SIPA: TEST-TIME REINFORCEMENT LEARNING THROUGH SPECULATIVE POLICY ADAPTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement learning (RL) has become a central paradigm for language-model post-training, yet its gains do not always transfer to unfamiliar deployment environments. Experience-based test-time adaptation has emerged as a promising approach, drawing on accumulated experience to guide subsequent decisions. However, collecting relevant experience is computationally demanding, and the resulting guidance may be inaccurate or poorly matched to the current state. We introduce Speculative Inference-Time Policy Adaptation (SIPA), a test-time RL framework that adapts the current action distribution using immediate execution feedback, without parameter updates or pre-collected trajectories. While the actor reasons, a lightweight branch proposes and tests candidate actions in isolated copies of the current environment state. Asymmetric Policy Adaptation (APA) combines the actor's selection preferences with the critic's execution-based rejection feedback. A KL-regularized logit update incorporates these signals before action commitment while bounding deviation from the actor's prior. We establish stability bounds under feedback error and sufficient conditions for expected-return improvement. Across five multi-turn benchmarks, SIPA achieves a 22.1% relative improvement in average accuracy over the strongest baseline using only 69.6% as many committed steps.

1 INTRODUCTION

Recent advances such as DeepSeek-R1 and Search-R1 have established reinforcement learning (RL) as a key driver of reasoning and tool use in large language models (LLMs) (Guo et al., 2025; Jin et al., 2025). However, policies optimized on a given task distribution may not generalize reliably to unfamiliar deployment environments (Kim et al.). Adapting to these environments through further RL training requires additional interaction data and parameter updates, making continual adaptation costly (Kim et al.; Li et al., 2026a). This motivates test-time adaptation (TTA), in which agents use information and feedback available during deployment to adjust their behavior to the task at hand (Zuo et al., 2026; Li et al., 2026a).

To enable such adaptation, one line of work updates model parameters using rewards derived from sampled responses or collected interaction trajectories (Zuo et al., 2026; Kim et al.). Training-free alternatives instead augment the context with strategies distilled from past experience (Shinn et al., 2023; Ouyang et al., 2026; Li et al., 2026a). This reliance on experience poses two challenges. First, acquiring task-relevant experience incurs sampling or interaction costs and can delay adaptation when transferable experience is scarce (Zuo et al., 2026; Li et al., 2026a). Second, experience-derived feedback can misdirect adaptation: errors in evaluating outcomes or transferring past experience to the current context can reinforce mistaken behavior rather than correct it (Wang et al., 2026; Cao et al., 2026). These challenges raise a central question:

Q. *Can an agent adapt from immediate environment feedback before committing its next action, without relying on previously collected trajectories?*

To address this question, we propose Speculative Inference-Time Policy Adaptation (SIPA), a test-time RL framework that adapts action selection through speculative execution. As shown in Figure 1, while the actor reasons about its next action, a lightweight branch proposes candidate actions in parallel and executes them in isolated copies of the current environment state. A critic evaluates

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

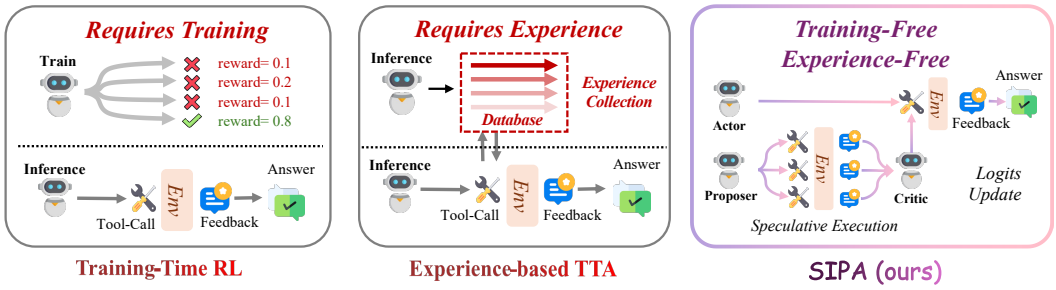

Figure 1: **Policy learning and adaptation.** Training-time RL updates parameters from trajectory rewards; experience-based TTA reuses completed trajectories. SIPA tests current-state actions during actor reasoning and adapts the action distribution before commitment, without parameter updates or past trajectories.

the observed outcomes and refines the actor's action distribution before commitment, with model weights remaining frozen. SIPA thus derives its adaptation signals from immediate environment feedback rather than previously accumulated experience. Moreover, by overlapping feedback acquisition with actor reasoning, this speculative design incurs 10.25% additional per-step wall-clock latency over Actor-only across five benchmarks with a single GPU.

Even with immediate execution feedback, the lightweight critic may struggle to identify the best action. Yet execution failures or unmet prerequisites can provide direct evidence against particular candidates. We therefore introduce Asymmetric Policy Adaptation (APA): the actor supplies a distribution over actions to take, while the critic supplies a distribution over actions to avoid. Drawing on group-relative feedback in RL (Yao et al., 2026), APA compares rejection probabilities within the proposal group to obtain nonpositive logit corrections for actor candidates with matching trials. A Kullback–Leibler (KL)-regularized update incorporates these corrections while explicitly bounding deviation from the actor's prior. The critic thus refines the actor's preferences through execution evidence rather than replacing its deliberative judgment. Empirically, SIPA delivers consistent gains across five multi-turn benchmarks, improving average accuracy by 22.1% over baselines with 10.25% additional per-step latency in the single-GPU comparison across five benchmarks.

Our contributions are threefold:

- We introduce SIPA, a test-time RL framework that adapts the current action distribution before commitment using execution feedback acquired in parallel with actor reasoning, without parameter updates or previously collected trajectories.
- We propose APA to combine the actor's selection preferences with the critic's execution-based rejection feedback through a constrained, KL-regularized update. We derive its exponential-form solution, establish stability bounds under feedback error, and identify sufficient conditions for expected-return improvement.
- Across five benchmarks, SIPA consistently improves average accuracy over Actor-only with Qwen3 models from 4B to 14B. At 4B, it achieves a 22.1% relative improvement in average accuracy over the strongest baseline ReasoningBank using only 69.6% as many committed steps.

2 Preliminaries

At step t, let s_t denote the underlying environment state and h_t the observable interaction history of the current task. An action policy $\pi(\cdot \mid h_t)$ defines a distribution over the action space. Executing an action $a_t \sim \pi(\cdot \mid h_t)$ yields $(s_{t+1}, o_t) \sim \mathcal{E}(\cdot \mid s_t, a_t)$, where $\mathcal{E}$ denotes the environment's possibly stochastic execution kernel (Yao et al., 2022). The objective is to maximize the expected task return, $J(\pi) = \mathbb{E}_{\tau\sim(\pi,\mathcal{E})}[\sum_{t=1}^{T} r_t]$, where τ is the induced trajectory, $T \leq H$ is its length, and r_t is the task reward at step t.

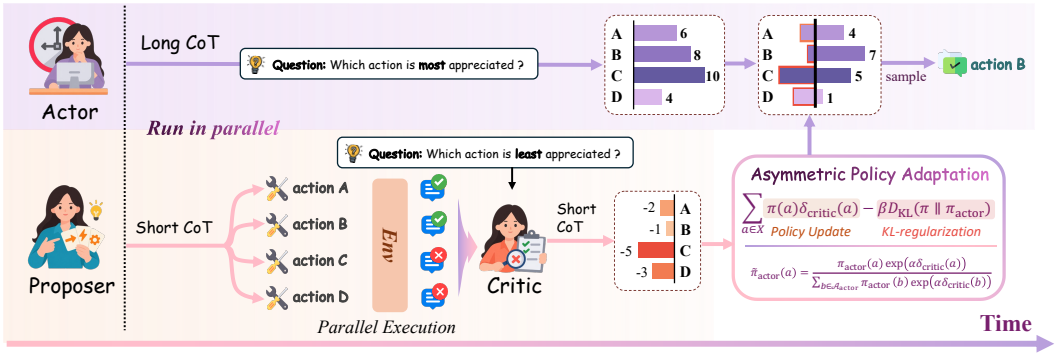

Figure 2: **One decision with SIPA.** The actor generates and scores actions while the proposer tests alternatives in isolated environment copies. The critic evaluates outcomes; APA updates the action distribution before execution.

Let p_θ denote the autoregressive token distribution of a frozen LLM. Given a nonempty finite candidate set $\mathcal{A}$ and a scoring prompt C containing the decision context, the candidates, and a selection instruction, we obtain an action distribution by normalizing complete-action likelihoods (Brohan et al., 2023):

$$\ell_\theta(a; C) = \sum_{k=1}^{|a|} \log p_\theta(a_k \mid C, a_{<k}),$$
$$\pi(a \mid C, \mathcal{A}) = \frac{\exp(\ell_\theta(a; C)/\mathcal{T})}{\sum_{b \in \mathcal{A}} \exp(\ell_\theta(b; C)/\mathcal{T})}, \qquad a \in \mathcal{A}. \tag{1}$$

Here a_k is the k-th token of the canonical action serialization, including its terminator, and $\mathcal{T} > 0$ is a scoring temperature shared across candidates in the same prompt. Token log-likelihoods are summed without length normalization. The selection instruction determines whether the distribution expresses preferences for actions to take or to avoid.

3 Speculative Inference-Time Policy Adaptation

As illustrated in Figure 2, SIPA combines a deliberative actor with a lightweight proposer–critic branch that acquires execution feedback in parallel. Asymmetric Policy Adaptation (APA) incorporates this feedback into the actor's action distribution before commitment. All three roles share the frozen model p_θ, with role-specific prompts and reasoning budgets.

3.1 Speculative Evidence Acquisition

Inspired by Speculative Actions (Ye et al., 2026), we obtain execution evidence for the current decision while the actor deliberates. We omit the step index t below. Given the shared decision context constructed from h, the actor and proposer concurrently generate candidate sets:

$$\mathcal{A}_{\text{actor}} \leftarrow \text{Generate}(h; \text{actor}),$$
$$\mathcal{A}_{\text{proposer}} \leftarrow \text{Generate}(h; \text{proposer}). \tag{2}$$

Here Generate denotes prompted list generation under the specified role, followed by schema validation and deduplication. Both candidate sets are nonempty and contain at most G distinct actions. The proposer does not wait for the actor's candidates; the two sets may overlap but need not coincide. Schema validity does not guarantee successful execution.

SIPA assumes an execution interface that supports isolated trials from the current environment state s without modifying the ongoing environment. Each proposal $a \in \mathcal{A}_{\text{proposer}}$ is executed once in a separate copy of s, yielding

$$(s'_a, o_a) \sim \mathcal{E}(\cdot \mid s, a), \qquad a \in \mathcal{A}_{\text{proposer}}. \tag{3}$$

The critic receives the action–observation pairs $\mathcal{O} = \{(a, o_a) : a \in \mathcal{A}_{\text{proposer}}\}$ and assesses public trial observations in the shared context, excluding benchmark scores and score-derived labels. The branch states s'_a are discarded, and trial outcomes do not become committed transitions. After APA, the selected action is executed afresh in the ongoing environment.

The two paths synchronize before APA. Apart from integration costs and resource contention, the auxiliary branch adds waiting only if it outlasts the actor, but still requires additional inference and trials. Appendix D details scheduling; Section 4 evaluates efficiency.

3.2 Asymmetric policy adaptation

Execution feedback can provide evidence against an action without identifying the best alternative. APA therefore combines the actor's preferences for actions to take with the critic's preferences for actions to avoid.

The scoring prompt C_{actor} combines the decision context constructed from h, the candidates $\mathcal{A}_{\text{actor}}$, and the actor's reasoning, asking "Which action is most appropriate?" The critic prompt C_{critic} combines the same decision context with $\mathcal{A}_{\text{proposer}}$, the trial evidence $\mathcal{O}$, and the critic's assessment, asking "Which action is least appropriate?" Both prompts request one complete candidate action. Applying Equation (1) gives

$$\begin{aligned} \pi_{\text{actor}}(a) &= \pi(a \mid C_{\text{actor}}, \mathcal{A}_{\text{actor}}), \qquad a \in \mathcal{A}_{\text{actor}}, \\ \pi_{\text{critic}}(a) &= \pi(a \mid C_{\text{critic}}, \mathcal{A}_{\text{proposer}}), \quad a \in \mathcal{A}_{\text{proposer}}. \end{aligned} \tag{4}$$

A larger $\pi_{\text{actor}}(a)$ favors executing a, whereas a larger $\pi_{\text{critic}}(a)$ favors rejecting it. The critic distribution represents relative rejection preferences within the proposal group, not calibrated probabilities of action failure.

Inspired by GRPO's use of within-group feedback (Yao et al., 2026), APA compares each rejection probability with that of the least-rejected proposal. For each actor candidate $a \in \mathcal{A}_{\text{actor}}$, we define

$$\delta_{\text{critic}}(a) = \begin{cases} \min\limits_{b \in \mathcal{A}_{\text{proposer}}} \pi_{\text{critic}}(b) - \pi_{\text{critic}}(a), & a \in \mathcal{A}_{\text{proposer}}, \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

Matching requires identical complete actions, including arguments. The minimum baseline makes every applied correction nonpositive: matched least-rejected proposals and untried actor candidates receive zero correction, while more strongly rejected candidates receive more negative corrections.

APA incorporates these corrections through a KL-regularized policy update. Let π range over probability distributions on $\mathcal{A}_{\text{actor}}$. We solve

$$\begin{aligned} \widetilde{\pi}_{\text{actor}} = \arg\max_{\pi} \quad & \sum_{a \in \mathcal{A}_{\text{actor}}} \pi(a)\delta_{\text{critic}}(a) - \beta D_{\text{KL}}(\pi \| \pi_{\text{actor}}), \\ \text{subject to} \quad & D_{\text{KL}}(\pi \| \pi_{\text{actor}}) \le \kappa. \end{aligned} \tag{6}$$

Here $D_{\text{KL}}(\pi \| \pi_{\text{actor}}) = \sum_{a \in \mathcal{A}_{\text{actor}}} \pi(a) \log[\pi(a)/\pi_{\text{actor}}(a)]$. The penalty weight $\beta > 0$ discourages deviation from the actor prior, while $\kappa \ge 0$ imposes a hard upper bound. This finite-action relative-entropy formulation (Peters et al., 2010; Abdolmaleki et al., 2018) admits the following update.

Proposition 1 (Exponential-form policy update)**.** *For a positive prior π_{actor} on $\mathcal{A}_{\text{actor}}$, finite feedback δ_{critic}, $\beta > 0$, and $\kappa \ge 0$, Equation* (6) *has a unique solution:*

$$\widetilde{\pi}_{\text{actor}}(a) = \frac{\pi_{\text{actor}}(a) \exp(\alpha \delta_{\text{critic}}(a))}{\sum_{b \in \mathcal{A}_{\text{actor}}} \pi_{\text{actor}}(b) \exp(\alpha \delta_{\text{critic}}(b))}, \tag{7}$$

where $0 \le \alpha \le \beta^{-1}$. For $\kappa > 0$, $\alpha = (\beta + \lambda)^{-1}$, where $\lambda \ge 0$ is the multiplier satisfying complementary slackness for the KL constraint. At $\kappa = 0$, the solution is $\widetilde{\pi}_{\text{actor}} = \pi_{\text{actor}}$, represented by $\alpha = 0$.

