INTERQ-RL: INTERACTION-AWARE CORRECTION OF EXECUTION MISMATCH IN QUANTIZED AGENTIC RL

Anonymous authors
Paper under double-blind review

ABSTRACT

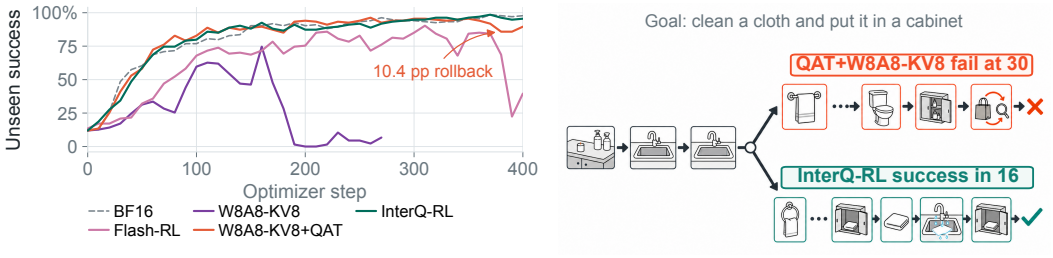
Multi-turn agent reinforcement learning (RL) relies on long closed-loop rollouts, making generation and KV-cache costs a major bottleneck. Quantized rollout reduces these costs, but its low-precision executor also serves as the behavior policy and can induce execution mismatch with the learner. Existing quantized-RL methods are designed primarily for single-turn generation. They align the learner forward with quantized rollout or correct the current-token rollout–learner ratio, without modeling the policy history that produces each agent state. Even after quantization alignment, residual rollout–learner mismatch accumulates along the state-reaching policy history, so a current action may look locally aligned despite a mismatched history. We introduce **InterQ-RL** (Interaction-aware Quantized Reinforcement Learning) to correct this post-alignment mismatch. It applies bounded execution-prefix correction over the state-reaching policy history, allocates correction strength across action segments by effective support under a fixed displacement budget, and requires no additional model forward pass. We evaluate dense and mixture-of-experts (MoE) backbones from 4B to 30B on ALFWorld, WebShop, and ScienceWorld, with all quantized arms sharing the same rollout with 8-bit weights, activations, and KV cache (W8A8-KV8). Across four environment–backbone settings, InterQ-RL achieves the strongest quantized endpoint result, matching BF16 to within 0.7 points and exceeding it in two settings. It matches or exceeds the learning speed of alignment alone while avoiding its late performance regression.

1 INTRODUCTION

On-policy PPO/GRPO optimizes language-agent outcomes using fresh trajectories from the current policy at every update (Guo et al., 2025; Shao et al., 2024). These trajectories can span multiple turns of search, environment interaction, and tool use (Yao et al., 2023; Cheng et al., 2026; Feng et al., 2025). To scale this data-generation loop, modern systems decouple rollout generation from learning (Sheng et al., 2025; Kwon et al., 2023; Hu et al., 2025; Fu et al., 2025). As interactions grow longer, autoregressive generation and the expanding KV cache become major compute and memory bottlenecks on the rollout side. W8A8 reduces linear-operator cost, and KV8 reduces the cache footprint of long-context rollout (Micikevicius et al., 2022; Qiu et al., 2026). Yet the low-precision rollout executor is also the behavior policy. When the executor and learner evaluate the same checkpoint along different numerical paths, they can define different action distributions. We call this *execution mismatch*, which violates the behavior–old-policy consistency assumed by PPO/GRPO (Schulman et al., 2017; Shao et al., 2024).

Prior quantized-RL work addresses training–inference mismatch (Zhong et al., 2026) in two ways: alignment matches the learner forward to quantized rollout (Gu et al., 2026; Xi et al., 2026a), whereas correction clips or reweights the rollout–learner ratio at the current token (Yao et al., 2025; Zhou et al., 2026; Li et al., 2026b; Qiu et al., 2026). Both lines focus on single-turn generation. They do not address residual rollout–learner mismatch that remains after alignment and accumulates over the policy history of multi-turn agents. In a closed-loop agent, earlier actions determine later states. Along the same realized trajectory, shared environment-transition factors cancel, but the ratios of policy actions remain (Sutton & Barto, 2018; Precup et al., 2000). Environment feedback changes later contexts but contributes no policy-likelihood factor. We call this *interaction-structured execution mismatch*.

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.



(a) Alignment reduces collapse but leaves rollback **(b)** Same task and prefix, different outcome

Figure 1: *(a)* ALFWorld `valid_unseen` success, evaluated every 10 steps. W8A8-KV8 is the unaligned run and collapses; W8A8-KV8+QAT is aligned but drops 10.4 points from step 260 to 380 and only partially recovers by step 400. Flash-RL destabilizes late, while InterQ-RL remains stable and approaches BF16 at step 400. *(b)* For one step-400 task, the two arms diverge after sharing their first three actions: W8A8-KV8+QAT exhausts the 30-action budget, whereas InterQ-RL finishes in 16. Across all 134 tasks, 10.45% of W8A8-KV8+QAT episodes and 4.48% of InterQ-RL episodes fail after exhausting the action budget.

We therefore propose InterQ-RL (Interaction-aware Quantized Reinforcement Learning). The method first uses quantization-aware training (QAT) to align the learner forward with rollout quantization. It then forms a bounded execution-prefix correction by accumulating rollout–learner log-ratios along the state-reaching policy history. Each policy-generated action is treated as one allocation segment. For each segment, InterQ-RL computes an effective-support score from the prefix weights and uses it to allocate correction under a fixed squared-displacement budget. The procedure reuses existing log probabilities and requires no additional model forward pass. Figure 1 motivates correction beyond alignment: the aligned baseline remains trainable but regresses late and exhausts the action budget more often than InterQ-RL.

We evaluate InterQ-RL on three agentic environments and 4B–30B dense and MoE Qwen3 backbones (Yang et al., 2025) under a common W8A8-KV8 rollout. It outperforms QAT by 6.0 and 2.8 points on ALFWorld success and WebShop score (Yao et al., 2022) while matching BF16 within 0.7 points on ALFWorld success, 0.3 on WebShop score, and 0.5 on WebShop success. On both ScienceWorld (Wang et al., 2022) backbones, it attains the highest macro and solved endpoints. Its normalized area under the learning curve (AULC) is within 0.33 of QAT on ALFWorld and highest among quantized arms in the other three settings.

This paper makes the following contributions.

1. We characterize *interaction-structured execution mismatch* in quantized agentic RL: W/A+KV alignment leaves residual rollout–learner mismatch that compounds along the state-reaching policy history.
2. We propose InterQ-RL, which applies bounded execution-prefix correction over the state-reaching policy history, allocates correction strength across action segments by effective support under a fixed displacement budget, and requires no additional model forward pass.
3. Using a common W8A8-KV8 rollout, we evaluate three agentic environments and 4B–30B dense and MoE backbones. Across all four settings, InterQ-RL achieves the strongest quantized endpoint result; its AULC is within 0.33 of QAT on ALFWorld and highest among quantized arms in the other three settings.

2 Preliminaries

2.1 Policy Optimization with Quantized Rollouts

Let q_{old} be the learner policy at the rollout checkpoint and q_θ the current policy. In quantized rollout (Li et al., 2026b; Qiu et al., 2026; Zhou et al., 2026), a token x_t in context h_t is sampled from the behavior policy r_{exec} realized by the low-precision executor. Even at a shared checkpoint, r_{exec} and

q_{old} may differ, so the current-policy-to-behavior likelihood ratio factors as

$$\frac{q_\theta(x_t \mid h_t)}{r_{\text{exec}}(x_t \mid h_t)} = \underbrace{\frac{q_\theta(x_t \mid h_t)}{q_{\text{old}}(x_t \mid h_t)}}_{\text{optimization ratio}} \cdot \underbrace{\frac{q_{\text{old}}(x_t \mid h_t)}{r_{\text{exec}}(x_t \mid h_t)}}_{\text{rollout–learner ratio}}. \tag{1}$$

The optimization ratio is clipped by the PPO/GRPO objective (Schulman et al., 2017; Shao et al., 2024); the rollout–learner ratio already differs from one before any parameter update, because the executor evaluates the same weights in low precision.

QAT simulates W8A8-KV8 in the learner forward while retaining high-precision master parameters, gradients, and optimizer states (Liu et al., 2024a; Gu et al., 2026).

2.2 Agentic Trajectories

A language agent alternates policy actions with tool or environment feedback (Yao et al., 2023; Cheng et al., 2026). For task z, an S-step trajectory is

$$\tau = (z, o_0, a_1, o_1, \ldots, a_S, o_S), \qquad R = R(\tau), \tag{2}$$

where o_0 is the initial observation, a_s a policy-generated action segment, o_s the subsequent observation, and R the trajectory-level reward. After flattening into tokens, set $m_t = 1$ at policy-generated positions and $m_t = 0$ at observations and other non-policy positions. Observations enter neither the policy likelihood nor the policy-gradient loss, but remain in the context h_t of later actions.

