CAGPO: CRITIC-ANCHORED GROUP POLICY OPTIMIZATION FOR MULTI-TURN LLM AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Credit assignment is a central challenge in training multi-turn LLM agents: the final outcome provides little direct feedback on the contribution of individual actions. Many group-based methods obtain finer-grained credit by comparing multiple executed rollouts of the same task. Yet real interactions may not be replayable: a message already sent to a user or a completed external transaction cannot simply be undone to test another action under identical conditions. Even when replay is possible, repeated environment interaction can be costly, requiring paid API calls, time-consuming simulator runs, or repeated physical robot trials. We propose Critic-Anchored Group Policy Optimization (CAGPO), which combines temporal credit on executed trajectories with local comparisons that require no additional candidate execution. At selected interaction states, CAGPO samples alternative responses and derives group-relative advantages from their shared critic's scores. Progress-aware critic training constrains adjacent-turn value changes using real return targets. The actor learns from both real and candidate responses, while only executed trajectories supervise the critic. With Qwen2.5-7B, CAGPO achieves 99.2% success on ALFWorld and 89.8% peak success on WebShop. On ScienceWorld, it reaches 78.6% peak success, 21.6 percentage points above the reported ScalingInter result. Our implementation is available at `https://anonymous.4open.science/r/CAGPO-F620/README.md`.

1 INTRODUCTION

LLM agents solve interactive tasks by repeatedly generating responses, executing actions, and incorporating environmental observations. Unlike single-turn completion, a successful interaction may require a sequence of decisions whose consequences become apparent only much later. Reinforcement learning can optimize such behavior from task feedback, but a final success or failure does not identify the contribution of each intermediate action. A successful trajectory may contain redundant steps, whereas a failed one may contain useful decisions undermined by later mistakes (Feng et al., 2025; Cheng et al., 2026). Learning effectively from each environment interaction therefore requires more than assigning the same outcome-based signal to an entire trajectory.

Existing methods approach this problem through value estimation or comparisons among sampled outcomes. PPO with generalized advantage estimation (GAE) propagates returns through a learned critic (Schulman et al., 2017; 2016). Outcome-based GRPO instead derives relative advantages from groups of responses without an independent critic (Shao et al., 2024); applied directly to complete agent episodes, it assigns a shared outcome advantage across their decisions. Recent methods recover finer credit: GiGPO groups repeated states, HGPO accounts for historical context, and GraphGPO exploits cross-trajectory transition structure (Feng et al., 2025; He et al., 2026; Cheng et al., 2026). These advances improve how collected experience is used, but their local comparisons still depend on which comparable states and alternative transitions appear in that experience.

Obtaining more executed alternatives is not always an attractive solution. Each agent rollout entails external interactions as well as text generation. In real systems, resetting may require manual intervention or specialized mechanisms, and irreversible transitions can prevent recovery (Dulac-Arnold et al., 2019; Nguyen & Cheng, 2023; Zhang & Weihs, 2023). If comparable initial conditions cannot be restored or replicated, standard per-task multi-rollout sampling cannot be applied directly.

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.

This motivates a complementary question: *can an agent obtain useful local comparisons at a visited decision context without executing additional candidate responses?*

We propose **Critic-Anchored Group Policy Optimization (CAGPO)**. The term *critic-anchored* captures the link between real experience and unexecuted alternatives: the critic is trained on return targets from executed trajectories and then serves as a shared evaluator for candidate responses. Candidate comparisons are thus anchored to a value function grounded in observed task outcomes, without executing the candidates or using their predicted scores as critic-training targets. The same critic supports two complementary forms of credit assignment. On executed trajectories, token-level GAE provides temporal credit across response boundaries. At selected contexts, the policy generates additional responses that are scored by the critic, not the environment; group-normalized scores provide local candidate advantages. To incorporate task progress into this shared critic, we adapt AgentPRM's promise-and-progress perspective (Xi et al., 2025b) into a sign-hinge constraint on adjacent-turn value changes. Real and candidate tokens then contribute to the same actor update, while only real trajectories supervise the critic. The mechanism does not require repeated state matches or environment execution for its candidate groups, but trades extra model computation and reliance on critic estimates for these additional comparisons.

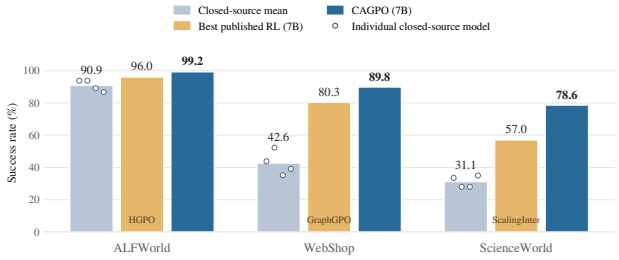

Figure 1: **CAGPO achieves strong performance across three agent benchmarks.** Closed-source bars show the mean across Gemini-3.1-Pro, Claude-Opus-4.8, Qwen3.8-Max, and GPT-5.5; dots show individual model results, not seed uncertainty. The strongest published 7B RL baselines are HGPO (He et al., 2026), GraphGPO (Cheng et al., 2026), and ScalingInter (Xi et al., 2025a), respectively; CAGPO uses Qwen2.5-7B, with reporting protocols detailed in Section 5.1.

We evaluate CAGPO on three agent benchmarks: ALFWorld, WebShop, and ScienceWorld. With Qwen2.5-7B, CAGPO achieves 99.2% success on ALFWorld and 89.8% peak success on WebShop. On ScienceWorld, it reaches 78.6% peak success, 21.6 percentage points above the reported ScalingInter result. Figure 1 highlights CAGPO's near-perfect ALFWorld success and its strong WebShop and ScienceWorld results: the reported scores exceed the strongest listed 7B RL baselines by 3.2, 9.5, and 21.6 percentage points, respectively, and are above the four-model closed-source average on every benchmark. These results highlight the promise of combining temporal credit with critic-anchored local comparisons without additional candidate execution in the environment. Our contributions are:

- A local comparison mechanism that constructs group-relative policy supervision from critic-scored, unexecuted candidates at shared contexts, complementing temporal credit on real trajectories.
- A progress-aware shared-critic design that integrates adjacent-turn directional consistency with real-return regression and reuses the critic for candidate evaluation.
- An empirical evaluation on ALFWorld, WebShop, and ScienceWorld demonstrating strong task performance and the effectiveness of critic-guided credit assignment for multi-turn agents.