APA implements this solution as an action-level logit update: $\widetilde{\pi}_{\text{actor}} = \text{softmax}(\log \pi_{\text{actor}} + \alpha \delta_{\text{critic}})$, with normalization over $\mathcal{A}_{\text{actor}}$. For $\kappa > 0$, we use the nominal step $\alpha = \beta^{-1}$ if it satisfies

the KL limit; otherwise, scalar bisection finds the largest feasible step. Appendix B provides the derivation and boundary cases.

For $\alpha > 0$, more negative corrections reduce a candidate's odds relative to less penalized candidates, while equal corrections preserve prior probability ratios. A zero correction therefore does not imply an unchanged probability. Proposer-only actions contribute to the critic's comparison but do not enter the final sampling support. If the correction is constant on $\mathcal{A}_{\text{actor}}$, including when the candidate sets do not overlap, the actor distribution remains unchanged.

At each step, the agent samples $a_t \sim \widetilde{\pi}_{\text{actor},t}$ and executes it afresh in the ongoing environment. This defines π_+; the Actor-only policy π_0 instead samples from $\pi_{\text{actor},t}$, with identical actor reasoning, candidate generation, and scoring.

3.3 Theoretical analysis

We establish APA's stability under feedback error and sufficient conditions for local return improvement. The analysis fixes the actor's reasoning context, candidate set $\mathcal{A}_{\text{actor}}$, and positive prior π_{actor}, with $\beta > 0$ and $\kappa \geq 0$. Full proofs and supporting analyses appear in Appendix B.

Theorem 2 (Stability under feedback error)**.** *Let $\delta_\star$ be a finite reference signal, and let $\pi_\star$ and $\mathcal{L}_\star$ denote the optimizer and objective of Equation* (6) *with δ_{critic} replaced by $\delta_\star$, keeping the prior, β, and κ unchanged. Define the offset-invariant feedback error*

$$e = \min_{b \in \mathbb{R}} \max_{a \in \mathcal{A}_{\text{actor}}} \left| \delta_{\text{critic}}(a) - \delta_\star(a) - b \right|. \tag{8}$$

Then

$$\mathcal{L}_\star(\pi_\star) - \mathcal{L}_\star(\widetilde{\pi}_{\text{actor}}) \leq \frac{e^2}{2\beta}, \tag{9}$$

$$\max\{D_{\text{KL}}(\pi_\star \| \widetilde{\pi}_{\text{actor}}), D_{\text{KL}}(\widetilde{\pi}_{\text{actor}} \| \pi_\star)\} \leq \frac{e^2}{2\beta^2}, \tag{10}$$

$$d_{\text{TV}}(\pi_\star, \widetilde{\pi}_{\text{actor}}) \leq \frac{e}{2\beta}, \tag{11}$$

where d_{TV} denotes total variation distance.

The reference signal is used only for analysis. A larger KL penalty tightens these worst-case bounds, even when the two updates have different effective steps.

For return analysis, let $Q(a) = Q^{\pi_0}(\xi, a)$ denote the finite expected remaining return after taking a at a full environment-and-agent decision state ξ and following π_0 thereafter. The Actor-only policy π_0 disables APA but retains the actor's reasoning, candidate generation, and scoring.

Theorem 3 (Local return improvement)**.** *For a fixed decision state ξ and realized critic correction, define*

$$\Delta(\xi) = \sum_{a \in \mathcal{A}_{\text{actor}}} \left[\widetilde{\pi}_{\text{actor}}(a) - \pi_{\text{actor}}(a)\right] Q(a). \tag{12}$$

If the correction and continuation values order all actor candidates consistently,

$$\big(\delta_{\text{critic}}(a) - \delta_{\text{critic}}(b)\big)\big(Q(a) - Q(b)\big) \geq 0, \qquad a, b \in \mathcal{A}_{\text{actor}},$$

then $\Delta(\xi) \geq 0$.

More generally, take $\delta_\star(a) = Q(a)/c$ for $c > 0$, with e defined by Equation (8)*. For $\alpha > 0$,*

$$\Delta(\xi) \geq c \left[\frac{D_{\text{KL}}(\widetilde{\pi}_{\text{actor}} \| \pi_{\text{actor}}) + D_{\text{KL}}(\pi_{\text{actor}} \| \widetilde{\pi}_{\text{actor}})}{\alpha} - e \sum_{a \in \mathcal{A}_{\text{actor}}} \left| \widetilde{\pi}_{\text{actor}}(a) - \pi_{\text{actor}}(a) \right| \right]. \tag{13}$$

At $\alpha = 0$, the policy is unchanged and $\Delta(\xi) = 0$.

Alignment is required across the entire actor support, including untried candidates assigned zero correction; correct rankings within the proposal set alone are insufficient. Appendix B.3 establishes how nonnegative conditional local gains imply nonnegative expected task-return improvement under shared dynamics and fresh-execution assumptions.

4 EXPERIMENTS

Benchmarks and metrics. We evaluate on ToolSandbox (Lu et al., 2025), ScienceWorld (Wang et al., 2022), BFCL (Patil et al., 2025), ALFWorld (Shridhar et al., 2020), and BabyAI-Text (Carta et al., 2023), covering tool use, scientific interaction, and embodied instruction following. Metrics are task accuracy (Acc., %) and mean committed steps per successful task (Steps).

Models and baselines. For the main comparison, all methods use frozen Qwen3-4B. All baselines enable native thinking; within SIPA, only the actor uses native thinking. Actor-only disables APA but retains SIPA's candidate scoring and resampling; Critic-only selects trialed proposals using critic feedback. We also compare two categories of TTA: *retrieval-based TTA*, which draws experience from historical trajectories, including ReasoningBank (Ouyang et al., 2026), ACE (Zhang et al., 2026), and JitRL (Li et al., 2026a); and *generation-based TTA*, which acquires experience through repeated trials at test time, including Reflexion (Shinn et al., 2023) and SMURFS (Chen et al., 2025). Methods use matched task sets and interaction limits within each benchmark. Baseline descriptions, implementation details, and prompts appear in Appendix D.

Table 1: Performance across five benchmarks. Improvement is SIPA's gain over each method in Avg. (percentage points), with darker green indicating larger gains. Bold marks the best accuracy.

Method	ToolSandbox		ScienceWorld		BFCL		ALFWorld		BabyAI		Avg. Acc.↑	Avg. Steps↓	Improv. (Acc)
	Acc.↑	Steps↓	Acc.↑	Steps↓	Acc.↑	Steps↓	Acc.↑	Steps↓	Acc.↑	Steps↓			
Actor-only	15.98	24.91	16.02	21.66	19.17	9.65	76.12	25.00	57.89	19.00	37.04	20.04	+11.60
Critic-only	5.48	27.00	3.91	6.20	6.67	16.38	20.90	31.79	81.58	23.03	23.71	20.88	+24.93
ReasoningBank	**18.72**	34.63	12.50	24.69	20.00	9.85	82.09	22.94	65.79	21.16	39.82	22.65	+8.82
ACE	10.05	6.45	10.16	17.54	22.08	10.13	82.84	22.97	47.37	13.72	34.50	14.16	+14.14
JitRL	13.70	27.07	13.67	23.09	18.75	11.96	78.36	24.19	71.05	15.81	39.11	20.42	+9.53
Reflexion	15.53	24.38	**20.31**	26.42	25.83	10.92	72.39	30.20	31.58	13.92	33.13	21.17	+15.51
SMURFS	14.16	23.97	2.34	16.00	2.92	31.00	27.61	30.14	52.63	20.75	19.93	24.37	+28.71
SIPA (ours)	**18.72**	8.51	**20.31**	19.83	**30.00**	10.28	**87.31**	24.24	**86.84**	15.94	**48.64**	15.76	—

Steps average successful tasks only and count committed actions, including retries when used.

4.1 OVERALL PERFORMANCE

SIPA achieves the highest or tied-highest accuracy on all five benchmarks (Table 1), reaching 48.64% on average. This exceeds Actor-only by 11.60 percentage points and the strongest baseline, ReasoningBank, by 8.82 points. The largest gain occurs on BabyAI, where accuracy improves by 28.95 points over Actor-only, reducing the observed failure rate by more than two-thirds.

Retrieval-based TTA achieves higher average accuracy than generation-based TTA, but requires additional trajectory collection. Both families benefit from additional computation for experience acquisition: retrieval-based baselines receive two rounds of historical trajectory collection before evaluation (Appendix D), while generation-based baselines make repeated attempts serially at test time, introducing additional latency (Section 4.3). Nevertheless, SIPA outperforms both families, acquiring useful feedback at the current state in parallel with actor reasoning rather than accumulating trajectories or waiting for successive attempts.

SIPA also reports fewer committed steps per successful task than Actor-only. This suggests that the critic's corrective feedback helps agents avoid unproductive actions and unnecessary detours, enabling more efficient task completion. Figure 3 illustrates a case study on BabyAI. At step 4, the actor assigns 99.32% probability to `toggle`, whose trial leaves the green key two cells ahead. Independent execution shows that `go forward` instead satisfies the goal condition. The critic's rejection feedback raises the probability of `go forward` from 0.68% to above 99.99%, and the committed action completes the task in four steps. Additional cases in Appendix D.1–D.7 illustrate other ways that feedback changes the current decision.

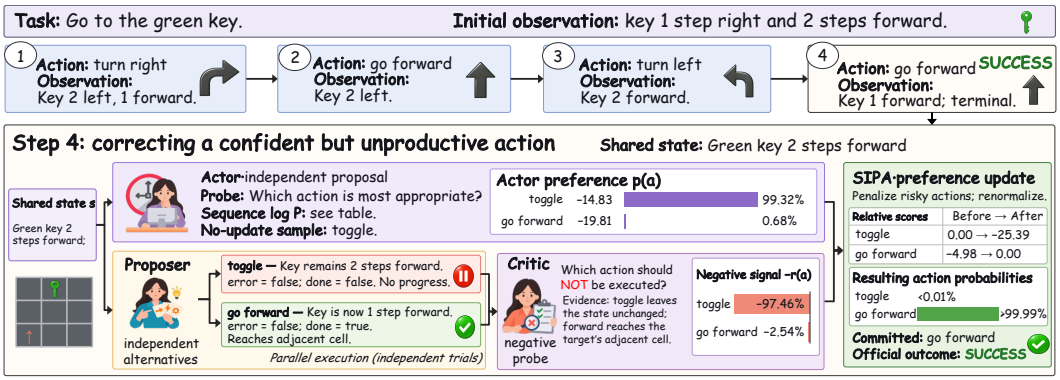

Figure 3: **BabyAI: reaching the green key.** Current-state trials identify a successful forward move over an ineffective toggle. APA raises its probability from 0.68% to above 99.99%.

4.2 Ablation Study

Table 2 compares four variants under the same evaluation protocol. **w/o APA** is Actor-only (Table 1): it retains candidate generation, complete-action probability scoring, and resampling with $\alpha = 0$, isolating feedback-based adaptation. **w/o negative probe** substitutes a positive critic probe while retaining the actor prior and trial observations, testing the value of rejection feedback. **w/o KL** removes the hard KL cap but retains the KL penalty. **w/o exec.** withholds trial observations from the critic while retaining the policy update. The latter two test the roles of bounded updates and execution evidence. We report accuracy (%) and its benchmark average (Avg.); TS, SW, and ALF abbreviate ToolSandbox, ScienceWorld, and ALFWorld. Full results appear in Appendix C.1.

SIPA leads all ablations in average accuracy, with the largest loss from disabling APA, followed by withholding execution observations. APA combines feedback with actor preferences, while speculative scheduling reduces feedback-acquisition latency (Section 4.3). Figure 4(b) shows that SIPA achieves higher accuracy than Actor-only with a smaller step budget on ALFWorld.

Table 2: Ablation results with Qwen3-4B.

Method	TS	SW	BFCL	ALF	BabyAI	Avg.
SIPA (full)	**18.72**	**20.31**	**30.00**	**87.31**	86.84	**48.64**
w/o APA	15.98	16.02	19.17	76.12	57.89	37.04
w/o negative probe	12.33	17.58	27.92	84.33	78.95	44.22
w/o KL	16.44	14.45	26.67	83.58	**89.47**	46.12
w/o exec.	13.70	13.67	29.17	84.33	73.68	42.91

Table 3: Efficiency comparison.

Method	t/step	t/task	Steps
Actor-only	11.81	248.29	21.02
Critic-only	6.92	160.67	23.22
Reflexion	19.81	460.66	23.25
SIPA, 1 GPU	13.02	226.60	17.40
SIPA, 2 GPUs	12.34	214.77	17.40

4.3 Efficiency Analysis

Table 3 reports latency (seconds) and successful-task steps across all five benchmarks. Steps are weighted by each benchmark's success count. Main experiments use one GPU, matching the baselines; the two-GPU variant separates the actor from the proposer and critic.

Figure 4(a) summarizes stage timings across the same five benchmarks, showing how speculative scheduling overlaps feedback acquisition with actor reasoning. Although the auxiliary branch finishes earlier on average, occasional slow proposer or critic samples delay commitment, producing the marked *extra critic latency* relative to Actor-only. This overhead differs from the critic's full runtime and the auxiliary branch's idle waiting time. It depends on the slowest branch at each decision, which average stage durations can obscure.

SIPA achieves higher accuracy and lower per-step latency than Reflexion, whose serial attempts place experience acquisition directly on the decision path. Two GPUs further reduce latency, show-

ing that immediate feedback can improve decisions without the full delay of sequential trial-and-feedback cycles.

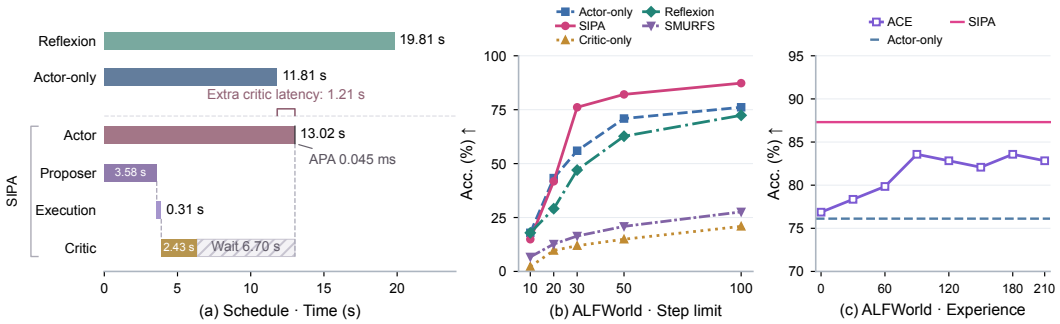

Figure 4: **(a)** Baseline latency and SIPA schedule. **(b)** ALFWorld accuracy versus step limit. **(c)** ACE experience scaling versus Actor-only and SIPA.