2.3 Trajectory-Prefix Likelihood Ratios

Let target policy π and behavior policy μ share the task distribution, environment transition kernel, and interaction protocol, and induce trajectory distributions P_π and P_μ. If $P_\pi \ll P_\mu$, the likelihood ratio of a realized behavior prefix is

$$\frac{P_\pi(\tau_{\le t})}{P_\mu(\tau_{\le t})} = \prod_{\substack{u \le t \\ m_u = 1}} \frac{\pi(x_u \mid h_u)}{\mu(x_u \mid h_u)}. \tag{3}$$

This is the standard trajectory change-of-measure identity (Sutton & Barto, 2018; Precup et al., 2000). The policies are evaluated on the same realized history and share the environment kernel, so initial-state and transition factors cancel. The product therefore retains only policy positions from the realized prefix, while feedback remains action-conditioned and enters later contexts.

Equation 3 is the exact prefix likelihood-ratio identity; it determines the policy-history scope of the bounded training surrogate in §3.

3 InterQ-RL: Aligning Execution and Correcting Interaction History

InterQ-RL first aligns the quantization semantics of learner and rollout, then constructs a bounded correction surrogate by detached reweighting of the base PPO/GRPO loss. *Scope* determines which policy history leading to the current state enters the surrogate; *allocation* determines how fixed correction strength is distributed across action segments. InterQ-RL addresses these choices with execution-prefix correction and support-aware allocation, respectively (Figure 2).

3.1 Aligning the Policy that Generated the Data

The first stage reproduces the rollout quantization semantics in the learner forward. Rollout uses blockwise W8A8 linear operators and stores K/V in FP8 E4M3 with per-token-per-head scaling. Standard QAT covers the W/A path. For the KV path we match the learner quantize–dequantize rule to the rollout cache-write kernel, at the same points in the layer. Appendix A gives the full specification. The two forward paths thereby share quantization semantics, while their conditional distributions may still differ at a shared checkpoint.

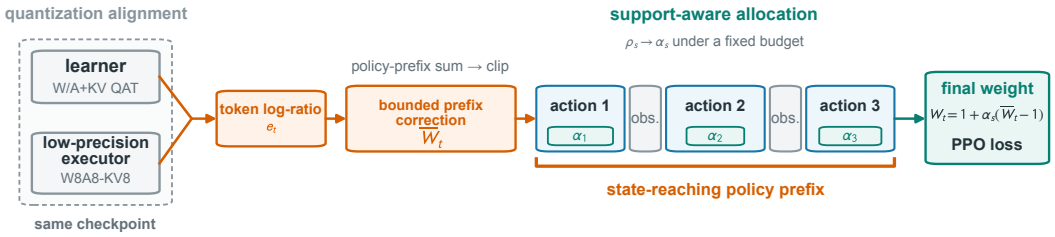


Figure 2: InterQ-RL aligns quantization, corrects the state-reaching policy history, and allocates correction strength by effective support under a fixed displacement budget. Grey: the learner forward matches rollout W/A and KV quantization at a shared checkpoint. Orange: per-token rollout–learner log-ratios form the prefix correction; feedback changes later contexts but adds no policy-likelihood term. Green: segment-level effective support ρ_s determines retention α_s under this budget.

For a rollout token x_t in context h_t, we quantify this residual mismatch by $\delta_t := \log q_{\text{old}}(x_t \mid h_t) - \log r_{\text{exec}}(x_t \mid h_t)$. On executor samples, we report the standard nonnegative K3 estimator for $D_{\text{KL}}(r_{\text{exec}} \| q_{\text{old}})$ (Schulman, 2020; Shao et al., 2024). We use K3 as a diagnostic; Appendix A gives its formula and expectation. Over the final 20 steps in Figure 3, median K3 falls from 1.03 for W8A8-KV8 to 0.029 for W8A8-KV8+QAT, a reduction of roughly one and a half orders of magnitude. The aligned value nevertheless remains about 20× the BF16 value of 0.0014. Alignment therefore removes most of the token-local mismatch; InterQ-RL uses the remainder to construct its history correction.

3.2 Correcting the History that Produced the Current State

After alignment, token-local correction still compares only the conditional probabilities of the current token x_t in the realized context h_t; it omits how the policies differ in reaching that context.

Take the target and behavior policies of §2.3 to be q_{old} and r_{exec}, two evaluations of the same checkpoint along different numerical paths. For a realized history h_t and current token x_t,

$$\frac{P_{q_{\text{old}}}(h_t, x_t)}{P_{r_{\text{exec}}}(h_t, x_t)} = \underbrace{\frac{P_{q_{\text{old}}}(h_t)}{P_{r_{\text{exec}}}(h_t)}}_{\text{history factor}} \cdot \underbrace{\frac{q_{\text{old}}(x_t \mid h_t)}{r_{\text{exec}}(x_t \mid h_t)}}_{\text{current-token factor}}. \tag{4}$$

Equation 4 isolates the term omitted by token-local correction: existing token-local rules retain only the current-token factor (Qiu et al., 2026; Zhou et al., 2026). A current-token factor near one does not imply a history factor near one. Earlier actions determine the current observation and reachable state.

Let $m_t = 1$ mark policy tokens and $m_t = 0$ observations or other non-policy positions. After an implementation safety clip, InterQ-RL accumulates token log-ratios only over policy positions:

$$\begin{aligned} e_t &:= \text{clip}(\delta_t, -B, B), \qquad n_t := \sum_{u \le t} m_u, \\ E_t &:= \sum_{\substack{u \le t \\ m_u = 1}} e_u, \qquad \overline{W}_t := \exp[\text{clip}(E_t, -\epsilon_{\text{low}}\sqrt{n_t}, \epsilon_{\text{high}}\sqrt{n_t})]\,. \end{aligned} \tag{5}$$

The mask excludes environment feedback from the log-ratio sum while retaining earlier action factors in later state-reaching prefixes, matching the policy-position scope of the exact product in §2.3. To keep clipping strength comparable across policy positions, we scale the log-space envelope with $\sqrt{n_t}$ (Zhang et al., 2026b); here it bounds the correction for residual mismatch between the two forward paths. We reuse its envelope parameters, $\epsilon_{\text{low}} = 0.025$ and $\epsilon_{\text{high}} = 0.05$; the per-token log-ratio clip uses verl's default numerical-safety bound B. The weight $\overline{W}_t$ enters as a detached coefficient, so gradients flow only through the current-token score function.

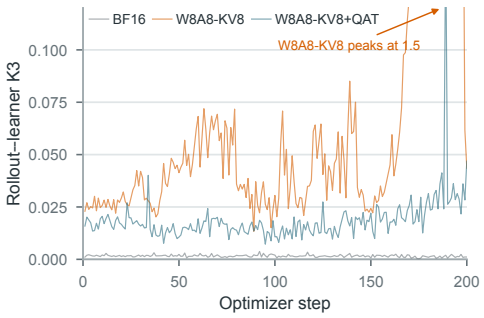


Figure 3: W8A8-KV8+QAT reduces rollout–learner K3 relative to unaligned W8A8-KV8 at shared weights, but does not eliminate it. Each curve compares the learner and executor at matched checkpoints.



Figure 4: A current action can look locally aligned despite a mismatched state-reaching history. 18,683 action segments from ten consecutive updates continued from an aligned checkpoint; color encodes interaction turn.

Figure 4 shows current-action and state-reaching prefix ratios separating with interaction depth, motivating correction over the policy history rather than the current token alone.

3.3 Allocating Correction by Support, Not Magnitude

Let an action segment s be a maximal contiguous run of $m_t = 1$ with n_s tokens. For a segment with $n_s > 1$, define

$$\mu_s := \frac{1}{n_s} \sum_{t \in s} \overline{W}_t, \qquad \nu_s := \frac{1}{n_s} \sum_{t \in s} \overline{W}_t^2, \qquad \mathrm{ESS}_s := \frac{\left(\sum_{t \in s} \overline{W}_t\right)^2}{\sum_{t \in s} \overline{W}_t^2},$$
$$\rho_s := \left[\frac{\mathrm{ESS}_s - 1}{n_s - 1} \cdot \frac{\nu_s}{\nu_s + \varepsilon_{\mathrm{mom}}} \right]_{[0,1]}, \quad C_s := \frac{1}{n_s} \sum_{t \in s} \left(\overline{W}_t - 1\right)^2. \tag{6}$$

Here C_s measures correction magnitude as mean squared displacement from identity. The score ρ_s summarizes prefix-weight concentration within an action segment through length-normalized ESS; higher values indicate broader support. InterQ-RL uses ρ_s as the allocation priority. The second-moment factor stabilizes near-zero moments, and uniform weights give $\rho_s \approx 1$. Segment normalization treats each action as one unit rather than favoring longer text. The score follows normalized ESS (Elvira et al., 2022; Zhou et al., 2026).

