BAPO: BEHAVIOR-AWARE POLICY OPTIMIZATION FOR FINE-GRAINED CREDIT ASSIGNMENT IN LONG-HORIZON AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement learning (RL) has become an effective approach for improving the sequential decision-making capabilities of large language model (LLM) agents. While outcome-based methods such as Group Relative Policy Optimization (GRPO) offer a simple way to optimize agent policies using task-level rewards, their trajectory-level supervision often fails to distinguish the contributions of intermediate interactions. Recent methods therefore exploit finer-grained structures within agent trajectories to improve credit assignment. A key challenge is to distinguish intermediate states that reflect meaningful task progress from those that contribute little to the final outcome. Changes in an agent's tool-use behavior can provide useful signals for locating such task-relevant states along the trajectory. Building on this insight, we propose **Behavior-Aware Policy Optimization** (BAPO), a simple method for identifying task-relevant intermediate states and using them to construct fine-grained learning signals for policy optimization. BAPO first detects coarse changes in an agent's tool-use behavior to localize candidate transition regions. It then attributes task-level outcomes to these candidate transitions by contrasting behavioral patterns across successful and failed trajectories, identifying transitions associated with task success. Based on this attribution, BAPO redistributes trajectory-level advantages toward actions associated with reaching these states. In this way, BAPO uses task outcomes to provide differentiated supervision for intermediate interactions. Experiments on ALFWorld and Terminal-Bench with Qwen-family models show that BAPO consistently improves agent performance, with success-rate absolute gains of up to 4.5 %. Beyond policy optimization, we find that these transition signals can also guide inference-time trajectory compression, suggesting that they capture behaviorally meaningful structure useful for both training and inference.

1 INTRODUCTION

Large language model (LLM) agents have evolved beyond basic conversational interfaces to execute complex tasks in domains like coding, search, and multimodal applications (Yang et al., 2024; Jin et al., 2025; Yao et al., 2022; Li et al., 2025). In this context, reinforcement learning (RL) Sutton et al. (1998); Ouyang et al. (2022) offers a highly natural framework, optimizing these agents to solve long horizon problems through continuous environmental interaction.

Currently, reinforcement learning algorithms like Proximal Policy Optimization (PPO) (Schulman et al., 2017) and Group Relative Policy Optimization (GRPO) (Shao et al., 2024) serve as the primary engines for optimizing LLM policies in general reasoning tasks. Motivated by their success, a recent wave of work has extended these methods to interactive agents. For example, frameworks such as Search-R1 (Jin et al., 2025), ReSearch (Chen et al., 2025), and ReTool (Feng et al., 2026a) leverage these algorithms to drive multi-turn search and tool utilization.

However, many of these methods rely primarily on task outcomes for supervision. In tasks spanning dozens of interactions, terminal rewards indicate overall success but provide limited guidance about the contributions of intermediate actions (Arjona-Medina et al., 2019; Lightman et al., 2024). For example, locating relevant information, performing an effective operation, and repeating an un-

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.

successful search may all receive the same trajectory-level feedback. Applying a shared advantage signal (Shao et al., 2024) across these interactions does not distinguish their respective contributions, creating a credit assignment challenge in long-horizon learning.

Recent work addresses this challenge by exploiting structure within or across trajectories. GiGPO (Feng et al., 2026b) groups actions at recurring environment states to estimate step-level relative advantages; G2PO (Wang et al., 2026a) uses graphs of state relationships to construct fine-grained advantage signals; and ECHO (Xie et al., 2026) uses source-indexed turn memories to route positive credit back to earlier interactions. These approaches highlight the value of relationships among interactions for credit assignment. We explore a complementary structure by grouping consecutive interactions into behavioral intervals defined by shifts in the agent's execution patterns. These intervals help localize candidate critical states and provide structured units for credit assignment.

Long-horizon tasks often unfold through phases in which consecutive interactions serve a shared intermediate goal (Peng et al., 2026). For example, an agent may initially alternate between searching and browsing to gather information, then shift to editing files, executing programs, and verifying results. Organizing such interactions into behavioral intervals is motivated by temporal abstraction in reinforcement learning, where temporally extended behaviors serve as units for learning and decision-making (Sutton et al., 1999). From this perspective, changes in tool-use patterns offer observable cues for identifying candidate boundaries between behavioral intervals. Yet a behavioral shift does not necessarily indicate progress: it may also reflect a failed attempt or ineffective exploration. We therefore consider behavioral structure together with task outcomes, using the former to organize the trajectory and the latter to assess which intervals are associated with success. This motivates our central question: *Can behavioral shifts reveal useful units for credit assignment, and can task outcomes guide how credit is distributed across these units?*

In this paper, we propose **B**ehavior-**A**ware **P**olicy **O**ptimization (BAPO), a framework that combines behavioral segmentation with outcome-conditioned interval scoring. BAPO first encodes tool calls and their execution feedback into behavioral representations, then detects changes in their local distributions to partition a trajectory into behavioral intervals. It then performs interval-level credit attribution using evidence from successful and failed trajectories. Finally, BAPO prioritizes the highest-scoring interval through a centered advantage adjustment that preserves the mean advantage over generated tokens. This design uses outcome information to differentiate learning signals across behavioral intervals without requiring an external reward model.

Beyond policy optimization, this behavior-aware perspective naturally extends to a targeted inference harness tailored for long-horizon deployments. By dynamically compressing the historical steps within these identified behavioral intervals, this harness enables a continuously evolving state representation. Consequently, it establishes a unified mechanism for history management that strictly aligns with the agent's current execution phase, seamlessly bridging short-term execution and long-term planning across both training and inference paradigms.

Extensive experiments demonstrate the superiority of our approach. BAPO not only significantly accelerates convergence and enhances training stability, but also achieves an additional 4% performance gain over the strong Qwen-family baselines on challenging long-horizon benchmarks. Furthermore, the proposed harness effectively mitigates trajectory drifting during extended rollouts. Ultimately, BAPO provides unified guidance for identifying and leveraging the actual critical states that dictate success in complex agentic tasks.

In summary, the main contributions of this paper are as follows:

1. We present Behavior-Aware Policy Optimization (BAPO), to the best of our knowledge, an effective reinforcement learning framework that addresses long-horizon credit assignment by autonomously defining critical states through the agent's intrinsic behavioral evolution.

2. We propose a lightweight advantage redistribution mechanism to pinpoint pivotal transitions without external reward models, alongside a complementary inference harness that compresses history into summaries to prevent trajectory drift.

