}}\|^2\right] \leq D^2_{\text{cal}} := 4\,\eta^2\epsilon^2_{\text{cal}}\sigma^2, \tag{13}$$

  where $\eta > 0$ maps the bounded reweighting drift to the parameter space.

- ***On-policy Regime (Step-wise Variance):*** *The update step is given by $\Delta\theta^k_{\text{grpo}} = \alpha_k w_{\text{grpo}} \nabla \mathcal{J}_i(\theta)$, and its second moment scales with the learning rate α_k and the weight upper bound $(1 + \epsilon_{\text{clip}})$:*

$$\mathbb{E}\left[\|\Delta\theta^k_{\text{grpo}}\|^2\right] \leq \alpha^2_k(1 + \epsilon_{\text{clip}})^2\sigma^2 \tag{14}$$

Proof. Following the L-smoothness assumption of the objective function, the iterates of our algorithm satisfy the following expected lower bound:

$$\mathbb{E}[\mathcal{J}(\theta^{k+1})] \geq \mathbb{E}\Big[\mathcal{J}(\theta^k) + \langle \nabla \mathcal{J}(\theta^k), \theta^{k+1} - \theta^k\rangle - \frac{L}{2}\|\theta^{k+1} - \theta^k\|^2\Big] \tag{15}$$

By defining the actual parameter update step as $\Delta\theta^k = \theta^{k+1} - \theta^k$, we substitute this definition and the expected proportional gain lower bound into the smoothness inequality:

$$\mathbb{E}[\mathcal{J}(\theta^{k+1})] \geq \mathbb{E}[\mathcal{J}(\theta^k)] + \alpha_k(1 - \epsilon_{\text{clip}})\mathbb{E}\left[\|\nabla \mathcal{J}(\theta^k)\|^2\right] - \frac{L}{2}\,\mathbb{E}\left[\|\Delta\theta^k\|^2\right] \tag{16}$$

Now, using the mixture decomposition $\Delta\theta^k = \lambda\,\Delta\theta^k_{\text{cal}} + (1-\lambda)\,\Delta\theta^k_{\text{grpo}}$ together with the convexity of $\|\cdot\|^2$ and the further relaxation $1-\lambda \le 1$, we substitute the decoupled second-moment bounds from Definition 5 into $\mathbb{E}\left[\|\Delta\theta^k\|^2\right]$:

$$\begin{aligned}&\mathbb{E}[\mathcal{J}(\theta^{k+1})] \ge \mathbb{E}[\mathcal{J}(\theta^k)] + \alpha_k(1-\epsilon_{\text{clip}})\\ &\mathbb{E}\left[\|\nabla\mathcal{J}(\theta^k)\|^2\right] - \frac{L}{2}\left(\lambda\mathbb{E}\left[\|\Delta\theta^k_{\text{cal}}\|^2\right] + \alpha_k^2(1+\epsilon_{\text{clip}})^2\sigma^2\right)\end{aligned} \tag{17}$$

Rearranging the terms to isolate the target gradient norm:

$$\alpha_k(1-\epsilon_{\text{clip}})\mathbb{E}\left[\|\nabla\mathcal{J}(\theta^k)\|^2\right] \le \mathbb{E}[\mathcal{J}(\theta^{k+1}) - \mathcal{J}(\theta^k)] + \frac{L}{2}\lambda\mathbb{E}\left[\|\Delta\theta^k_{\text{cal}}\|^2\right] + \frac{L}{2}\alpha_k^2(1+\epsilon_{\text{clip}})^2\sigma^2. \tag{18}$$

Now, we sum this inequality across all iterations from $k=0$ to $K-1$, assuming a horizon-dependent constant learning rate $\alpha_k \equiv \alpha$, and invoke the decoupled boundedness assumptions in Definition 5:

$$\begin{aligned}&\alpha(1-\epsilon_{\text{clip}})\sum_{k=0}^{K-1}\mathbb{E}\left[\|\nabla\mathcal{J}(\theta^k)\|^2\right] \le \left[\mathcal{J}(\theta^K) - \mathcal{J}(\theta^0)\right] + \frac{L}{2}\lambda\underbrace{\sum_{k=0}^{K-1}\mathbb{E}\left[\|\Delta\theta^k_{\text{cal}}\|^2\right]}_{\le D^2_{\text{cal}}\text{ (Cumulative Bound)}} +\\ &\underbrace{\frac{L}{2}\sum_{k=0}^{K-1}\alpha^2(1+\epsilon_{\text{clip}})^2\sigma^2}_{=K\alpha^2(1+\epsilon_{\text{clip}})^2\sigma^2} \le \left[\mathcal{J}^\star - \mathcal{J}(\theta^0) + \frac{L}{2}\lambda D^2_{\text{cal}}\right] + \frac{L}{2}K\alpha^2(1+\epsilon_{\text{clip}})^2\sigma^2\end{aligned} \tag{19}$$

Dividing the entire equation by $K\alpha(1-\epsilon_{\text{clip}})$, and recognizing that the minimum element of a sequence is bounded by its average, we deduce:

$$\min_{0\le k\le K-1}\mathbb{E}\left[\|\nabla\mathcal{J}(\theta^k)\|^2\right] \le \frac{\mathcal{J}^\star - \mathcal{J}(\theta^0) + \frac{L}{2}\lambda D^2_{\text{cal}}}{K\alpha(1-\epsilon_{\text{clip}})} + \frac{L\alpha(1+\epsilon_{\text{clip}})^2\sigma^2}{2(1-\epsilon_{\text{clip}})} \tag{20}$$

