CoWISE: Co-Evolving World Model with Policy via Semantic–Behavioral Coordination

Anonymous authors
Paper under double-blind review

Abstract

World models enable language agents to predict the consequences of their actions and improve their policies through simulated interaction. To train these models, existing approaches typically use trajectories collected from agent-environment interactions to reproduce real observations. However, even semantically similar simulated observations can still induce different policy actions (e.g., simulated observations closely match the real observation in overall content but miss key objects, leading the policy to take incorrect actions), which we identify as behavioral mismatch. This behavioral mismatch grows as the policy continues learning while the world model remains fixed, thereby further misleading the policy into choosing wrong actions and degrading its performance. To reduce this mismatch, we propose **CoWISE**, the first dual-level co-evolution framework that continually adapts the world model to the evolving policy through semantic-behavioral coordination. Specifically, at the policy–world level, CoWISE enables the world model and the policy to co-evolve by leveraging each other's predictions. At the semantic–behavioral level, it adaptively balances real-world observational supervision and policy-derived behavioral supervision. To evaluate CoWISE, we conduct extensive experiments across 8 benchmarks, demonstrating that CoWISE improves policy performance and learns a world model with higher behavioral fidelity. Furthermore, using this learned world model for test-time scaling yields further gains, outperforming combinations of separately trained policy and world model[1].

1 Introduction

World models simulate the consequences of an LLM-based agent's actions by generating observations (Li et al., 2026b; Lu et al., 2026), enabling the policy to evaluate candidate actions before execution (Liu et al., 2026a) and learn through simulated interaction (Sun et al., 2026a), highlighting the importance of world models in advancing agent capabilities (Li et al., 2026c).

To train these models, existing approaches primarily use agent–environment trajectories to reproduce real observations (Li et al., 2026c; Yu et al., 2026; Zuo et al., 2026; Lu et al., 2026; Chen et al., 2026). However, simulated and real observations can lead the same policy to take different actions even when they are semantically similar (Figure 1(a)). For example, when the policy searches for a film's producer, **the simulated search results contain most of the same distractor information as the real ones but omit the correct film producer information**, preventing the model from taking the correct action (Appendix E). To reduce differences in policy actions under real and simulated observations, recent work uses a frozen reference policy to provide behavioral supervision for world model training (Huang et al., 2026), thereby improving behavioral fidelity. However, since this supervision is derived from a frozen reference policy, the world model can become **outdated as the policy evolves**, exhibiting declining behavioral fidelity as the policy continues to learn while the world model remains fixed (Figure 1(b)). This degradation in behavioral fidelity limits the utility of the world model throughout policy optimization. Therefore, to continually provide the policy with high-quality simulated observations, the world model should adapt to policy changes while maintaining high semantic and behavioral fidelity in these observations.

To achieve this, we propose **CoWISE**, the first dual-level co-evolution framework that adapts the world model to the evolving policy through semantic–behavioral coordination. Specifically, at the

[1]Our code will be publicly available. See the Reproducibility Statement for details.

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.

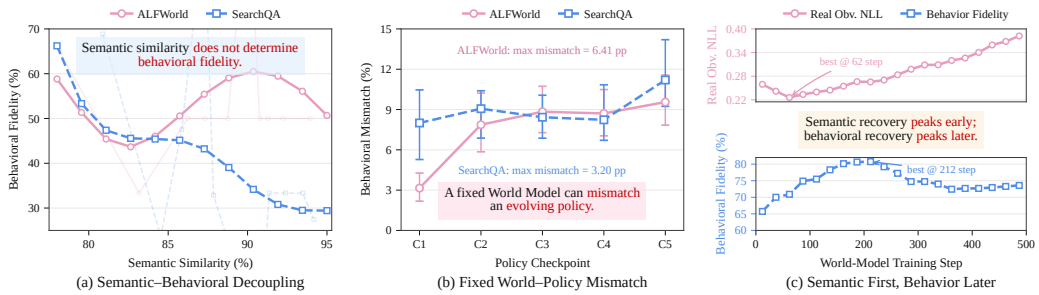


Figure 1: **Motivating studies for CoWISE. (a)** Semantic fidelity does not reliably predict behavioral fidelity. **(b)** A frozen world model's behavioral fidelity degrades as the policy evolves. **(c)** Behavioral fidelity continues to improve after real-observation NLL starts rising.