This distinction occurs in practice: Appendix B gives a matched pair of 24-token segments with $C_s = 0.128882$ and 0.128889, a difference of only 7×10^{-6}. Their six largest squared-displacement contributions carry 76% and 26% of the total, respectively; the latter is close to the uniform share of $6/24 = 25\%$.

To set the overall correction strength, InterQ-RL first defines a uniform reference. Let $\mathcal{S}$ contain all action segments in one rollout/update batch, and define

$$\mu_g := \frac{1}{|\mathcal{S}|} \sum_{s \in \mathcal{S}} \mu_s, \qquad \nu_g := \frac{1}{|\mathcal{S}|} \sum_{s \in \mathcal{S}} \nu_s, \qquad \alpha_g := \mathrm{clip}\left(\frac{\mu_g}{\sqrt{\nu_g + \varepsilon_{\mathrm{mom}}}}, 0, 1 \right). \tag{7}$$

Up to the clip and $\varepsilon_{\mathrm{mom}}$, $\alpha_g^2 = \mu_g^2 / \nu_g$, the moment ratio underlying relative ESS. The scalar α_g sets the uniform-reference strength. InterQ-RL redistributes this strength by solving

$$\max_{\{0 \le \alpha_s \le 1\}} \sum_{s \in \mathcal{S}} C_s \rho_s \alpha_s \qquad \text{subject to} \qquad \sum_{s \in \mathcal{S}} C_s \alpha_s^2 \le \alpha_g^2 \sum_{s \in \mathcal{S}} C_s. \tag{8}$$

The utility $C_s \rho_s \alpha_s$ favors retaining correction with broader support, while $C_s \alpha_s^2$ measures its resulting displacement. The constraint fixes total segment-normalized displacement to the uniform reference; at the optimum it saturates, so allocation redistributes rather than increases correction strength (Appendix B).

Proposition 1. If every segment with $n_s > 1$ satisfies $C_s > 0$ and $\rho_s \in (0, 1]$, and $\alpha_g \in (0, 1)$, then equation 8 has the unique solution $\alpha_s = \min(c\rho_s, 1)$, where $c > 0$ is determined uniquely by the budget equality and can be found by one-dimensional bisection. If all ρ_s are equal then $\alpha_s = \alpha_g$ and InterQ-RL reduces exactly to the uniform reference. The proof and the $n_s = 1$ case are given in Appendix B.

Higher-ρ_s segments therefore retain more prefix correction, whereas lower-ρ_s segments are pulled toward identity. Since $0 \le \alpha_s \le 1$, every final weight lies between 1 and $\overline{W}_t$.

3.4 The InterQ-RL Objective

For a fixed executor batch, rewards and group-relative advantages are detached. Let $\mathrm{seg}(t)$ denote the action segment containing token t, and let $\ell_t^{\mathrm{base}}(\theta)$ be the token-level loss of the underlying PPO/GRPO objective. InterQ-RL defines the empirical objective

$$W_t := 1 + \alpha_{\mathrm{seg}(t)} \left(\overline{W}_t - 1\right), \qquad \mathcal{L}_{\text{InterQ-RL}} := \mathrm{Agg}_{\mathrm{base}} \left[m_t \, \mathrm{stopgrad}(W_t) \ell_t^{\mathrm{base}}(\theta) \right]. \tag{9}$$

Equation 3 gives the exact $q_{\mathrm{old}}/r_{\mathrm{exec}}$ ratio for the state-reaching policy prefix associated with each token; W_t is its bounded, support-allocated counterpart. Equation 9 reweights only the base token loss: W_t accounts for residual execution mismatch after alignment, while the current-policy PPO/GRPO ratio, clipping rule, executor-sampled advantage, and aggregation remain unchanged. With no mismatch, $W_t = 1$ and the objective reduces to the base PPO/GRPO objective. Prefix sums, segment statistics, and the one-dimensional solver are linear in policy-token count and reuse existing rollout and learner log probabilities, requiring no additional model forward pass. Algorithm 1 gives the per-batch procedure.

Algorithm 1 Execution-prefix correction with support-aware allocation. The returned W_t is the coefficient applied in equation 9. Constants: $\epsilon_{\mathrm{low}} = 0.025$, $\epsilon_{\mathrm{high}} = 0.05$, $\varepsilon_{\mathrm{mom}} = 10^{-8}$, and 80 bisection steps.

$e_t \leftarrow \mathrm{clip}(\log q_{\mathrm{old}}(x_t \mid h_t) - \log r_{\mathrm{exec}}(x_t \mid h_t),\ -B,\ B)$
$E_t \leftarrow \sum_{u \le t} m_u e_u$ (state-reaching policy prefix)
$n_t \leftarrow \sum_{u \le t} m_u$
$\overline{W}_t \leftarrow \exp\left[\mathrm{clip}\left(E_t, -\epsilon_{\mathrm{low}}\sqrt{n_t}, \epsilon_{\mathrm{high}}\sqrt{n_t}\right)\right]$
$\mathcal{S} \leftarrow$ maximal runs of consecutive $m_t = 1$ (the action segments)
For each $s \in \mathcal{S}$, compute $\mu_s, \nu_s, \rho_s, C_s$ from equation 6
$\alpha_g \leftarrow \mathrm{clip}\left(\mu_g / \sqrt{\nu_g + \varepsilon_{\mathrm{mom}}}, 0, 1\right)$ (batch means μ_g, ν_g from equation 7)
$c \leftarrow$ root of $\sum_s C_s \min(c\rho_s, 1)^2 = \alpha_g^2 \sum_s C_s$, by bisection
$\alpha_s \leftarrow \min(c\rho_s, 1)$ (Appendix B.5 for $n_s = 1$)
$W_t \leftarrow 1 + \alpha_{\mathrm{seg}(t)} \left(\overline{W}_t - 1\right)$
return W_t

4 Experiments

4.1 Experimental Setup

Settings and comparisons. We evaluate on three multi-turn agent environments: ALFWorld (Shridhar et al., 2021), WebShop (Yao et al., 2022), and ScienceWorld (Wang et al., 2022). ALFWorld and WebShop both use Qwen3-4B-Instruct-2507 and report on 134 `valid_unseen` tasks and 500 official test tasks, respectively. Because Qwen3-4B provides limited learning signal on the more demanding ScienceWorld setting, we use Qwen3-8B and Qwen3-30B-A3B-Instruct-2507, covering dense and MoE backbones (Yang et al., 2025); endpoint evaluation uses 1,819 official test instances. All runs use GRPO (Shao et al., 2024). Table 1 is the primary end-to-end comparison: the published baselines retain their objectives; all arms share task IDs, evaluator, training horizon, and optimizer settings; and all quantized arms share the W8A8-KV8 rollout. Baselines are BF16, Flash-RL with token-local TIS (Yao et al., 2025), AIS (Zhou et al., 2026), QuRL with ACR+UAQ (Li et al., 2026b), and QAT with W/A+KV alignment. Flash-RL uses verl's built-in rollout importance-sampling path; AIS and QuRL are reimplemented from their papers. Appendix C gives method-specific settings.

Table 1: End-to-end comparison of complete methods under a common training budget. ALFWorld and WebShop report the mean over the final six checkpoint evaluations; ScienceWorld reports the full official test at a fixed endpoint checkpoint. ALFWorld and WebShop use Qwen3-4B; ScienceWorld headers give the backbone size. Bold marks the best quantized result. † QuRL collapsed at step 278 and has no common-horizon endpoint.

	ALFWorld	WebShop		ScienceWorld (8B)		ScienceWorld (A3B)	
Method	Success ↑	Score ↑	Success ↑	Macro ↑	Solved ↑	Macro ↑	Solved ↑
BF16	97.1	78.0	54.0	66.9	74.1	67.0	74.3
Flash-RL	64.1	72.8	44.4	63.2	69.2	65.8	71.5
AIS	92.3	76.5	51.5	59.6	64.2	65.0	70.6
QuRL	Coll.†	75.3	51.7	57.1	63.9	64.0	69.6
QAT	90.4	74.9	49.7	65.5	75.8	69.3	76.6
InterQ-RL	**96.4**	**77.7**	**53.5**	**67.8**	**76.1**	**69.9**	**76.9**

Protocol and metrics. All settings train for 400 optimizer steps. ALFWorld and WebShop endpoints are the mean of the final six checkpoint evaluations. ScienceWorld reports the full official test at a fixed endpoint checkpoint. We report the ALFWorld success rate, the WebShop score and success rate, and the ScienceWorld macro normalized score and per-episode solved rate. Success and solved rates count fully completed tasks, whereas the WebShop score and the ScienceWorld macro score credit partial progress. To separate endpoint quality from learning speed, we report normalized AULC: the trapezoidal area under the evaluation curve, with scores recorded every 10 steps and the area divided by the common 400-step horizon. ALFWorld uses the same `valid_unseen` set, WebShop the same 500-instruction official test set, and ScienceWorld the 259-instance development monitor. Interaction cost is the mean number of turns over successful endpoint episodes; cap exhaustion counts unsuccessful episodes that consume the environment's full action budget. Appendix C specifies baseline rules, data splits, evaluation protocols, hardware, and quantization.