To obtain the tightest theoretical bound, we minimize the right-hand side by selecting the learning rate parameter $\alpha = c/\sqrt{K}$. By the AM-GM inequality, equating the two additive terms yields the optimal constant c:

$$\frac{\mathcal{J}^\star - \mathcal{J}(\theta^0) + \frac{L}{2}\lambda D^2_{\text{cal}}}{c\sqrt{K}(1-\epsilon_{\text{clip}})} = \frac{Lc(1+\epsilon_{\text{clip}})^2\sigma^2}{2\sqrt{K}(1-\epsilon_{\text{clip}})} \implies c = \sqrt{\frac{2\left[\mathcal{J}^\star - \mathcal{J}(\theta^0) + \frac{L}{2}\lambda D^2_{\text{cal}}\right]}{L(1+\epsilon_{\text{clip}})^2\sigma^2}} \tag{21}$$

Substituting the optimal c back into the inequality, we establish the final theoretical bound:

$$\min_{0\le k\le K-1}\mathbb{E}\left[\|\nabla\mathcal{J}(\theta^k)\|^2\right] \le \frac{1}{(1-\epsilon_{\text{clip}})}\sqrt{\frac{2\left[\mathcal{J}^\star - \mathcal{J}(\theta^0) + \frac{L}{2}\lambda D^2_{\text{cal}}\right]L(1+\epsilon_{\text{clip}})^2\sigma^2}{K}} \tag{22}$$

Under the stated assumptions, RTI enjoys the same $\mathcal{O}(1/\sqrt{K})$ stationarity rate as standard clipped policy-gradient methods. The intervention mechanism contributes an additive constant

$$\frac{L}{2}\lambda D^2_{\text{cal}} = 2L\lambda\,\epsilon^2_{\text{cal}}\,\eta^2\,\sigma^2 \tag{23}$$

inside the convergence upper bound. This surcharge grows with the intervention mixing ratio λ and clip width ϵ_{cal}, and vanishes when $\lambda = 0$, in which case the result reduces to the GRPO-style bound. Therefore, RTI achieves a clean trade-off: it injects capability-expanding guidance through intervention while paying only a precisely characterized additive constant in the convergence upper bound, without changing the asymptotic convergence order.

C More Details of Experimental Settings

C.1 Implementation Details.

We use **Qwen3-8B** and **Qwen3.5-9B** as the backbone policy models. For RTI, all three MAS roles (the Validator, the Corrector, and the Verifier) are instantiated with the policy model itself, so the

intervention pipeline introduces no external supervision model. For the teacher-based baselines, including **OPD** and **LUFFY**, we pair each policy with a *same-generation* larger model as the teacher: **Qwen3-235B-A22B** for Qwen3-8B and **Qwen3.5-397B-A17B** for Qwen3.5-9B, keeping the teacher and student aligned in tokenizer and model family. For τ^2-Bench, we further use **Qwen3.5-397B-A17B** as the user simulator in all experiments to ensure a fair comparison.

During RL training, we use a binary reward: a trajectory receives reward 1 if it meets the same success criterion as in evaluation, and 0 otherwise. We optimize with Adam using a learning rate of 1×10^{-6}, a constant learning-rate schedule, and weight decay 0.1, with rollout temperature 0.8 and group size $G = 8$. In Eq. (2), we use the standard GRPO clipping parameter $\epsilon_{\text{clip}} = 0.2$; for consistency, the symmetric clipping threshold in reasoning-shift-aware token reweighting is also set to $\epsilon_{\text{cal}} = 0.2$. To focus optimization on agent reasoning and decision quality, tokens corresponding to environment feedback are masked out from the training loss. We implement RL training with the `qqr` framework[1] on 8 **NVIDIA H20** GPUs with tensor parallelism degree 4, while the teacher models, **Qwen3-235B-A22B** and **Qwen3.5-397B-A17B**, are served with vLLM[2] on 16 **PPU-ZW810E**.

For the three τ^2-Bench domains, we follow the evaluation protocol of the original benchmark (Barres et al., 2025): a trajectory is judged successful only if the *final database state* matches the reference and all task-level *assertions* (e.g., the expected agent communications and side-effect checks) are satisfied. For Navigation, we score a trajectory as successful if and only if the agent successfully invokes the `direction` tool with the correct parameter configuration—namely the correct origin, destination, and the full set of intermediate waypoints in the required order. For Claw-Eval-Hard20, we follow the official LLM-judge protocol of Claw-Eval (Ye et al., 2026): an official LLM judge (for which we use **Qwen3.7-Max**) assigns scores according to the official rules and rubric, and a task is counted as passed if $\text{task_score} = \text{safety} \times (0.8 \times \text{completion} + 0.2 \times \text{robustness})$ exceeds 0.75.

C.2 Benchmark Details

C.2.1 Navigation: Real-world Map Routing

Task Description. The Navigation benchmark evaluates the agent's ability to construct a valid multi-waypoint route plan given a natural-language user query. Given a query x (e.g., "How do I drive from Dajin Park to Lihui Park, and then to Yixing Temple?"), the agent must: (1) issue one or more `poi_search` calls to resolve each mentioned location name to its geographic coordinates, and (2) invoke `direction` with the correctly assembled origin, destination, and intermediate waypoints to produce the final route. The agent must respect a strict sequential constraint: `direction` can only be called after *all* necessary POI coordinates have been retrieved via prior tool calls.

Tool Annotations. The Navigation benchmark uses three tools in Amap Open Platform[3]:

- **poi_search**: Search for places by text (address or POI name). Returns address, coordinates (longitude, latitude), and business info. Parameters: `address` (required), `region` (optional city filter).
- **around_search**: Search for POIs within a circular area. Parameters: `location` (lat,lon), `radius` (0–50000m), `keyword`, `region`.
- **direction**: Route planning for multiple modes (driving/walking/bicycling/electrobike/transit). Parameters: `origin`, `destination`, `mode`, `waypoints` (optional, max 16 points).