4.4 Adaptation without accumulated experience

Figure 4(c) tests whether additional historical experience closes the gap to SIPA on ALFWorld. ACE improves initially, but further accumulation yields fluctuating gains and remains below SIPA throughout the tested range. The advantage therefore persists beyond cold start and is not resolved by enlarging the experience pool. This suggests that experience volume alone is insufficient: adaptation also benefits from evidence directly relevant to the current decision. SIPA obtains such evidence through current-state trials, avoiding a separate trajectory-collection phase.

4.5 Scaling across model sizes

SIPA improves average accuracy over both standalone controls with Qwen3 backbones from 4B to 14B, while its own accuracy increases with model size (Table 4). Its gains are therefore not confined to compensating for a small actor. These results suggest that model capacity and feedback-based adaptation are complementary: larger backbones still benefit from execution evidence when selecting actions. Combining the actor's preferences with the critic's judgments remains more effective on average than relying on either alone, supporting SIPA's applicability across the tested model sizes.

Table 4: Scaling to Qwen3-8B and Qwen3-14B: accuracy (%) / successful-task mean steps.

	Qwen3-8B			**Qwen3-14B**		
Benchmark	Actor-only	Critic-only	SIPA	Actor-only	Critic-only	SIPA
BFCL	27.50 / 9.74	9.58 / 8.30	**28.75** / 10.43	32.08 / 10.73	17.08 / 23.85	**34.58** / 11.16
ToolSandbox	**20.55** / 5.00	7.31 / 3.44	20.09 / 6.64	23.29 / 3.35	12.79 / 3.50	**26.94** / 3.81
ALFWorld	63.43 / 33.67	52.24 / 29.81	**88.06** / 18.71	70.90 / 24.64	75.37 / 26.16	**92.54** / 20.38
BabyAI	81.58 / 17.48	78.95 / 23.90	**100.00** / 15.21	55.26 / 8.14	**100.00** / 8.55	97.37 / 9.35
ScienceWorld	4.30 / 45.18	2.34 / 21.17	**33.59** / 19.80	5.86 / 27.33	10.16 / 20.31	**39.45** / 16.50
Avg.	39.47	30.08	**54.10**	37.48	43.08	**58.18**

4.6 Parameter sensitivity

On ToolSandbox, we vary the nominal update strength $\eta = 1/\beta$, KL cap κ, and candidate ceiling G individually around $(32, 10, 6)$. The reference achieves the highest accuracy in each tested sweep (Figure 5). Increasing G beyond six or strengthening the update reduces accuracy, showing that more candidates or larger corrections alone do not guarantee better decisions. These local comparisons do not establish a universal optimum.

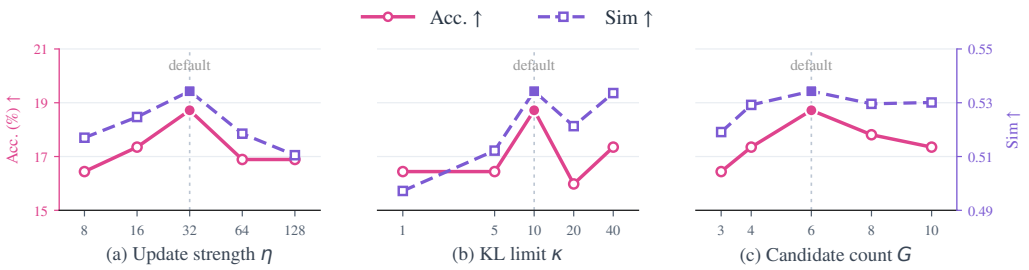

Figure 5: **ToolSandbox parameter sensitivity.** Acc. (pink) and similarity (purple), with 219 tasks per point. Each sweep varies one parameter around $(\eta, \kappa, G) = (32, 10, 6)$ (filled markers). η and κ use $\log_2$ spacing.

5 Related Work

Test-time adaptation of language agents. Recent test-time RL methods update model parameters using self-derived rewards or environment feedback (Zuo et al., 2026; Kim et al.; Liu et al., 2026). With frozen weights, Self-Refine and Reflexion use iterative feedback to refine outputs or guide subsequent attempts (Madaan et al., 2023; Shinn et al., 2023). More recent methods develop persistent context: ReasoningBank distills reasoning memories, ACE evolves playbooks, and ReMe refines procedural experience (Ouyang et al., 2026; Zhang et al., 2026; Cao et al., 2026). JitRL converts retrieved trajectories into action-logit adjustments (Li et al., 2026a). SIPA instead acquires feedback from isolated trials at the current state and updates the next-action distribution before commitment, without parameter updates or previously collected trajectories.

Search and speculative execution. Tree of Thoughts explores intermediate reasoning paths (Yao et al., 2023); recent agent search uses Monte Carlo planning guided by memory, reflection, or tool feedback (Shi et al., 2025; Yu et al., 2025; Yang et al., 2026). Speculative decoding accelerates generation through parallel draft verification while preserving the target distribution (Leviathan et al., 2023), with EAGLE-3 improving speculative drafting (Li et al., 2026b). At the action level, Speculative Actions executes predicted actions concurrently and commits only when they match the actor's choice (Ye et al., 2026). SIPA uses concurrent trials to obtain evidence for changing the actor's action distribution, rather than validating an unchanged choice. Its speculation supports adaptation before commitment while overlapping feedback acquisition with actor reasoning.

Feedback-guided policy optimization. Relative-entropy and trust-region methods constrain policy changes (Peters et al., 2010; Schulman et al., 2015; Abdolmaleki et al., 2018), providing foundations for stable policy optimization. More recently, InfAlign incorporates inference procedures into KL-regularized reward maximization (Balashankar et al., 2024). APA applies a KL constraint to the current action distribution with frozen weights. Its feedback design combines actor preferences with execution-based rejection probabilities, applying nonpositive corrections to candidate actions. Our analysis characterizes the resulting update's stability under feedback error and sufficient conditions for expected-return improvement.

6 Conclusion

We introduced SIPA, a test-time RL framework that adapts action selection using current-state execution feedback without parameter updates or pre-collected trajectories. Speculative trials overlap actor reasoning, and APA combines actor preferences with critic rejection feedback through a constrained KL-regularized update. We establish stability bounds under feedback error and sufficient conditions for expected-return improvement. Across five benchmarks, SIPA improves average accuracy by 22.1% relative to ReasoningBank, the strongest baseline at 4B, while gains over Actor-only persist from 4B to 14B. Ablations support execution evidence and asymmetric feedback, establishing speculative execution as a mechanism for policy adaptation.

AI USE STATEMENT

We used AI tools to assist with drafting and polishing the manuscript and to retrieve and discover related work. The authors take responsibility for the final content.

ETHICS STATEMENT

This work introduces SIPA, a test-time adaptation framework for language agents. Our experiments use existing benchmarks for tool use and embodied instruction following, with speculative actions executed in isolated copies of the environment state. More capable agents may also amplify unintended or harmful actions. We discourage misuse and emphasize that deployment requires appropriate access controls, reliable isolation of speculative execution, and human oversight for consequential actions.

REPRODUCIBILITY STATEMENT

We describe SIPA's candidate generation, execution feedback, and policy update in the main text, together with the benchmarks and evaluation metrics. Appendix B provides theoretical assumptions and proofs. Appendix C reports supplementary results, including ablations, parameter sensitivity, and interaction-budget experiments, to support replication and further analysis. Appendix D documents the algorithm, model and inference configurations, hyperparameters, execution protocol, and exact prompts.

REFERENCES

Abbas Abdolmaleki, Jost Tobias Springenberg, Yuval Tassa, Remi Munos, Nicolas Heess, and Martin Riedmiller. Maximum a posteriori policy optimisation. *arXiv preprint arXiv:1806.06920*, 2018.

Ananth Balashankar, Ziteng Sun, Jonathan Berant, Jacob Eisenstein, Michael Collins, Adrian Hutter, Jong Lee, Chirag Nagpal, Flavien Prost, Aradhana Sinha, et al. Infalign: Inference-aware language model alignment. *arXiv preprint arXiv:2412.19792*, 2024.

Anthony Brohan, Yevgen Chebotar, Chelsea Finn, Karol Hausman, Alexander Herzog, Daniel Ho, Julian Ibarz, Alex Irpan, Eric Jang, Ryan Julian, et al. Do as i can, not as i say: Grounding language in robotic affordances. In *Conference on robot learning*, pp. 287–318. Pmlr, 2023.

Zouying Cao, Jiaji Deng, Li Yu, Weikang Zhou, Zhaoyang Liu, Bolin Ding, and Hai Zhao. Remember me, refine me: A dynamic procedural memory framework for experience-driven agent evolution. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 16803–16822, 2026.

Thomas Carta, Clément Romac, Thomas Wolf, Sylvain Lamprier, Olivier Sigaud, and Pierre-Yves Oudeyer. Grounding large language models in interactive environments with online reinforcement learning. In *International conference on machine learning*, pp. 3676–3713. PMLR, 2023.

Junzhi Chen, Juhao Liang, and Benyou Wang. Smurfs: Multi-agent system using context-efficient dfsdt for tool planning. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 3281–3298, 2025.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638, 2025.

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.

Sungjune Kim, Gyeongrok Oh, Heeju Ko, Daehyun Ji, Dongwook Lee, Byung-Jun Lee, Sujin Jang, and Sangpil Kim. Test-time adaptation for online vision-language navigation with feedback-based reinforcement learning. In *Forty-second International Conference on Machine Learning*.

Yaniv Leviathan, Matan Kalman, and Yossi Matias. Fast inference from transformers via speculative decoding. In *International conference on machine learning*, pp. 19274–19286. PMLR, 2023.

Yibo Li, Zijie Lin, Ailin Deng, Xuan Zhang, Yufei He, Shuo Ji, Tri Cao, and Bryan Hooi. Just-in-time reinforcement learning: Continual learning in llm agents without gradient updates. *arXiv preprint arXiv:2601.18510*, 2026a.

Yuhui Li, Fangyun Wei, Chao Zhang, and Hongyang Zhang. Eagle-3: Scaling up inference acceleration of large language models via training-time test. *Advances in Neural Information Processing Systems*, 38:136737–136756, 2026b.

Changyu Liu, Yiyang Liu, Taowen Wang, Qiao Zhuang, James Chenhao Liang, Wenhao Yang, Renjing Xu, Qifan Wang, Dongfang Liu, and Cheng Han. On-the-fly vla adaptation via test-time reinforcement learning. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 40107–40125, 2026.

Jiarui Lu, Thomas Holleis, Yizhe Zhang, Bernhard Aumayer, Feng Nan, Haoping Bai, Shuang Ma, Shen Ma, Mengyu Li, Guoli Yin, et al. Toolsandbox: A stateful, conversational, interactive evaluation benchmark for llm tool use capabilities. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 1160–1183, 2025.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. Self-refine: Iterative refinement with self-feedback. *Advances in neural information processing systems*, 36:46534–46594, 2023.

Siru Ouyang, Jun Yan, I Hsu, Yanfei Chen, Ke Jiang, Zifeng Wang, Rujun Han, Long Le, Samira Daruki, Xiangru Tang, et al. Reasoningbank: Scaling agent self-evolving with reasoning memory. In *International Conference on Learning Representations*, volume 2026, pp. 94327–94354, 2026.

Shishir G Patil, Huanzhi Mao, Fanjia Yan, Charlie Cheng-Jie Ji, Vishnu Suresh, Ion Stoica, and Joseph E Gonzalez. The berkeley function calling leaderboard (bfcl): From tool use to agentic evaluation of large language models. In *Forty-second International Conference on Machine Learning*, 2025.

Jan Peters, Katharina Mulling, and Yasemin Altun. Relative entropy policy search. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 24, pp. 1607–1612, 2010.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International conference on machine learning*, pp. 1889–1897. Pmlr, 2015.

Zijing Shi, Meng Fang, and Ling Chen. Monte carlo planning with large language model for text-based game agents. In *International Conference on Learning Representations*, volume 2025, pp. 52924–52944, 2025.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *Advances in neural information processing systems*, 36:8634–8652, 2023.

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.

Ru Wang, Wei Huang, Qi Cao, Yusuke Iwasawa, Yutaka Matsuo, and Jiaxian Guo. Self-harmony: Learning to harmonize self-supervision and self-play in test-time reinforcement learning. In *International Conference on Learning Representations*, volume 2026, pp. 83112–83144, 2026.

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.

Shuo Yang, Caren Han, Yihao Ding, Shuhe Wang, and Eduard H Hovy. Tooltree: Efficient llm tool planning via dual-feedback monte carlo tree search and bidirectional pruning. In *International Conference on Learning Representations*, volume 2026, pp. 30750–30772, 2026.

Chaorui Yao, Yanxi Chen, Yuchang Sun, Yushuo Chen, Wenhao Zhang, Xuchen Pan, Yaliang Li, and Bolin Ding. Group-relative reinforce is secretly an off-policy algorithm: Demystifying some myths about grpo and its friends. In *International Conference on Learning Representations*, volume 2026, pp. 66179–66203, 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.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems*, 36:11809–11822, 2023.

Naimeng Ye, Arnav Ahuja, Georgios Liargkovas, Yunan Lu, Kostis Kaffes, and Tianyi Peng. Speculative actions: A lossless framework for faster ai agents. In *The Fourteenth International Conference on Learning Representations*, 2026.

Xiao Yu, Baolin Peng, Vineeth Vajipey, Hao Cheng, Michel Galley, Jianfeng Gao, and Zhou Yu. Exact: Teaching ai agents to explore with reflective-mcts and exploratory learning. In *International Conference on Learning Representations*, volume 2025, pp. 65157–65184, 2025.

Qizheng Zhang, Changran Hu, Shubhangi Upasani, Boyuan Ma, Fenglu Hong, Vamsidhar Kamanuru, Jay Rainton, Chen Wu, Mengmeng Ji, Hanchen Li, et al. Agentic context engineering: Evolving contexts for self-improving language models. In *International Conference on Learning Representations*, volume 2026, pp. 86069–86100, 2026.

Yuxin Zuo, Kaiyan Zhang, Li Sheng, Shang Qu, Ganqu Cui, Xuekai Zhu, Haozhan Li, Xinwei Long, Ermo Hua, Biqing Qi, et al. Ttrl: Test-time reinforcement learning. *Advances in Neural Information Processing Systems*, 38:131459–131483, 2026.