4.2 End-to-End Results

Endpoint performance and training dynamics. InterQ-RL gives the strongest quantized result on every metric in Table 1, including the four that count fully completed tasks. It improves over QAT by 6.0 and 2.8 points on ALFWorld success and WebShop score and by 2.3 and 0.6 on the two ScienceWorld macro scores, and over the strongest of Flash-RL, AIS, and QuRL by 1.2 to 6.9 points. It matches BF16 within 0.7 points on ALFWorld success, 0.3 on WebShop score, and 0.5 on WebShop success, and exceeds it by 0.9 and 2.9 on the ScienceWorld macro scores and 2.0 and 2.6 on the solved rates. Figure 5 places the ALFWorld endpoint in its training context. QAT reaches 95.5% at step 300 but falls to 89.6% by step 400, while InterQ-RL is still at 95.5%. The endpoint separation shows that interaction correction sustains task performance. Flash-RL drops from 84.3% at step 370 to 39.6% at step 400. QuRL falls to 6.0% at step 270 as its training reward falls from a peak of 0.992 to 0.141, and terminates at step 278. AIS avoids collapse but learns more slowly and unevenly: it reaches 69.4% at step 150 versus 88.1% for InterQ-RL, drops from 79.9% at step 210 to 67.9% ten steps later, and reaches a 92.3% last-six endpoint versus 96.4% for InterQ-RL.

Learning speed and interaction cost. In Table 2, InterQ-RL has the highest quantized AULC in three settings and is within 0.33 of QAT on ALFWorld (81.96 versus 82.29), while remaining above Flash-RL, AIS, and QuRL in all four. While these three baselines are 2.0–17.2 points below BF16 across common-horizon results, InterQ-RL stays within 1.9 points in the first three settings and exceeds BF16 by 3.3 on ScienceWorld/30B-A3B. InterQ-RL also reduces interaction cost. It yields the shortest successful trajectories on ALFWorld and WebShop, reducing the QAT averages from 12.72 to 10.06 and from 6.51 to 4.60 turns, respectively. On the 117 ALFWorld tasks solved by both methods, InterQ-RL averages 9.64 turns versus 12.79 for QAT. In both ScienceWorld settings, it remains within 0.11 turns of the shortest quantized arm. Cap exhaustion also falls from 10.45% to 4.48% on ALFWorld and from 1.60% to 0.40% on WebShop. On ALFWorld and both ScienceWorld settings, the arm with the lowest AULC also has the longest successful trajectories, whereas InterQ-RL combines higher AULC with shorter trajectories.

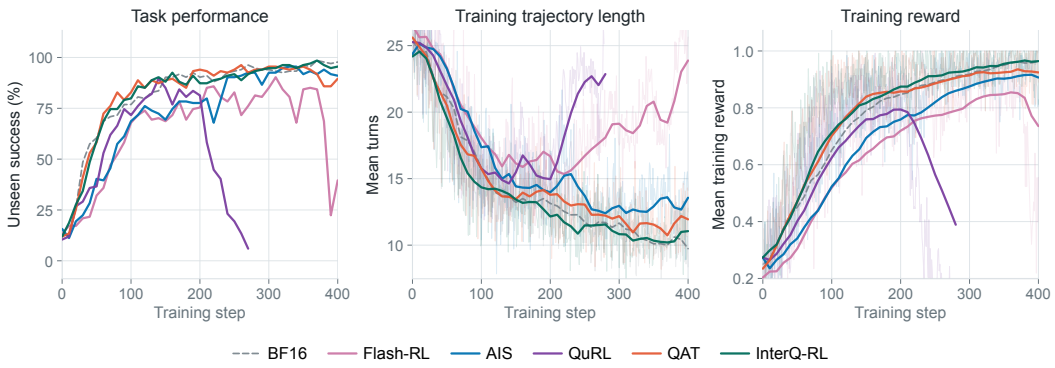


Figure 5: On ALFWorld, InterQ-RL remains stable while QAT rolls back, Flash-RL destabilizes, QuRL collapses, and AIS learns more slowly. Mean turns in the middle panel are averaged over all training trajectories. For trajectory length and reward, smoothed trends use 10-step block medians followed by light EMA.

Table 2: Learning speed and the interaction cost of successful trajectories. ALFWorld and WebShop use Qwen3-4B; ScienceWorld headers give the backbone size. Turns average over successful episodes only; bold marks the best quantized AULC. QuRL collapsed on ALFWorld at step 278 and has no result on the common horizon.

	ALFWorld		WebShop		ScienceWorld (8B)		ScienceWorld (A3B)	
Method	AULC ↑	Turns ↓	AULC ↑	Turns ↓	AULC ↑	Turns ↓	AULC ↑	Turns ↓
BF16	82.1	10.98	75.2	5.23	56.4	20.45	54.1	19.04
Flash-RL	64.9	20.34	72.5	5.09	47.6	31.93	52.1	20.31
AIS	72.5	12.43	72.8	6.61	45.4	22.32	49.6	18.07
QuRL	—	—	72.2	6.23	50.8	21.19	45.5	32.32
QAT	**82.3**	12.72	72.3	6.51	54.4	22.38	54.1	20.20
InterQ-RL	82.0	10.06	**73.3**	4.60	**54.6**	21.20	**57.4**	18.18

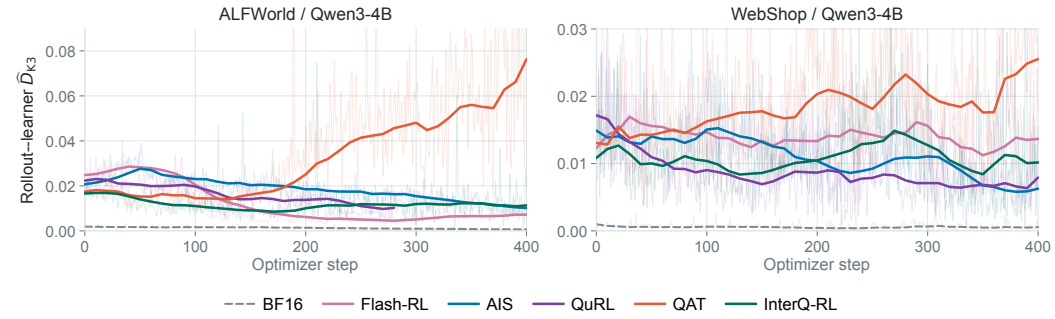


Figure 6: During continued optimization, K3 grows under QAT and remains stable under InterQ-RL. K3 is computed on the trajectories of each arm and shown as smoothed trends. The QuRL curve on ALFWorld ends at step 278.

Rollout–learner K3. Over the first 100 ALFWorld steps, QAT and InterQ-RL have similar median K3 (0.0155 versus 0.0125); by steps 301–400 they separate to 0.0598 and 0.0119 (Figure 6). WebShop shows the same pattern, with QAT and InterQ-RL ending at 0.0210 and 0.0081. On each arm's rollouts, QAT's rollout–learner gap widens, whereas InterQ-RL's remains stable. A low K3 alone does not imply strong task performance: QuRL's K3 falls as it collapses, while Flash-RL

and AIS end with K3 values of 0.0071 and 0.0107 despite lower success (64.1% and 92.3%) than InterQ-RL (0.0119 at 96.4%); AIS also has lower AULC (72.5 versus 82.0).

Additional design analyses. Appendix C.4 provides two additional analyses on ALFWorld. Prefix applies the bounded prefix weights directly, Global uses one coefficient across all segments, and InterQ-RL allocates segment-specific coefficients by effective support (Figure 9). Prefix and Global reach 88.8% and 94.8% success at steps 290 and 230 but end at 37.3% and 22.4% at step 400, whereas InterQ-RL ends at 95.5%. Global and InterQ-RL use the same segment-normalized displacement budget. These results show that prefix correction provides useful learning signal, while support-aware allocation sustains late-stage performance under a fixed correction budget. As an additional ablation, Table 6 pairs Flash-RL's TIS rule and the AIS correction rule with the same QAT learner used by InterQ-RL. QAT+TIS and QAT+AIS reach 90.55% and 92.66% last-six success, compared with 90.42% for QAT alone, while their AULCs fall from 82.29 to 70.59 and 76.27. Combining QAT with token-local correction therefore does not resolve the residual mismatch in multi-turn agent RL. InterQ-RL reaches 96.39% success with 81.96 AULC and lowers cap exhaustion to 4.48%, showing the effectiveness of our method for quantized agent RL.