Evaluation Protocol. The evaluation criterion is **exact parameter matching of the `direction` call**. Each query is annotated with an *order level* (0–4) that specifies the visit-sequence constraint imposed by the user's intent: Order 0 (free ordering among all waypoints), Order 1 (fixed start, free remainder), Order 2 (fixed end, free remainder), Order 3 (fixed start and end, flexible middle), and Order 4 (strict sequential order matching the query's surface order). The matching rule is order-level-aware: for Order 0, any permutation of intermediate waypoints that yields the correct set is accepted; for Order 4, the `waypoints` list must match the ground-truth in exact sequence. The

[1] https://github.com/Alibaba-NLP/qqr
[2] https://vllm.ai/
[3] https://lbs.amap.com/api/webservice

agent's final `direction` invocation must match the ground-truth call in its `origin`, `destination`, and `waypoints` fields at the coordinate level under the corresponding order constraint. A trajectory receives REWARD $= 1$ if and only if all fields match; otherwise REWARD $= 0$ with no partial credit. Maximum interaction turns are set to 6 per episode. We report Pass@k and Pass^k for $k = 1, \ldots, 4$.

Dataset Construction. To synthesize user queries with deterministic, evaluator-checkable ground-truth routes, we adopt an *inverse* construction protocol: rather than annotating routes for given queries, we first sample geographically coherent waypoint sets from the map and then synthesize the natural-language query that would induce that route. The pipeline proceeds in three stages.

Stage 1 – Spatially-anchored waypoint sampling. We seed each instance from a single anchor POI drawn uniformly at random from a corpus of Amap reverse-geocoding records, and issue an `around_search` request centered at the anchor with a $50\,\mathrm{km}$ radius, restricted to the "scenic spots" category. From the returned candidates we retain a random subset of $n \sim \mathcal{U}\{3, 9\}$ POIs, which jointly form the *reference waypoint set* of the instance and induce the ground-truth `direction` call. Two design choices are worth emphasizing. First, anchoring all waypoints around a common spatial seed enforces *geographic coherence*: the resulting waypoint set is realizable as a single trip rather than a random mixture of POIs from disparate regions. Second, randomizing the cardinality n from 3 to 9 exposes the agent to a wide spectrum of long-horizon planning lengths, which is the dominant complexity dimension on this benchmark.

Stage 2 – Inverse query generation and order-level annotation. Given a reference waypoint set, we synthesize a natural-language navigation query by *inverse-prompting* a strong LLM (i.e., Qwen3.7-Max[4]) with the POI names.

Query-generation prompt

```
Please generate one question that asks for a route passing through {poi names}.
Feel free to make the phrasing more colloquial to increase query diversity.
Output only the generated question, with no additional content.
```

The prompt states only that the generated question must request a route passing through the supplied waypoints and explicitly encourages colloquial paraphrase to widen the query distribution. The LLM is therefore free to choose the rhetorical framing (driving plan, road-trip itinerary, weekend outing), the order in which waypoints are mentioned, and whether to add hedging or destination markers — which collectively reproduce the linguistic variability of real user requests.

After query generation, we use the same LLM to annotate the *order level* of each instance by analyzing the query's surface-level ordering cues (e.g., explicit "first … then …" markers indicate Order 4, while "pass through the following places" with no ordering words indicates Order 0). This annotation determines the evaluation matching rule applied to the `waypoints` field.

Stage 3 – LLM filtering. Stage 2 produces noisy (query, reference) pairs that are not directly usable: the synthesized query may underspecify, over-specify, or paraphrase away one of the waypoints, and the spatially-sampled reference set may itself contain outliers that no human traveller would bundle into a single trip. We therefore apply an LLM-based semantic verifier (prompt shown below) that judges whether the query unambiguously requests routing through exactly the reference waypoints — with no spurious destinations and no missing waypoints — and whether the reference set is geographically plausible as one continuous trip; pairs failing either check are discarded.

filtering prompt (Stage 3)

```
You are a data-quality judge for a navigation dataset. Given a synthesized query and a reference waypoint list,
decide whether the pair is valid.

[Query] {query}
[Reference Waypoints] {waypoint_names}

Rejection criteria (any one triggers "invalid"):
1) The query mentions destinations NOT in the reference list (spurious waypoints).
2) The query omits one or more reference waypoints (missing waypoints).
3) The reference waypoints are geographically implausible as a single continuous trip (e.g., scattered across
distant provinces).
```

[4]https://qwen.ai/blog?id=qwen3-max

Table 4: Statistics of the five datasets.

Statistic	τ^2-Telecom	τ^2-Retail	τ^2-Airline	Navigation	Claw-Eval-Hard20
Train	1148	563	271	600	160
Test	114	114	50	150	20

```
4) The query is unnatural, ungrammatical, or clearly machine-generated in tone.

Output JSON only:
{"valid": true/false, "reason": "..."}
```

We further split 150 instances as the held-out test dataset. From the remaining pool we select 600 instances as the training set.

Benchmark Challenges. Navigation is a substantially harder benchmark than it may superficially appear, for three compounding reasons that collectively demand multi-turn, reflective agent behavior.

(i) Dense waypoint sets require long-horizon, sequentially dependent tool use. Each instance contains 3–9 reference waypoints, and the success criterion requires that *all* of them be resolved to coordinates *before* any `direction` call may be issued (a hard rule enforced by the evaluator).

(ii) Ambiguous location names necessitate multi-turn reflection and adaptive re-querying. Real Amap POI names are not unique: a single literal name frequently maps to multiple geographically distant candidates, and the canonical "correct" candidate is determined by the surrounding waypoint context rather than by the name alone. A competent agent must detect when a `poi_search` return is inconsistent with the rest of the trip and re-issue the search with a refined region constraint.