(1) policy–world level, CoWISE enables the world model and the policy to co-evolve by leveraging each other's predictions. The evolving policy provides its latest interaction trajectories and behavioral supervision to update the world model, while the world model progressively feeds simulated observations and world knowledge back into policy learning. At the **(2) semantic–behavioral level**, CoWISE adaptively balances real-world observational supervision and policy-derived behavioral supervision. As semantic and behavioral learning follow different dynamics, with behavioral fidelity peaking after the best fit to real observations (Figure 1(c)), CoWISE dynamically adjusts their relative weights to improve behavioral fidelity while maintaining semantic fidelity.

To evaluate CoWISE, we conduct extensive experiments across eight benchmarks. The results show that CoWISE improves both policy performance and world model behavioral fidelity. The policy alone outperforms policy optimization baselines on average, while the world model achieves higher action agreement and task retention than world model baselines. CoWISE further combines these capabilities for test-time scaling, using the same model in both roles to simulate the observations of candidate actions when the policy is uncertain. This yields further gains over policy-only execution and better performance than approaches that combine separately trained policies and world models, without requiring an additional model. In summary, the key contributions of this work are as follows:

- To the best of our knowledge, this work is the first to identify that existing world models struggle to continually maintain both semantic and behavioral fidelity in simulated observations as the policy evolves, posing the fundamental bottleneck in using world models to guide policy acting.
- To mitigate this bottleneck, we propose CoWISE, the first dual-level co-evolution framework that jointly trains the policy and world model, enabling the world model to continuously adapt to the evolving policy while maintaining both semantic and behavioral fidelity.
- To evaluate CoWISE, we conduct extensive experiments across 8 benchmarks, demonstrating improvements in both policy performance and world-model behavioral fidelity. Moreover, using the co-evolved policy and world model for test-time scaling yields further gains over policy-only execution and combinations of separately trained policies and world models.

2 Preliminaries and Motivating Studies

Before introducing CoWISE, we formalize agent–environment interaction and world model prediction, semantic and behavioral fidelity (§2.1), and present studies for evolving policies (§ 2.2).

2.1 Preliminaries

Agent–environment interaction. We consider an agent policy interacting with an environment $\mathcal{E}$ to complete a task $x \sim \mathcal{D}$. At step t, the policy generates $a_t \sim \pi_\theta(\cdot \mid h_t)$, where h_t is the interaction history before generating a_t. For nonterminal a_t, the environment returns an observation o_{t+1}, updating the history as $h_{t+1} = h_t \oplus (a_t, o_{t+1})$, where $\oplus$ denotes concatenation. The resulting sequence of policy responses and observations forms a trajectory τ, for which $\mathcal{E}$ provides a task reward $R(\tau)$ upon termination.

World model. A world model predicts the simulated observation $\hat{o}_{t+1} \sim W_\theta(\cdot \mid c_t)$, where the *World Context* c_t combines the world model system prompt, interaction history h_t, and current action a_t. In CoWISE, π_θ and W_θ share a single backbone (Xu et al., 2026; Lu et al., 2026).

Semantic vs. behavioral fidelity. **(1) Semantic fidelity** measures how faithfully a simulated observation preserves the semantic content of the corresponding real environmental feedback. We quantify it by $s_{\text{sem}} = \cos(e(\hat{o}_{t+1}), e(o_{t+1}))$ with a text encoder $e(\cdot)$. **(2) Behavioral fidelity** measures how faithfully simulated feedback preserves the action-level decision signal induced by real feedback. Following Huang et al. (2026), we modify their behavioral discrepancy measure by taking absolute differences in token log-probabilities before averaging across tokens. Specifically, for a reference policy π, let a_{t+1} be its next response generated under real feedback and $\mathcal{I}_{t+1}$ the scored token positions. Writing the token log-probability under observation o as $b_{t,i}(o) = \log \pi(a_{t+1,i} \mid h_t \oplus (a_t, o), a_{t+1,<i})$, we define the behavioral discrepancy as

$$d_t(\hat{o}_{t+1}; \pi) = \frac{1}{|\mathcal{I}_{t+1}|} \sum_{i \in \mathcal{I}_{t+1}} \left| b_{t,i}(o_{t+1}) - b_{t,i}(\hat{o}_{t+1}) \right|. \tag{1}$$

Lower discrepancy indicates higher behavioral fidelity under the reference policy. Taking absolute differences before averaging prevents cancellation across tokens.