2 Related Work

Policy optimization and value estimation. PPO (Schulman et al., 2017) is commonly paired with a learned value function and GAE (Schulman et al., 2016) to estimate credit along a sequence. GRPO

(Shao et al., 2024) replaces the independent critic with group statistics; its outcome-supervised variant supplies a coarse signal when a complete agent trajectory is treated as one response. This limitation concerns the source and granularity of feedback, not every possible group-relative objective. Value-based approaches remain relevant for multi-turn agents: ArCHer separates utterance-level decisions from token-level generation (Zhou et al., 2024), Turn-PPO studies turn-level value estimation (Li et al., 2026), and SAO develops single-rollout asynchronous optimization with skip-observation GAE (Hou et al., 2026). CAGPO builds on real-trajectory value learning rather than claiming a new single-rollout paradigm; its additional use of the critic is to compare unexecuted responses.

Fine-grained credit from collected trajectories. RAGEN, Search-R1, and ReSearch establish multi-turn RL settings involving environment interaction and tool use (Wang et al., 2025; Jin et al., 2025; Chen et al., 2025). GiGPO refines episode-level feedback through anchor-state groups, but requires identifiable repeated states; without repeated states, its local advantage vanishes (Feng et al., 2025). HGPO addresses the fact that matching current states may have different histories, trading context consistency against the sample size of hierarchical groups (He et al., 2026). GraphGPO uses an aggregated transition graph to recognize progress even within failed trajectories; however, its comparisons use observed outgoing transitions, and singleton groups receive zero graph advantage supplemented by episode-level credit (Cheng et al., 2026). These methods already provide fine-grained supervision without additional rollouts beyond their underlying group collection. CAGPO addresses a different constraint: it generates alternatives at selected visited prefixes rather than waiting for comparable executed alternatives to occur.

Progress-aware reward and value modeling. AgentPRM (Xi et al., 2025b) distinguishes future success potential from local progress and combines value regression with adjacent-step difference matching. This directly inspires our critic-progress objective. We use a directional sign-hinge variant and integrate it into the critic shared by real-trajectory estimation and candidate scoring. Thus our contribution is an adaptation and use of progress-aware training, not the first introduction of progress supervision. This also differs from GraphGPO's progress feedback derived from distances in an empirical graph.

Environment interaction constraints. Real-world RL must account for limited interaction and operational constraints (Dulac-Arnold et al., 2019). Reset-free RL makes explicit that restoring initial states may require human intervention or engineered recovery, while irreversible transitions can halt continued learning (Nguyen & Cheng, 2023; Zhang & Weihs, 2023). Consequently, per-task groups of executed episodes may be costly or infeasible when comparable initial conditions cannot be reproduced, including through independent copies. CAGPO removes the need to execute its additional candidates, not the need for real training feedback. It is therefore complementary to, rather than a solution for, safe exploration and recovery in non-resettable systems.

3 PRELIMINARIES

Multi-turn interaction. We use the notation of Figure 2. At turn $n \in \{0, \ldots, N-1\}$, the agent receives an input context s_n containing the task and available interaction history, and generates a response $R_n \sim \pi_\theta(\cdot \mid s_n)$. Executing its action produces an observation o_{n+1} and task reward; the next context incorporates this feedback. Thus an episode follows $s_0, R_0, o_1, R_1, \ldots, R_{N-1}, o_N$. The context is the policy's information state, not necessarily a fully observed physical state. Uppercase R_n denotes a response, not a reward. Rewards may be terminal outcomes or intermediate feedback, depending on the task.

Let a_t, $t = 0, \ldots, T-1$, index only policy-generated action tokens across the episode. Observation tokens remain in the conditioning context but do not occupy this action-token index. The reward r_t equals the environment reward for a turn at its final action token and is zero at other token positions. The critic V_ϕ supplies token values v_t. We reserve $V_n = V_\phi(s_n \oplus R_n)$ for its output at the final token of response R_n, where $\oplus$ denotes token concatenation. The rollout policy is $\pi_{\text{old}} = \pi_{\theta_{\text{old}}}$ and remains fixed during a collection/update iteration. The task objective is to maximize expected accumulated environment reward, $J(\theta) = \mathbb{E}_{\tau \sim \pi_\theta}[\sum_{t=0}^{T-1} r_t]$; the discount γ below specifies the credit estimator. Since each turn reward appears at one token position, this sum does not count observations or duplicate a turn's reward across its response.

Clipped policy optimization. For the context h_t preceding action token a_t, define the probability ratio and negative PPO surrogate as

$$\rho_t = \frac{\pi_\theta(a_t \mid h_t)}{\pi_{\text{old}}(a_t \mid h_t)}, \tag{1}$$

$$\ell_{\text{PPO}}(\rho_t, A_t) = -\min\left(\rho_t A_t, \text{clip}(\rho_t, 1-\epsilon_{\text{clip}}, 1+\epsilon_{\text{clip}})A_t\right). \tag{2}$$

The masked token mean of this loss is denoted L_{PPO}. The clipping threshold ϵ_{clip} is distinct from the normalization stabilizer ϵ. GAE uses temporal differences from V_ϕ to estimate A_t (Schulman et al., 2016).

Outcome-based group comparison. For comparison, outcome-based GRPO samples K executed trajectories for the same task and standardizes their outcome rewards within the group (Shao et al., 2024). Every action token in one trajectory receives that trajectory's scalar advantage. We use K only for executed-trajectory groups and reserve G for CAGPO's unexecuted candidate responses; the two group sizes do not denote the same interaction budget.