3. Experimental results demonstrate that BAPO significantly enhances training stability and achieves a notable 4.5% performance gain over strong Qwen-family baselines on complex long-horizon benchmarks.

2 Related Work

Fine-Grained Credit Assignment for LLM Agents. Reinforcement learning has been extended to interactive LLM agents through frameworks such as Search-R1 (Jin et al., 2025) and ReTool (Feng et al., 2026a), which optimize multi-turn tool use from task feedback. A central challenge is assigning this feedback to intermediate interactions. Process supervision provides local signals by training reward models to evaluate intermediate reasoning steps (Lightman et al., 2024). Rewarding Progress (Setlur et al., 2025) further characterizes useful process rewards through changes in the probability of eventual success under a prover policy, connecting intermediate evaluation to task outcomes. Another line of work derives credit from relationships within or across interaction trajectories. GiGPO (Feng et al., 2026b) groups actions at recurring environment states to estimate step-level relative advantages. G2PO (Wang et al., 2026a) aggregates matching observations into a shared state-transition graph and constructs edge-level advantages from state-value estimates. GraphGPO (Cheng et al., 2026) instead evaluates transitions according to their estimated reduction in graph-based distance to the task goal. ECHO (Xie et al., 2026) uses the source indices of selected memories to route positive outcome credit to earlier interactions and memory-selection actions.

Temporal Abstraction and Segment-Level Optimization. Temporal abstraction provides a foundation for organizing sequential behavior beyond individual actions. The options framework formalizes temporally extended policies (Sutton et al., 1999), while CompILE learns reusable, variable-length behavioral segments from demonstration trajectories (Kipf et al., 2019). These approaches establish behavioral segments as useful units for learning and execution, although their segmentation objectives differ from credit assignment in LLM agents. Recent work explicitly incorporates intermediate temporal granularity into LLM optimization. Segment Policy Optimization (SPO) (Guo et al., 2026) partitions reasoning sequences and estimates segment-level advantages through Monte Carlo sampling. HiPER (Peng et al., 2026) organizes agent execution around explicit subgoals and estimates advantages at both the planning and action levels. BEACON (Wang et al., 2026b) partitions trajectories at milestone boundaries, combining within-segment reward shaping with advantages at two temporal scales. SPACE (Yang et al., 2026) derives action-chunk boundaries from programmatic skills induced from successful trajectories and uses chunk-aware optimization to learn variable-length action sequences.

Context Management for Long-Horizon Agents Context management addresses the growing interaction histories produced during long-horizon execution. MemGPT (Packer et al., 2023) organizes memory into multiple tiers and allows the model to manage information movement between its active context and external storage. More recent approaches directly compress ongoing interaction histories. ReSum (Wu et al., 2025) periodically summarizes accumulated context to support continued exploration; its ReSum-GRPO variant additionally trains agents on trajectories segmented at summarization points by broadcasting outcome-based advantages across segments. AgentFold (Ye et al., 2025) learns to fold selected interactions into summaries at multiple temporal scales, including groups of consecutive interactions. ACON (Kang et al., 2025) optimizes compression guidelines using task feedback to improve information retention in compressed histories and observations.

3 Methodology

In this section, we introduce the Behavior-Aware Policy Optimization (BAPO) framework. We first formalize the interval-based perspective to expose the credit assignment bottleneck in long-horizon tasks (Section 3.1). Subsequently, we detail how BAPO autonomously discovers intrinsic behavioral shifts without external supervision (Section 3.2), rigorously redistributes global credit via contrastive empirical signals and zero-sum constraints (Section 3.3), and ultimately extends this paradigm into a unified behavior-driven inference harness (Section 3.4).

3.1 Problem Formulation

In standard RL for LLM agents, a long-horizon task is modeled as a Partially Observable Markov Decision Process (POMDP). The agent, parameterized by an LLM policy $\pi_\theta(a_t \mid h_t)$, interacts with the environment over N steps, generating a flat sequence trajectory $\tau = \{(o_1, a_1), \ldots, (o_N, a_N)\}$ and receiving a sparse reward $r \in \{0, 1\}$ at the end.Standard policy-gradient methods (e.g., GRPO)

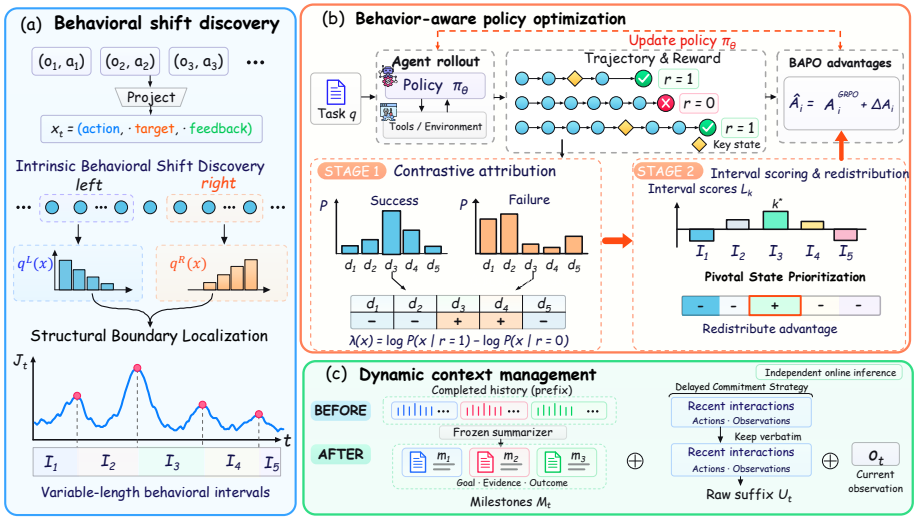

Figure 1: Overview of BAPO. (a) Behavioral boundary detection identifies coherent macro-states. (b) Advantage redistribution prioritizes intervals associated with task success. (c) Behavior-aware context compression summarizes completed intervals while retaining recent interactions.

derive their optimization signal from the trajectory-level outcome, providing limited supervision for distinguishing the relative importance of different temporal regions within the same rollout.

However, in complex open-ended agentic tasks, task progression does not accumulate uniformly across atomic steps. Instead, it exhibits strong *temporal abstraction*: dense, exploratory reasoning steps are punctuated by sparse, decisive bottleneck transitions (e.g., successfully executing a pivotal tool or retrieving critical context). However, trajectory-level outcome supervision does not explicitly reveal which temporal regions are most strongly associated with task success, leaving substantial ambiguity in fine-grained credit assignment.