5 Related Work

Quantized LLM reinforcement learning. Quantized LLM RL builds on weight, activation, and KV-cache quantization (Dettmers et al., 2022; Xiao et al., 2023; Liu et al., 2024b; DeepSeek-AI et al., 2025), and spans learner–rollout alignment and mismatch correction. FP8-RL quantizes rollout W/A and the KV cache (Qiu et al., 2026). QaRL and Jet-RL align the learner forward with quantized rollout (Gu et al., 2026; Xi et al., 2026a). Flash-RL, AIS, and QuRL control training–inference mismatch through clipping, adaptive weighting, or update-aware quantization (Yao et al., 2025; Zhou et al., 2026; Li et al., 2026b), all in single-turn generation. InterQ-RL addresses residual execution mismatch along the policy histories of multi-turn agents.

Policy optimization under distribution shift. Off-policy RL corrects distribution shift through importance weighting (Precup et al., 2000; Munos et al., 2016; Espeholt et al., 2018). LLM methods span response-level objectives, cumulative-token ratios, prefix surrogates, and whole-sequence weighting or masking (Zheng et al., 2025; Zhang et al., 2026b; Lei et al., 2026; Zhang et al., 2026a; Li et al., 2026a). These methods address general rollout or policy-version mismatch at different granularities. InterQ-RL instead targets quantization-induced mismatch that remains after alignment and allocates bounded correction across action segments by effective support under a fixed displacement budget.

Agentic reinforcement learning. GiGPO and LOOP extend policy optimization to multi-turn interaction, targeting credit assignment and training efficiency (Feng et al., 2025; Chen et al., 2025); Agent-R1, AgentGym-RL, and SkyRL-Agent provide open training frameworks (Cheng et al., 2026; Xi et al., 2026b; Cao et al., 2025). InterQ-RL addresses a complementary execution-level source of distribution shift: mismatch introduced when the behavior policy runs in low precision and propagated through action-conditioned history.

6 Conclusion

We identified interaction-structured execution mismatch in quantized agentic RL: even after W/A+KV alignment, residual rollout–learner mismatch compounds along the state-reaching policy history. InterQ-RL applies bounded execution-prefix correction along this history and allocates correction strength across action segments by effective support under a fixed displacement budget, reusing existing log probabilities without another forward pass. Across four settings spanning three environments and 4B–30B dense and MoE backbones under a common W8A8-KV8 rollout, InterQ-RL achieves the strongest quantized endpoint result in every setting, matches BF16 to within 0.7 points and exceeds it in two. Its AULC is within 0.33 of QAT on ALFWorld and highest among quantized arms in the other three settings.

AI USE STATEMENT

During the preparation of this manuscript we used large language model assistants only to polish the writing, at the level of grammar, flow, and tone within sentences and paragraphs. They were not used to generate ideas, design experiments, or draw conclusions. All technical content, methodology, and interpretation were independently written, thoroughly verified, and approved by the authors. To guard against factual and citation errors, the authors reviewed every edited sentence and checked each citation against its primary source. The authors take full responsibility for the accuracy and integrity of this manuscript.

ETHICS STATEMENT

This work uses only publicly available simulated environments and open-weight models that are already established in the research literature. No human subjects are involved and no personal data is collected. We release no new text, image, or audio corpus, and use the environments and checkpoints for research purposes only.

REPRODUCIBILITY STATEMENT

To let the community replicate our findings, this project will be released as open-source software. The Supplementary Material includes our code. The method and per-batch procedure are specified in §3 and Algorithm 1; Appendices A and B give the quantization details, proof, and edge cases. Section 4.1 and Appendix C give the complete experimental protocol, from data splits and hyperparameters to hardware.

REFERENCES

Shiyi Cao, Dacheng Li, Fangzhou Zhao, Shuo Yuan, Sumanth R. Hegde, Connor Chen, Charlie Ruan, Tyler Griggs, Shu Liu, Eric Tang, Richard Liaw, Philipp Moritz, Matei Zaharia, Joseph E. Gonzalez, and Ion Stoica. SkyRL-Agent: Efficient RL training for multi-turn LLM agent, 2025. URL https://arxiv.org/abs/2511.16108.

Kevin Chen, Marco Cusumano-Towner, Brody Huval, Aleksei Petrenko, Jackson Hamburger, Vladlen Koltun, and Philipp Krähenbühl. Reinforcement learning for long-horizon interactive LLM agents, 2025. URL https://arxiv.org/abs/2502.01600.

Mingyue Cheng, Shuo Yu, Daoyu Wang, Qingchuan Li, Xiaoyu Tao, Jie Ouyang, Yucong Luo, Yitong Zhou, Qi Liu, and Enhong Chen. Agent-R1: A unified and modular framework for agentic reinforcement learning, 2026. URL https://arxiv.org/abs/2511.14460.

DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, et al. DeepSeek-V3 technical report, 2025. URL https://arxiv.org/abs/2412.19437.

Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. GPT3.int8(): 8-bit matrix multiplication for transformers at scale. In *Advances in Neural Information Processing Systems*, volume 35, pp. 30318–30332. Curran Associates, Inc., 2022. doi: 10.52202/068431-2198. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/c3ba4962c05c49636d4c6206a97e9c8a-Paper-Conference.pdf.

Víctor Elvira, Luca Martino, and Christian P. Robert. Rethinking the effective sample size. *International Statistical Review*, 90(3):525–550, 2022. doi: https://doi.org/10.1111/insr.12500. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/insr.12500.

Lasse Espeholt, Hubert Soyer, Remi Munos, Karen Simonyan, Vlad Mnih, Tom Ward, Yotam Doron, Vlad Firoiu, Tim Harley, Iain Dunning, Shane Legg, and Koray Kavukcuoglu. IMPALA: Scalable distributed deep-RL with importance weighted actor-learner architectures. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pp. 1407–1416. PMLR, 2018. URL https://proceedings.mlr.press/v80/espeholt18a.html.

Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. Group-in-group policy optimization for LLM agent training. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=QXEhBMNrCW.

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*, 2025. URL https://openreview.net/forum?id=X9diEuva9R.

Hao Gu, Hao Wang, Jiacheng Liu, Lujun Li, Qiyuan Zhu, Bei Liu, Binxing Xu, Lei Wang, Xintong Yang, Sida Lin, Sirui Han, and Yike Guo. QaRL: Rollout-aligned quantization-aware RL for fast and stable training under training–inference mismatch. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 17051–17064. Association for Computational Linguistics, July 2026. doi: 10.18653/v1/2026.findings-acl.841. URL https://aclanthology.org/2026.findings-acl.841/.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning. *Nature*, 645:633–638, 2025. doi: 10.1038/s41586-025-09422-z. URL https://doi.org/10.1038/s41586-025-09422-z.

Jian Hu, Xibin Wu, Wei Shen, Jason Klein Liu, Weixun Wang, Songlin Jiang, Haoran Wang, Hao Chen, Bin Chen, Wenkai Fang, Xianyu, Yu Cao, Haotian Xu, and Yiming Liu. OpenRLHF: A ray-based easy-to-use, scalable and high-performance RLHF framework. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 656–666. Association for Computational Linguistics, November 2025. doi: 10.18653/v1/2025.emnlp-demos.48. URL https://aclanthology.org/2025.emnlp-demos.48/.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with PagedAttention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, pp. 611–626. Association for Computing Machinery, 2023. doi: 10.1145/3600006.3613165. URL https://doi.org/10.1145/3600006.3613165.

Shiye Lei, Zhihao Cheng, and Dacheng Tao. A step back: Prefix importance ratio stabilizes policy optimization, 2026. URL https://arxiv.org/abs/2601.22718.

Yingru Li, Jiacai Liu, Jiawei Xu, Yuxuan Tong, Ziniu Li, Qian Liu, and Baoxiang Wang. Trust region masking for long-horizon llm reinforcement learning, 2026a. URL https://arxiv.org/abs/2512.23075.

Yuhang Li, Reena Elangovan, Xin Dong, Priyadarshini Panda, and Brucek Khailany. QuRL: Low-precision reinforcement learning for efficient reasoning. In *The Fourteenth International Conference on Learning Representations*, 2026b. URL https://openreview.net/forum?id=eG0bpCwdKn.

Zechun Liu, Barlas Oguz, Changsheng Zhao, Ernie Chang, Pierre Stock, Yashar Mehdad, Yangyang Shi, Raghuraman Krishnamoorthi, and Vikas Chandra. LLM-QAT: Data-free quantization aware training for large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 467–484. Association for Computational Linguistics, August 2024a. doi: 10.18653/v1/2024.findings-acl.26. URL https://aclanthology.org/2024.findings-acl.26/.

Zirui Liu, Jiayi Yuan, Hongye Jin, Shaochen Zhong, Zhaozhuo Xu, Vladimir Braverman, Beidi Chen, and Xia Hu. KIVI: A tuning-free asymmetric 2bit quantization for KV cache. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pp. 32332–32344. PMLR, 2024b. URL https://proceedings.mlr.press/v235/liu24bz.html.

Paulius Micikevicius, Dusan Stosic, Neil Burgess, Marius Cornea, Pradeep Dubey, Richard Grisenthwaite, Sangwon Ha, Alexander Heinecke, Patrick Judd, John Kamalu, Naveen Mellempudi, Stuart Oberman, Mohammad Shoeybi, Michael Siu, and Hao Wu. FP8 formats for deep learning, 2022. URL https://arxiv.org/abs/2209.05433.