4 Critic-Anchored Group Policy Optimization

CAGPO combines temporal credit from executed behavior with model-based comparisons at selected decision contexts (Figure 2). The real batch B_{real} contains executed trajectories and their rewards. Candidate rows $B_{\text{candidate}}$ provide additional actor supervision, but never critic targets. Candidate construction requires only one executed trajectory to provide a prefix; a training batch may still contain multiple real trajectories, including repeated tasks.

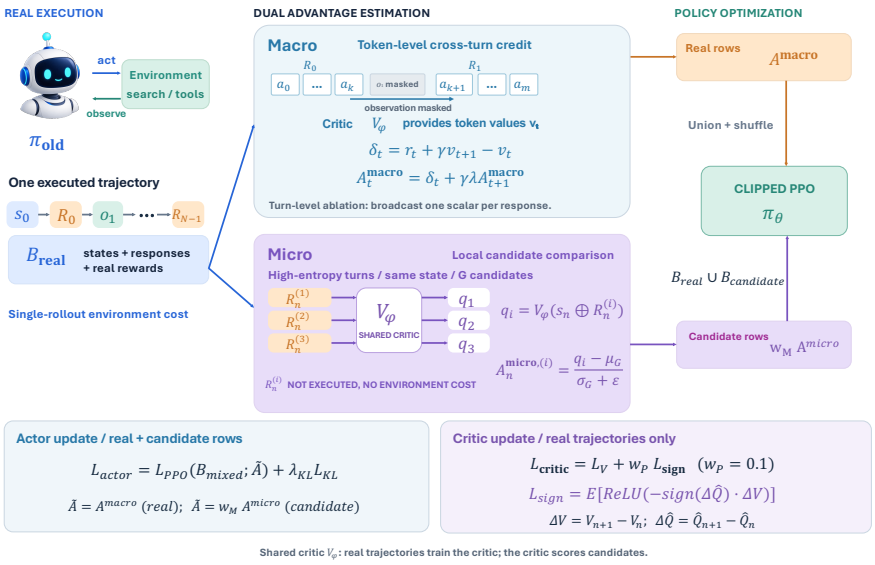

Figure 2: **CAGPO overview.** The macro branch assigns temporal credit to executed trajectories using token-level GAE, with turn-level estimation also supported. The micro branch scores unexecuted candidates with the shared critic to obtain group-relative advantages. Real and candidate rows train the actor, while only real returns and progress consistency train the critic; update order is specified in Section 4.4.

4.1 Temporal Credit from Executed Trajectories

The macro branch propagates task feedback along the real action-token sequence. With terminal bootstrap $v_T = A_T^{\text{macro}} = 0$, it computes

$$\delta_t = r_t + \gamma v_{t+1} - v_t, \qquad A_t^{\text{macro}} = \delta_t + \gamma\lambda A_{t+1}^{\text{macro}}. \tag{3}$$

At a turn boundary, the successor is the next action token, skipping the intervening observation tokens as in SAO (Hou et al., 2026). Thus rewards propagate across turns without treating environment text as policy actions. Before normalization, the return target is $\hat{G}_t = A_t^{\text{macro}} + v_t$. We whiten macro advantages over valid action tokens for the actor while retaining the unnormalized return targets for the critic. In the mixed actor objective, A^{macro} denotes these whitened advantages, matching the real-row label in the figure. Alternatively, turn-level estimation assigns one advantage to every action token of a response. The discount and GAE settings for both granularities are specified in Section 5.1.

4.2 Local Comparison without Candidate Execution

At a selected prefix s_n, sample $R_n^{(i)} \sim \pi_{\text{old}}(\cdot \mid s_n)$ for $i = 1, \ldots, G$. These are candidate responses, not additional executed trajectories. Reading the critic at each candidate's final token gives

$$q_i = V_\phi(s_n \oplus R_n^{(i)}), \qquad A_n^{\text{micro},(i)} = \frac{q_i - \mu_G}{\sigma_G + \epsilon}, \tag{4}$$

where $\mu_G = G^{-1}\sum_i q_i$ and $\sigma_G^2 = G^{-1}\sum_i (q_i - \mu_G)^2$. The symbols q_i, μ_G, σ_G are local to the selected turn, as in Figure 2. A zero-spread group contributes zero advantage. Each scalar advantage is broadcast to the generated tokens of its candidate, excluding the shared prefix from the policy loss.

This construction supplies comparisons even when no other executed trajectory revisits s_n. Because all candidates share the same prefix, they are evaluated under the same available context. Group centering removes a common score offset, and standardization reduces sensitivity to positive score rescaling, up to ϵ. Neither operation corrects ranking errors: q_i is a learned estimate, not an observed candidate return. We prioritize turns with high mean response entropy for candidate construction; the selection budget and group size are specified in Section 5.1. Additional implementation details are in Appendix A.

4.3 Progress-Aware Critic Training

The critic's role in both branches motivates supervision of relative changes as well as absolute returns. Inspired by AgentPRM's promise-and-progress formulation (Xi et al., 2025b), we constrain adjacent-turn value differences on real trajectories. Let $\hat{Q}_n$ be the detached return target $\hat{G}_t$ at the final token of executed response R_n. It is not the candidate prediction q_i. Define

$$\Delta V = V_{n+1} - V_n, \qquad \Delta\hat{Q} = \hat{Q}_{n+1} - \hat{Q}_n.$$

AgentPRM motivates matching predicted and target differences. We instead use the directional variant shown in the framework:

$$L_{\text{sign}} = \mathbb{E}_{\text{adjacent real turns}}\left[\text{ReLU}\left(-\text{sign}(\Delta\hat{Q})\,\Delta V\right)\right], \qquad L_{\text{critic}} = L_V + w_P L_{\text{sign}}. \tag{5}$$

Here L_V is the base PPO value-regression loss on real return targets; its implementation retains value clipping, and w_P controls the strength of progress regularization. The auxiliary loss penalizes opposite-signed changes without matching their exact magnitude. It does not require values to increase monotonically: negative target differences allow decreases, and equal targets produce no penalty. In particular, terminal-only returns with $\gamma = \lambda = 1$ can yield equal targets throughout a trajectory, so the term need not be active on every training example.