To bridge this gap, we introduce a temporally abstracted view of the trajectory. Rather than treating τ as a monolithic sequence, we partition its temporal horizon into K non-overlapping intervals $\mathcal{I} = I_1, \ldots, I_K$ ($K \ll N$). Each interval $I_k = [s_k, e_k]$ induces a macro-state S_k, defined as the ordered sequence of atomic interactions within that interval:

$$S_k = (o_{s_k}, a_{s_k}), \ldots, (o_t, a_t), \ldots, (o_{e_k}, a_{e_k}). \tag{1}$$

These macro-states sequentially span the entire trajectory without gaps or overlaps. Under this abstraction, BAPO uses the resulting temporal structure to guide both credit assignment during training and context management during inference, as illustrated in Fig. 1:

- **Behavioral boundary detection** identifies $\mathcal{B} = \{e_1, e_2, \ldots, e_{K-1}\}$ from local changes in tool-use patterns, grouping consecutive interactions into behaviorally coherent macro-states;
- **Advantage redistribution** scores macro-states by their empirical association with task success and reallocates credit across generated tokens while preserving the trajectory-level advantage;
- **Behavior-aware context compression** compresses completed macro-states into summaries and evidence-grounded entries, retaining recent interactions as context for subsequent decisions.

3.2 Intrinsic Behavioral Shift Discovery

Localizing the boundary set $\mathcal{B}$ requires identifying changes in the agent's interaction patterns along the trajectory. Because raw textual interactions contain substantial semantic variation that is not directly relevant to temporal structure, we first project each interaction step into a compact behavioral descriptor. Specifically, we define a functional projection operator $f : \mathcal{O} \times \mathcal{A} \to \mathcal{X}$ that maps each interaction into $x_t = (\text{action}, \text{target}, \text{feedback})$.

BAPO treats x as a random variable over behavioral descriptors. At each timestep t, we construct a symmetric sliding window of size W, with the left window $W_t^{\mathrm{L}} = \{x_{t-W}, \ldots, x_{t-1}\}$ and the right window $W_t^{\mathrm{R}} = \{x_{t+1}, \ldots, x_{t+W}\}$. We denote the empirical distribution of x within these windows by q_t^{L} and q_t^{R}.

We then quantify the behavioral shift between the two windows using the Jensen-Shannon (JS) divergence, which measures local changes in behavioral composition and serves as a signal for potential transitions between behavioral phases:

$$J_t = D_{\mathrm{JS}}(q_t^{\mathrm{L}}(x) \parallel q_t^{\mathrm{R}}(x)) = \frac{1}{2} D_{\mathrm{KL}}(q_t^{\mathrm{L}} \parallel M_t) + \frac{1}{2} D_{\mathrm{KL}}(q_t^{\mathrm{R}} \parallel M_t), \tag{2}$$

where $M_t = \frac{1}{2}(q_t^{\mathrm{L}} + q_t^{\mathrm{R}})$. The resulting score $J_t \in [0, 1]$ measures the degree of local change in behavioral composition. A sufficiently strong local maximum in the sequence J_t is treated as a candidate behavioral boundary. To retain only salient local changes while keeping both comparison windows inside the trajectory, we define the boundary set as

$$\mathcal{B} = \left\{ t \in \{W+1, \ldots, N-W\} \mid J_t > \delta \text{ and } J_t = \max_{t' \in \{t-w, \ldots, t+w\}} J_{t'} \right\}, \tag{3}$$

where δ is the detection threshold and w is the radius of the local peak-detection neighborhood. The detected boundaries partition the trajectory into $K = |\mathcal{B}| + 1$ non-overlapping intervals, each of which induces a macro-state used in the subsequent scoring and advantage redistribution stages.

3.3 Contrastive Attribution Credit Redistribution

Given a partitioned trajectory $\mathcal{I} = \{I_1, \ldots, I_K\}$, BAPO identifies which interval contributes most to task success. We achieve this by contrasting successful and failed rollouts to distribute the final trajectory reward into interval-specific learning signals.

Contrastive Outcome Attribution. We first estimate the distribution of behavioral descriptors across successful and failed trajectories. For each descriptor x, we compute its contrastive score via a log-likelihood ratio:

$$\lambda(x) = \log P(x \mid r = 1) - \log P(x \mid r = 0). \tag{4}$$

A higher $\lambda(x)$ simply means that descriptor x appears more often in successes than failures, making it a practical indicator of its positive impact on the task.

Fine-Grained Token-Level Alignment. To use these behavioral scores for token-level policy optimization, we must map them back to individual generation steps. We first assign a score to each interaction step t based on the occurrences of its descriptors $d_t(x)$:

$$g_t = \frac{\sum_x d_t(x)\lambda(x)}{\sum_x d_t(x)}. \tag{5}$$

We then average these step scores within each interval, weighted by the number of generated tokens n_t. Letting $m_k = \sum_{t \in I_k} n_t$ be the token count of interval I_k, and $M = \sum_k m_k$ be the total trajectory tokens, the interval score L_k and the trajectory mean $\bar{L}$ are:

$$L_k = \frac{1}{m_k} \sum_{t \in I_k} n_t g_t, \qquad \bar{L} = \frac{1}{M} \sum_{k=1}^{K} m_k L_k. \tag{6}$$

Zero-Sum Advantage Redistribution. To pinpoint the critical bottleneck that drives task progression, BAPO next identifies the interval with the strongest empirical association with task success:

$$k^\star = \arg\max_{1 \le k \le K} L_k. \tag{7}$$

BAPO concentrates the redistributed credit on $I_{k^\star}$, with its advantage modifier u_k determined by how much the interval score exceeds the trajectory-level mean:

$$u_k = \alpha \, \mathbb{I}(k = k^\star)\big(L_{k^\star} - \bar{L}\big), \tag{8}$$

where $\alpha \geq 0$ controls the redistribution strength. The modifier is then applied to the original token-level advantages. A mean-preserving centering step ensures that the resulting adjustment remains zero-sum across all generated tokens:

$$\bar{u} = \frac{1}{M} \sum_{k=1}^{K} m_k u_k, \qquad \widehat{A}_i = A_i^{\mathrm{GRPO}} + u_{k(i)} - \bar{u}, \tag{9}$$

where $k(i)$ is the interval containing token i. By design, the total adjustment is zero-sum, allowing BAPO to heavily reward the breakthrough state while preserving the stable convergence properties of the base GRPO estimator.

3.4 Behavior-Aware Dynamic Context Management

Beyond policy optimization, the behavior-aware abstraction naturally serves as a plug-and-play inference harness. In long-horizon tasks, maintaining a strictly expanding history inevitably exacerbates context fragmentation and triggers catastrophic trajectory drift. To counteract this, BAPO dynamically compresses historical interactions at detected behavioral boundaries, replacing raw sequences with dense semantic milestones. Crucially, this mechanism operates purely during inference, requiring neither policy retraining nor trajectory-level contrastive scores.