APPENDIX CONTENTS

A Notation **14**

B Test-time policy optimization and core proofs **15**
B.1 Test-time policy optimization and proof of Proposition 1 15
B.2 Proof of Theorem 2 16
B.3 Proof of Theorem 3 and expected-return corollary 16

C Supplementary Experimental Results **18**
C.1 Ablations and sensitivity 18
C.2 Mechanism diagnostics 19
C.3 Paired performance differences 22
C.4 Performance under interaction budgets 23

D Algorithm and implementation semantics **25**
D.1 BabyAI: advancing an ongoing trajectory 27
D.2 ALFWorld: directing search toward the target 27
D.3 ScienceWorld: revising a concentrated action preference 28
D.4 ScienceWorld: retaining an untrialed alternative 28
D.5 ScienceWorld: shifting from selection to inspection 29
D.6 ALFWorld: rejecting a valid but misdirected action 29
D.7 ALFWorld: ending repeated inspection 30
D.8 Prompts 30

E Supplementary theoretical analysis **31**
E.1 Sharpness of the stability bounds 31
E.2 Sensitivity of the rejection correction 32
E.3 Limits of the return guarantee 32

A NOTATION

Table 5 collects the notation for the method, core proofs, and retained supplementary analyses. In a single-decision calculation, we omit the step index and identify distributions on the actor candidates with vectors. The token distribution p_θ is distinct from the actor probability vector p; the scoring prompt C is distinct from the feasible set $\mathcal{C}$.

Table 5: Notation. Definitions with a step index also apply without that index at a fixed decision.

Symbol	Meaning
Environment and interaction	
$t, H, T \le H$	Decision index, horizon, and realized trajectory length.
s_t, h_t	Underlying environment state and observable interaction history.
a_t, o_t, r_t	Committed action, resulting observation, and task reward.
τ, π, $J(\pi)$	Trajectory, action policy, and expected task return $\mathbb{E}_{\tau\sim(\pi,\mathcal{E})}[\sum_{t=1}^{T} r_t]$.
$\mathcal{E}$	Environment execution kernel for (s_{t+1}, o_t) given (s_t, a_t).
s'_a, o_a, $\mathcal{O}$	Trial successor state, public trial observation, and evidence set $\mathcal{O} = \{(a, o_a) : a \in S\}$. Trial states are discarded.
Frozen model, candidates, and feedback	
θ, p_θ	Frozen model parameters and autoregressive token distribution.
a, $\lvert a\rvert$, a_k, $a_{<k}$	Complete serialized action, its token length, token k, and preceding tokens. The serialization includes its terminator.
C, C_{actor}, C_{critic}	Generic, actor, and critic scoring prompts.
$\ell_\theta(a; C)$	Complete-action log-likelihood $\sum_{k=1}^{\lvert a\rvert} \log p_\theta(a_k \mid C, a_{<k})$, without length normalization.
$\mathcal{T} > 0$, $T_C > 0$	Scoring temperature in Equation (1) and critic scoring temperature in Appendix E.2. The trajectory length is denoted by T.
$\mathcal{A}$, X, S	Generic finite scoring set; actor set $X = \mathcal{A}_{\text{actor}}$; proposer set $S = \mathcal{A}_{\text{proposer}}$.
a_i, b_j, K, M, G	Actor candidate i, proposal j, set sizes $K = \lvert X\rvert$, $M = \lvert S\rvert$, and candidate ceiling G; $1 \le K, M \le G$.
π_{actor}, p_i	Actor prior and its coordinate $p_i = \pi_{\text{actor}}(a_i) > 0$.
$\widetilde{\pi}_{\text{actor}}$, q_i	APA distribution and its coordinate $q_i = \widetilde{\pi}_{\text{actor}}(a_i)$.
π_{critic}, r_j, z_j	Rejection distribution on S, $r_j = \pi_{\text{critic}}(b_j)$, and critic log-score $z_j = \ell_\theta(b_j; C_{\text{critic}})$. Thus $r = \text{softmax}(z/T_C)$.
δ_{critic}, g_i	Applied correction $g_i = \delta_{\text{critic}}(a_i)$ from Equation (5). Unmatched actor candidates receive zero correction.
Local optimization and feedback error	
Δ_K, u	Probability simplex $\{u \in \mathbb{R}^K : u_i \ge 0,\ \sum_i u_i = 1\}$ and a candidate distribution on X.
β, κ	KL penalty weight $\beta > 0$ and hard KL budget $\kappa \ge 0$.
η, α, λ	Nominal step $\eta = \beta^{-1}$, effective step $0 \le \alpha \le \eta$, and KL multiplier $\lambda \ge 0$. For $\kappa > 0$, $\alpha = (\beta + \lambda)^{-1}$.
$\mathcal{C}$	Feasible set $\{u \in \Delta_K : D_{\text{KL}}(u\Vert p) \le \kappa\}$.
F_g, $\mathcal{L}_\star$	Local objective $F_g(u) = \langle u, g\rangle - \beta D_{\text{KL}}(u\Vert p)$ and reference objective $\mathcal{L}_\star = F_{g_\star}$.
$q(\alpha)$, $Z(\alpha)$, $\psi(\alpha)$	Exponential family $q_i(\alpha) = p_i e^{\alpha g_i}/Z(\alpha)$, normalizer $Z(\alpha) = \sum_i p_i e^{\alpha g_i}$, and $\psi(\alpha) = \log Z(\alpha)$.
$\delta_\star$, $g_\star$, $\pi_\star$, $q_\star$	Reference feedback $g_{\star,i} = \delta_\star(a_i)$ and its optimizer $q_{\star,i} = \pi_\star(a_i)$ with the same p, β, κ.
e, b, ε	Offset-invariant error $e = \min_b \lVert g - g_\star - b\mathbf{1}\rVert_\infty$; a minimizing offset b gives residual $\varepsilon = g - g_\star - b\mathbf{1}$ with $\lVert\varepsilon\rVert_\infty = e$.
$\widehat{A}_{g,i}$	Centered proxy advantage $g_i - \langle p, g\rangle$; centering does not change the APA optimizer.
$D_{\text{KL}}(u\Vert v)$, $d_{\text{TV}}(u, v)$	KL divergence $\sum_i u_i \log(u_i/v_i)$ and total variation $\frac{1}{2}\lVert u - v\rVert_1$.
$\langle u, v\rangle$, $\lVert v\rVert_1$, $\lVert v\rVert_\infty$, $\mathbf{1}$	Inner product $\sum_i u_i v_i$, sum of absolute coordinates, maximum absolute coordinate, and all-ones vector.
d, f_i, Z	Proof-local abbreviations $d = \lVert q_\star - q\rVert_1$, $f_i = e^{\alpha g_i}$, and $Z = Z(\alpha)$ at the selected step.
Reference values and sequential return	

Table 5 continued.

Symbol	Meaning
ξ_t	Full analytical decision state: environment state, history, time, retained agent state, and current actor reasoning, candidates, and prior.
ω_t	Auxiliary randomness and evidence from proposing, trial execution, and critic evaluation.
$p_t(a \mid \xi)$, $q_t(a \mid \xi, \omega)$	Actor prior and APA distribution conditional on the indicated decision state and evidence.
π_0, π_+	Actor-only reference and adapted policies: $\pi_0 = p_t$, $\pi_+ = \mathbb{E}_{\omega\mid\xi}[q_t]$. They share the actor candidate interface.
$\mathcal{K}_t$, $\mathcal{L}(\cdot \mid \cdot)$	Shared kernel for the committed reward and next full decision state; conditional probability law.
$Q_t^{\pi_0}$, $V_t^{\pi_0}$, Q_i	Reference action and state values; locally $Q_i = Q^{\pi_0}(\xi, a_i)$ and $V^{\pi_0}(\xi) = \langle p, Q\rangle$.
$A_i^{\pi_0}$, c	Reference advantage $Q_i - \langle p, Q\rangle$ and positive scale in the comparison $g_\star = Q/c$.
$\Delta(\xi)$, $\Delta_t(\xi, \omega)$, $\overline{\Delta}_t(\xi)$	Local value gain $\langle q - p, Q\rangle$, its evidence-conditioned form, and its conditional mean over ω.
$d_t^{\pi_+}$	Distribution of full decision states at step t under π_+.
Timing and interaction budgets	
T_{actor}, T_{aux}	Actor reasoning time and auxiliary-path time in the scheduling model.
T_{merge}	Residual scoring/integration time.
T_{SIPA}, $T_{\text{actor-only}}$	Decision latency for SIPA and Actor-only. These are elapsed times, not scoring temperatures.
N_i, L	Committed length of task i and a committed-step limit.

All logarithms are natural, with $0 \log 0 = 0$. Expectations and variances are written $\mathbb{E}$ and Var; a subscript identifies the sampling distribution. Primed quantities denote perturbed versions of the same objects.

B TEST-TIME POLICY OPTIMIZATION AND CORE PROOFS

We first connect APA to local policy improvement, then prove Proposition 1 and Theorems 2 and 3. Throughout the local analysis, the actor reasoning, nonempty support X, and prior p are fixed, with $p_i > 0$, finite feedback, $\beta > 0$, and $\kappa \geq 0$. Using Appendix A, write

$$F_g(u) = \langle u, g\rangle - \beta D_{\mathrm{KL}}(u\|p), \qquad \mathcal{C} = \{u \in \Delta_K : D_{\mathrm{KL}}(u\|p) \leq \kappa\}.$$

The applied signal is $g_i = \delta_{\text{critic}}(a_i)$. The optimization and stability proofs also hold for arbitrary finite signals, without independence or unbiasedness assumptions on their errors.

B.1 TEST-TIME POLICY OPTIMIZATION AND PROOF OF PROPOSITION 1

The policy-improvement objective. At a fixed decision state ξ, let $Q_i = Q^{\pi_0}(\xi, a_i)$ and $A_i^{\pi_0} = Q_i - \langle p, Q\rangle$. Replacing p by u for this decision changes reference continuation value by $\langle u, A^{\pi_0}\rangle = \langle u - p, Q\rangle$. APA uses execution-based feedback in place of these unknown advantages. Define the centered proxy

$$\widehat{A}_{g,i} = g_i - \langle p, g\rangle.$$

For every $u \in \mathcal{C}$,

$$\langle u, \widehat{A}_g\rangle - \beta D_{\mathrm{KL}}(u\|p) = F_g(u) - \langle p, g\rangle. \tag{14}$$

The subtracted baseline is independent of u. Thus Equation (6) is exactly a KL-regularized local policy-improvement problem with a proxy advantage; centering is an analytical identity, not an extra algorithmic step. If $g = Q/c + b\mathbf{1}$ for $c > 0$, then $\widehat{A}_g = A^{\pi_0}/c$. In general, the proxy need not equal the true advantage, and Theorem 3 specifies the resulting return conditions.

This gives SIPA its test-time RL interpretation: current-state execution feedback enters a policy-improvement objective that is solved directly over action probabilities before commitment, with θ frozen. Group-relative feedback motivates APA, but its independent proposer, rejection gaps, and exact candidate-level KL update are not an on-policy GRPO gradient estimator.

Proof of Proposition 1. If $\kappa = 0$, nonnegativity of KL gives $\mathcal{C} = \{p\}$, so $q = p$ and $\alpha = 0$. Suppose $\kappa > 0$. For $0 \le \alpha \le \beta^{-1}$, define

$$q_i(\alpha) = \frac{p_i e^{\alpha g_i}}{Z(\alpha)}, \qquad Z(\alpha) = \sum_i p_i e^{\alpha g_i}, \qquad \psi(\alpha) = \log Z(\alpha).$$

These distributions are positive. Since $\psi' = \mathbb{E}_{q(\alpha)}[g]$ and $\psi'' = \mathrm{Var}_{q(\alpha)}(g)$,

$$D_{\mathrm{KL}}(q(\alpha)\|p) = \alpha\psi'(\alpha) - \psi(\alpha), \qquad \frac{d}{d\alpha} D_{\mathrm{KL}}(q(\alpha)\|p) = \alpha \,\mathrm{Var}_{q(\alpha)}(g) \ge 0. \tag{15}$$

Use $\alpha = \beta^{-1}$ if it satisfies the budget. Otherwise, g is nonconstant and the KL increases continuously and strictly for $\alpha > 0$, starting at zero. There is therefore a unique $\alpha \in (0, \beta^{-1})$ attaining κ, found by bisection. Constant feedback, including $K = 1$, gives $q = p$ at the nominal step.

For the selected $\alpha > 0$, set $q = q(\alpha)$ and $\lambda = \alpha^{-1} - \beta$. Then $\lambda \ge 0$ and $\lambda(D_{\mathrm{KL}}(q\|p) - \kappa) = 0$. Substituting $\log(q_i/p_i) = \alpha g_i - \log Z(\alpha)$ yields, for every feasible u,

$$\begin{aligned} F_g(q) - F_g(u) &= (\beta + \lambda) D_{\mathrm{KL}}(u\|q) + \lambda\big[D_{\mathrm{KL}}(q\|p) - D_{\mathrm{KL}}(u\|p)\big] \\ &\ge (\beta + \lambda) D_{\mathrm{KL}}(u\|q). \end{aligned} \tag{16}$$

The inequality follows from feasibility and complementary slackness. It proves optimality, and equality of the objectives requires $u = q$, proving uniqueness. The equality holds on the full simplex and shows that q maximizes the associated Lagrangian. Together with feasibility and complementary slackness, this certifies λ as a KL-constraint multiplier. Hence $\alpha = (\beta + \lambda)^{-1}$ and $q = \mathrm{softmax}(\log p + \alpha g)$, as claimed. □

B.2 Proof of Theorem 2

Proof. Let q and $q_\star$ maximize F_g and $F_{g_\star}$ on the same $\mathcal{C}$. The case $\kappa = 0$ is immediate. For $\kappa > 0$, Equation (16) implies

$$F_g(q) - F_g(u) \ge \beta D_{\mathrm{KL}}(u\|q), \qquad u \in \mathcal{C}, \tag{17}$$

and likewise for $(g_\star, q_\star)$. Put $v = g - g_\star$. The offset-invariant error is $e = (\max_i v_i - \min_i v_i)/2$, attained at $b = (\max_i v_i + \min_i v_i)/2$. This common offset cancels against $q_\star - q$. With $d = \|q_\star - q\|_1$, Equation (17) and Pinsker's inequality give

$$\begin{aligned} F_{g_\star}(q_\star) - F_{g_\star}(q) &= F_g(q_\star) - F_g(q) + \langle q_\star - q, g_\star - g\rangle \\ &\le -\beta D_{\mathrm{KL}}(q_\star\|q) + ed \le -\frac{\beta}{2} d^2 + ed \le \frac{e^2}{2\beta}. \end{aligned} \tag{18}$$

This is Equation (9). Applying Equation (17) to $(g_\star, q_\star)$ also gives

$$\beta D_{\mathrm{KL}}(q\|q_\star) \le F_{g_\star}(q_\star) - F_{g_\star}(q) \le \frac{e^2}{2\beta}.$$

Exchanging the two signals preserves e and bounds the reverse KL identically. Finally, Pinsker gives $d_{\mathrm{TV}}(q_\star, q) \le \sqrt{D_{\mathrm{KL}}(q\|q_\star)/2} \le e/(2\beta)$, proving Equations (10) and (11). □

The two optimizers share p, X, β, and κ, but may have different multipliers and effective steps. The result controls sensitivity to feedback error, not its alignment with task return.