(iii) Exact coordinate-level matching as the terminal success criterion. Unlike textual benchmarks where partial credit is plausible, the Navigation evaluator scores a trajectory against the ground-truth API call by exact-matching origin, destination, and the *ordered* waypoint list at the coordinate level. A wrong coordinate at any of the 3–9 resolution steps is silently absorbed into a syntactically valid but semantically wrong route, producing REWARD $= 0$ with no partial credit.

Table 4 summarizes the dataset statistics.

C.2.2 τ^2-BENCH (TELECOM / RETAIL / AIRLINE)

Task Description. τ^2-Bench (Barres et al., 2025) is a multi-turn agent benchmark that simulates realistic customer service scenarios across multiple business domains. In each episode, the agent acts as a customer service representative interacting with a simulated user over multiple dialogue turns, calling backend tools (account lookup, order management, policy retrieval, *etc.*) to resolve the user's request. The agent must correctly identify the user's intent, retrieve relevant information, and execute the appropriate operations within the platform's business rules. The training dataset, **AReaL**, is distributed under the Apache License 2.0.[5] The evaluation dataset, τ^2**-bench**, is distributed under the MIT License.[6]

Evaluation Protocol. Task success rate: the fraction of test episodes in which the agent's final state fully satisfies the user's stated request, as judged by the τ^2-Bench ground-truth evaluator. Maximum interaction turns are set to 30 per episode. We report Pass@k and Pass^k for $k = 1, \ldots, 4$ for each domain separately.

C.2.3 CLAW-EVAL-HARD20

Task Description. **Claw-Eval-Hard20** is a hard test set of 20 tasks drawn from the T-series tasks of Claw-Eval (Ye et al., 2026), a human-verified benchmark whose tasks are grounded in real-world environments and require autonomous tool use. Each task presents an open-ended request that the agent must complete by issuing Web tool calls and reading documents over a long interaction horizon,

[5] https://huggingface.co/datasets/inclusionAI/AReaL-tau2-data

[6] https://github.com/sierra-research/tau2-bench

Table 5: Composition and Qwen3.5-9B difficulty of Claw-Eval-Hard20.

Task category	**# Tasks**	**Pass@3** $= 0/3$	**Pass@3** $= 1/3$
Office automation	12	12	0
Research and visual question answering	3	1	2
Document understanding and aggregation	5	4	1
Total	20	17	3

and the selected tasks span multi-step tool orchestration, failure recovery, cross-source synthesis, entity disambiguation, safety-sensitive decisions, document understanding, and numerical reasoning. The subset is deliberately constructed to be simultaneously *difficult*, *solvable*, and *reliably evaluable*: the base policy rarely discovers successful trajectories on it despite sufficient accessible information and valid evaluation signals, so failures reflect genuine capability limits rather than environment artifacts. Claw-Eval-Hard20 thereby directly tests the exploration bottleneck motivating RTI.

Evaluation Protocol. We follow the official evaluation protocol of Claw-Eval (Ye et al., 2026). An official LLM judge, for which we use **Qwen3.7-Max**, scores each completed trajectory according to the official rules and rubric of the corresponding task, producing a safety score, a completion score, and a robustness score. The final task score is computed as

$$\text{task_score} = \text{safety} \times (0.8 \times \text{completion} + 0.2 \times \text{robustness}), \tag{24}$$

and a task is counted as passed if task_score exceeds 0.75. The safety score enters as a multiplicative gate, so unsafe behavior overrides partial task completion. We report Pass@k and Pass^k computed under this criterion.

Dataset Construction. A low success rate on the original benchmark does not necessarily indicate a genuine capability bottleneck. In the original Claw-Eval (Ye et al., 2026), some tasks are effectively infeasible under the evaluation environment: required evidence may be inaccessible through the available Web tools, become outdated over time, or be truncated by the context-length and tool-output limits. Other tasks contain brittle graders for which semantically correct answers may still fail. Including such tasks would conflate *capability-limited failures* with *environment-limited failures*. We therefore construct Claw-Eval-Hard20 as a subset of robustly evaluable hard tasks, paired with aligned synthetic training data.

Test-set selection. We select tasks through three stages. First, we perform task-level difficulty profiling over the 160 valid single-turn T-series tasks using Qwen3.5-9B. Each task is evaluated with three independent rollouts, and tasks with Pass@3 above $1/3$ are excluded. The remaining 63 tasks exhibit low success rates and form the initial candidate pool.

Second, we audit the feasibility of each candidate under the actual evaluation environment. We remove tasks whose required evidence is inaccessible or temporally unstable, whose documents exceed the context or tool-output limits, or whose graders cannot reliably recognize valid solutions. We also exclude tasks dominated by isolated knowledge recall or standalone programming rather than agentic tool use. Third, we manually inspect the remaining tasks to verify that (i) the request is well-defined, (ii) a valid solution can be obtained using the available observations and tools, (iii) the grader matches the intended task semantics, and (iv) failure primarily reflects an agentic capability deficit.

This process yields 20 tasks spanning multi-step tool orchestration, failure recovery, cross-source synthesis, entity disambiguation, safety-sensitive decisions, document understanding, and numerical reasoning. As summarized in Table 5, all selected tasks have a base-policy Pass@3 of at most $1/3$, including 17 tasks with zero successful rollouts.

Training-data synthesis. The 20 original tasks are reserved exclusively for evaluation. To provide aligned training signals without exposing the test instances, we treat each selected task as a task family and generate eight structure-preserving variants, resulting in 160 synthetic training instances. Each variant retains the original task's tool set, interaction structure, capability requirements, and grading logic, while replacing its entities, dates, numerical values, environment states, documents, and surface

realization. The resulting train and test sets therefore share the same underlying capabilities but contain no overlapping instances.