Categorical State Space Projection. To ensure statistical robustness during online boundary detection, high-dimensional textual interactions are first projected into a compact categorical space $\mathcal{C}$. Let $c_t = \phi(a_t) \in \mathcal{C}$ denote the categorical identifier of the action at step t. This ensures functionally equivalent operations on distinct targets map to a shared behavioral distribution.

Causal Phase Detection with Delayed Commitment. Unlike offline training, on-the-fly inference dictates a strictly causal boundary detection mechanism. At any step t, the agent has access to past interactions up to step $t-1$. Let p_t^L and p_t^R denote the normalized empirical histograms of the categorical identifiers over two adjacent causal windows: the antecedent window $c_{t-2H:t-H-1}$ and the subsequent window $c_{t-H:t-1}$, respectively. A structural phase boundary flag $b_{t-H} \in \{0, 1\}$ is formally registered at the lagging step $t-H$ if and only if the streaming divergence breaches a critical threshold δ:

$$b_{t-H} = \mathbb{I}\left(D_{\mathrm{JS}}\left(p_t^L \parallel p_t^R\right) > \delta\right). \tag{10}$$

This H-step lag prevents premature segmentation and ensures statistically behavioral shifts.

Hierarchical Context Restructuring. Upon confirming a boundary at step $t-H$ (i.e., $b_{t-H} = 1$), the history is split into a finalized segment $\mathcal{S}_{t-H}$ ending at step $t-H-1$ and a pending suffix $\mathcal{U}_{t-1}$ containing the most recent H interactions. A frozen LLM compressor f_{LLM} summarizes $\mathcal{S}_{t-H}$ into a compact milestone, yielding the delayed memory update:

$$\mathcal{M}_t = \begin{cases} \mathcal{M}_{t-1} \oplus f_{\mathrm{LLM}}\left(\mathcal{S}_{t-H}\right), & \text{if } b_{t-H} = 1, \\ \mathcal{M}_{t-1}, & \text{otherwise.} \end{cases} \tag{11}$$

Without a new boundary, the memory remains static and the pending suffix $\mathcal{U}_{t-1}$ continues to accrue recent operations. Consequently, the context for the agent's decision at step t is dynamically restructured into a hierarchical state representation:

$$\tilde{h}_t = \mathcal{M}_t \oplus \mathcal{U}_{t-1} \oplus o_t, \tag{12}$$

where $\mathcal{U}_{t-1}$ preserves immediate operational details and o_t is the current observation. By systematically replacing finalized sub-trajectories with compact semantic abstractions, this mechanism helps limit the growth of the active context and mitigates the accumulation of localized noise over long interaction horizons.

4 Experiments

In this section, we systematically evaluate the effectiveness of the BAPO framework across complex, long-horizon agentic benchmarks. The empirical study is designed to validate whether explicitly identifying behavioral shifts can effectively resolve the credit assignment problem during training, and whether this abstraction translates into more robust state representations during deployment. Specifically, the evaluation aims to answer three core research questions:

- **(RQ1):** Does BAPO improve policy optimization on long-horizon tasks?
- **(RQ2):** Can the behavior aware harness improve long-horizon performance?
- **(RQ3):** How do segmentation strategy and boundary-detection hyperparameters affect BAPO?

4.1 Experimental Setup

Datasets and Benchmarks. We evaluate long-horizon agent performance on two benchmarks: ALFWorld (Shridhar et al., 2020) for text-based household interaction and Terminal-Bench 2.1 (Merrill et al., 2026) for terminal-based problem solving. ALFWorld requires agents to complete household goals through sequential navigation and object manipulation. We evaluate Qwen2.5-1.5B-Instruct Yang et al. (2025) and Qwen2.5-7B-Instruct spanning six categories, using the reported results of PPO, RLOO (Ahmadian et al., 2024), GRPO, and GiGPO as references. Terminal-Bench 2.1 evaluates agents on 89 tasks that require executing commands and interacting with a terminal environment. We use Qwen3.5-9B and compare BAPO with the base model and DAPO Yu et al. (2026). For both benchmarks, we report success rate, determined by environment goal completion in ALFWorld and task-specific verifiers in Terminal-Bench.

Training Details. For ALFWorld, we follow the GiGPO training protocol. Each rollout batch contains 16 groups of eight trajectories, with a maximum of 50 environment steps per trajectory. The per-step prompt and response limits are 2,048 and 512 tokens, respectively. We use an actor learning rate of 10^{-6}, a minibatch size of 256, and a KL coefficient of 0.01. Rollout and evaluation temperatures are set to 0.4. For terminal tasks, we train Qwen3.5-9B (Qwen Team, 2026) on TMax-15K following the TMax training recipe (Ivison et al., 2026). Each rollout batch samples eight tasks with 32 trajectories per task. when training, we use a learning rate of 10^{-6} and a rollout temperature of 0.4. Furthermore, environment rollouts are powered by sglang Zheng et al. (2024), enforcing a strict generation limit of 16,384 tokens per turn and a maximum trajectory length of 65,536 tokens.

Evaluation Protocol. For ALFWorld, we follow GiGPO and evaluate with a sampling temperature of 0.4 and a maximum of 50 environment interactions per trajectory. We evaluate models on tasks spanning six categories, with success determined by the environment's goal-completion signal. We report both category-level success rates and the overall success rate computed across all tasks. For Terminal-Bench 2.1, we follow TMax and evaluate on 89 tasks using its terminus harness with a persistent shell, executed through Harbor with a Podman backend. We retain the benchmark's default task-specific timeouts and determine success using the provided verifiers. Each evaluation run measures the fraction of tasks successfully completed, and we report the mean success rate over three evaluation runs with different random seeds.

4.2 Main Results: Enhancing Policy Optimization (RQ1)

Performance on Terminal-Bench. Table 1 reports results on TB 2.1 using Qwen3.5-9B. BAPO achieves a success rate of 33.3%, compared with 16.1% for the base model and 28.8% for DAPO. This corresponds to a 17.2-percentage-point improvement over the base model and a further 4.5-point gain over DAPO. Together with the ALFWorld results, this comparison shows that BAPO improves overall task success over the compared RL baselines in both household interaction and terminal problem solving.

Table 1: Performance on Terminal-Bench 2.1 with Qwen3.5-9B, averaged over three random seeds.