B.3 Proof of Theorem 3 and expected-return corollary

Proof of Theorem 3. Fix ξ and the realized correction g. If $\alpha = 0$, $q = p$ and $\Delta(\xi) = 0$. Otherwise set $f_i = e^{\alpha g_i}$ and $Z = \sum_i p_i f_i$. Expanding the double sum gives

$$\Delta(\xi) = \frac{\mathbb{E}_p[fQ] - \mathbb{E}_p[f]\mathbb{E}_p[Q]}{Z} = \frac{1}{2Z} \sum_{i,j} p_i p_j (f_i - f_j)(Q_i - Q_j). \tag{19}$$

If $(g_i - g_j)(Q_i - Q_j) \ge 0$ for all i, j, monotonicity of the exponential makes every summand nonnegative. This proves the ordering claim.

For the error bound, take $g_\star = Q/c$, $c > 0$, and write $g = Q/c + b\mathbf{1} + \varepsilon$ with $\|\varepsilon\|_\infty \le e$. Since $\langle q - p, \mathbf{1}\rangle = 0$,

$$\Delta(\xi) = c\langle q - p, g - \varepsilon\rangle \ge c\big[\langle q - p, g\rangle - e\|q - p\|_1\big].$$

The exponential form gives the identity

$$D_{\mathrm{KL}}(q\|p) + D_{\mathrm{KL}}(p\|q) = \sum_i (q_i - p_i)\log\frac{q_i}{p_i} = \alpha\langle q - p, g\rangle. \tag{20}$$

Substitution proves Equation (13), including when the KL constraint is active. □

Both the ordering condition and the error bound concern all actor candidates, including untried actions assigned zero correction. Accurate one-step execution does not by itself establish either condition for reference continuation values.

From local gains to task return. Let ξ_t include (s_t, h_t), time, retained agent state, and the current actor reasoning, candidates, and prior, so that it is sufficient to describe future evolution. This analytical conditioning does not reveal hidden state to the models. Let ω_t collect the auxiliary randomness and evidence. The induced policies are

$$\pi_0(a \mid \xi) = p_t(a \mid \xi), \qquad \pi_+(a \mid \xi) = \mathbb{E}_{\omega|\xi}[q_t(a \mid \xi, \omega)].$$

The reference is the Actor-only baseline: it retains the same actor reasoning, candidate generation, complete-action probability scoring, and resampling, with APA disabled.

Assume a common initial distribution, a finite horizon H, and integrability of all terms below. Both policies use a shared committed-transition kernel satisfying

$$\mathcal{L}(r_t, \xi_{t+1} \mid \xi_t = \xi, \omega_t = \omega, a_t = a) = \mathcal{K}_t(\cdot \mid \xi, a). \tag{21}$$

The kernel includes the next history, retained state, and actor quantities. Isolated trials, fresh committed randomness, and trial evidence that does not otherwise alter retained state suffice for this condition. Selecting a favorable trial outcome as the committed successor does not generally satisfy it.

Set $V^{\pi_0}_{H+1} = 0$, padding early termination with zero-reward absorbing transitions. The reference values satisfy $Q^{\pi_0}_t(\xi, a) = \mathbb{E}_{\mathcal{K}_t}[r_t + V^{\pi_0}_{t+1}(\xi_{t+1}) \mid \xi, a]$ and $V^{\pi_0}_t(\xi) = \sum_a p_t(a \mid \xi) Q^{\pi_0}_t(\xi, a)$. Write

$$\overline{\Delta}_t(\xi) = \mathbb{E}_{\omega|\xi}\left[\sum_a \big(q_t(a \mid \xi, \omega) - p_t(a \mid \xi)\big) Q^{\pi_0}_t(\xi, a)\right].$$

Corollary 4 (Expected task return)**.** *Under the preceding assumptions,*

$$J(\pi_+) - J(\pi_0) = \sum_{t=1}^{H} \mathbb{E}_{\xi \sim d^{\pi_+}_t}[\overline{\Delta}_t(\xi)]. \tag{22}$$

Consequently, nonnegative conditional mean gain at $d^{\pi_+}_t$-almost every state, for each t, implies $J(\pi_+) \ge J(\pi_0)$.

Proof. The shared initial law gives $\mathbb{E}[V^{\pi_0}_1(\xi_1)] = J(\pi_0)$. Telescoping and Equation (21) yield

$$\begin{aligned} J(\pi_+) - J(\pi_0) &= \mathbb{E}_{\pi_+}\left[\sum_{t=1}^{H}\big(r_t + V^{\pi_0}_{t+1}(\xi_{t+1}) - V^{\pi_0}_t(\xi_t)\big)\right] \\ &= \sum_{t=1}^{H} \mathbb{E}_{\xi \sim d^{\pi_+}_t}\left[\sum_a \pi_+(a \mid \xi) Q^{\pi_0}_t(\xi, a) - V^{\pi_0}_t(\xi)\right]. \end{aligned}$$

Substituting $\pi_+ = \mathbb{E}_{\omega|\xi}[q_t]$ identifies the bracket with $\overline{\Delta}_t(\xi)$. Nonnegativity gives the conclusion. □

Theorem 3 supplies sufficient conditions for the local gains in this identity. They need to hold on states visited by the adapted policy; nonnegative gain for every evidence realization is sufficient, but nonnegative conditional mean gain is enough.

C SUPPLEMENTARY EXPERIMENTAL RESULTS

This appendix presents additional experimental results supporting Section 4.

C.1 ABLATIONS AND SENSITIVITY

We evaluate all ablations on the same five benchmarks as full SIPA, keeping the task sets, backbone, resource limits, and stopping rules fixed. Table 6 reports strict task accuracy and mean committed interaction steps over successful tasks. Avg. summarizes accuracy across benchmarks; successful-task steps are reported separately for each benchmark. TS, SW, and ALF denote ToolSandbox, ScienceWorld, and ALFWorld, respectively.

The four variants isolate the roles of adaptation, feedback direction, the KL constraint, and execution evidence. **w/o APA** is the Actor-only baseline from Table 1, with the same reported accuracy and successful-task steps. It retains SIPA's actor candidate generation, complete-action probability scoring, and resampling but sets the effective update strength to $\alpha = 0$, so actions are sampled from the unmodified actor prior. **w/o negative probe** retains the actor prior and trial observations but uses a positive critic probe to favor supported candidates. **w/o KL** removes the hard KL budget while retaining the KL penalty. **w/o exec.** preserves the candidate interface but withholds trial observations from the critic. These comparisons complement the timing study of overlapping evidence acquisition with actor reasoning.

Ablation outcomes. Full SIPA reaches 48.64% average accuracy in Table 6. Disabling APA reduces this average by 11.60 percentage points, compared with 5.73 points when withholding execution observations, 4.42 when replacing the negative probe, and 2.52 when removing the hard KL cap. The effects are not uniform across benchmarks: removing the cap reduces accuracy on four benchmarks but raises BabyAI accuracy from 86.84% to 89.47%. Thus, the cap improves the benchmark average without being optimal for every setting. Successful-task steps also require joint interpretation with accuracy. For example, removing the cap lowers ToolSandbox steps from 8.51 to 7.44 but reduces accuracy from 18.72% to 16.44%. Since each step mean conditions on that variant's own successful tasks, a lower value alone does not establish faster completion of the same tasks.

Table 6: Qwen3-4B ablations: strict Acc. (%) / successful-task mean interaction steps. Avg. reports mean Acc.; bold marks the best accuracy. w/o APA is Actor-only; w/o negative probe uses a positive critic probe.

Method	TS	SW	BFCL	ALF	BabyAI	Avg.
Full SIPA	**18.72** / 8.51	**20.31** / 19.83	**30.00** / 10.28	**87.31** / 24.24	86.84 / 15.94	**48.64**
w/o APA	15.98 / 24.91	16.02 / 21.66	19.17 / 9.65	76.12 / 25.00	57.89 / 19.00	37.04
w/o negative probe	12.33 / 9.26	17.58 / 19.60	27.92 / 10.30	84.33 / 21.53	78.95 / 18.43	44.22
w/o KL	16.44 / 7.44	14.45 / 23.95	26.67 / 10.03	83.58 / 22.35	**89.47** / 15.74	46.12
w/o exec.	13.70 / 7.87	13.67 / 22.94	29.17 / 10.64	84.33 / 22.97	73.68 / 16.79	42.91

Parameter sensitivity. The sensitivity study uses all 219 ToolSandbox tasks and varies one parameter at a time around $(\eta, \kappa, G) = (32, 10, 6)$. The grids are $\eta \in \{8, 16, 32, 64, 128\}$, $\kappa \in \{1, 5, 10, 20, 40\}$, and $G \in \{3, 4, 6, 8, 10\}$; the other two parameters remain fixed. Each panel in Figure 5 shows strict accuracy and mean similarity on separately labeled axes, with consistent metric ranges across panels. Sim is the mean official ToolSandbox score in $[0, 1]$, measuring trajectory alignment with task milestones and assigning zero when a minefield is triggered, thus capturing partial progress beyond binary success (Lu et al., 2025). Update strength and KL budget use logarithmic spacing; candidate count uses linear spacing. Table 7 lists the exact performance values plotted in these sweeps.

The reference configuration achieves 41 strict successes, whereas the alternatives achieve 35–39. Accuracy and partial progress need not move together: increasing κ from 10 to 40 leaves Sim nearly unchanged (0.5343 versus 0.5336) but reduces strict successes from 41 to 38. Likewise,

$\eta = 64$ and $\eta = 128$ both solve 37 tasks, although Sim falls from 0.5185 to 0.5105. Increasing the candidate ceiling from six to eight or ten yields 39 or 38 successes, respectively, so a larger candidate allowance does not by itself improve performance. These one-parameter comparisons identify a useful reference within the tested grid; they do not measure interactions between parameters or establish a global optimum.

Table 7: Parameter sensitivity on ToolSandbox. Each sweep varies one parameter with the others fixed as indicated. n denotes strict successes out of 219 tasks; Acc. is reported in percent. Shading marks the shared reference configuration, and bold marks the best Acc. and Sim in each sweep.

Update strength η ($\kappa = 10, G = 6$)				**KL budget** κ ($\eta = 32, G = 6$)				**Candidate ceiling** G ($\eta = 32, \kappa = 10$)			
Value	n	Acc. ↑	Sim ↑	Value	n	Acc. ↑	Sim ↑	Value	n	Acc. ↑	Sim ↑
8	36	16.44	0.5170	1	36	16.44	0.4972	3	36	16.44	0.5191
16	38	17.35	0.5247	5	36	16.44	0.5122	4	38	17.35	0.5292
32	41	**18.72**	**0.5343**	10	41	**18.72**	**0.5343**	6	41	**18.72**	**0.5343**
64	37	16.89	0.5185	20	35	15.98	0.5213	8	39	17.81	0.5296
128	37	16.89	0.5105	40	38	17.35	0.5336	10	38	17.35	0.5301

C.2 Mechanism diagnostics

We examine how APA modifies the actor's distribution in an additional Qwen3-4B evaluation across the five benchmarks. Table 8 summarizes coverage and constraint activity, while Figure 6 summarizes the frequency of sampled-action changes, the actor probability mass covered by speculative trials, and the magnitude of policy adaptation. A sampled-action change occurs when sampling from the actor prior and the adapted distribution with the same uniform random draw and candidate ordering selects different actions. Trial coverage spans 28.52–54.82% of actor mass, yet sampled actions change in only 0.17–15.53% of decisions. Coverage therefore does not imply intervention: the effect of feedback also depends on the actor prior and the relative corrections across candidates. BabyAI has the highest action-change rate (15.53%), whereas BFCL changes only 0.17% of sampled actions. Average KL utilization remains between 2.35% and 11.94% of the configured budget. The entropy panels further show that adaptation does not uniformly sharpen the policy: entropy increases on four benchmarks and decreases slightly on BabyAI, consistent with feedback redistributing probability according to the observed evidence.

Figure 7 shows that these small averages coexist with occasional substantial updates. The policy is effectively unchanged in 72.68–91.53% of decisions, giving a zero median total-variation (TV) distance on every benchmark. The upper tail is markedly different: the 99th-percentile normalized KL reaches the constraint boundary throughout, and the KL bound actively limits 1.27–11.09% of decisions. ALFWorld has the highest constraint activity (11.09%), while BabyAI has the largest 90th-percentile TV distance (0.94). Together, the quantiles and cumulative distributions characterize APA as an intermittent intervention: most decisions retain the actor distribution, while a smaller subset receives substantial probability reallocation. The KL constraint controls the magnitude of those updates; their usefulness still depends on the execution feedback.

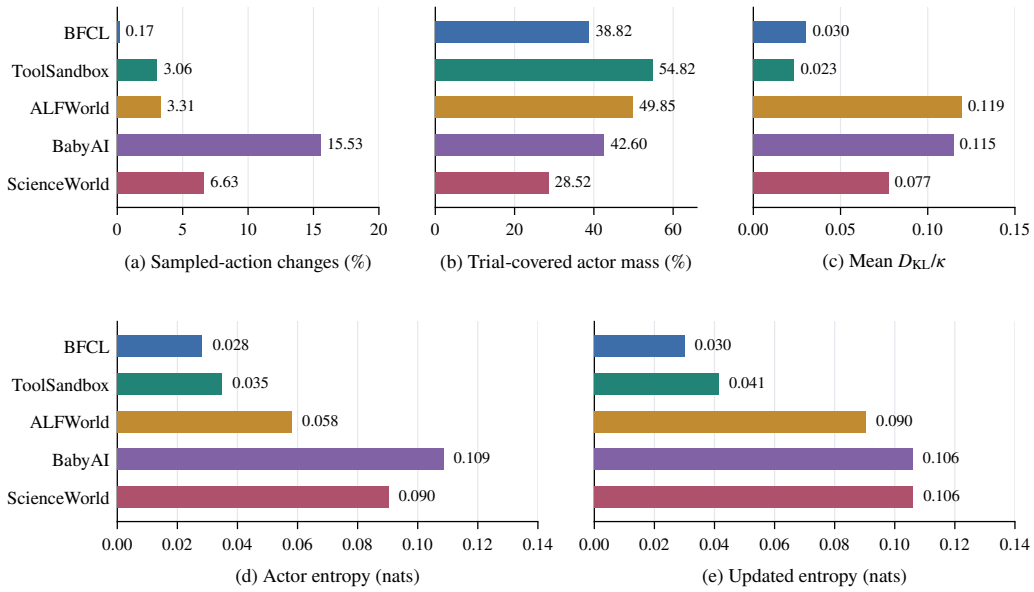

Figure 6: **How APA intervenes.** Decision-level means from 18,383 valid probability records across 887 tasks. Panels show (a) recorded sampled-action changes, (b) actor probability mass assigned to trialed actions, (c) realized KL relative to its budget, and (d–e) entropy before and after adaptation. Action changes measure intervention, not whether an action was corrected.