The design integrates progress supervision into the critic rather than adding a separate actor reward. Its intended role is to guide value changes using real outcomes; it does not establish that unexecuted candidates are ranked correctly. The difference-matching alternative and training layout are described in Appendix A.

4.4 Joint Policy Optimization

Real and candidate rows are concatenated and shuffled to form $B_{\text{mixed}} = B_{\text{real}} \cup B_{\text{candidate}}$. The union denotes batch concatenation, not merging responses at one token position. For valid action tokens, the actor uses

$$\widetilde{A} = \begin{cases} A^{\text{macro}}, & \text{real rows,} \\ w_M A^{\text{micro}}, & \text{candidate rows,} \end{cases} \qquad L_{\text{actor}} = L_{\text{PPO}}(B_{\text{mixed}}; \widetilde{A}) + \lambda_{\text{KL}} L_{\text{KL}}. \tag{6}$$

The KL term regularizes against the reference policy with weight λ_{KL}. The weight w_M scales candidate advantages before clipping, rather than weighting a separately normalized candidate loss. The implementation averages over valid tokens within each mixed micro-batch and accumulates gradients across micro-batches and devices. Observations, padding, and candidate-prefix tokens are masked out.

Training procedure. Each iteration first collects real trajectories with π_{old} and computes macro advantages and return targets with the current critic. The critic is then updated on B_{real} using its own optimizer. Selected prefixes generate candidate responses, which the updated critic scores using Equation 4. Finally, the actor is updated on B_{mixed} using Equation 6. Macro estimation and candidate scoring therefore share a network but occur before and after its critic update, respectively. Targets and candidate advantages are detached: actor gradients do not pass through the critic. Additional candidate computation introduces no additional environment execution.

5 Experiments

5.1 Setup

We evaluate CAGPO along three dimensions: benchmark performance across multi-turn tasks, the contribution of its credit-assignment components, and training dynamics and computational cost.

Benchmarks. We use three public agent benchmarks spanning household interaction, online shopping, and scientific experimentation. **ALFWorld** (Shridhar et al., 2021) contains six household task types; we evaluate success on 128 in-distribution games with a 50-turn limit, following the GiGPO protocol. **WebShop** (Yao et al., 2022) requires free-text search and navigation to purchase an item matching a user instruction. We evaluate 128 test instructions with a 15-turn limit and report both normalized purchase-match score and success, defined as a perfect match. **ScienceWorld** (Wang et al., 2022) tests multi-step laboratory procedures. We use the 200 released AgentGym-RL test tasks (Xi et al., 2025a) and report strict success, requiring a task score of 100 rather than partial completion. Task-family mappings and checkpoint-specific evaluation horizons are documented in Table 2; environment details are provided in Appendix B.

Compared methods. We compare against three groups of baselines. First, proprietary prompting agents include Gemini-3.1-Pro, Claude-Opus-4.8, Qwen3.8-Max, and GPT-5.5, evaluated by us on the same environments and held-out tasks. Second, open-source prompting baselines include Qwen2.5, ReAct (Yao et al., 2023), and Reflexion (Shinn et al., 2023) on ALFWorld and WebShop, together with the open-source models reported by AgentGym-RL on ScienceWorld. Third, RL baselines include PPO, RLOO (Ahmadian et al., 2024), GRPO, GiGPO, HCAPO, HGPO, and GraphGPO on ALFWorld and WebShop (Feng et al., 2025; Tan et al., 2026; He et al., 2026; Cheng et al., 2026), and AgentGym-RL (GRPO) and ScalingInter on ScienceWorld (Xi et al., 2025a).

Implementation details. We train Qwen2.5-1.5B-Instruct and Qwen2.5-7B-Instruct with a target budget of 300 steps on $8\times$A100-80GB. ALFWorld and WebShop use token-level macro, candidate supervision, and the progress loss; ScienceWorld uses token-level macro and candidate supervision without progress regularization. Full hyperparameters and sequence budgets are listed in Appendix C.

Validation runs every 10 steps (ScienceWorld: every 25) on the held-out sets described above, also used for our closed-source evaluations. Our runs are single-seed; validation peaks and checkpoint evaluations are identified in the result descriptions. Checkpoint selection on these sets introduces selection bias. Published baselines retain their source protocols, including multi-seed reporting, so cross-paper comparisons are descriptive rather than controlled reruns.

5.2 Main Results

Tables 1 and 2 report benchmark performance, and Figure 1 summarizes the 7B results. We discuss each benchmark below; component comparisons are presented separately in Section 5.3.

Table 1: **ALFWorld and WebShop performance:** success rates and WebShop scores on a 0–100 scale, grouped by backbone. – denotes unreported task-level results.