Method	success rates (%)
Base model	$16.1_{\pm 3.7}$
DAPO	$28.8_{\pm 1.4}$
BAPO	$\mathbf{33.3}_{\pm 1.72}$

Performance on ALFWorld. Table 2 compares success rates on ALFWorld across two model sizes. BAPO achieves the highest overall success rate among the compared RL methods, reaching 90.8% with Qwen2.5-1.5B-Instruct and 94.0% with Qwen2.5-7B-Instruct. These results exceed GiGPO by 4.1 and 3.2 percentage points, respectively, and GRPO by 18.0 and 16.4 points. The category-level results reveal where these improvements arise. At 1.5B, BAPO improves Look from 67.5% to 98.2% and Pick2 from 76.4% to 86.3% relative to GiGPO. At 7B, the largest gains occur on Heat and Pick2, increasing from 83.7% to 98.4% and from 79.2% to 92.2%, respectively. BAPO

Table 2: Performance on ALFWorld across strong baselines, averaged over three random seeds. Subscript values indicate standard deviations, and the best results are highlighted in bold.

Type	Method	ALFWorld						
		Pick	Look	Clean	Heat	Cool	Pick2	All
Off-the-Shelf Model								
Prompting	Qwen3.5-397B	96.6	72.7	70.4	0.0	70.4	85.7	71.9
Prompting	DeepSeek-V3.2	79.3	54.6	48.2	46.2	29.6	57.1	53.1
Prompting	Gemini-2.5-Pro	92.8	63.3	62.1	69.0	26.6	58.7	60.3
Qwen2.5-1.5B-Instruct								
Prompting	Qwen2.5	5.9	5.5	3.3	9.7	4.2	0.0	4.1
Prompting	ReAct	17.4	20.5	15.7	6.2	7.7	2.0	12.8
Prompting	Reflexion	35.3	22.2	21.7	13.6	19.4	3.7	21.8
RL Training	PPO (with critic)	$64.8_{\pm 3.5}$	$40.5_{\pm 6.9}$	$57.1_{\pm 4.9}$	$60.6_{\pm 6.6}$	$46.4_{\pm 4.0}$	$47.4_{\pm 1.9}$	$54.4_{\pm 3.1}$
RL Training	RLOO	$88.3_{\pm 3.0}$	$52.8_{\pm 8.6}$	$71.0_{\pm 5.9}$	$62.8_{\pm 8.7}$	$66.4_{\pm 5.5}$	$56.9_{\pm 4.7}$	$69.7_{\pm 2.5}$
RL Training	GiGPO	$\mathbf{94.4}_{\pm 5.9}$	$67.5_{\pm 4.6}$	$\mathbf{94.8}_{\pm 3.8}$	$\mathbf{94.4}_{\pm 7.8}$	$79.8_{\pm 4.7}$	$76.4_{\pm 5.4}$	$86.7_{\pm 1.7}$
RL Training	GRPO	$85.3_{\pm 1.5}$	$53.7_{\pm 8.0}$	$84.5_{\pm 6.8}$	$78.2_{\pm 7.9}$	$59.7_{\pm 5.0}$	$53.5_{\pm 5.6}$	$72.8_{\pm 3.6}$
RL Training	**BAPO**	$93.1_{\pm 4.8}$	$\mathbf{98.2}_{\pm 3.2}$	$89.3_{\pm 3.7}$	$88.9_{\pm 2.5}$	$\mathbf{88.9}_{\pm 2.8}$	$\mathbf{86.3}_{\pm 3.4}$	$\mathbf{90.8}_{\pm 1.9}$
Qwen2.5-7B-Instruct								
Prompting	Qwen2.5	33.4	21.6	19.3	6.9	2.8	3.2	14.8
Prompting	ReAct	48.5	35.4	34.3	13.2	18.2	17.6	31.2
Prompting	Reflexion	62.0	41.6	44.9	30.9	36.3	23.8	42.7
RL Training	PPO (with critic)	$92.3_{\pm 4.0}$	$64.0_{\pm 8.4}$	$92.5_{\pm 2.4}$	$89.5_{\pm 7.0}$	$80.3_{\pm 2.0}$	$68.8_{\pm 8.3}$	$80.4_{\pm 2.7}$
RL Training	RLOO	$87.6_{\pm 4.3}$	$78.2_{\pm 8.3}$	$87.3_{\pm 5.8}$	$81.3_{\pm 7.6}$	$71.9_{\pm 5.2}$	$48.9_{\pm 8.4}$	$75.5_{\pm 4.6}$
RL Training	GiGPO	$\mathbf{97.7}_{\pm 1.6}$	$82.7_{\pm 7.9}$	$\mathbf{98.8}_{\pm 1.6}$	$83.7_{\pm 7.2}$	$89.3_{\pm 8.2}$	$79.2_{\pm 6.6}$	$90.8_{\pm 1.3}$
RL Training	GRPO	$90.8_{\pm 5.1}$	$66.1_{\pm 6.7}$	$89.3_{\pm 5.4}$	$74.7_{\pm 6.9}$	$72.5_{\pm 5.4}$	$64.7_{\pm 7.3}$	$77.6_{\pm 5.2}$
RL Training	**BAPO**	$94.4_{\pm 5.6}$	$\mathbf{90.3}_{\pm 4.8}$	$95.7_{\pm 4.9}$	$\mathbf{98.4}_{\pm 2.8}$	$\mathbf{92.8}_{\pm 2.5}$	$\mathbf{92.2}_{\pm 6.8}$	$\mathbf{94.0}_{\pm 1.4}$

Table 3: Performance on ScienceWorld across Qwen-family models, averaged over three random seeds. Subscript values indicate standard deviations, and the best results are highlighted in bold.

	Qwen3.5-9B			Qwen3.5-35B-A3B		
Method	SR↑	PR↑	Cost↓	SR↑	PR↑	Cost↓
ReAct	$57.0_{\pm 1.7}$	$86.1_{\pm 0.9}$	$1.0\times$	$62.6_{\pm 2.2}$	$86.0_{\pm 1.0}$	$1.0\times$
ClaudeCode	$58.9_{\pm 2.2}$	$76.0_{\pm 1.2}$	$\mathbf{0.55}\times$	$61.1_{\pm 1.6}$	$79.7_{\pm 0.7}$	$\mathbf{0.51}\times$
Behavior-aware harness	$\mathbf{66.3}_{\pm 4.0}$	$\mathbf{88.3}_{\pm 1.4}$	$\mathbf{0.55}\times$	$\mathbf{70.0}_{\pm 4.0}$	$\mathbf{88.6}_{\pm 0.8}$	$0.52\times$

outperforms GiGPO in three of six categories at 1.5B and four at 7B, while GiGPO retains higher scores on Pick and Clean at both scales.