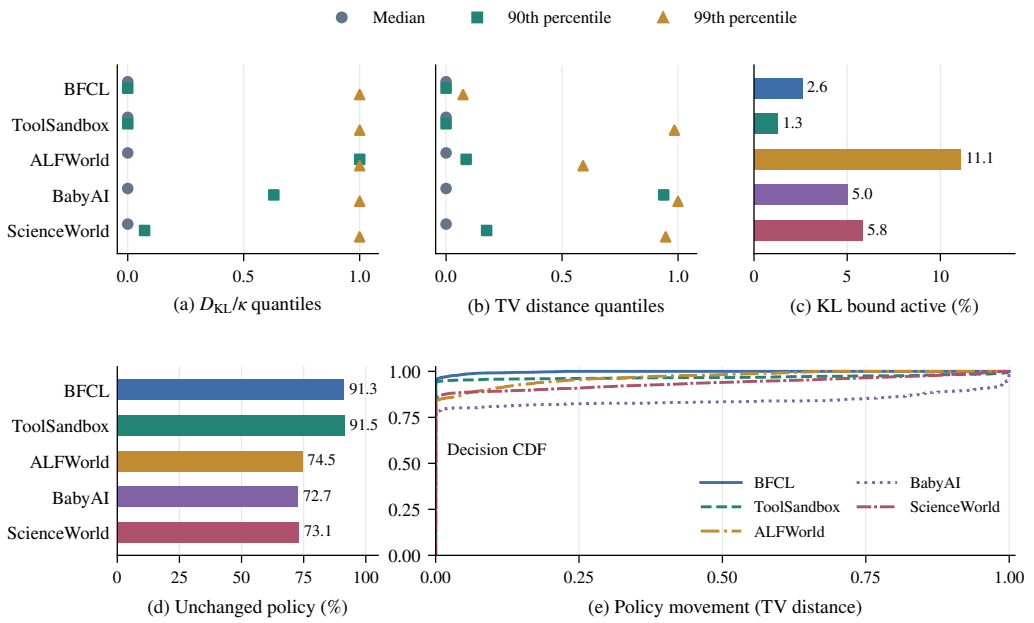

Figure 7: **Constraint activity and policy movement.** (a–b) Median, 90th, and 99th percentiles of normalized KL and TV distance. (c) Fraction of decisions where the effective step is reduced at the KL boundary. (d) Fraction with TV distance below 10^{-10}. (e) Empirical cumulative distributions of TV distance, retaining the mass at zero and the full upper tail. All summaries weight recorded decisions equally.

Table 8: Mechanism diagnostics on Qwen3-4B. Active denotes a binding KL constraint; unchanged denotes TV distance below 10^{-10}. Trial mass is the actor probability mass covered by speculative execution. Decision-level summaries use valid probability records.

Benchmark	Tasks	Valid decisions	Active (%)	Unchanged (%)	Trial mass
BFCL	240	3028	2.58	91.28	0.39
ToolSandbox	219	2291	1.27	91.53	0.55
ALFWorld	134	5077	11.09	74.49	0.50
BabyAI	38	721	4.99	72.68	0.43
ScienceWorld	256	7266	5.81	73.08	0.29

Coverage and configuration. Table 8 contains 18,383 valid probability records from 887 tasks; 28 of the 18,411 recorded decisions are excluded because their probability records fail validation. Means and quantiles weight valid decisions equally, so longer trajectories contribute more observations. The diagnostic runs use $(\eta, \kappa, G) = (16, 0.1, 6)$ for BFCL, $(32, 10, 6)$ for ToolSandbox, $(64, 1, 4)$ for ALFWorld, $(32, 10, 2)$ for BabyAI, and $(64, 4, 4)$ for ScienceWorld. These additional runs characterize the recorded behavior of APA under their respective configurations, rather than reproduce the main comparison under a common setting. In particular, differences in constraint activity across benchmarks reflect both the encountered decisions and the configured update strength, KL budget, and candidate ceiling.

Observed execution errors. Table 9 examines decisions for which both the unadapted actor sample and the selected action have trial evidence. Errors here are environment execution errors; benchmark scores and score-derived task-failure labels are excluded, including in ScienceWorld. Within this subset, adaptation removes 22 recorded execution errors and introduces two, with most removals occurring in ScienceWorld. ALFWorld and BabyAI contain no execution errors in the observed pairs, showing that their policy changes concern choices among executable actions. This distinction matters: avoiding an execution error is only one possible benefit of feedback, and an executable action can still fail to advance the task.

The paired evidence covers 25.08–53.38% of valid decisions, depending on the benchmark. The error counts therefore describe only states where both sampled actions were tested, not an error-correction rate over all decisions. The remaining errors are also informative: on BFCL, both samples produce execution errors in 337 of the 1,167 observed pairs, while only one pair changes from an erroneous to a valid action. Speculative evidence thus does not ensure that the selected action is executable. Conversely, the zero-error ALFWorld and BabyAI pairs leave task relevance and progress as distinct dimensions that execution validity alone cannot resolve.

Table 9: Environment-execution error transitions for observed action pairs. Each pair compares the actor sample before adaptation with the selected action at the same state, using trial evidence for both. Coverage is relative to valid decisions. Execution validity does not imply task correctness.

Benchmark	Observed pairs	Both valid	Error removed	Error introduced	Both errors	Coverage (%)
BFCL	1167	829	1	0	337	38.54
ToolSandbox	1223	1205	5	0	13	53.38
ALFWorld	2445	2445	0	0	0	48.16
BabyAI	255	255	0	0	0	35.37
ScienceWorld	1822	1759	16	2	45	25.08

C.3 PAIRED PERFORMANCE DIFFERENCES

Figure 8 compares SIPA with Actor-only and Critic-only at 8B and 14B. Table 10 gives the available success overlaps. Against Actor-only, accuracy differences on ALFWorld, BabyAI, and ScienceWorld are positive at both model scales. The largest gap is on BabyAI at 14B: 42.11 percentage points, with a 95% interval of [23.68, 57.89]. On ALFWorld at 14B, SIPA and Actor-only solve 124 and 95 of 134 tasks, respectively, giving a 21.64-point difference with a 95% interval of [13.43, 29.85]. The Actor-only comparisons on BFCL and ToolSandbox have intervals spanning zero, including the small negative difference on ToolSandbox at 8B (-0.46 points). Against Critic-only, all intervals lie above zero except BabyAI at 14B, where SIPA solves one fewer task (-2.63 points; interval $[-7.89, 0.00]$). The gains are thus strongest in the interactive environments, while combining the two signals does not guarantee improvement over the stronger standalone policy on every benchmark.

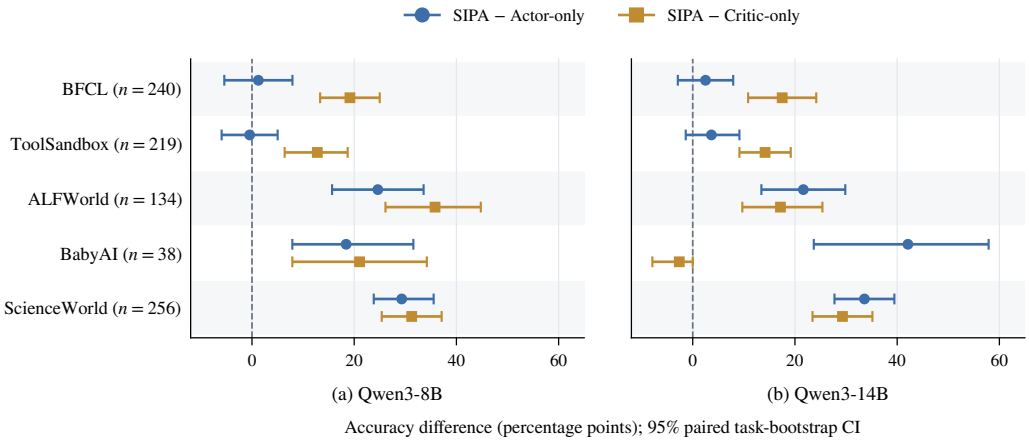

Figure 8: **Paired accuracy differences.** Points show SIPA minus each standalone control on the same task set; bars are pointwise 95% paired task-bootstrap confidence intervals. Positive values favor SIPA. Both panels share the same horizontal scale. Intervals describe task-sampling uncertainty within the evaluated runs, rather than variation across repeated runs.

Table 10: Paired task outcomes at 8B and 14B. Both and neither indicate joint success and failure; SIPA-only and control-only count discordant outcomes. Gains are accuracy differences in percentage points, with 95% paired task-bootstrap intervals.

Benchmark	Control	N	Both	SIPA only	Control only	Neither	Gain (pp)	95% CI
Qwen3-8B								
BFCL	Actor-only	240	35	34	31	140	1.25	[-5.42, 7.92]
BFCL	Critic-only	240	17	52	6	165	19.17	[13.33, 25.00]
ToolSandbox	Actor-only	219	27	17	18	157	-0.46	[-5.94, 5.02]
ToolSandbox	Critic-only	219	5	39	11	164	12.79	[6.39, 18.72]
ALFWorld	Actor-only	134	78	40	7	9	24.63	[15.67, 33.58]
ALFWorld	Critic-only	134	65	53	5	11	35.82	[26.12, 44.78]
BabyAI	Actor-only	38	31	7	0	0	18.42	[7.89, 31.58]
BabyAI	Critic-only	38	30	8	0	0	21.05	[7.89, 34.21]
ScienceWorld	Actor-only	256	8	78	3	167	29.30	[23.83, 35.55]
ScienceWorld	Critic-only	256	5	81	1	169	31.25	[25.39, 37.11]
Qwen3-14B								
BFCL	Actor-only	240	58	25	19	138	2.50	[-2.92, 7.92]
BFCL	Critic-only	240	27	56	14	143	17.50	[10.83, 24.17]
ToolSandbox	Actor-only	219	37	22	14	146	3.65	[-1.37, 9.13]
ToolSandbox	Critic-only	219	26	33	2	158	14.16	[9.13, 19.18]
ALFWorld	Actor-only	134	90	34	5	5	21.64	[13.43, 29.85]
ALFWorld	Critic-only	134	96	28	5	5	17.16	[9.70, 25.37]
BabyAI	Actor-only	38	20	17	1	0	42.11	[23.68, 57.89]
BabyAI	Critic-only	38	37	0	1	0	-2.63	[-7.89, 0.00]
ScienceWorld	Actor-only	256	14	87	1	154	33.59	[27.73, 39.45]
ScienceWorld	Critic-only	256	21	80	5	150	29.30	[23.44, 35.16]

Table 10 separates net gains from changes in the solved-task set. The accuracy difference is the number of SIPA-only successes minus control-only successes, divided by the task count. At 8B, SIPA gains 34 BFCL tasks relative to Actor-only but loses 31, leaving a net advantage of only three tasks (1.25 points). On ALFWorld, the corresponding counts are 40 and seven, yielding 24.63 points; on ScienceWorld, they are 78 and three, yielding 29.30 points. The larger aggregate gains therefore coincide with a stronger imbalance between newly solved and lost tasks, rather than an absence of regressions. BabyAI also warrants attention to its small denominator: with 38 tasks, one outcome changes accuracy by 2.63 points. The reported intervals come from 4,000 bootstrap resamples of paired task outcomes within each comparison; they do not quantify variation across random seeds or provide a simultaneous guarantee over all comparisons.

C.4 Performance under interaction budgets

Figure 9 compares accuracy at committed-action limits of 10, 20, 30, 50, and 100. On BFCL, SIPA rises from 21.67% at limit 10 to 30.00% at limit 100, with most gains achieved by limit 30, and remains above all evaluated baselines at every budget. BabyAI benefits substantially from additional interaction: SIPA increases from 34.21% to 86.84% over limits 10–100. Its advantage over Actor-only grows from 10.53 percentage points at limit 10 to 36.84 at limit 50, then narrows to 28.95 at limit 100 as Actor-only improves. Reflexion fluctuates with the budget, while SMURFS improves more steadily; both remain below SIPA throughout. Table 11 reports their exact success counts and successful-task steps.

On ToolSandbox, SIPA increases from 15.53% at limit 10 to 18.72% at limits 50 and 100. It trails Actor-only at limits 10 and 20, overtakes it from limit 30 onward, and leads by 2.74 percentage points at limit 100. Reflexion achieves the highest accuracy at limits 20 and 30, whereas SIPA leads at limits 50 and 100. On ALFWorld, SIPA trails Actor-only at limits 10 and 20 but overtakes it from limit 30 onward, reaching 87.31% at limit 100 versus 76.12% for Actor-only, 72.39% for Reflexion, and 27.61% for SMURFS. Additional interaction therefore benefits the methods differently across environments; the independent budget-specific runs need not produce monotonic curves.

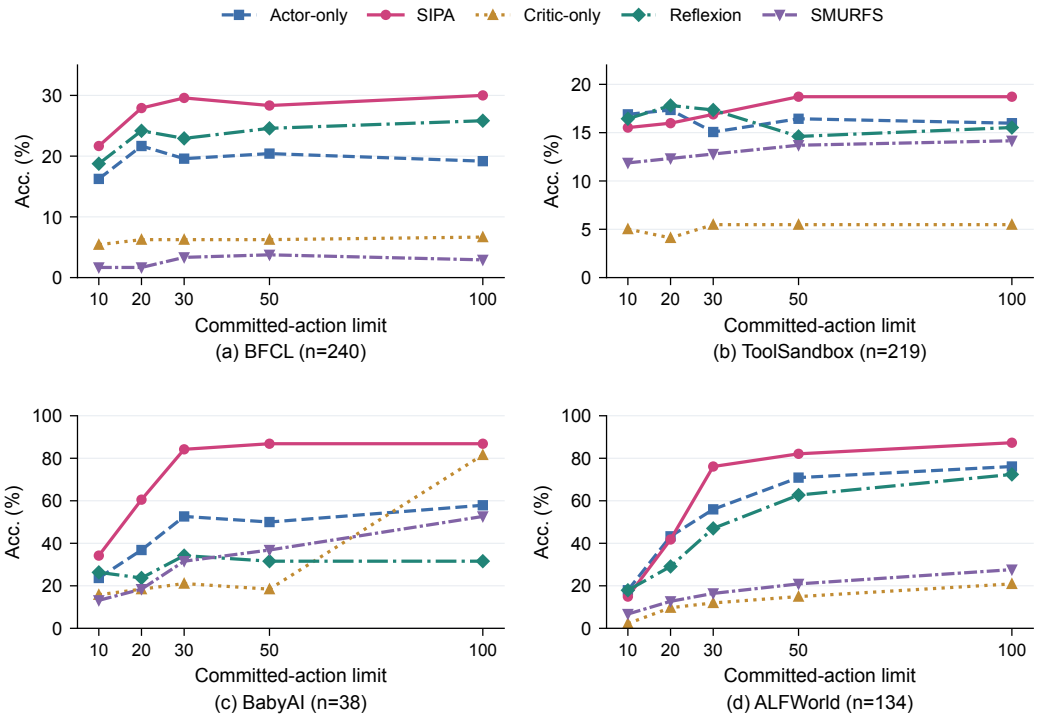

Figure 9: **Accuracy under interaction budgets on Qwen3-4B.** Each point evaluates the full benchmark at the indicated limit. All benchmarks count committed actions. Budget settings are evaluated in separate runs, rather than by truncating a shared trajectory. Vertical scales differ across benchmarks.