Type	Method	ALFWorld							WebShop	
		Pick	Look	Clean	Heat	Cool	Pick2	**All**	Score	Succ.
Closed-source										
Prompt	Gemini-3.1-Pro	100.0	85.7	85.7	94.4	92.3	100.0	93.8	52.5	43.8
Prompt	Claude-Opus-4.8	92.9	92.9	100.0	94.4	88.5	95.2	93.8	60.6	52.3
Prompt	Qwen3.8-Max	89.3	92.9	85.7	88.9	92.3	85.7	89.1	39.4	35.2
Prompt	GPT-5.5	89.3	92.9	76.2	94.4	92.3	76.2	86.7	47.8	39.1
Qwen2.5-1.5B-Instruct										
Prompt	Qwen2.5	5.9	5.5	3.3	9.7	4.2	0.0	4.1	23.1	5.2
Prompt	ReAct	17.4	20.5	15.7	6.2	7.7	2.0	12.8	40.1	11.3
Prompt	Reflexion	35.3	22.2	21.7	13.6	19.4	3.7	21.8	55.8	21.9
RL	PPO (with critic)	64.8	40.5	57.1	60.6	46.4	47.4	54.4	73.8	51.5
RL	RLOO	88.3	52.8	71.0	62.8	66.4	56.9	69.7	73.9	52.1
RL	GRPO	85.3	53.7	84.5	78.2	59.7	53.5	72.8	75.8	56.8
RL	GiGPO	94.4	67.5	94.8	94.4	79.8	76.4	86.7	83.1	65.0
RL	HCAPO	88.6	75.0	97.6	90.7	84.2	74.2	87.0	83.8	68.5
RL	HGPO	–	–	–	–	–	–	94.9	90.6	78.1
RL	GraphGPO	95.2	85.7	100.0	96.3	85.3	93.7	92.7	89.3	78.7
RL	**CAGPO (ours)**	**100.0**	92.9	**100.0**	**100.0**	**100.0**	**100.0**	**99.2**	**91.4**	**82.8**
Qwen2.5-7B-Instruct										
Prompt	Qwen2.5	33.4	21.6	19.3	6.9	2.8	3.2	14.8	26.4	7.8
Prompt	ReAct	48.5	35.4	34.3	13.2	18.2	17.6	31.2	46.2	19.5
Prompt	Reflexion	62.0	41.6	44.9	30.9	36.3	23.8	42.7	58.1	28.8
RL	PPO (with critic)	92.3	64.0	92.5	89.5	80.3	68.8	80.4	81.4	68.7
RL	RLOO	87.6	78.2	87.3	81.3	71.9	48.9	75.5	80.3	65.7
RL	GRPO	90.8	66.1	89.3	74.7	72.5	64.7	77.6	79.3	66.1
RL	GiGPO	97.7	82.7	98.8	83.7	89.3	79.2	90.8	84.4	72.8
RL	HCAPO	99.1	90.3	97.3	81.8	90.8	81.9	91.4	85.1	73.8
RL	HGPO	–	–	–	–	–	–	96.0	88.5	79.3
RL	GraphGPO	100.0	92.9	100.0	94.4	91.4	92.1	95.3	86.9	80.3
RL	**CAGPO (ours)**	**100.0**	**100.0**	**100.0**	**100.0**	95.7	**100.0**	**99.2**	**91.4**	**89.8**

ALFWorld (Table 1). CAGPO achieves 99.2% success at both model scales, completing 127 of 128 tasks. At 7B, this exceeds the reported HGPO and GraphGPO results by 3.2 and 3.9 points; at 1.5B, the corresponding gains are 4.3 and 6.5 points. Strong performance at both scales shows that the method is effective even with a smaller backbone. The 7B model also achieves 100.0% on Pick2, which requires completing a compositional instruction.

WebShop (Table 1). CAGPO reaches peak success rates of 89.8% at 7B and 82.8% at 1.5B, exceeding the reported GraphGPO results by 9.5 and 4.1 points. The 7B score of 91.4 comes from a different checkpoint and is not the score at the 89.8% success peak. WebShop's open-ended search and navigation choices make it a useful setting for testing local candidate supervision; the component comparisons below examine its contribution.

ScienceWorld (Table 2). CAGPO reaches 78.6% peak success, compared with the reported 50.5% for AgentGym-RL (GRPO) and 57.0% for ScalingInter. The task-family breakdown comes from an extended checkpoint with 80.5% overall success: Find and Lifespan reach 100.0%, while Chem-Mix remains unsolved. Under the reference 20-turn limit, that checkpoint scores 62.0%, so the cross-paper gap also reflects evaluation-horizon differences. These experiments use macro and micro supervision without progress loss.

5.3 Ablations

Ablation design. We examine candidate supervision, candidate score sources, and critic-progress regularization. The micro-off variant uses only executed trajectories, while zero-score and random-score controls retain candidate rows but replace their scoring signal.

Table 2: **ScienceWorld strict success (%)** on 200 test tasks, with open-source and RL baselines from AgentGym-RL (Xi et al., 2025a) and Qwen2.5-7B-Instruct for all RL methods. Task-family mappings are detailed in Appendix B.

Model	Measure	Test-Cond.	Find	Chem-Mix	Lifespan	**Overall**
Closed-source						
Gemini-3.1-Pro	52.83	15.58	15.91	60.00	87.50	33.50
Claude-Opus-4.8	30.19	18.18	15.91	60.00	81.25	28.00
GPT-5.5	60.38	16.88	11.36	60.00	87.50	35.00
Qwen3.8-Max	50.94	5.19	20.45	20.00	75.00	28.00
Open-source ≥100B						
Qwen3-235B-A22B	11.32	4.82	59.09	20.00	66.67	23.50
DeepSeek-V3-0324	0.00	0.00	2.27	0.00	0.00	0.50
DeepSeek-R1-0528	1.89	0.00	11.36	0.00	20.00	4.50
Open-source <100B						
Qwen2.5-7B-Instruct	1.89	0.00	0.00	0.00	13.33	1.50
Qwen3-8B	9.43	0.00	18.18	0.00	46.67	10.00
Qwen3-32B	5.66	1.20	31.82	0.00	66.67	14.00
Llama-3.1-70B-Instruct	24.53	4.82	40.91	40.00	86.67	25.00
RL (Qwen2.5-7B)						
AgentGym-RL-7B (GRPO)	24.53	59.04	65.91	0.00	66.67	50.50
ScalingInter-7B	33.96	55.42	88.64	0.00	73.33	57.00
CAGPO (ours)	**88.7**	**66.3**	**100.0**	0.0	**100.0**	**78.6**

Table 3: Micro layer on/off: peak validation success (%) within 300 steps, same macro estimator and identical environment interaction.

Environment	Macro	Micro on	Micro off	Δ
WebShop 7B	token	**89.8**	78.9	**+10.9**
WebShop 7B	turn	**85.9**	70.3	**+15.6**
ScienceWorld 7B	turn	**79.4**	72.0	+7.4
ScienceWorld 7B	token	78.6	80.0	−1.4
ALFWorld 7B	token	**100.0**	97.7	+2.3

Unless stated otherwise, single-factor ablations use WebShop 7B as the base cell (token-level macro, progress loss off, micro on with G=4, top-15% entropy selection, w_M=0.3, critic scoring; 85.2% at step 300, 89.1% peak), with a 300-step budget. We report the last point and the peak, with checkpoint results identified in the table captions. Token-versus-turn comparisons also change γ (1 versus 0.95), and therefore compare two estimator configurations rather than isolating granularity alone.