2.2 Motivating Studies

With these definitions, we ask three questions: **(1)** Does higher semantic fidelity consistently correspond to higher behavioral fidelity? **(2)** Can a frozen world model trained with a fixed reference policy maintain behavioral fidelity as the agent policy evolves? **(3)** During joint training, do the world model's fit to real observations and its behavioral fidelity improve together? Experimental details for all three studies are in Appendix A.

(1) Semantic and behavioral fidelity. To test whether semantic fidelity reliably indicates behavioral fidelity, we evaluate ZeroSearch (Sun et al., 2026a) on SearchQA and W2W (Li et al., 2026c) on ALFWorld, both trained via SFT to reproduce observations from real interactions. For each task, we sample 128 transitions (h_t, a_t, o_{t+1}) from trajectories collected by the GRPO-trained policy. Given (h_t, a_t), the world model predicts $\hat{o}_{t+1}$. We compare each prediction with o_{t+1} using semantic similarity and behavioral fidelity e^{-d_t}. As shown in Figure 1(a), **higher semantic fidelity does not consistently correspond to higher behavioral fidelity** on either task, as semantic similarity captures overall semantic correspondence but does not guarantee that decision-critical information affecting the policy's action is fully preserved. This mismatch motivates complementing semantic supervision with explicit supervision of behavioral fidelity.

(2) Behavioral fidelity under policy evolution. We next test whether behavioral fidelity learned with a fixed reference policy persists as the policy evolves. Following BehR (Huang et al., 2026), we train world models on SearchQA and ALFWorld by combining semantic supervision from real-environment observations with behavioral supervision from a fixed reference policy (training details in Appendix C.2). We then freeze the world models and evaluate behavioral fidelity under five successive GRPO policy checkpoints on the same fixed set of transitions. Figure 1(b) shows that **the frozen world models do not maintain behavioral fidelity as the policies evolve**: behavioral mismatch (one minus mean fidelity) increases from 3.2% to 9.6% on ALFWorld and from approximately 8.0% to 11.2% on SearchQA between the first and last checkpoints. This degradation motivates updating world models with behavioral supervision from the evolving policy.

(3) Learning dynamics of semantic and behavioral fidelity. Finally, we test whether the world model's fit to real observations and its behavioral fidelity improve together during joint training. We track real-observation negative log-likelihood (NLL) and behavioral fidelity when training the SearchQA world model from the second motivating study, which uses equal weights on the two objectives. Figure 1(c) shows that **behavioral fidelity continues to improve even as the fit to real observations deteriorates**: NLL increases after reaching its minimum around step 62, while behavioral fidelity peaks around step 212. This divergence motivates adapting the objective weights to improve behavioral fidelity while maintaining the model's fit to real observations.