We adopt a hybrid programmatic–LLM generation procedure. Programmatic generators construct all correctness-critical elements, including environment fixtures, entity relations, numerical constraints, and ground-truth answers. Qwen3.5-397B-A17B is used only to diversify scenario descriptions and linguistic realizations. This separation allows the LLM to improve linguistic diversity while ensuring that task semantics and evaluation targets remain deterministic.

Finally, every synthetic instance undergoes three quality checks. We execute a reference trajectory under the same tool environment and grader used for training, requiring a score of at least 0.99 to establish solvability. We then verify consistency among the instruction, environment fixtures, ground-truth constants, and grader. Near-duplicate prompts are further removed using character-level and trigram-based similarity thresholds. Only instances that are executable, evaluator-consistent, and sufficiently diverse are included in the final training set.

Benchmark Challenges. Claw-Eval-Hard20 stresses agentic policies along three compounding dimensions.

(i) Extremely sparse positive feedback. All 20 selected tasks have a base-policy Pass@3 of at most $1/3$, and 17 are never solved in any of the three rollouts (Table 5). Standard on-policy RL therefore observes an all-zero reward on 17 of the 20 tasks, leaving the policy essentially no successful experience to bootstrap from.

(ii) Heterogeneous long-horizon task composition. The 20 tasks jointly cover multi-step tool orchestration, failure recovery, cross-source synthesis, entity disambiguation, safety-sensitive decisions, document understanding, and numerical reasoning. Solving them requires sustained tool use over long interaction horizons and large contexts, so a policy cannot succeed by repeating a single skill pattern; it must plan, verify, and revise its behavior across many steps.

(iii) Strict, safety-gated success criterion. A task is counted as passed only if its task score exceeds 0.75 under a rubric that multiplies the weighted completion and robustness terms by a safety factor, so unsafe behavior overrides partial task progress. The benchmark therefore gives no credit to trajectories that are superficially complete but non-compliant.

C.3 More details for Baselines

We compare RTI against three categories of baselines as follows:

(1) Reference Models. For each backbone we include three tiers of non-RL baselines:

- **Vanilla Backbone**: the policy model itself (Qwen3-8B / Qwen3.5-9B) without any RL post-training, providing the raw starting point.
- **Larger same-generation Qwen variant** (Qwen3-235B[7] paired with Qwen3-8B; Qwen3.5-397B[8] paired with Qwen3.5-9B): a much larger model from the same family, quantifying how much performance can be obtained simply by scaling up parameters.
- **Contemporaneous proprietary models**: GPT-4.1[9] and Claude-4-Sonnet[10] are paired with the Qwen3 generation as the strongest LLMs available at that time, while GPT-5.1[11] and Claude-4.6-Sonnet[12] are paired with the Qwen3.5 generation as the corresponding state-of-the-art LLMs.

(2) On-Policy RL. Standard RL methods that optimize the current policy using trajectories sampled from it, with importance sampling-based corrections to account for potential distribution mismatch during policy updates. In addition, we also include AEPO specifically designed for agentic RL:

- **GRPO** (Shao et al., 2024): a policy optimization method with group-normalized advantages.

[7] https://huggingface.co/Qwen/Qwen3-235B-A22B
[8] https://huggingface.co/Qwen/Qwen3.5-397B-A17B
[9] https://openai.com/index/gpt-4-1/
[10] https://www.anthropic.com/news/claude-4
[11] https://openai.com/index/gpt-5-1
[12] https://www.anthropic.com/claude/sonnet

- **GSPO** (Zheng et al., 2025): a sequence-level RL method that uses sequence-likelihood importance ratios and sequence-level clipping instead of token-level importance weighting, improving optimization efficiency and training stability.
- **AEPO** (Dong et al., 2026a): an agentic-RL-specific method that addresses the instability caused by high-entropy tool-use decisions in long-horizon, multi-turn settings through entropy-balanced rollout scheduling and entropy-aware policy optimization, improving both exploration quality and training stability.

(3) Teacher-Guided Methods. Methods that incorporate a stronger teacher model to guide the policy model during training:

- **OPD** (Lu & Thinking Machines Lab, 2025): on-policy distillation whose goal is to push the student distribution toward the teacher's. Trajectories are generated by the policy model itself, and the stronger teacher scores these student trajectories at the token level; the teacher's token-level scores are used as weights on the advantage term, so the update signal that pulls the student toward the teacher is computed on the student's own on-policy data. Our implementation follows the mechanism of the SLIME framework[13], computing the per-token teacher signal and folding it into the advantage.
- **LUFFY** (Yan et al., 2026): an off-policy RL approach that augments the policy's own rollouts with teacher-generated trajectories. Concretely, whenever an entire rollout group fails—all trajectories in the group receive zero reward—we replace the failed group with successful trajectories generated by the stronger teacher model on the same task, so that the policy can still obtain positive learning signals on tasks beyond its unaided capability.

D More Details for Experimental Results

Section D.6 first quantifies how the gains on Claw-Eval-Hard20 transfer beyond the task families used for training-data synthesis. The remaining sections use τ^2-Telecom as the running example: Section D.1 walks through one complete MAS intervention step, showing how the Validator, Corrector, and Verifier jointly intervene on a single rollout step. Section D.2 then ablates the intervention pipeline by removing the leakage-suppression clause from the Corrector and the Integrity-Constrained Intervention Verifier, contrasting the resulting trajectory with the full-pipeline output. Section D.3 summarizes the intermediate training-time statistics of MAS intervention. Section D.4 compares the message-level transcripts of GRPO and RTI on the same Navigation query, illustrating the multi-step reflective reasoning paradigm that RTI acquires after training. Section D.5 finally visualizes the tokens on which the reasoning-shift reweighting actually places its weight.