Candidate supervision and training dynamics. Table 3 shows that candidate supervision improves WebShop peak success by 10.9 and 15.6 points under token- and turn-level macro, respectively. The turn-level ScienceWorld gain is 7.4 points, while the token-level peak is 1.4 points lower. Learning curves in Figure 5 further show faster convergence and reduced collapse in the tested settings.

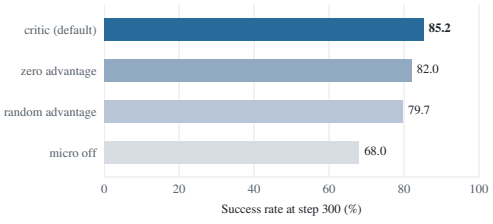

Figure 3: **Candidate score sources** on WebShop 7B at step 300. Learned scores outperform zero and random scores by 3.2 and 5.5 points.

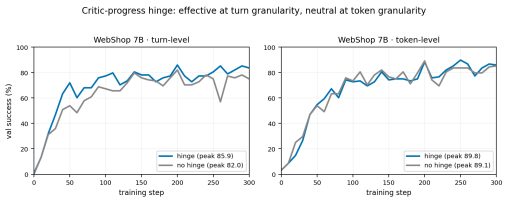

Figure 4: **Critic progress loss** on WebShop 7B. The turn-level peak rises from 82.0 to 85.9, with a smaller token-level gain.

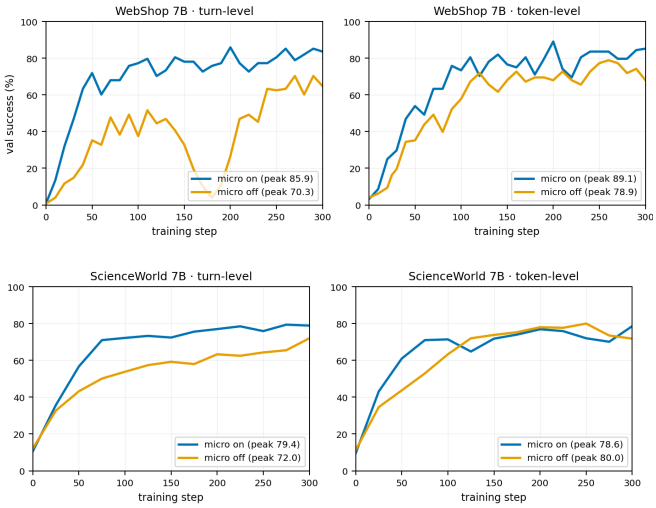

Figure 5: **Micro layer on/off** at matched environment interaction (128 episodes per step, 300 steps), for the same macro estimator. On WebShop the micro layer lifts the whole curve and prevents the turn-level collapse between steps 110 and 190. On ScienceWorld it doubles the early learning speed.

Candidate score sources. Keeping candidate rows fixed, critic scoring reaches 85.2% at step 300, compared with 82.0% for zero scores and 79.7% for random scores (Figure 3, Appendix D). The gains support learned critic feedback over merely adding candidate rows or arbitrary signed updates, without establishing perfect ranking.

Table 4: Critic progress loss (sign-hinge, w_P=0.1): peak validation success (%) within 300 steps.

Environment	Macro	Progress loss on	Off	Δ
WebShop 7B	turn	**85.9**	82.0	+3.9
WebShop 1.5B	turn	**82.8**	82.0	+0.8
WebShop 7B	token	**89.8**	89.1	+0.7
WebShop 1.5B	token	**82.8**	81.2	+1.6
ALFWorld 7B	token	100.0	100.0	0.0
ALFWorld 1.5B	token	98.4	99.2	stabilizes (none drifts to 84.4)

Critic-progress loss. Progress regularization improves turn-level WebShop 7B success by 3.9 points, with smaller token-level gains and no change on saturated ALFWorld 7B (Figure 4, Table 4). On ALFWorld 1.5B, its peak is lower but it avoids the observed decline to 84.4%. The benefit is configuration-dependent: equal-target pairs yield no progress penalty, and these results do not establish better candidate ranking.

Additional score/success comparisons, an offline scoring probe, hyper-parameter sensitivity, and learning curves are provided in Appendix D; computational costs are discussed in Appendix E.

6 Conclusion

CAGPO combines temporal credit on executed trajectories with local comparisons of critic-scored, unexecuted responses. Its progress-aware shared critic learns adjacent-turn directional consistency from real return targets. Experiments show gains from candidate supervision at fixed real interaction budgets, with score-source controls favoring learned critic feedback. CAGPO thus provides additional policy supervision from visited contexts without additional candidate execution.

AI Use Statement. We used generative AI for language polishing. The authors take full responsibility for the manuscript.

Reproducibility Statement. Anonymized code is available at `https://anonymous.4open.science/r/CAGPO-F620/README.md`. All benchmark datasets are public; evaluation protocols are described in Section 5.

REFERENCES

Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. Back to basics: Revisiting reinforce-style optimization for learning from human feedback in llms. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12248–12267, 2024. URL https://aclanthology.org/2024.acl-long.662/.

Mingyang Chen, Linzhuang Sun, Tianpeng Li, Haoze Sun, Yijie Zhou, Chenzheng Zhu, Haofen Wang, Jeff Z. Pan, Wen Zhang, Huajun Chen, Fan Yang, Zenan Zhou, and Weipeng Chen. ReSearch: Learning to reason with search for LLMs via reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/7b3a0d3a0864b66f229c0a9c84f9e950-Abstract-Conference.html.

Xin Cheng, Shuo He, Lang Feng, HaiYang Xu, Ming Yan, Lei Feng, and Bo An. Beyond trajectory-level attribution: Graph-based credit assignment for agentic reinforcement learning. *arXiv preprint arXiv:2605.26684*, 2026.