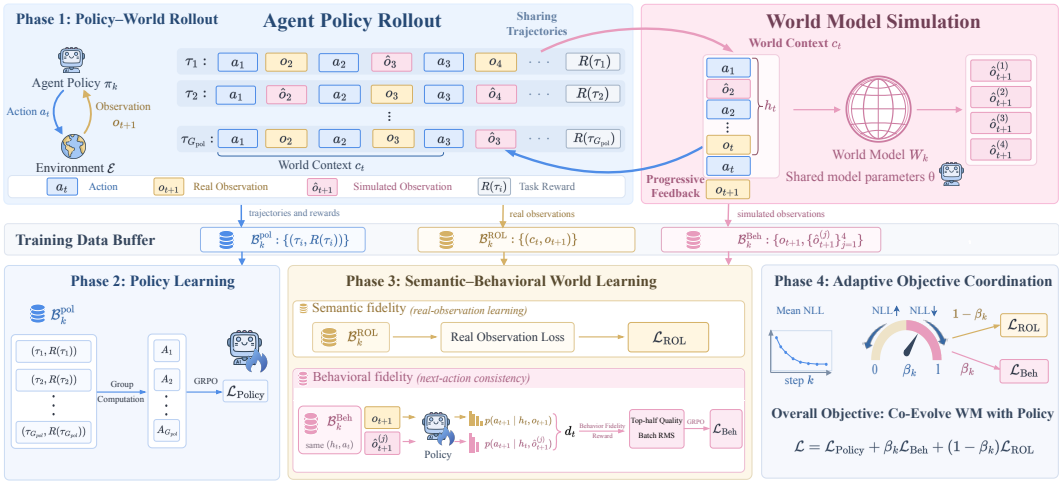


Figure 2: **Overview of CoWISE.** The evolving policy provides the latest behavioral supervision, while the world model returns simulated observations for policy learning. Adaptive weighting balances semantic and behavioral objectives according to the world model's fit to real observations.

3 CoWISE: Co-Evolving World Models and Policies

Motivated by findings in Sec. 2.2, we introduce CoWISE, a dual-level framework that co-evolves the policy and world model while adaptively coordinating semantic and behavioral learning. As illustrated in Fig. 2, CoWISE comprises four phases: policy-world rollout (§3.1), policy learning (§3.2), semantic-behavioral world learning (§3.3), and adaptive objective coordination (§3.4).

3.1 Policy–World Rollout

To adapt the world model to the evolving policy (Fig. 1(b)), CoWISE collects fresh interaction data at each iteration k. The policy π_k and world model W_k share parameters θ_k, and π_k (Xu et al., 2026) generates G_{pol} trajectories $\{\tau_i\}_{i=1}^{G_{\text{pol}}}$ per task x for GRPO-based policy learning (Shao et al., 2024).

Along each trajectory, the policy acts on its observed history h_t, and the environment executes each nonterminal action a_t to return a real observation o_{t+1}, forming a transition (h_t, a_t, o_{t+1}). To reduce computational cost and focus world model learning on transitions most consequential for subsequent policy decisions, we retain a subset with high next-action uncertainty (Lu et al., 2026). For each retained transition, W_k samples G_{wm} candidate observations $\hat{o}_{t+1}^{(j)}$ conditioned on the world context c_t. To train the policy to use simulated observations, we update $h_{t+1} = h_t \oplus (a_t, \tilde{o}_{t+1})$, where $\tilde{o}_{t+1}$ is a uniformly sampled candidate with probability p_k and o_{t+1} otherwise. We increase p_k from zero, limiting early exposure to low-quality simulated observations.

The collected data populate three training buffers. $\mathcal{B}_k^{\text{pol}}$ stores trajectories and environment-provided task rewards $(\tau, R(\tau))$ for policy learning (§3.2). For world model learning (§3.3), $\mathcal{B}_k^{\text{ROL}}$ stores (c_t, o_{t+1}) for real-observation supervision, and $\mathcal{B}_k^{\text{Beh}}$ stores $(c_t, o_{t+1}, \{\hat{o}_{t+1}^{(j)}\}_{j=1}^{G_{\text{wm}}})$ for behavioral supervision. Transition selection and feedback-schedule details are provided in Appendix B.2. Appendix B.5 describes the rollout scheduling optimizations used to accelerate data collection.