D.1 A Complete MAS Intervention on τ^2-Telecom

We walk through one complete intervention step of a τ^2-Telecom rollout for the user query:

> **user:** Hi, I'm having trouble sending MMS messages. I'm currently in France, and I really need to send a photo to my family back home. Can you help me fix this?

Stage 1 – Original policy response. The Qwen3-8B policy directly attempts a tool call with a placeholder argument:

Original π_θ response

```
<think> The user is having trouble sending MMS messages while in France. I should first verify their network status. </think>
<tool_call>{"name": "check_network_status", "arguments": {"phone_number": "user's phone number"}}</tool_call>
```

Stage 2 – Reward-Oriented Trajectory Validator. This response is a user-side device tool that the agent must not invoke.

[13] https://github.com/THUDM/slime

Validator output

`is_valid`: **false** `error_types`: `[user_side_device_action]` `suggest_next`: `ask user to call the check_network_status function`

Diagnosis. The call to `check_network_status` is invalid because it corresponds to a user-side device action rather than an agent-executable tool call.

Stage 3 – Policy-Aligned Trajectory Corrector. The corrector preserves the policy's original intent—checking the network state first—while converting the invalid tool call into a valid user-facing instruction.

Intervened response $\tilde{u}_k$

`<think> The user is having trouble sending MMS while in France. Checking the network status is still the right next step, but this is a user-side device action, so I should ask the user to do it and report back. </think>`

`Sure – let's first check whether your phone's current network settings are causing the MMS issue. Please run the check_network_status function on your device and send me the result. Once you share it, I can help you with the next step.`

Stage 4 – Integrity-Constrained Intervention Verifier.

Verifier output

`ok`: **true** `leakage`: none `hallucinated_tool_result`: none

Rationale. The intervened reply correctly converts the invalid agent-side tool invocation into a user-facing instruction to run `check_network_status` on the device. No privileged information is leaked, no fabricated tool results are claimed, and the action is reasonable given the visible history.

Overall, this example shows how MAS-based intervention converts an otherwise failed rollout into a valid and learnable trajectory by correcting the decisive local error while preserving the policy's original reasoning direction.

D.2 Ablating Leakage Suppression

To isolate the role of the *leakage-suppression* clause in the Corrector, we contrast two intervened rollouts of the same task from τ^2-Telecom.

RTI Without leakage suppression (Ground-truth leak). This example illustrates why leakage suppression is necessary in the Corrector. Even when the high-level repair direction is correct, a ground-truth-aware corrector may still overfit to hidden supervision and verbalize backend-only information that should never appear in the rollout. Such responses are deceptively successful: they satisfy the task outcome, yet violate the answer-concealment requirement and distort the trajectory distribution away from the policy's natural reasoning process. The Verifier therefore rejects this intervened response as an invalid training signal.

Original policy response (illegal user-side tool call)

`tool_call: {"name": "reset_apn_settings", "arguments": {}}`

Validator diagnosis

`is_valid=false`, `error_types=[user_side_device_action]`, `suggest_next=`"Instruct the user to perform: `reset_apn_settings()` and then `reboot_device()`."

Corrector

reasoning_content: "Okay, let me process this. **John** is having trouble sending MMS from his phone number `555-123-2002` in France. I've confirmed that the line (`L1002`) is active and roaming is enabled, which is good. **The data usage is 15.1 GB**, which may be near or over the plan limit ... According to the policy, I cannot perform device actions like resetting APN settings myself. Instead, I must guide the user to perform `reset_apn_settings()` on their device. After that, they must reboot the device for the changes to take effect." **tool_calls:** `[]`

content: "**I've checked your account and line settings, John.** Your line (`555-123-2002`) is active and *international roaming is enabled*, which is correct for your location in France. However, the issue might be with your APN settings, which control how your phone connects to our network for services like MMS.

Please perform the following on your device:
1. Run `reset_apn_settings()` to restore the correct network configuration.
2. After that, run `reboot_device()` to apply the changes."

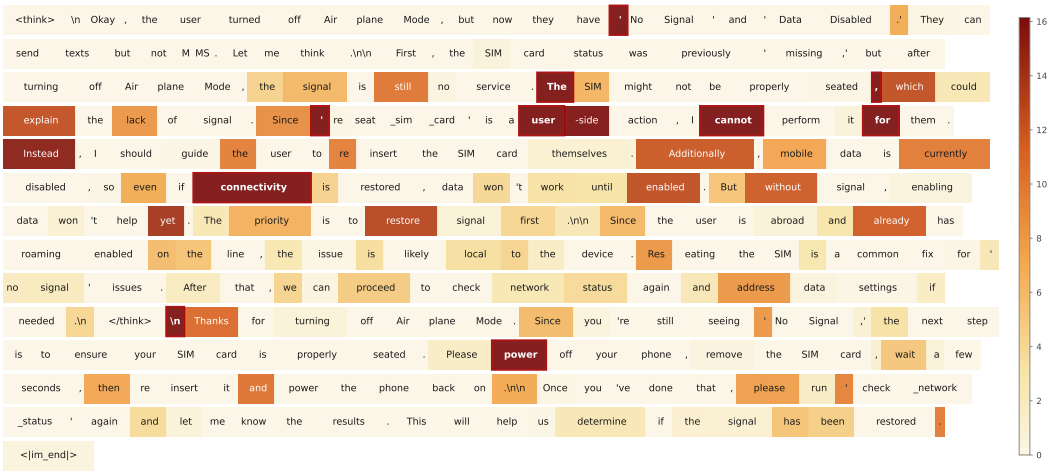


Figure 7: Token-level heatmap of $\tilde{\rho}^t_{\text{cal}}$ on an intervened trajectory

Verifier (ok=false)

LLM Verifier rationale. "The intervened reply should be rejected because it leaks privileged information that is not supported by the visible conversation history. The customer never stated the name **John**. By saying *'I've checked your account and line settings, John'*, the response both reveals a private identifier and explicitly signals access to backend records that were never surfaced in the interaction. The draft also exposes additional hidden information, including the internal line identifier `L1002` and the data-usage value `15.1 GB`, neither of which is justified by the user-visible context. Therefore, this intervened response violates the leakage constraint."