Table 11: BabyAI budget results for Reflexion and SMURFS ($N = 38$). Steps average strictly successful tasks.

Limit	Reflexion			SMURFS		
	Successes	Acc. (%)	Steps	Successes	Acc. (%)	Steps
10	10/38	26.32	4.70	5/38	13.16	4.60
20	9/38	23.68	5.22	7/38	18.42	8.43
30	13/38	34.21	7.54	12/38	31.58	15.75
50	12/38	31.58	8.83	14/38	36.84	19.93
100	12/38	31.58	13.92	20/38	52.63	20.75

Table 11 adds the success counts and conditional step means behind the BabyAI baseline curves. SMURFS increases from five successes at limit 10 to 20 at limit 100, while its mean steps among successes rise from 4.60 to 20.75. Reflexion instead has 12 successes at both limits 50 and 100, with mean steps increasing from 8.83 to 13.92. The larger allowance therefore produces additional successes for SMURFS in these runs, but no net increase for Reflexion over the latter range. Equal success counts do not imply identical solved-task sets, and the changing success-conditioned means should not be read as latency changes on a fixed set of tasks.

D ALGORITHM AND IMPLEMENTATION SEMANTICS

Algorithm 1 uses the role distributions and correction defined in Section 3.2. The nominal update step is $\eta = 1/\beta$.

Implementation details. We implement SIPA with a shared, frozen Qwen3-4B backbone for the Actor, Proposer, and Critic. Inference is served through vLLM in FP16 on eight NVIDIA A100 GPUs, with a maximum context length of 32,768 tokens. Independent evaluation episodes are distributed across inference workers. The Actor uses native thinking with a 1,024-token reasoning budget, while the Proposer and Critic disable native thinking and generate explicit assessments capped at 128 tokens. All baselines enable native thinking. All SIPA roles use a generation temperature of 0.7 and share the task context and available-action schema. The context retains the original request, extracted earlier context, and the 16 most recent messages.

Unless otherwise specified, we use a nominal update strength of $\eta = \beta^{-1} = 32$, a KL budget of $\kappa = 10$, and a candidate ceiling of $G = 6$. The Actor and Proposer independently generate nonempty candidate sets through schema-constrained decoding, followed by validation and deduplication. Each set contains at most G distinct actions; neither role is required to fill the candidate budget. We allocate 256 tokens per action and $256G + 32$ tokens per candidate array. Actions are serialized as canonical JSON with sorted keys and a newline terminator. Within each role, complete actions are scored from a common probe prefix by summing token log-probabilities without length normalization. The Actor probe asks which action to take, whereas the Critic probe asks which action to avoid given the trial feedback. The Actor uses a scoring temperature of 1 and the Critic uses a scoring temperature of 2, with normalization over their respective candidate sets.

The Actor path runs concurrently with proposal generation, speculative execution, and Critic assessment. Each proposal is tested once in an isolated copy of the current environment state. APA applies corrections only to exactly matching actions, including their arguments, and normalizes the updated distribution over the entire Actor candidate set. If the nominal update exceeds the KL budget, we use 60 bisection iterations to find the largest feasible step in $[0, \eta]$. The next action is sampled from the updated distribution and executed afresh in the ongoing environment; speculative branch states are discarded.

Execution feedback. Trial feedback contains public observations and, where provided by the interface, termination status and execution-error indicators. The critic does not receive benchmark evaluation scores or success/failure labels derived from them. In ScienceWorld, `error` denotes an environment execution error and is never constructed from a task-failure score. Termination alone does not establish success. Scores and final-success annotations in the case studies report evaluation outcomes, not additional critic inputs.

Algorithm 1 Speculative Inference-Time Policy Adaptation (one decision)

Require: Frozen model p_θ; history h_t; environment state s_t; shared action interface; candidate ceiling $G \geq 1$; KL penalty $\beta > 0$ and budget $\kappa \geq 0$
Ensure: Committed action a_t, next state s_{t+1}, and updated history h_{t+1}

Snapshot s_t; hold the shared history and action interface fixed
in parallel
  Actor path: selection preferences
    $X \leftarrow \mathrm{Generate}(h_t; \mathrm{actor})$
    Form C_{actor} from h_t, X, and actor reasoning (positive probe)
    $p \leftarrow \pi(\cdot \mid C_{\mathrm{actor}}, X)$
  Proposer–critic path: execution evidence
    $S \leftarrow \mathrm{Generate}(h_t; \mathrm{proposer})$
    for all $b \in S$ **in parallel do**
      $(s'_b, o_b) \sim \mathcal{E}(\cdot \mid s_t, b)$ ▷ Separate copy of the snapshot
    end for
    $\mathcal{O} \leftarrow \{(b, o_b) : b \in S\}$; discard all trial states s'_b
    Briefly assess $\mathcal{O}$ given h_t and S; form C_{critic} (negative probe)
    $r \leftarrow \pi(\cdot \mid C_{\mathrm{critic}}, S)$
synchronize both paths
  Asymmetric policy adaptation
$g(a) \leftarrow 0$ for all $a \in X$
$g(a) \leftarrow \min_{b \in S} r(b) - r(a)$ for all $a \in X \cap S$ ▷ Exact full-action match
Define $q(\alpha) = \mathrm{softmax}_X(\log p + \alpha g)$; set $\eta \leftarrow \beta^{-1}$
$\alpha \leftarrow \eta$
if $\kappa = 0$ **then**
  $\alpha \leftarrow 0$
else if $D_{\mathrm{KL}}(q(\eta) \| p) > \kappa$ **then**
  Bisect $[0, \eta]$ for $D_{\mathrm{KL}}(q(\alpha) \| p) = \kappa$; use the feasible endpoint
end if
  Action commitment
Sample $a_t \sim q(\alpha)$ on X
$(s_{t+1}, o_t) \sim \mathcal{E}(\cdot \mid s_t, a_t)$ ▷ Fresh execution in the ongoing environment
$h_{t+1} \leftarrow \mathrm{Append}(h_t, a_t, o_t)$

Our SIPA evaluation comprises 219 ToolSandbox tasks, 256 ScienceWorld tasks, 240 BFCL instances, 134 ALFWorld tasks, and 38 BabyAI-Text tasks. Each episode permits at most 100 committed actions. Speculative trials are counted separately as additional environment calls.

Baseline descriptions. Actor-only uses the same reasoning, candidate generation, complete-action probability scoring, and resampling as SIPA, with the effective update strength set to $\alpha = 0$. It generates and validates the actor candidates, sums token log-probabilities under the actor probe without length normalization, normalizes over these candidates using the same scoring temperature, and samples from the resulting unmodified actor prior. This is the w/o APA setting in Tables 2 and 6; its results are identical to the Actor-only row in Table 1. The same Actor-only definition applies throughout the scaling, interaction-budget, and efficiency experiments. Critic-only tests proposals and samples from a positive critic distribution over these proposals without an actor prior. Among retrieval-based TTA methods, ReasoningBank (Ouyang et al., 2026) retrieves reasoning memories derived from historical trajectories; ACE (Zhang et al., 2026) accumulates experience in a playbook; and JitRL (Li et al., 2026a) retrieves past trajectories to adjust action logits. For all three retrieval-based baselines, we collect two rounds of historical trajectories before evaluation to supply their experience. Among generation-based TTA methods, Reflexion (Shinn et al., 2023) retains reflections across attempts within a task, and SMURFS (Chen et al., 2025) combines task decomposition, verification, and backtracking. Their successive attempts acquire experience serially during test-time interaction. SIPA instead uses current-state execution feedback to adapt the actor's action distribution in parallel with actor reasoning and retains no cross-task memory.

D.1 BABYAI: ADVANCING AN ONGOING TRAJECTORY

Figure 10 shows a correction that advances the task without immediately completing it. At step 2, the actor assigns 99.9825% probability to `toggle`, whose trial leaves the observation unchanged. Moving forward instead brings the agent closer to the grey box. Rejection feedback raises `go forward` from 0.0008% to 73.30%, and SIPA commits to this movement. A subsequent left turn completes the mission at step 3. The case illustrates how immediate execution evidence can identify useful intermediate progress.

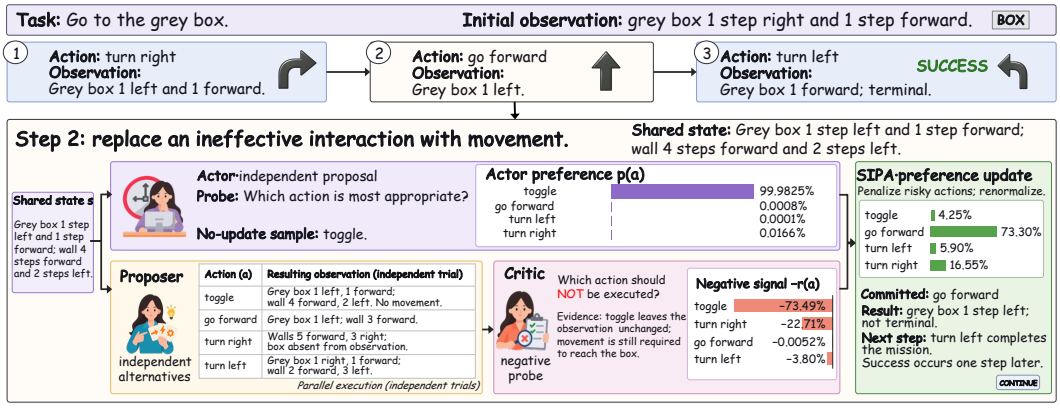

Figure 10: **BabyAI: approaching the grey box.** The three-step trajectory appears above the decision at step 2. Feedback suppresses an ineffective toggle and favors moving forward; the mission succeeds on the following step.

D.2 ALFWORLD: DIRECTING SEARCH TOWARD THE TARGET

In Figure 11, the agent has not yet located the requested saltshaker. The actor favors another shelf, but a speculative visit to cabinet 1 reveals the target. APA increases the cabinet action's probability to 69.48%, enabling the agent to find the saltshaker and subsequently place it in the drawer. The example also illustrates candidate support: the proposer-only drawer action cannot enter the updated actor distribution, while untrialed actor candidates remain available after normalization.

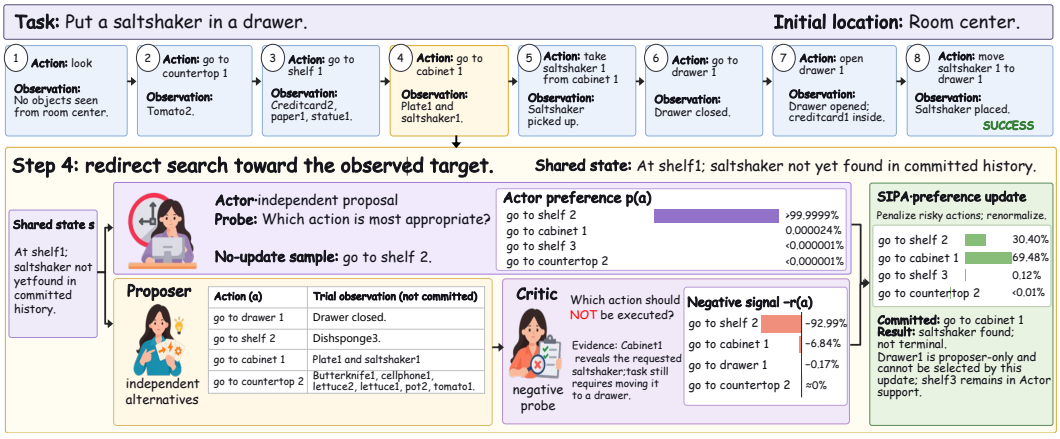

Figure 11: **ALFWorld: locating a saltshaker before placing it in a drawer.** At step 4, current-state trials reveal a useful search destination. The selected cabinet visit discovers the target; the remaining committed actions complete the task at step 8.

D.3 SCIENCEWORLD: REVISING A CONCENTRATED ACTION PREFERENCE

Figure 12 illustrates how APA revises a highly concentrated actor prior. The actor assigns 99.92% probability to `focus on B`. The critic assigns greater rejection probability to B than to J, and APA raises `focus on J` to 96.83%. The committed selection completes the task. Only the two shared focus actions receive direct corrections, while both inspection actions remain in the actor support. The displayed observations acknowledge the selected planes without revealing their angles; this case illustrates the recorded preference update rather than a correctness certificate supplied by the environment.

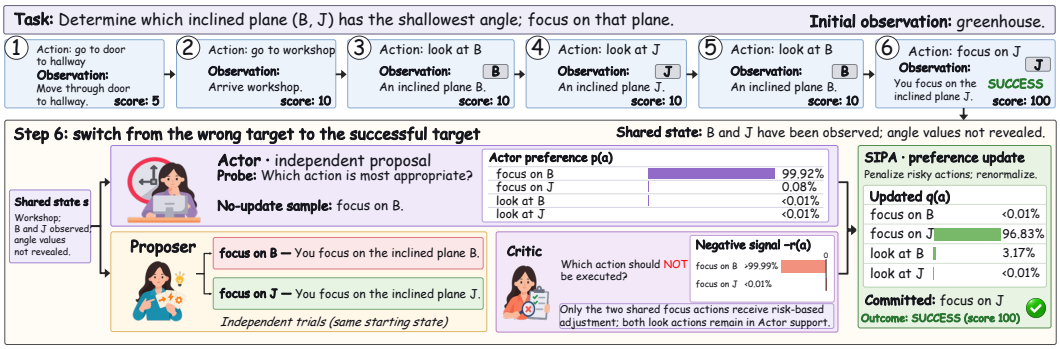

Figure 12: **ScienceWorld: revising the target choice.** At step 6, critic rejection shifts probability from B to J. The committed J action completes the episode; the final score is shown only as an evaluation outcome.