4.3 Inference Robustness: Mitigating Trajectory Drift (RQ2)

Settings and Baselines. The efficacy of the dynamic context-compression harness is validated on ScienceWorld (Wang et al., 2022), strictly adhering to the task selection and environmental parameters of AgentCL Shu et al. (2026). The protocol encompasses 90 tasks requiring extensive multi-step scientific reasoning, bounded at 50 action steps per task. Evaluations are executed on Qwen3.5-9B and Qwen3.5-35B-A3B natively, without supplementary policy training. Following AgentCL, thinking mode is enabled with a sampling temperature of 0.6, top-p of 0.95, and top-k of 20. The proposed harness—which condenses completed behavioral intervals into semantic milestones while retaining the raw active suffix—is benchmarked against ReAct and ClaudeCode. Performance is quantified via Success Rate (SR), Progress Rate (PR), and inference Cost (normalized to ReAct within each respective backbone).

Performance. As demonstrated in Table 3, the Behavioral aware harness achieves SRs of 66.3% and 70.0% on the 9B and 35B-A3B models, yielding absolute improvements of 9.3 and 7.4 percentage points over standard ReAct. Correspondingly, PR metrics advance to 88.3% and 88.6%. Remarkably, these performance surges coincide with drastic reductions in normalized inference Cost (dropping to $0.55\times$ and $0.52\times$), translating to computational savings of up to 48%. When juxtaposed with ClaudeCode, the proposed harness elevates SR by 7.3 points at an identical cost footprint

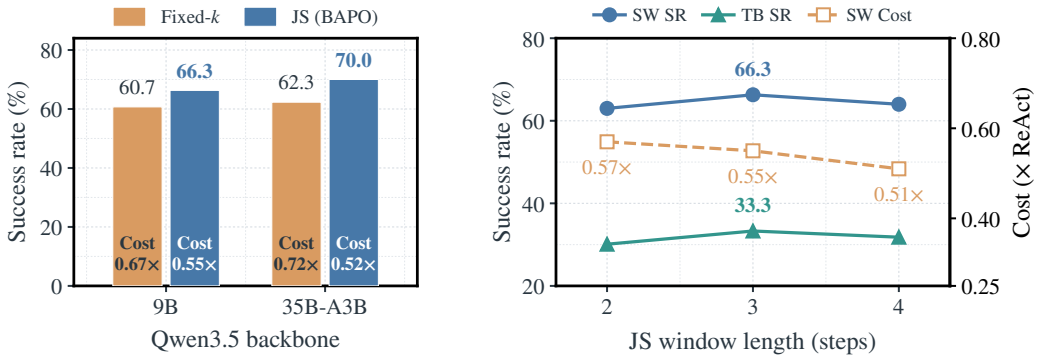

Figure 2: Effects of segmentation strategy (left) and window size (right) on success rate and cost.

for the 9B model, and by 8.9 points with negligible cost overhead for the 35B-A3B backbone. These results definitively prove that hierarchical context restructuring mathematically insulates the policy from compounding drift, driving superior task completion while slashing computational overhead.

4.4 Ablation Studies and Analysis (RQ3)

Effect of Behavioral Segmentation. We compare JS-based boundaries with fixed-k intervals in the ScienceWorld context-compression harness. Fixed-k partitions histories at regular intervals, whereas JS adapts boundaries to changes in tool-use behavior. As shown in Figure 2 (left), JS improves success from 60.7% to 66.3% on Qwen3.5-9B and from 62.3% to 70.0% on Qwen3.5-35B-A3B, yielding gains of 5.6 and 7.7 percentage points, respectively. Normalized cost also decreases from $0.67\times$ to $0.55\times$ and from $0.72\times$ to $0.52\times$. These results show that aligning compression boundaries with behavioral changes achieves higher task success at lower cost than the evaluated fixed-k schedule on both backbones.

Effect of Window Length. We vary the JS comparison window length over $\{2, 3, 4\}$ interaction steps, using Qwen3.5-9B in both experiments. For RL, we vary the offline window W during BAPO training under the TMax setup and evaluate on Terminal-Bench 2.1. For the inference harness, we vary the online window H on ScienceWorld. Each length specifies the number of steps in each window used for the JS comparison. Figure 2 (right) shows that a three-step window achieves the highest success rates among the tested lengths: 66.3% on ScienceWorld and 33.3% on Terminal-Bench 2.1, both 3% above the two-step setting. Increasing the window to four lowers success to 64.0% and 31.8%, respectively, although ScienceWorld cost decreases further from $0.55\times$ to $0.51\times$. These results support our default window length of three when prioritizing task success across the two settings.

5 Conclusion

In this paper, we propose Behavior-Aware Policy Optimization (BAPO), a framework for modeling the temporal structure of long-horizon agent trajectories. BAPO detects intrinsic behavioral shifts to partition trajectories into macro-states, estimates their association with task success through contrastive attribution, and redistributes credit toward the most informative interval under a zero-sum constraint. The same behavioral abstraction is further extended to inference-time context management, where completed interaction segments are compressed while recent information is preserved. Experiments across ALFWorld, Terminal-Bench 2.1, and ScienceWorld consistently demonstrate the effectiveness of BAPO in improving task performance and inference efficiency. Overall, BAPO provides a unified way to exploit behavioral structure for both training-time credit assignment and inference-time context management in long-horizon agents. In the future, we plan to explore richer behavioral representations and more adaptive boundary detection strategies to further improve credit assignment and context management in longer and more diverse agent trajectories.

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.

Jose A Arjona-Medina, Michael Gillhofer, Michael Widrich, Thomas Unterthiner, Johannes Brandstetter, and Sepp Hochreiter. Rudder: Return decomposition for delayed rewards. *Advances in Neural Information Processing Systems*, 32, 2019.

Mingyang Chen, Tianpeng Li, Haoze Sun, Yijie Zhou, Chenzheng Zhu, Haofen Wang, Jeff Z Pan, Wen Zhang, Huajun Chen, Fan Yang, et al. Research: Learning to reason with search for llms via reinforcement learning. *arXiv preprint arXiv:2503.19470*, 2(3):7, 2025.

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.

Jiazhan Feng, Shijue Huang, Xingwei Qu, Ge Zhang, Yujia Qin, Baoquan Zhong, Chengquan Jiang, Jinxin Chi, and Wanjun Zhong. Retool: Reinforcement learning for strategic tool use in llms. In *International Conference on Learning Representations*, volume 2026, pp. 37909–37926, 2026a.

Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. Group-in-group policy optimization for llm agent training. *Advances in Neural Information Processing Systems*, 38:46375–46408, 2026b.

Yiran Guo, Lijie Xu, Jie Liu, Dan Ye, and Shuang Qiu. Segment policy optimization: Effective segment-level credit assignment in rl for large language models. *Advances in Neural Information Processing Systems*, 38:114399–114431, 2026.

Hamish Ivison, Junjie Oscar Yin, Rulin Shao, Teng Xiao, Nathan Lambert, and Hannaneh Hajishirzi. Tmax: A simple recipe for terminal agents. *arXiv preprint arXiv:2606.23321*, 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.

Minki Kang, Wei-Ning Chen, Dongge Han, Huseyin A Inan, Lukas Wutschitz, Yanzhi Chen, Robert Sim, and Saravan Rajmohan. Acon: Optimizing context compression for long-horizon llm agents. *arXiv preprint arXiv:2510.00615*, 2025.

Thomas Kipf, Yujia Li, Hanjun Dai, Vinicius Zambaldi, Alvaro Sanchez-Gonzalez, Edward Grefenstette, Pushmeet Kohli, and Peter Battaglia. Compile: Compositional imitation learning and execution. In *International Conference on Machine Learning*, pp. 3418–3428. PMLR, 2019.

Jiaze Li, Jingyang Chen, Yuxun Qu, Shijie Xu, Zhenru Lin, Junyou Zhu, Boshen Xu, Wenhui Tan, Pei Fu, Jianzhong Ju, et al. Xiaomi mimo-vl-miloco technical report. *arXiv preprint arXiv:2512.17436*, 2025.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *International Conference on Learning Representations*, volume 2024, pp. 39578–39601, 2024.

Mike Merrill, Alexander Shaw, Nicholas Carlini, Boxuan Li, Harsh Raj, Ivan Bercovich, Lin Shi, Jeong Shin, Thomas Walshe, E Kelly Buchanan, et al. Terminal-bench: Benchmarking agents on hard, realistic tasks in command line interfaces. In *International Conference on Learning Representations*, volume 2026, pp. 40903–40986, 2026.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35: 27730–27744, 2022.

Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G Patil, Ion Stoica, and Joseph E Gonzalez. Memgpt: Towards llms as operating systems. *arXiv preprint arXiv:2310.08560*, 2023.

Jiangweizhi Peng, Yuanxin Liu, Ruida Zhou, Charles Fleming, Zhaoran Wang, Alfredo Garcia, and Mingyi Hong. Hiper: Hierarchical reinforcement learning with explicit credit assignment for large language model agents. *arXiv preprint arXiv:2602.16165*, 2026.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026. URL `https://qwen.ai/blog?id=qwen3.5`.

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

Amrith Setlur, Chirag Nagpal, Adam Fisch, Xinyang Geng, Jacob Eisenstein, Rishabh Agarwal, Alekh Agarwal, Jonathan Berant, and Aviral Kumar. Rewarding progress: Scaling automated process verifiers for llm reasoning. In *International Conference on Learning Representations*, volume 2025, pp. 60808–60838, 2025.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. Alfworld: Aligning text and embodied environments for interactive learning. *arXiv preprint arXiv:2010.03768*, 2020.

Yiheng Shu, Bernal Jiménez Gutiérrez, Saisri Padmaja Jonnalagedda, Yuguang Yao, Huan Sun, and Yu Su. Agentcl: Toward rigorous evaluation of continual learning in language agents. *arXiv preprint arXiv:2606.02461*, 2026.

Richard S Sutton, Andrew G Barto, and Andrew Barto. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.

Richard S Sutton, Doina Precup, and Satinder Singh. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial intelligence*, 112(1-2):181–211, 1999.

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, 2022.

Yunan Wang, Minghui Song, Zihan Zhang, Shaohan Huang, Haizhen Huang, Furu Wei, Weiwei Deng, Feng Sun, and Qi Zhang. Group-graph policy optimization for long-horizon agentic reinforcement learning. *arXiv preprint arXiv:2606.22995*, 2026a.

Zixuan Wang, Yuchen Yan, Hongxing Li, Teng Pan, Dingming Li, Ruiqing Zhang, Weiming Lu, Jun Xiao, Yueting Zhuang, and Yongliang Shen. Milestone-guided policy learning for long-horizon language agents. *arXiv preprint arXiv:2605.06078*, 2026b.

Xixi Wu, Kuan Li, Yida Zhao, Liwen Zhang, Litu Ou, Huifeng Yin, Zhongwang Zhang, Xinmiao Yu, Dingchu Zhang, Yong Jiang, et al. Resum: Unlocking long-horizon search intelligence via context summarization. *arXiv preprint arXiv:2509.13313*, 2025.

Zijun Xie, Binbin Zheng, Enlei Gong, Jihua Liu, Yuyang You, Lingfeng Liu, Jiayao Tang, Guanqun Zhao, Aoqi Hu, and Zeyu Chen. Echo: Prune to act, trace to learn with selective turn memory in agentic rl. *arXiv preprint arXiv:2606.31650*, 2026.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

John Yang, Carlos Jimenez, Alexander Wettig, Kilian Lieret, Shunyu Yao, Karthik Narasimhan, and Ofir Press. Swe-agent: Agent-computer interfaces enable automated software engineering. *Advances in Neural Information Processing Systems*, 37:50528–50652, 2024.

Yanting Yang, Can Jin, Jinman Zhao, Jiahao Wu, Yang Zhou, Zhepeng Wang, Zhendong Wang, Mu Zhou, and Dimitris N Metaxas. Act more, decide less: Skill-guided adaptive action chunking for long-horizon llm agents. *arXiv preprint arXiv:2609.02042*, 2026.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2022.

Rui Ye, Zhongwang Zhang, Kuan Li, Huifeng Yin, Zhengwei Tao, Yida Zhao, Liangcai Su, Liwen Zhang, Zile Qiao, Xinyu Wang, et al. Agentfold: Long-horizon web agents with proactive context management. *arXiv preprint arXiv:2510.24699*, 2025.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *Advances in Neural Information Processing Systems*, 38:113222–113244, 2026.

Lianmin Zheng, Liangsheng Yin, Zhiqiang Xie, Chuyue Sun, Jeff Huang, Cody H Yu, Shiyi Cao, Christos Kozyrakis, Ion Stoica, Joseph E Gonzalez, et al. Sglang: Efficient execution of structured language model programs. *Advances in neural information processing systems*, 37:62557–62583, 2024.