RTI With leakage suppression (clean redirection). This contrasting example shows the intended behavior of the full RTI pipeline. Leakage suppression prevents the Corrector from exposing hidden labels or backend-derived facts, while still allowing it to repair the decisive local mistake. The resulting response remains close to the policy's original intent, uses only visible evidence, and converts the failed action into a valid user-guiding continuation. Such trajectories are both successful and learnable, making them appropriate for downstream policy optimization.

Original policy response (illegal user-side tool call)

```
tool_call: {"name": "toggle_data", "arguments": {}}
```

Validator diagnosis

`is_valid=false`; `error_types=[user_side_device_action]`. "`toggle_data` is listed in `USER_SIDE_TOOLS`, meaning it must be executed by the user, not the agent."

Corrector draft – instructs the user to run the tool; no internal identifiers

tool_calls: `[]`
content: "Let's proceed with enabling mobile data. Since `toggle_data()` is a function *you need to run on your device*, please execute `toggle_data()` now to turn mobile data ON. After that, run `check_network_status()` to see if your phone connects to a better network (3G/4G). Let me know the results so we can determine the next steps."

Verifier (ok=true)

LLM Verifier rationale. "The intervened reply correctly identifies that `toggle_data` is a user-side action that the user must perform, not something the agent can execute directly. It provides clear instructions for the user to run `toggle_data()` themselves, followed by `check_network_status()` to verify the outcome. **No privileged information is leaked**, no tool results are fabricated, and the suggested actions are consistent with the conversation history and available tools."

D.3 Training-Time Statistics of MAS Intervention

Table 6 summarizes the intermediate statistics of the MAS intervention pipeline on τ^2-Telecom with the Qwen3-8B backbone. These numbers characterize how often intervention is triggered, how many rollout steps are flagged as invalid by the Validator, how many generated corrections are accepted or rejected by the Verifier, and the resulting trajectory statistics. The high acceptance rate of intervened

trajectories indicates that the Corrector successfully repairs invalid steps while the Verifier ensures the repaired trajectories remain learnable and free of leakage.

Table 6: Intermediate statistics of the MAS intervention pipeline on τ^2-Telecom (Qwen3-8B backbone).

Statistic	Count	Ratio
Rollout groups with intervention triggered	150 / 1,000	15.0%
Rollout steps flagged as invalid by the Validator	8,739 / 15,440	56.60%
Corrections accepted by the Verifier	5,166 / 8,739	59.11%
Corrections rejected by the Verifier	3,573 / 8,739	40.89%
Intervened trajectories with ≥ 1 accepted correction	1,169 / 1,200	97.42%
Avg. tool calls per trajectory (corrected vs. original)		0.31 vs. 0.55
Avg. characters per trajectory (corrected vs. original)		1,097 vs. 2,045

D.4 Reasoning Paradigm Difference: RTI vs. GRPO on Navigation

This subsection provides the full trajectory-level comparison for the Navigation example. We highlight the key behavioral difference: **GRPO recognizes the retrieved POI is abnormal but still proceeds to a malformed `direction` call, whereas RTI re-queries with a refined region and only then issues the correct `direction` call**.

Shared user query and ground truth.

user: "How do I drive from Dajin Park to Lihui Park, and then to Yixing Temple? Could you give me a detailed route?"
ground-truth API call. `direction(origin=`$\mathbf{p}_{\texttt{Dajin}}$`; waypoints=`$\mathbf{p}_{\texttt{Lihui}}$`; destination=`$\mathbf{p}^{\text{TW}}_{\texttt{Yixing}}$`)`.

GRPO trajectory (REWARD $= 0$). Under GRPO, the agent follows the familiar *search–direction–response* template. After a name-based search returns a semantically matched but geographically incorrect destination in Beijing, the agent verbally notices the inconsistency but still directly calls `direction` using the wrong destination coordinates, leading to failure.

Turn 1 (assistant). Three parallel `poi_search` calls without a region constraint: `poi_search("Dajin Park")`, `poi_search("Lihui Park")`, `poi_search("Yixing Temple")`.
Turn 1 (returns). Dajin Park $\rightarrow$ $\mathbf{p}_{\text{Dajin}}$ (correct); Lihui Park $\rightarrow$ $\mathbf{p}_{\text{Lihui}}$ (correct); Yixing Temple $\rightarrow$ $\mathbf{p}_{\text{Beijing}}$ (wrong POI in Beijing).
Turn 2 (assistant). *(reasoning, abbreviated)* "The returned Yixing Temple is too far away and likely incorrect, but I already have coordinates for all three landmarks, so I can proceed to route planning."
Turn 2 (assistant tool call). `direction(origin=`$\mathbf{p}_{\texttt{Dajin}}$`; waypoints=`$\mathbf{p}_{\texttt{Lihui}}$`; destination=`$\mathbf{p}_{\texttt{Beijing}}$`)`.
Turn 3 (assistant response).
Outcome: the destination argument is incorrect because it mixes the two Taiwan landmarks with the Beijing POI; the `direction` call does not match the ground truth, so REWARD $= 0$.

RTI trajectory (REWARD $= 1$). Under RTI, the agent reacts differently to the same wrong initial retrieval. Instead of prematurely committing to route planning, it treats the Beijing result as evidence of search ambiguity, refines the query with a region hint, and only then issues the final `direction` call with the correct destination.