D.4 SCIENCEWORLD: RETAINING AN UNTRIALED ALTERNATIVE

In Figure 13, the agent must move an animal to a purple box, but the ambiguous reference `object` resolves to a pillow. The trial observation for `focus on object` identifies the selected object as a pillow, which does not match the requested animal. Strong rejection of the actor's preferred `focus on object` redistributes probability toward the actor's untrialed `look at object`, which reaches 82.53%. The committed inspection is nonterminal and confirms the object's identity; the subsequent trajectory retrieves the baby wolf and completes the task. This demonstrates how rejection can preserve a useful alternative outside the trialed set.

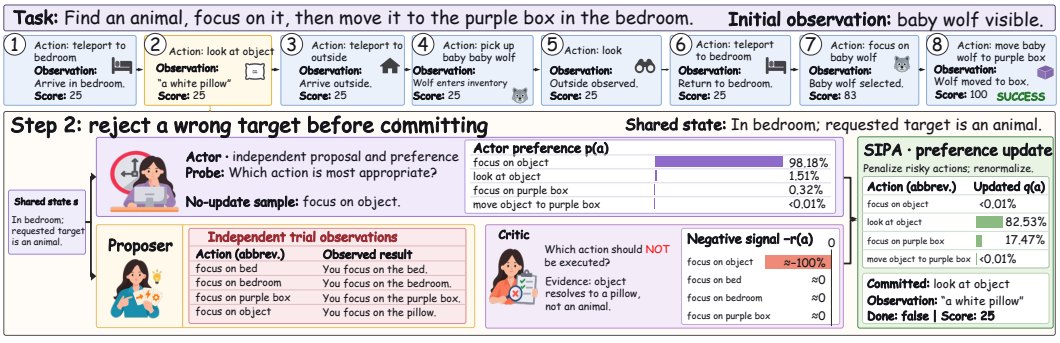

Figure 13: **ScienceWorld: avoiding a mistaken animal reference.** Suppression of the misleading focus action favors an inspection already present in the actor's candidates. The selected action is untrialed, and success occurs later in the eight-step trajectory.

D.5 SCIENCEWORLD: SHIFTING FROM SELECTION TO INSPECTION

Figure 14 shows rejection feedback redirecting a concentrated actor prior toward inspection. The actor overwhelmingly favors `focus on Al`. After adaptation, `look at Al` receives 98.59% probability, whereas the initially much less likely tin selection remains below 0.01%. SIPA commits to inspection and selects tin on the following step. The example illustrates how the actor prior shapes probability redistribution: rejecting the preferred focus action can favor gathering information rather than immediately selecting another target.

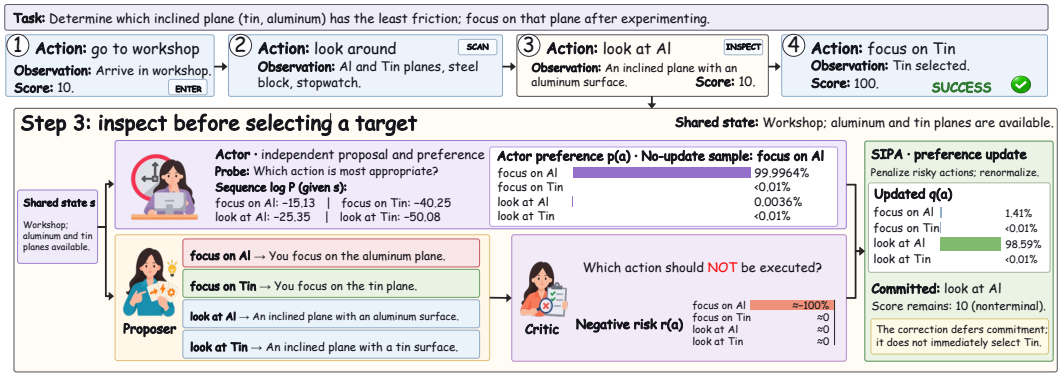

Figure 14: **ScienceWorld: favoring inspection before selection.** At step 3, APA reduces the probability of focusing on aluminum and favors inspection. The subsequent tin selection completes the task at step 4.

D.6 ALFWORLD: REJECTING A VALID BUT MISDIRECTED ACTION

The candidate actions in Figure 15 produce different state changes with different relevance to the goal. Returning the held pillow to the armchair is valid yet conflicts with the requested sofa destination. The critic rejects this action, and APA raises `go to sofa 1` from 0.0013% to 91.58%. The agent then transports both pillows and succeeds at step 9. This case shows why evaluating execution observations must account for the task: a valid state change does not necessarily advance the requested goal.

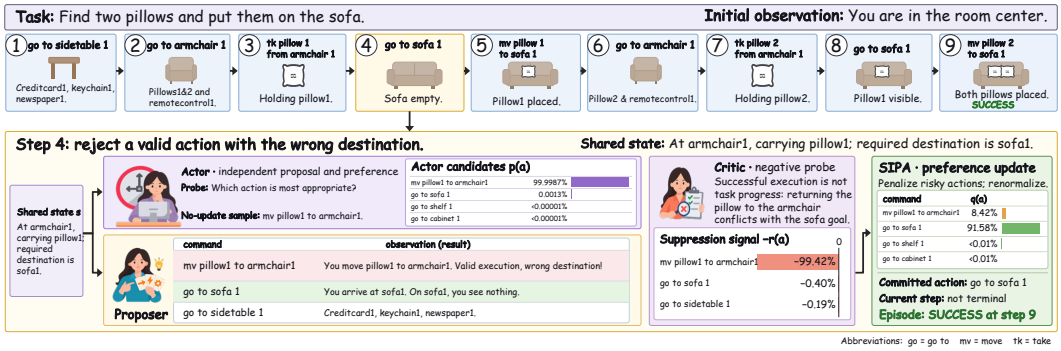

Figure 15: **ALFWorld: preserving the intended destination.** At step 4, feedback suppresses returning a pillow to its original armchair and favors moving toward the sofa. The highlighted action advances the task without immediately terminating it.

D.7 ALFWORLD: ENDING REPEATED INSPECTION

Figure 16 shows an agent repeatedly inspecting a desk after turning on the lamp. The actor still assigns 62.42% probability to another inspection and 37.58% to taking the CD. The trial observations distinguish repeated information from a new state change: inspecting the desk repeats its contents, using the lamp repeats an earlier operation, and taking the CD adds it to the inventory. APA raises the latter action above 99.99%, and the committed action completes the episode at step 6. The update favors a relevant state change over repeated observation; success is established by the subsequent committed execution.

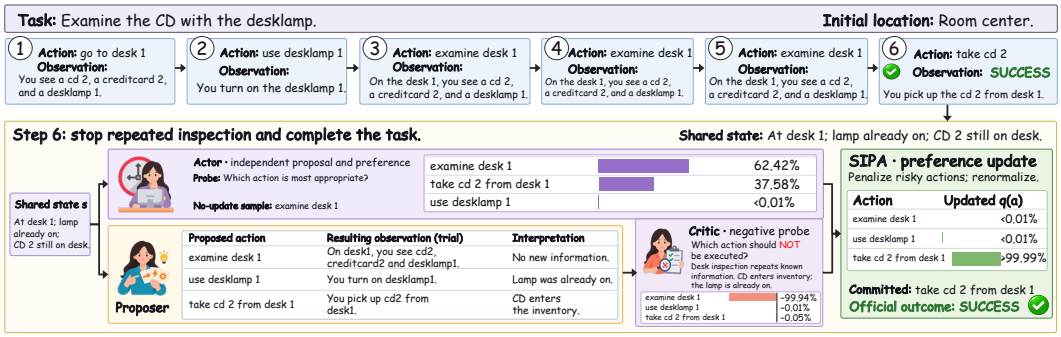

Figure 16: **ALFWorld: completing the CD-and-desklamp task.** At step 6, trial observations contrast redundant inspection with taking the CD. APA favors the latter, which succeeds when committed.

D.8 PROMPTS

The following prompts are reproduced verbatim. Line wrapping, bold emphasis, and placeholder colors are for presentation only.

Shared reasoning instruction: Actor and Proposer

Privately assess the **CURRENT** request using the public history and available actions. Determine what, if anything, remains to be done. Identify the next actions actually worth trying at this same state. Do not invent additional user goals. A candidate is an alternative to execute now, not a later step in a plan. Do not assign letters, ranks, scores or a required number of alternatives.

Actor role instruction

Identify **ALL** actions that are genuinely worth executing now, based on your assessment. Do not stop listing worthwhile alternatives merely because you already have a preferred action.

Proposer role instruction

Identify distinct alternatives worth testing through execution. Cover different plausible interpretations of the current state or next step. Avoid paraphrases and meaningless padding. If the current request may already be fulfilled, also consider replying to the user using the available interface. Do not assume the task is complete without evidence.

Shared candidate-generation template

Using the assessment below, output a compact JSON object with an actions array containing the next actions you consider worth trying. `{role_instruction}` The array must be nonempty; its maximum size is `{maximum}`, which is only a resource ceiling, not a target. Stop once the worthwhile alternatives have been listed. Do not add actions solely for variety. Do not output rankings or explanations. Use the provided action interface exactly.
Private assessment:
`{reasoning}`
Candidate JSON schema:
`{schema}`

Actor probability probe

Given the current request, history, assessment and alternatives above, the single most reasonable action to actually take **NOW** is: Respond with exactly one complete action JSON object from the alternatives, followed by a newline. No explanation, label or ranking.

Positive probe

Given this evidence, which alternative is **MOST** appropriate to actually execute **NOW** to advance or complete the **CURRENT** request? Prefer supported progress over redundant activity. Return exactly its complete action JSON object followed by a newline; no explanation, label or ranking.

Critic negative probability probe

Which of the alternatives is **LEAST** appropriate to execute **NOW**, given the current request, history and observed execution evidence? Return the action to **AVOID**, not its repair or a better alternative. Respond with exactly that complete action JSON object from the alternatives, followed by a newline. No explanation, label or ranking.

E SUPPLEMENTARY THEORETICAL ANALYSIS

We supplement the core proofs in Appendix B with the sharpness of the stability bounds, sensitivity of the rejection correction, and two limits of the return guarantee. We use the notation in Appendix A.

E.1 SHARPNESS OF THE STABILITY BOUNDS

The leading constants in Theorem 2 are asymptotically attained by a two-action construction. Fix $p = (1/2, 1/2)$, $\beta > 0$, and $\kappa > 0$, and take $g = (0, 0)$ and $g_\star = (e, -e)$ with $e > 0$. The offset-invariant error is exactly e. For sufficiently small e, both unconstrained optimizers satisfy the KL budget, giving

$$q = p, \qquad q_\star = \frac{(\exp(e/\beta), \exp(-e/\beta))}{2\cosh(e/\beta)}.$$

Direct substitution and Taylor expansion, with β fixed, yield

$$F_{g_\star}(q_\star) - F_{g_\star}(p) = \beta \log\cosh(e/\beta) = \frac{e^2}{2\beta} + O(e^4),$$
$$D_{\mathrm{KL}}(p\|q_\star) = \log\cosh(e/\beta) = \frac{e^2}{2\beta^2} + O(e^4),$$
$$D_{\mathrm{KL}}(q_\star\|p) = \frac{e}{\beta}\tanh(e/\beta) - \log\cosh(e/\beta) = \frac{e^2}{2\beta^2} + O(e^4),$$
$$d_{\mathrm{TV}}(q_\star, p) = \frac{1}{2}\tanh(e/\beta) = \frac{e}{2\beta} + O(e^3).$$

Thus the objective-gap, KL, and total-variation bounds are tight to leading order, so their constants cannot be decreased uniformly over the theorem's problem class. This construction uses inactive constraints; it does not establish sharpness restricted to cases where both constraints are active.

E.2 Sensitivity of the rejection correction

Fix the actor set X, proposer set $S = \{b_1, \dots, b_M\}$, their exact-action matching, and $T_C > 0$. For complete-action critic log-scores $z_j = \ell_\theta(b_j; C_{\mathrm{critic}})$, write $r = \mathrm{softmax}(z/T_C)$. Equation (5) gives

$$g_i = \begin{cases} \min_k r_k - r_j, & a_i = b_j \in S, \\ 0, & a_i \notin S. \end{cases}$$

Let r' and g' be obtained from another score vector z' on the same candidates. The softmax Jacobian satisfies

$$\sum_k \left|\frac{\partial r_j}{\partial z_k}\right| = \frac{2r_j(1-r_j)}{T_C} \le \frac{1}{2T_C}.$$

Integrating along the segment from z to z' gives $\|r - r'\|_\infty \le \|z - z'\|_\infty/(2T_C)$. The minimum is 1-Lipschitz in this norm, even when the least-rejected proposal changes. Unmatched candidates have zero correction in both cases, so

$$\|g - g'\|_\infty \le 2\|r - r'\|_\infty \le \frac{\|z - z'\|_\infty}{T_C}. \tag{23}$$

For the corresponding APA optimizers q, q' with the same p, β, κ, Theorem 2 and $\min_b \|g - g' - b\mathbf{1}\|_\infty \le \|g - g'\|_\infty$ imply

$$d_{\mathrm{TV}}(q, q') \le \frac{\|z - z'\|_\infty}{2\beta T_C}. \tag{24}$$

The effective steps may differ. This is a fixed-candidate sensitivity bound, not a guarantee that rejection scores are aligned with continuation values.

E.3 Limits of the return guarantee

The following constructed examples use APA's actual rejection-gap rule, not arbitrary corrections. Fix $\beta > 0$ and $\kappa > 0$; the nonconstant feedback in each example then gives $\alpha > 0$ by Proposition 1.

A KL budget cannot correct reversed feedback. Let $X = S = \{a_1, a_2\}$, $p = (1/2, 1/2)$, and $Q = (1, 0)$. A critic distribution $r = (3/4, 1/4)$ produces $g = (-1/2, 0)$ and therefore

$$q_1 = \frac{1}{1 + \exp(\alpha/2)} < \frac{1}{2}, \qquad \Delta(\xi) = q_1 - \frac{1}{2} < 0.$$

Tightening the KL budget can reduce the loss but cannot reverse its sign. At $\kappa = 0$, the prior is retained. Controlling the magnitude of an update is therefore distinct from validating its direction.

Correct rankings among tried actions are insufficient. Let $X = \{a_1, a_2, a_3\}$, $S = \{a_1, a_2\}$, $p = (1/3, 1/3, 1/3)$, and $Q = (3/4, 1, 0)$. The critic distribution $r = (3/4, 1/4)$ correctly rejects the lower-value tried action more strongly, but gives $g = (-1/2, 0, 0)$ on X. Hence

$$q = \frac{(\exp(-\alpha/2), 1, 1)}{\exp(-\alpha/2) + 2}, \qquad \Delta(\xi) = \frac{\exp(-\alpha/2) - 1}{6\big(\exp(-\alpha/2) + 2\big)} < 0.$$

Global normalization sends part of the removed mass to the untried, lowest-value action a_3. A zero correction does not preserve its probability. This is why the alignment or error condition in Theorem 3 must cover the entire actor support, not only the tried candidates.