3.2 Policy Learning

Using the environment-provided rewards in $\mathcal{B}_k^{\text{pol}}$, we compute each trajectory's group-relative advantage as $A_\tau^{\text{pol}} = [R(\tau) - \bar{R}_x]/(\sigma_x + \varepsilon)$, where $\bar{R}_x$ and σ_x are the mean and standard deviation of rewards across the G_{pol} trajectories for task x. We use the following GRPO loss (Shao et al., 2024) for policy learning and reuse it for behavioral world model learning (§3.3). For a model distribution

p_θ, scalar advantage A, and optimized token positions $\mathcal{T}$, we define the token-normalized loss

$$\ell_{\mathrm{GRPO}}(\theta; A) = -\frac{1}{|\mathcal{T}|}\sum_{i\in\mathcal{T}} \min\{\rho_i A,\ \mathrm{clip}(\rho_i, 1-\epsilon_-, 1+\epsilon_+)A\} + \lambda_{\mathrm{KL}} D_{\mathrm{KL}}(p_\theta \| p_{\mathrm{ref}}), \quad (2)$$

where ρ_i is the token likelihood ratio between the current and rollout models and ϵ_- and ϵ_+ specify the clipping bounds. For CoWISE training, we set $\lambda_{\mathrm{KL}} = 0$, so the KL term is inactive and no fixed-reference regularization is applied. For policy learning, $p_\theta = \pi_\theta$ and $\mathcal{T}$ contains only policy-response tokens, with real and simulated observations serving as conditioning context. Averaging over $\mathcal{B}_k^{\mathrm{pol}}$ gives $\mathcal{L}_{\mathrm{Policy}}(\theta) = \mathbb{E}_{\tau\sim\mathcal{B}_k^{\mathrm{pol}}}[\ell_{\mathrm{GRPO}}(\theta; A_\tau^{\mathrm{pol}})]$, which contributes to the joint update of the shared model parameters (§3.4).

3.3 Semantic–Behavioral World Learning

Since semantic similarity alone does not ensure behavioral fidelity (Fig. 1(a)), we combine real-observation and current-policy supervision using the two world model buffers from §3.1.

Semantic fidelity. To learn environment dynamics, we train W_θ to predict the real observations in $\mathcal{B}_k^{\mathrm{ROL}}$ from their world contexts c_t. We use the token-normalized Real-Observation Loss (ROL), $\mathcal{L}_{\mathrm{ROL}}(\theta) = \mathbb{E}_{(c_t, o_{t+1})\sim\mathcal{B}_k^{\mathrm{ROL}}}\left[-\frac{1}{|o_{t+1}|}\log W_\theta(o_{t+1} \mid c_t)\right]$, where $|o_{t+1}|$ is the token count.

Behavioral fidelity. Because behavioral fidelity depends on the policy, we score candidates in $\mathcal{B}_k^{\mathrm{Beh}}$ with the current π_k. For the next response a_{t+1} generated under real feedback, d_t in Eq. (1) compares its token log-probabilities under the real and candidate observations. Each candidate receives $r_{t,j}^{\mathrm{beh}} = \exp[-d_t(\hat{o}_{t+1}^{(j)}; \pi_k)]$, with higher rewards indicating closer behavioral agreement.

For these candidate rewards, standard GRPO normalization can amplify small within-group differences even when all candidates poorly preserve the current policy's behavior under real feedback (Appendix B.3). We therefore weight centered rewards by $q_t = \frac{2}{G_{\mathrm{wm}}}\sum_{j=1}^{G_{\mathrm{wm}}/2} r_{t,(j)}^{\mathrm{beh}}$, the mean reward of the highest-scoring half in each group, to favor groups whose better candidates achieve higher behavioral fidelity (Garg et al., 2026). A shared batch RMS then preserves these relative weights and reward gaps by applying the same normalization scale across groups (Liu et al., 2025):

$$v_{t,j} = q_t\left(r_{t,j}^{\mathrm{beh}} - \bar{r}_t^{\mathrm{beh}}\right), \qquad A_{t,j}^{\mathrm{beh}} = \frac{v_{t,j}}{\mathrm{RMS}_{\mathcal{B}_k^{\mathrm{Beh}}}(v) + \varepsilon}. \quad (3)$$

Here, $r_{t,(j)}^{\mathrm{beh}}$ denotes rewards in descending order, $\bar{r}_t^{\mathrm{beh}}$ is the group mean, and the RMS covers all batch candidates. With $p_\theta = W_\theta$ and $\mathcal{T}$ restricted to generated observation tokens in Eq. (2), the behavioral objective is $\mathcal{L}_{\mathrm{Beh}}(\theta) = \mathbb{E}_{\mathcal{B}_k^{\mathrm{Beh}}}\left[\frac{1}{G_{\mathrm{wm}}}\sum_{j=1}^{G_{\mathrm{wm}}} \ell_{\mathrm{GRPO}}(\theta; A_{t,j}^{\mathrm{beh}})\right]$.