A USAGE OF LLM

In preparing this manuscript, the authors made only use of a large language model (ChatGPT). Its role was limited to refining grammar, finding typos and mis-spelled words to improve readability. The study design, experimental procedures, data analysis, interpretation of results, and development of scientific ideas were carried out entirely by the authors without any reliance on the LLM. All substantive intellectual contributions remain solely the work of the authors, who carefully reviewed and verified the final text.

B TRAINING AND EVALUATION DETAILS

In this section, we collect the training and evaluation settings specified in the experimental protocol. ALFWorld follows the shared GiGPO setup; terminal training follows the TMax data and rollout configuration, with SGLang as the rollout engine in our implementation. The ScienceWorld harness is evaluated separately without additional policy training. Reference prompts are provided in Appendix C.

Table 4: RL training settings stated in the experimental protocol.

Setting	ALFWorld	Terminal training
Backbone	Qwen2.5-1.5B / 7B-Instruct	Qwen3.5-9B
Training data / environment	ALFWorld	TMax-15K
Tasks or groups per rollout batch	16	8
Trajectories per group / task	8	32
Trajectories per rollout batch	128	256
Maximum environment / tool interactions	50	64
Actor learning rate	10^{-6}	10^{-6}
Rollout temperature	0.4	0.4
Per-step / turn response limit (tokens)	512	16,384

ALFWorld evaluation. We evaluate on test tasks spanning Pick, Look, Clean, Heat, Cool, and Pick2, with at most 50 environment interactions and sampling temperature 0.4. A task succeeds when the environment reports goal completion. The All column is computed over the full test set, rather than as an unweighted average of the six category percentages.

Terminal-Bench evaluation. The evaluation uses 89 Terminal-Bench 2.1 tasks, the terminus-2 persistent-shell harness, Harbor, and the Podman backend. Task-specific verifiers determine success, and benchmark-default task timeouts are retained. Reported scores average three evaluation runs with different random seeds; this does not by itself denote three independent training runs. The 65,536-token training limit applies to generated responsest.

Table 5: ScienceWorld inference settings following the AgentCL task and decoding protocol. The two model sizes are evaluated separately.

Setting	Value
Backbones	Qwen3.5-9B; Qwen3.5-35B-A3B
Evaluation subset	full ScienceWorld tasks from AgentBoard
Maximum action steps per task	50
Additional policy training	None
Thinking mode	Enabled
Sampling temperature	0.6
Top-p / top-k	0.95 / 20
Default online JS window H	3

Metrics and aggregation. For n evaluation tasks, let s_i indicate goal completion and $p_i \in [0, 1]$ denote the final ScienceWorld progress score. Following the AgentBoard convention, the reported metrics are

$$\mathrm{SR} = \frac{100}{n} \sum_{i=1}^{n} s_i, \qquad \mathrm{PR} = \frac{100}{n} \sum_{i=1}^{n} p_i. \tag{13}$$

SR therefore measures complete task success, whereas PR also reflects partial progress.

C PROMPTS TEMPLATE

We provide all prompt templates used for training and evaluation.

ALFWorld Training and Evaluation Template (Reference)

```
You are an expert agent operating in the ALFRED Embodied
Environment. Your task is to: {task_description}
Prior to this step, you have already taken {step_count} step(s).
Below are the most recent {history_length} observations and the
corresponding actions you took: {action_history}
You are now at step {current_step} and your current observation
is: {current_observation}
Your admissible actions of the current situation are:
[{admissible_actions}].

Now it's your turn to take an action.
You should first reason step-by-step about the current situation.
This reasoning process MUST be enclosed within <think> </think>
tags.
Once you've finished your reasoning, you should choose an
admissible action for current step and present it within
<action> </action> tags.
```

Figure 3: ALFWorld prompt template following official GiGPO implementation.

TMax Terminal-Agent Training Template

```
[System: upstream instructions]
You are a helpful assistant that can interact with a computer.

Your response must include a THOUGHT section before your action
where you explain your reasoning. After the THOUGHT, you must
call the `bash` tool with EXACTLY ONE bash command (multiple
commands chained with `&&` or `||` count as a single action).

Failure to follow these rules - calling no tool, calling a tool
other than `bash`, or omitting the THOUGHT - will cause your
response to be rejected.
```

Figure 4: TMax training prompt template

ScienceWorld Interaction Template

```
[System]
You are a helpful agent that interacts with the virtual science
school environment to solve the given task.

[User]
Task: {task_description}
Observation: {initial_observation}
Inventory: {inventory}

Available commands include:
look around; look at {OBJ}; look in {OBJ}; read {OBJ};
open {OBJ}; close {OBJ}; pick up {OBJ}; put down {OBJ};
move {OBJ} to {OBJ}; pour {OBJ} into {OBJ};
activate {OBJ}; deactivate {OBJ}; use {OBJ} [on {OBJ}];
go to {LOC}; focus on {OBJ}; wait [DURATION]; task; inventory.

[Interaction history]
ACTION: {previous_action}
OBSERVATION: {environment_feedback}

[Assistant]
ACTION: {next_action}
```

Figure 5: ScienceWorld interaction template.

Behavior aware context engineering prompt Template

```
[System]
Task: {goal}

Boundary metadata (selected deterministically before this call):
{boundary_json}

Already retained append-only memory:
<existing_memory>
{if not existed, then (none)}
</existing_memory>

Exact visible segment ending at the selected cut:
<sealed_segment>
[step i] observation / action / result
</sealed_segment>

The boundary came from tool-call distribution change. It is
not evidence of progress, correctness, failure, or task
completion. Record only what was visibly
observed. Do not choose a boundary, judge reward, plan a future
action, recommend recovery, or rewrite earlier memory.

Return one JSON object with exactly this schema:
{
  "schema_version": "predictive-tool-state-v0",
  "segment": {
    "summary": "concise description of the observed
    interaction and result",
    "evidence": [{"step": <first step>, "quote": "exact visible
    observation quote"}]
  },
  "memory_items": [
    {
      "value": "durable visible information that can affect
      later tool choices",
      "evidence_step": <first step>,
      "evidence_quote": "exact visible observation quote"
    }
  ]
}

Every evidence step must be one of {valid_steps}. Every quote
must be an exact substring of that step's observation or result
and at most 300 characters; select the shortest sufficient
exact clause instead of copying a whole observation. Use at
most 6 evidence entries and 8 memory items.
An empty memory_items list is valid when the segment contains
no durable information. Keep every JSON string on one line and
escape any newline, tab, quote, or backslash required by JSON.
Return JSON only.
```

Figure 6: Behavior aware compression prompt template.