Gabriel Dulac-Arnold, Daniel Mankowitz, and Todd Hester. Challenges of real-world reinforcement learning. *arXiv preprint arXiv:1904.12901*, 2019.

Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. Group-in-group policy optimization for llm agent training. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Shuo He, Lang Feng, Qi Wei, Xin Cheng, Lei Feng, and Bo An. HGPO: Hierarchy-of-groups policy optimization for long-horizon agentic tasks. In *International Conference on Learning Representations (ICLR)*, 2026. OpenReview id T8Dev99qnz.

Zhenyu Hou, Yujiang Li, Jie Tang, and Yuxiao Dong. Single-rollout asynchronous optimization for agentic reinforcement learning. *arXiv preprint arXiv:2607.07508*, 2026.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. Search-R1: Training LLMs to reason and leverage search engines with reinforcement learning. *arXiv preprint arXiv:2503.09516*, 2025.

Junbo Li, Peng Zhou, Rui Meng, Meet P. Vadera, Lihong Li, and Yang Li. Turn-PPO: Turn-level advantage estimation with PPO for improved multi-turn RL in agentic LLMs. In *Findings of the Association for Computational Linguistics: EACL 2026*, pp. 6227–6243, 2026. URL https://aclanthology.org/2026.findings-eacl.328/.

Hoai-An Nguyen and Ching-An Cheng. Provable reset-free reinforcement learning by no-regret reduction. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pp. 25939–25955, 2023.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. In *International Conference on Learning Representations*, 2016.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

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. *arXiv preprint arXiv:2402.03300*, 2024.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/1b44b878bb782e6954cd888628510e90-Abstract-Conference.html.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. Alfworld: Aligning text and embodied environments for interactive learning. In *International Conference on Learning Representations*, 2021.

Hui-Ze Tan, Xiao-Wen Yang, Hao Chen, Jie-Jing Shao, Yi Wen, Yuteng Shen, Weihong Luo, Xiku Du, Lan-Zhe Guo, and Yu-Feng Li. Hindsight credit assignment for long-horizon llm agents. *arXiv preprint arXiv:2603.08754*, 2026.

Ruoyao Wang, Peter Jansen, Marc-Alexandre Côté, and Prithviraj Ammanabrolu. Scienceworld: Is your agent smarter than a 5th grader? In *Proceedings of EMNLP*, 2022.

Zihan Wang, Kangrui Wang, Qineng Wang, Pingyue Zhang, Linjie Li, Zhengyuan Yang, Xing Jin, Kefan Yu, Minh Nhat Nguyen, Licheng Liu, Eli Gottlieb, Yiping Lu, Kyunghyun Cho, Jiajun Wu, Li Fei-Fei, Lijuan Wang, Yejin Choi, and Manling Li. RAGEN: Understanding self-evolution in LLM agents via multi-turn reinforcement learning. *arXiv preprint arXiv:2504.20073*, 2025.

Zhiheng Xi, Jixuan Huang, Chenyang Liao, Baodai Huang, Honglin Guo, Jiaqi Liu, 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: Training llm agents for long-horizon decision making through multi-turn reinforcement learning. *arXiv preprint arXiv:2509.08755*, 2025a.

Zhiheng Xi, Chenyang Liao, Guanyu Li, Yajie Yang, Wenxiang Chen, Zhihao Zhang, Binghai Wang, Senjie Jin, Yuhao Zhou, Jian Guan, Wei Wu, Tao Ji, Tao Gui, Qi Zhang, and Xuanjing Huang. AgentPRM: Process reward models for LLM agents via step-wise promise and progress. *arXiv preprint arXiv:2511.08325*, 2025b. URL https://arxiv.org/abs/2511.08325.

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, 2022.

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

Zichen Zhang and Luca Weihs. When learning is out of reach, reset: Generalization in autonomous visuomotor reinforcement learning. *arXiv preprint arXiv:2303.17600*, 2023.

Yifei Zhou, Andrea Zanette, Jiayi Pan, Sergey Levine, and Aviral Kumar. ArCHer: Training language model agents via hierarchical multi-turn RL. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pp. 62178–62209, 2024. URL https://proceedings.mlr.press/v235/zhou24t.html.

A Notation and Implementation Details

Table 5: Notation used in Figure 2 and the method.

Symbol	Meaning
s_n, R_n, o_{n+1}	Input prefix, executed response, and following observation at turn n
a_t, r_t, v_t	Action token, aligned reward, and critic output at action-token index t
V_ϕ, V_n	Shared critic and its output at the last token of executed response R_n
$R_n^{(i)}, q_i$	Unexecuted candidate i at prefix s_n and its predicted score
G, μ_G, σ_G	Candidate count, score mean, and population standard deviation within a turn
$A_t^{\text{macro}}, A_n^{\text{micro},(i)}$	Real-trajectory and candidate advantages
$\hat{Q}_n, \Delta\hat{Q}, \Delta V$	Real turn-end return target, target difference, and predicted value difference
$B_{\text{real}}, B_{\text{candidate}}, B_{\text{mixed}}$	Real rows, candidate rows, and their concatenated actor batch
$w_M, w_P, \lambda_{\text{KL}}$	Candidate-advantage weight, progress-loss weight, and actor KL weight

Packed trajectories. Each executed trajectory occupies one training row. The chat template is applied incrementally so that each turn's prompt is the token-exact prefix of the next. A loss mask excludes observations and padding, while turn boundaries retain the alignment of rewards and final-token values. Candidates use the corresponding real prefix as their generation prompt and are re-laid as separate rows. Only their newly generated tokens receive the micro advantage. Real and candidate rows are shuffled before the data-parallel split; only the real batch is passed to the critic optimizer.

Selection and estimator configurations. Turn selection ranks responses by mean entropy and applies a per-step cap. The turn-level macro variant reads V_n at response ends and broadcasts one GAE advantage per response. Selection budgets, discount factors, and learning rates are specified in Section 5.1.

Progress-loss alternative. The implementation also supports GAE-target difference matching, $\mathbb{E}[(\Delta V - \Delta\hat{Q})^2]$, following AgentPRM's design (Xi et al., 2025b). The adopted sign-hinge constrains direction instead. Candidate scores and real return targets are detached. Progress pairs are consecutive turns of the same real trajectory and never involve synthetic candidate rows.