Rémi Munos, Thomas Stepleton, Anna Harutyunyan, and Marc G. Bellemare. Safe and efficient off-policy reinforcement learning. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pp. 1054–1062. Curran Associates Inc., 2016.

Doina Precup, Richard S. Sutton, and Satinder P. Singh. Eligibility traces for off-policy policy evaluation. In *Proceedings of the Seventeenth International Conference on Machine Learning*, pp. 759–766. Morgan Kaufmann Publishers Inc., 2000.

Zhaopeng Qiu, Shuang Yu, Jingqi Zhang, Shuai Zhang, Xue Huang, Jingyi Yang, and Junjie Lai. FP8-RL: A practical and stable low-precision stack for LLM reinforcement learning, 2026. URL https://arxiv.org/abs/2601.18150.

John Schulman. Approximating KL divergence, 2020. URL http://joschu.net/blog/kl-approx.html.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017. URL https://arxiv.org/abs/1707.06347.

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.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. HybridFlow: A flexible and efficient RLHF framework. In *Proceedings of the Twentieth European Conference on Computer Systems*, pp. 1279–1297. Association for Computing Machinery, 2025. doi: 10.1145/3689031.3696075. URL https://doi.org/10.1145/3689031.3696075.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Cote, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. ALFWorld: Aligning text and embodied environments for interactive learning. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=0IOX0YcCdTn.

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 2018. URL http://incompleteideas.net/book/the-book-2nd.html.

Ruoyao Wang, Peter Jansen, Marc-Alexandre Côté, and Prithviraj Ammanabrolu. ScienceWorld: Is your agent smarter than a 5th grader? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 11279–11298. Association for Computational Linguistics, December 2022. doi: 10.18653/v1/2022.emnlp-main.775. URL https://aclanthology.org/2022.emnlp-main.775/.

Haocheng Xi, Charlie Ruan, Peiyuan Liao, Yujun Lin, Han Cai, Yilong Zhao, Shuo Yang, Kurt Keutzer, Song Han, and Ligeng Zhu. Jet-RL: Enabling on-policy FP8 reinforcement learning with unified training and rollout precision flow, 2026a. URL https://arxiv.org/abs/2601.14243.

Zhiheng Xi, Jixuan Huang, Chenyang Liao, Baodai Huang, Jiaqi Liu, Honglin Guo, yajie yang, Rui Zheng, Junjie Ye, Jiazheng Zhang, Wenxiang Chen, Wei He, Yiwen Ding, Guanyu Li, Zehui Chen, Zhengyin Du, Xuesong Yao, Yufei Xu, Jiecao Chen, Tao Gui, Zuxuan Wu, Qi Zhang, Xuanjing Huang, and Yu-Gang Jiang. AgentGym-RL: An open-source framework to train LLM agents for long-horizon decision making via multi-turn RL. In *The Fourteenth International Conference on Learning Representations*, 2026b. URL https://openreview.net/forum?id=ZgCCDwcGwn.

Guangxuan Xiao, Ji Lin, Mickael Seznec, Hao Wu, Julien Demouth, and Song Han. SmoothQuant: Accurate and efficient post-training quantization for large language models. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pp. 38087–38099. PMLR, 2023. URL https://proceedings.mlr.press/v202/xiao23c.html.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, et al. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Feng Yao, Liyuan Liu, Dinghuai Zhang, Chengyu Dong, Jingbo Shang, and Jianfeng Gao. On the rollout-training mismatch in modern RL systems. In *OPT 2025: Optimization for Machine Learning*, 2025. URL https://openreview.net/forum?id=xsonlP8DRV.

Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. Webshop: Towards scalable real-world web interaction with grounded language agents. In *Advances in Neural Information Processing Systems*, volume 35, pp. 20744–20757. Curran Associates, Inc., 2022. doi: 10.52202/068431-1508. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/82ad13ec01f9fe44c01cb91814fd7b8c-Paper-Conference.pdf.

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.

Wenhao Zhang, Yibo Xie, Rui Wang, Jiahua Yang, Lei Jiang, Zibo Yang, Yawei Wang, Jiali Xu, jasperawang, Haoyang Long, Huan Xiong, and alantzhao. Reusing rollouts under policy lag: Prefix-normalized policy optimization for LLM reinforcement learning, 2026a. URL https://arxiv.org/abs/2608.01418.

Yuheng Zhang, Chenlu Ye, Shuowei Jin, Changlong Yu, Wei Xiong, Saurabh Sahu, and Nan Jiang. Rethinking importance sampling in LLM policy optimization: A cumulative token perspective, 2026b. URL https://arxiv.org/abs/2605.07331.

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.

Tianle Zhong, Neiwen Ling, Yifan Pi, Zijun Wei, Tianshu Yu, Geoffrey Fox, Peng Wu, and Xiao Yu. Diagnosing training inference mismatch in LLM reinforcement learning, 2026. URL https://arxiv.org/abs/2605.14220.

Jiajun Zhou, Wei Shao, Lingchao Zheng, Yuwei Fan, and Ngai Wong. AIS: Adaptive importance sampling for quantized RL, 2026. URL https://arxiv.org/abs/2605.13907.

A Quantization Alignment and Execution Diagnostics

A.1 Linear weight and activation path

The rollout executor quantizes linear operators to FP8 E4M3 with block scaling: weight scales cover 128×128 blocks and activation scales cover 1×128 blocks. The learner forward applies the matching quantize–dequantize rule to the same operands, while master parameters, gradients, and optimizer states stay in high precision. Scales are stored in FP32. Quantization covers every linear layer except `lm_head`, the embedding, and the MoE gate.

A.2 KV cache path

The rollout executor stores the KV cache in `fp8_per_token_head` format. For the K or V vector x at a given token and attention head, the cache-write kernel computes

$$s(x) := \max\left(\frac{\max_j |x_j|}{448},\, 10^{-6}\right), \qquad \widehat{x} := s(x)\ \mathrm{FP8}_{\mathrm{E4M3}}\left[\mathrm{clip}\left(\frac{x}{s(x)}, -448, 448\right)\right]. \tag{10}$$

The scale depends on the current token–head slice alone. The learner applies the same quantize–dequantize rule at the corresponding cache-write sites: keys after key normalization and RoPE, values at the matching V cache-write position. The learner rule matches the vLLM cache-write kernel bitwise, and its straight-through backward is the identity.

A.3 The K3 diagnostic

For an executor sample $x_t \sim r_{\mathrm{exec}}(\cdot \mid h_t)$, the token log-ratio used to quantify the residual mismatch is $\delta_t := \log q_{\mathrm{old}}(x_t \mid h_t) - \log r_{\mathrm{exec}}(x_t \mid h_t)$. We report the standard K3 estimator (Schulman, 2020; Shao et al., 2024),

$$\widehat{D}_{\mathrm{K3}} := \frac{1}{\sum_t m_t} \sum_{t:\, m_t=1} \left[\exp(\delta_t) - \delta_t - 1\right], \tag{11}$$

in which every term is nonnegative because $\exp(\delta_t) \geq 1 + \delta_t$. On samples from r_{exec},

$$\mathbb{E}_{r_{\mathrm{exec}}}[\exp(\delta_t)] = 1, \qquad -\mathbb{E}_{r_{\mathrm{exec}}}[\delta_t] = D_{\mathrm{KL}}(r_{\mathrm{exec}}(\cdot \mid h_t) \,\|\, q_{\mathrm{old}}(\cdot \mid h_t)), \tag{12}$$

so the expectation of equation 11 is $D_{\mathrm{KL}}(r_{\mathrm{exec}} \| q_{\mathrm{old}})$, the rollout–learner divergence at a shared checkpoint.

B SUPPORT-AWARE ALLOCATION

B.1 THE DISPLACEMENT BUDGET

Fix one rollout/update batch and let $\mathcal{S}$ be its action segments, with μ_s, ν_s, ESS_s, ρ_s, and C_s as in equation 6, which applies to segments with $n_s > 1$ and takes $\varepsilon_{\mathrm{mom}} = 10^{-8}$. With $W_t = 1 + \alpha_s(\overline{W}_t - 1)$ from equation 9, each segment satisfies

$$\frac{1}{n_s}\sum_{t\in s}(W_t - 1)^2 = \alpha_s^2 \frac{1}{n_s}\sum_{t\in s}\left(\overline{W}_t - 1\right)^2 = C_s\alpha_s^2. \tag{13}$$

The constrained quantity in equation 8 is therefore $\sum_s C_s\alpha_s^2 = \sum_s n_s^{-1}\sum_{t\in s}(W_t - 1)^2$, the segment-normalized squared displacement of the applied weights from the identity weight 1. Giving every segment $\alpha_s = \alpha_g$ makes this quantity $\alpha_g^2\sum_s C_s$, which is exactly the budget. Support-aware allocation and the uniform reference therefore work under the same segment-normalized displacement budget.