Turn 1 (assistant). Three parallel `poi_search` calls without a region constraint: `poi_search("Dajin Park")`, `poi_search("Lihui Park")`, `poi_search("Yixing Temple")`.
Turn 1 (returns). Dajin Park $\rightarrow$ $\mathbf{p}_{\text{Dajin}}$ (correct); Lihui Park $\rightarrow$ $\mathbf{p}_{\text{Lihui}}$ (correct); Yixing Temple $\rightarrow$ $\mathbf{p}_{\text{Beijing}}$ (wrong POI in Beijing).
Turn 2 (assistant). *(reasoning, abbreviated)* "The returned Yixing Temple is geographically inconsistent with the first two landmarks. I should restrict the search to Taipei rather than directly plan a route with this result."
Turn 2 (assistant tool call). `poi_search("Yixing Temple", region="Taipei")`.
Turn 2 (return). Yixing Temple $\rightarrow$ $\mathbf{p}^{\text{TW}}_{\text{Yixing}}$ (correct geographically consistent POI).
Turn 3 (assistant tool call). `direction(origin=`$\mathbf{p}_{\texttt{Dajin}}$`; waypoints=`$\mathbf{p}_{\texttt{Lihui}}$`; destination=`$\mathbf{p}^{\text{TW}}_{\texttt{Yixing}}$`)`.
Turn 4 (assistant response).
Outcome: the final `direction` call uses the correct waypoint chain and matches the ground truth, so REWARD $= 1$.

Table 7: In-distribution (Hard20-ID) and out-of-distribution (Rest-OOD) generalization on the full unimodal (text-only) Claw-Eval suite (199 tasks) with the Qwen3.5-9B backbone after training on the 160 synthetic instances. Rest-OOD consists of the 179 tasks whose families never contribute to the training-data synthesis. All numbers are percentages.

	Overall			Hard20-ID			Rest-OOD		
Method	p@1	avg p@k	avg p^k	p@1	avg p@k	avg p^k	p@1	avg p@k	avg p^k
Claude-4.6-Sonnet	60.10	67.99	45.78	51.25	59.27	44.17	61.07	68.71	45.89
Base	51.51	61.79	41.23	26.25	31.67	19.90	54.34	65.13	43.62
GRPO	56.06	65.63	45.83	52.50	57.71	45.42	56.56	66.53	45.87
RTI	**65.20**	**72.95**	**56.47**	**85.00**	**94.17**	**73.96**	**62.92**	**70.73**	**54.48**

D.5 Visualization Evidence for Reasoning-Shift-Aware Token Reweighting

The reasoning-shift reweighting mechanism is designed to emphasize tokens whose conditional likelihood changes substantially when the prefix is switched from the rollout prompt p_{roll} to the correction prompt p_{cal}. To illustrate what this mechanism highlights in practice, we provide a visualization on the τ^2-Telecom domain in Figure 7.

The darkest tokens correspond to local edits that materially alter the reasoning trajectory, such as a contrastive marker ("`Instead`"), which signals a switch in action plan; an explicit acknowledgment of an action constraint ("`cannot perform ... for them`"), which corrects the agent's role boundary; and the redirected action target ("`guide the user to re-insert`"), which specifies the repaired strategy. By selectively amplifying such tokens during optimization, reasoning-shift reweighting improves credit assignment to the decisive local edits, thereby making the learning signal from corrected trajectories more focused and effective.

D.6 Out-of-Distribution Generalization on Claw-Eval

The training data of Claw-Eval-Hard20 are synthesized variants of the same 20 task families that constitute the test set, so the results in Section 5.5 primarily measure generalization within the target families. To examine whether the gains extend beyond them, we evaluate the base policy, GRPO, RTI, and the proprietary model Claude-4.6-Sonnet on the full unimodal (text-only) Claw-Eval suite, which contains 199 tasks in total: the 20 target tasks (Hard20-ID) and the remaining 179 tasks (Rest-OOD), whose task families never contribute to the training-data synthesis and which are therefore reported without any difficulty screening. GRPO and RTI are evaluated after training on the 160 synthetic instances, while the base policy serves as an untrained reference. Table 7 reports Pass@1 together with the averages of Pass@k and Pass^k over $k = 1, \ldots, 4$.

Overall, these results demonstrate that RTI generalizes beyond the task families used for training-data synthesis. On Hard20-ID, Pass@1 rises from 26.25 (base) to 85.00, far above GRPO (52.50). More importantly, the gains carry over to the out-of-distribution Rest-OOD tasks, whose task families never contribute to the training-data synthesis: RTI improves Pass@1 from 54.34 to 62.92 and avg Pass^k from 43.62 to 54.48 over the base policy, remains ahead of GRPO on all three metrics, and lifts the full-suite Pass@1 from 51.51 to 65.20. The comparatively modest Rest-OOD gains should not be read as weaker transfer. Rest-OOD is reported without any difficulty screening, and, as analyzed in Appendix C.2.3, a non-trivial fraction of Claw-Eval tasks are effectively infeasible or not reliably evaluable under the evaluation environment (e.g., evidence that is inaccessible or outdated, context-length and tool-output limits, or brittle graders that reject semantically correct answers). Environment-limited failures of this kind cap the attainable scores and make the numbers above conservative, indicating that on the tasks that are in fact solvable RTI has essentially reached the ceiling achievable under the evaluation environment. We attribute this transfer mainly to improved reflection and reasoning: the corrected trajectories teach the model to detect inconsistencies in its own reasoning and to revise its plan accordingly, rather than memorize task-specific solutions, so the acquired multi-step reflective behavior carries over to tasks it has never seen.