B Additional Benchmark Details

ALFWorld. The benchmark contains 3,827 task instances across Pick & Place, Examine in Light, Clean & Place, Heat & Place, Cool & Place, and Pick Two & Place. The agent uses templated commands and observes its location, visible objects, and admissible actions. Our reward is $+10$ on completion, with a -0.1 penalty per invalid action.

WebShop. The environment is built on 1.18M Amazon products and 12,087 human instructions. Actions take the form `search[query]` or `click[element]`, and rendered observations are truncated to 3,000 characters. Purchase reward measures attribute, option, and price match in $[0, 1]$; table scores are expressed on a 0–100 scale.

ScienceWorld. The environment contains 10 connected rooms, over 200 objects, and about 25 action templates. Tasks include measuring melting points, testing conductivity, finding organisms, mixing chemicals, and comparing lifespans. The AgentGym-RL reference groups the test tasks into Measure, Test-Conductivity, Find, Chem-Mix, and Lifespan, with counts 53/83/44/5/15. Our task-name mapping gives 53/77/44/5/16, with seven additional tasks contributing only to the overall result.

C Full Training Configuration

We use a target budget of 300 training steps, with some ALFWorld runs stopped early.

Backbones are Qwen2.5-1.5B-Instruct and Qwen2.5-7B-Instruct. Common settings are an actor learning rate of 10^{-6}, a critic learning rate of 10^{-5}, KL coefficient $\lambda_{\mathrm{KL}} = 0.01$, no entropy bonus, and rollout temperature 1.0, on 8×A100-80GB with vLLM rollout and FSDP actor and critic. The main configuration is token-level macro (γ=λ=1) with the micro layer (G=4, top-15% mean-response-entropy selection, w_M=0.3) and the critic progress loss (sign-hinge, w_P=0.1) on ALFWorld and WebShop. The turn-level variant uses $\gamma = 0.95$ and $\lambda = 1$, broadcasting each turn's advantage to its action tokens. The reported ScienceWorld token-level experiments use macro and micro without the progress loss, testing the two-level estimator in another domain. Candidate selection is capped at 128 turns per step for 7B and 256 for 1.5B.

Packed rows use a 2,048/20,480-token prompt/response budget on ALFWorld (22,528-token context) and 4,096/16,384 on WebShop (20,480-token context). ScienceWorld uses 30 turns during training; the AgentGym-RL reference trains with 20.

D Additional Ablations and Diagnostics

Candidate scoring. Figure 3 summarizes the candidate score-source controls discussed in the main text.

Candidate supervision: score and success. Table 6 further compares WebShop score and success for the micro on/off variants. Without the progress loss, enabling micro raises success from 75.0 to 82.0 at token granularity and from 65.6 to 70.3 at turn granularity. Score improves at the same time (85.8 to 90.6 and 80.5 to 82.6), so the gain is visible in both exact task completion and partial matching quality.

Table 6: WebShop 7B component comparison (128 episodes, one pass per configuration). Score and success are from the same evaluation; these are not training-validation peaks.

Macro	Progress loss	Micro	Score	Success (%)
Token	Off	Off	85.8	75.0
Token	Off	On	90.6	82.0
Turn	Off	Off	80.5	65.6
Turn	Off	On	82.6	70.3
Turn	On	On	91.3	82.0

Offline candidate-score analysis. To understand the scoring result, we ran an offline probe on WebShop 7B checkpoints (steps 150 and 300). At 419–454 real states we sampled G=4 candidates, scored them with the critic, executed every candidate from a replay of the state, and completed each episode with the policy M=2 or 4 times to estimate continuation returns. The within-group Spearman correlation between critic score and realized return is close to zero (0.035, −0.020, −0.073), and in 65–72% of groups all four candidates map to the same environment action (different wordings of one click or search).

The online controls and offline correlations answer different questions. The controls show a training-performance difference when the score source is changed; they do not establish reliable ranking of alternative actions. Within-group centering removes group-level offsets, and division by standard deviation removes positive scale differences up to the numerical stabilizer. Absolute value differences across states therefore cannot directly explain the normalized micro signal. Duplicate actions, finite continuation samples, and policy-dependent outcomes may affect the probe, but their contributions have not been isolated.

Micro hyper-parameters. Table 7 varies one factor at a time. The selection ratio achieves its highest peak at 15% among the three tested values. This is consistent with a trade-off between adding candidate feedback and changing the composition of the mixed actor batch, although the sweep does not isolate these mechanisms. Increasing G from 2 to 8 improves the listed last-step and peak results, reaching a 90.6 peak at $G = 8$, at the cost of more candidate computation. All tested

positive weights train successfully, but their results differ; the sweep supports the chosen defaults rather than invariance to candidate weight or universal hyper-parameter robustness.

Table 7: Micro-layer hyper-parameters on WebShop 7B (last@300 / peak, %). Defaults in bold.

Factor	Values: last@300 / peak		
w_M	0.1: 82.8 / 84.4	**0.3**: 85.2 / 89.1	1.0: 86.7 / 86.7
Selection ratio	0.05: 78.1 / 81.2	**0.15**: 85.2 / 89.1	0.30: 82.8 / 82.8
G	2: 84.4 / 84.4	**4**: 85.2 / 89.1	8: 88.3 / **90.6**

E Computational Cost

The micro layer adds no executed episodes to a given real rollout batch. Its cost is candidate generation, critic scoring, and actor training on additional rows. Observed WebShop 7B micro-stage times are 30–90 s, approximately 8–20% of recorded step time, with 160–1,000 candidate rows per step. This is a stage-time share, not an end-to-end overhead estimate. The practical advantage is that additional comparisons use model computation without replaying the environment. In the recorded training curves the micro layer reaches 70% on ScienceWorld about twice as fast at token granularity and about four times as fast at turn granularity, and on WebShop 7B the micro-off control never reaches 80% within the budget at all.