Weight concentration and correction magnitude. For the positive weights $\overline{W}_t$ within a segment,

$$\frac{\mathrm{ESS}_s}{n_s} = \frac{\mu_s^2}{\nu_s} = \frac{1}{1 + \mathrm{CV}_s^2}, \qquad \mathrm{CV}_s^2 := \frac{\nu_s - \mu_s^2}{\mu_s^2}. \tag{14}$$

Weight concentration therefore lowers normalized ESS and increases CV_s^2. The finite-length factor in ρ_s preserves this concentration ordering, while $\nu_s/(\nu_s + \varepsilon_{\mathrm{mom}})$ stabilizes a near-zero moment. The separate quantity C_s records correction magnitude away from identity.

B.2 EQUAL MAGNITUDE, DIFFERENT SUPPORT

Using the replay underlying Figure 4, Figure 7 shows how the two 24-token segments of §3.3 distribute their correction across tokens.

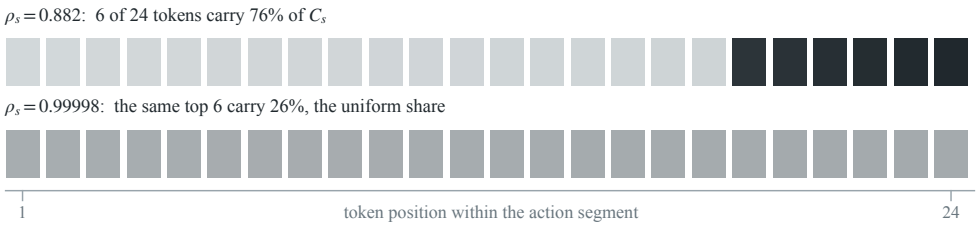


Figure 7: Two action segments with indistinguishable C_s and different ρ_s. Each block is one token, shaded by that token's share of squared displacement within its segment. The two values of C_s differ by 7×10^{-6}.

C_s therefore does not determine ρ_s. We selected the pair by searching for a near-tie in C_s.

B.3 CLOSED-FORM ALLOCATION

Proposition 1 (restated from §3.3). Let every segment with $n_s > 1$ satisfy $C_s > 0$ and $\rho_s \in (0, 1]$, and let $\alpha_g \in (0, 1)$. Consider

$$\max_{\alpha\in[0,1]^{|\mathcal{S}|}} \sum_{s\in\mathcal{S}} C_s\rho_s\alpha_s \qquad \text{subject to} \qquad \sum_{s\in\mathcal{S}} C_s\alpha_s^2 \le \alpha_g^2\sum_{s\in\mathcal{S}} C_s. \tag{15}$$

Then the optimum exists, is unique, and has the form

$$\alpha_s = \min(c\rho_s, 1), \qquad \text{where } c > 0 \text{ is the unique root of } \sum_s C_s\min(c\rho_s, 1)^2 = \alpha_g^2\sum_s C_s. \tag{16}$$

If all ρ_s are equal, then $\alpha_s = \alpha_g$.

Proof. Substitute $\beta_s := \sqrt{C_s}\,\alpha_s$, $v_s := \sqrt{C_s}\,\rho_s$, $u_s := \sqrt{C_s}$, and $R^2 := \alpha_g^2 \sum_s C_s$, which turns equation 15 into

$$\max_{\beta} \langle v, \beta \rangle \qquad \text{subject to} \qquad \|\beta\|_2 \le R, \quad 0 \le \beta_s \le u_s. \tag{17}$$

The feasible set is the intersection of a Euclidean ball with a box, hence convex and compact, so an optimum exists. Because $v_s > 0$ and $R^2 < \sum_s u_s^2$, any feasible point with $\|\beta\|_2 < R$ has a coordinate strictly below its box bound, and increasing β_s along that coordinate strictly increases the objective. Every optimum therefore satisfies $\|\beta\|_2 = R$ and exhausts the budget.

Write the KKT conditions for equation 17. The ball constraint is active, so its multiplier is positive, and the coordinatewise optimality condition gives $\beta_s = \min(c\,v_s, u_s)$ for a constant $c > 0$ fixed by that multiplier. In the original variables this condition is equation 16. Now set $g(c) := \sum_s C_s \min(c\rho_s, 1)^2$. Since every $\rho_s > 0$, the function g is continuous and strictly increasing from $g(0) = 0$ until it saturates at $\sum_s C_s$. Because $0 < R^2 = \alpha_g^2 \sum_s C_s < \sum_s C_s$, exactly one c satisfies $g(c) = R^2$, which also establishes uniqueness of the optimum. If all ρ_s equal a common $\rho > 0$, then all α_s coincide, and the budget equality gives $\alpha_s = \alpha_g$. □

B.4 Allocation over a full training run

The range of α_s within a batch never collapses: over one complete 400-step ALFWorld run, its width stays above 0.01 (Figure 8). The range shifts upward and narrows over training, from a median range of 0.634–0.915 with width 0.287 over steps 1–100 to 0.810–0.959 with width 0.151 over steps 301–400.

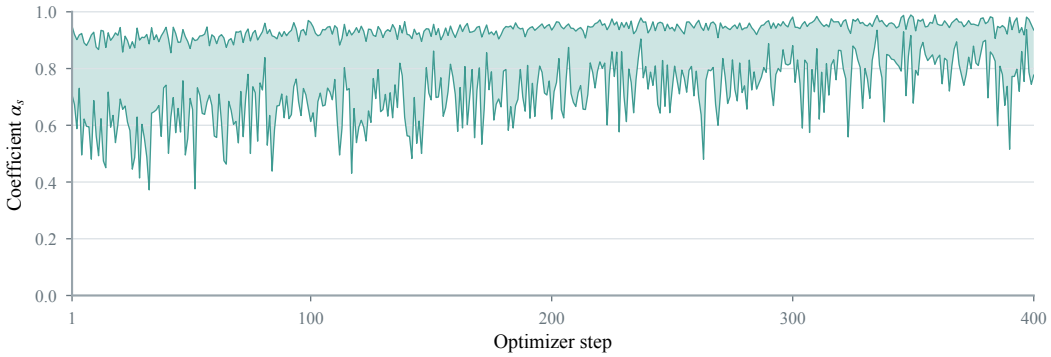


Figure 8: Range of the allocated α_s within each batch over one complete 400-step ALFWorld run: the band spans the smallest to the largest α_s in that batch. Across the 400 batches the median range is 0.717 to 0.942 with median width 0.215; over the whole run α_s reaches 0.372 and 0.989.

B.5 Premises of Proposition 1

Since equation 5 defines $\overline{W}_t$ as an exponential, $\overline{W}_t > 0$. Positivity makes $\mathrm{ESS}_s > 1$ strict for $n_s > 1$, so $\rho_s > 0$, and makes $\mu_g > 0$, so $\alpha_g > 0$. Jensen's inequality gives $\nu_s \ge \mu_s^2$ for each segment, and a second application after averaging over the batch gives $\nu_g \ge \mu_g^2$, hence $\alpha_g < 1$.

Every one of the 18,683 segments in the replay underlying Figure 4 has $C_s > 0$; $C_s = 0$ would require $\overline{W}_t = 1$ at all of a segment's tokens, and the smallest segment-mean $|\delta_t|$ there is 5.3×10^{-7}.

For $n_s = 1$ the factor $(\mathrm{ESS}_s - 1)/(n_s - 1)$ is $0/0$ and equation 6 leaves ρ_s undefined. We substitute the mean ρ_s over the segments with $n_s > 1$ in the same batch. Two of the 18,683 segments in that replay are of this kind.

C EXPERIMENTAL SETUP

C.1 ENVIRONMENTS, DATA, AND EVALUATION

Table 3 lists the four environment–backbone settings, their training horizons, AULC sets, and endpoint sets.

Table 3: Environments, backbones, training steps, and evaluation sets.

Environment	Backbone	Steps	AULC set	Endpoint
ALFWorld	Qwen3-4B	400	`valid_unseen` (10-step)	`valid_unseen` (final-six)
WebShop	Qwen3-4B	400	official test (10-step)	official test (final-six)
ScienceWorld	Qwen3-8B	400	259 monitor (10-step)	official test (step 400)
	Qwen3-30B-A3B	400	259 monitor (10-step)	official test (step 400)

All three environments use their public splits. ALFWorld trains on 3,553 games and uses the 134 `valid_unseen` games for its displayed curve, AULC, and endpoints (Shridhar et al., 2021). WebShop trains on 11,587 instructions and uses the 500 official test instructions for its displayed curve, AULC, and endpoints (Yao et al., 2022). ScienceWorld trains on the official training variations of each task and evaluates on the 1,819 official test variations (Wang et al., 2022).

Scores on the AULC sets are recorded every 10 steps. ALFWorld and WebShop endpoints are the mean of the final six checkpoint evaluations on the endpoint sets in Table 3. For periodic ScienceWorld evaluation, we use a fixed set of 259 held-out variations: the first ten by variation ID from each of the 30 task families, and all of them for the families that have fewer than ten. The full official test set of 1,819 (`task_name, variation_id`) pairs is run once, at the final training step. Both sets weight task families equally when forming the macro normalized score. For both ScienceWorld settings, each action is capped at 32 generated tokens (`max_generation_tokens`=32). The Qwen3-8B setting disables thinking (`enable_thinking=false`).

Within an environment, all arms share the training and evaluation task IDs, the initial checkpoint, the prompt order, the turn cap, and the evaluator. Training rollouts sample at temperature 1.0 with top-p 1.0 and top-k disabled; all evaluation decodes greedily. Each environment supplies its own reward: a binary success indicator for ALFWorld; for WebShop, the environment's final reward, with 0.05 subtracted per malformed tool call during training; and for ScienceWorld, the environment score clipped at zero and divided by 100, where an episode counts as solved at the full score of 100.

All turns of an episode are trained as one concatenated sequence: the prompt, each action, and each observation follow one another, and the policy-gradient loss acts only on action tokens (the mask m_t of §2.2). The response budget in Table 4 therefore covers all actions and observations of an episode together.

Table 4: Sequence budgets. All entries are token counts except the turn cap. Context is the rollout engine's window, Prompt caps the first-turn prompt, and Response caps the actions and observations appended after it; when appending the next observation would exceed this budget, the episode stops at the response boundary.

Environment	Turn cap	Context	Prompt	Response
ALFWorld	30	24,576	2,048	20,480
WebShop	15	24,576	4,096	20,480
ScienceWorld / 8B	100	40,960	1,024	39,936
ScienceWorld / 30B-A3B	100	50,176	1,024	49,152

An episode that neither succeeds nor terminates before the turn cap is reached counts as cap exhaustion. Reported cap-exhaust values and successful-episode turns are computed from the same step-400 episode records as the corresponding single-checkpoint success values. The training horizon and sequence budgets in Tables 3 and 4 are fixed by the environment and model size and shared across arms within each setting.

C.2 TRAINING HYPERPARAMETERS

Table 5 gives the common GRPO hyperparameters; §C.3 describes method-specific objective deviations.

Table 5: GRPO hyperparameters.

Hyperparameter	Value
Advantage estimator	GRPO
Optimizer	AdamW
Adam settings	$\beta = (0.9, 0.999)$; weight decay 0.01; clip norm 1.0
LR / schedule	1×10^{-6}; constant; no warmup
PPO clip	$\epsilon_{\text{PPO}} = 0.2$; dual-clip $c = 10$ (except QuRL)
KL	penalty in the loss, coefficient 0.001
Loss aggregation	`seq-mean-token-mean`
Responses per prompt	8
Train batch size	16

C.3 QUANTIZATION AND CORRECTION CONFIGURATIONS

Every quantized arm uses the same rollout stack: blockwise FP8 W8A8 linear operators and a per-token-per-head FP8 KV cache. QAT and InterQ-RL reproduce both rules in the learner forward, as specified in §A. For the ALFWorld design analysis, we additionally combine Flash-RL's TIS and AIS with this learner configuration while retaining their correction rules and reported parameters. InterQ-RL retains verl's default per-token log-ratio safety bound $B = 20$.

Baseline implementations follow each method's published objective, using reported parameter values where specified. Flash-RL applies token-local TIS to the rollout–learner ratio with bound $C = 2$ (Yao et al., 2025). AIS keeps a truncated weight, at bound $C = 5$, and gates how strongly it is applied by a per-batch coefficient that interpolates between the uncorrected and the fully weighted gradient; the coefficient combines an effective-sample-size term, a log-probability divergence saturating at $\delta = 0.02$, and a variance-amplification term with tolerance $\gamma = 1.2$ (Zhou et al., 2026).

QuRL has two components that act at different levels (Li et al., 2026b). Adaptive Clipping Range (ACR) scales the PPO upper bound by the gap between the quantized and the full-precision policy, $(1+\epsilon)/r_{i,t}$, which presumes a learner left at full precision. Our implementation follows Equation (9) of QuRL and does not apply dual-clip. Because QuRL does not specify the truncation cap C, we use $C = 2$ in all QuRL runs. Update-Aware Quantization (UAQ) rescales the base model once before training, at the published scale $s = 1.5$, so that each RL update is large enough for the rollout quantizer to capture rather than round away. UAQ needs the RMSNorm in front of a linear to absorb the rescale. In an MoE block that norm is shared with the router, and rescaling it would change routing, so on Qwen3-30B-A3B we apply UAQ to the attention group only. Table 1 evaluates this ACR+UAQ recipe, including the stated MoE adaptation. Because ACR is defined for a quantized rollout paired with a full-precision learner, this is the applicable QuRL configuration; substituting a QAT learner would change the policy pair that defines ACR.

Flash-RL uses verl's built-in rollout importance-sampling path. No public implementation of AIS or QuRL was available during our experiments; we therefore reimplemented both methods from their papers and will release our implementations with the code.

C.4 ADDITIONAL DESIGN ANALYSES

Support-aware allocation under a matched budget. Figure 9 provides the complete success and K3 trajectories for the allocation variants analyzed in §4.2, together with BF16 and QAT references; Prefix, Global, and InterQ-RL share the same bounded prefix weights. Over steps 301–400, median K3 is 0.0119 for InterQ-RL, 0.1172 for Prefix, and 0.1858 for Global. Appendix B reports the allocated α_s ranges over the full run.

Correction rules with learner alignment. As an additional ablation, the QAT+TIS and QAT+AIS rows pair Flash-RL's TIS rule and the AIS correction rule, respectively, with the QAT learner used by

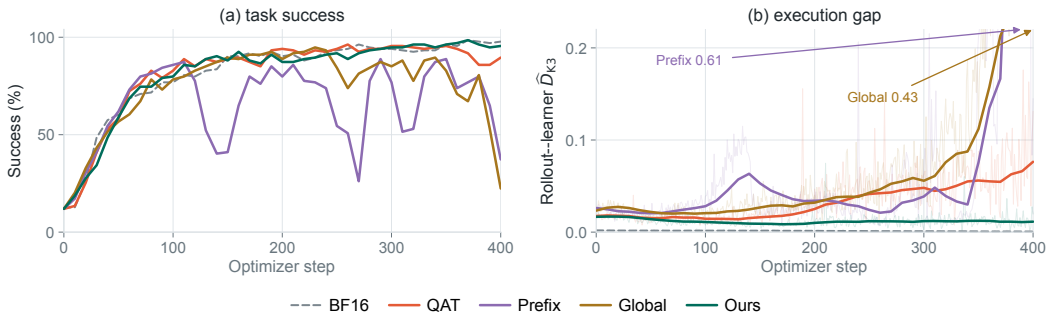


Figure 9: *(a)* ALFWorld `valid_unseen` success. *(b)* Smoothed rollout–learner K3; arrows mark off-panel peaks. All quantized arms share the W8A8-KV8+QAT substrate.

Table 6: Token-local corrections with learner alignment on ALFWorld; all quantized arms use the W8A8-KV8 rollout and W/A+KV-QAT learner.

Variant	Scope	Retention	Success ↑	AULC ↑	Turns ↓	Cap. ↓
BF16	—	—	97.14	82.07	10.98	2.24
QAT	none	none	90.42	**82.29**	12.72	10.45
QAT+TIS	current token	soft clip $C = 2$	90.55	70.59	9.86	5.22
QAT+AIS	current token	clip $C = 5$ + gate	92.66	76.27	12.46	6.72
InterQ-RL	segment prefix	$\alpha_s(\rho_s)$	**96.39**	81.96	10.06	**4.48**

Success averages the final six evaluations; Turns and Cap. use the step-400 evaluation. AULC covers steps 0–400; Turns averages successful episodes and Cap. is the cap-exhaust percentage.

InterQ-RL. With this learner configuration, InterQ-RL reaches 96.39% last-six success, 3.73 points above the strongest token-local variant and 5.97 above QAT (Table 6). Its AULC of 81.96 remains close to QAT's 82.29 and exceeds the token-local range of 70.59–76.27. Successful episodes average 10.06 turns, shorter than every variant except QAT+TIS, which reaches 9.86 turns at only 90.55% success. InterQ-RL's 4.48% cap-exhaust rate is below the token-local range of 5.22%–6.72%.

C.5 Implementation and hardware

Training runs on verl (Sheng et al., 2025) with vLLM (Kwon et al., 2023) as the rollout engine. The 4B runs use the FSDP backend; the 8B and 30B-A3B runs use the Megatron-LM backend with distributed checkpointing. All experiments run on NVIDIA H20 GPUs, with 8 GPUs per dense run and 16 for 30B-A3B.

D Limitations

Independent-seed replications of BF16, QAT, and InterQ-RL on ALFWorld and WebShop yield conclusions consistent with the reported comparisons. Compute constraints limited broader multi-seed coverage across the remaining arms and ScienceWorld settings; extending it would further strengthen the empirical assessment.