Sec. 3.4 balances $\mathcal{L}_{\mathrm{ROL}}$ and $\mathcal{L}_{\mathrm{Beh}}$ and combines them with $\mathcal{L}_{\mathrm{Policy}}$ to update the shared parameters.

3.4 Adaptive Objective Coordination

Real-observation NLL reaches its minimum before behavioral fidelity peaks (Fig. 1(c)), motivating adaptive weighting of the world model objectives. We jointly update the shared parameters using

$$\mathcal{L}_k(\theta) = \mathcal{L}_{\mathrm{Policy}}(\theta) + (1-\beta_k)\mathcal{L}_{\mathrm{ROL}}(\theta) + \beta_k \mathcal{L}_{\mathrm{Beh}}(\theta). \quad (4)$$

Here, $\beta_k \in (0, 1)$ balances behavioral learning and real-observation supervision.

To adapt this balance, we monitor the unweighted ROL on a fixed out-of-distribution calibration set, $\bar{\ell}_k = \mathcal{L}_{\mathrm{ROL}}(\theta_k; \mathcal{D}_{\mathrm{cal}})$, and its increase from the best observed value, $d_k = \bar{\ell}_k - \min_{0\le j\le k}\bar{\ell}_j$. Since e^{d_k} is the corresponding perplexity ratio, a relative perplexity tolerance δ gives an NLL budget of $\log(1+\delta)$. At scheduled updates, we use the remaining budget to adjust the behavioral weight:

$$\beta_{k+1} = \sigma[\mathrm{logit}\,\beta_k + \mathrm{clip}(\log(1+\delta) - d_k, -\log 2, \log 2)], \quad (5)$$

where σ is the sigmoid function. The update increases β_k when $d_k < \log(1+\delta)$ and shifts weight toward ROL when the budget is exceeded. Clipping bounds logit changes; σ keeps $\beta_k \in (0, 1)$. Derivations, hyperparameters, and the update schedule appear in Appendix B.4.

Table 1: **Main results** with exact match (EM, %) on SearchQA and success rate (SR, %) on ALFWorld and SciWorld. Best and second-best scores are **bold** and underlined.

Method	Open-Ended (EM)							Rule-Governed (SR)			Avg.
	NQ	TriviaQA	PopQA	HotpotQA	2Wiki	MuSiQue	Avg.	ALFWorld	SciWorld	Avg.	
Prompting											
Direct Prompt	13.4	40.8	14.0	18.3	25.0	3.1	19.1	15.0	4.2	9.6	16.7
ReAct	21.2	40.2	8.8	19.6	19.6	7.6	19.5	31.0	7.8	19.4	19.5
Policy Optimization											
PPO†	39.3	61.0	39.7	37.0	41.4	14.6	38.8	80.4	24.0	52.2	42.2
GRPO	47.9	63.9	47.8	43.9	43.6	18.3	44.2	88.7	49.1	68.9	50.4
GSPO†	47.0	64.5	46.1	40.0	38.2	19.2	42.5	89.5	50.7	70.1	49.4
GiGPO†	46.4	64.7	46.1	41.6	43.6	18.9	43.6	90.8	53.4	72.1	50.7
World-Model-Assisted Methods											
ZeroSearch†	43.6	65.2	**48.8**	34.6	35.2	18.4	41.0	—	—	—	—
PaW	48.9	64.9	48.5	44.9	**45.1**	18.9	45.2	80.6	53.0	66.8	50.6
TAPO	46.9	62.4	43.9	40.7	40.0	16.6	41.8	83.6	54.4	69.0	48.6
W2W + GRPO	48.1	64.0	47.6	44.3	43.8	18.1	44.3	89.1	49.7	69.4	50.6
BehR + GRPO	48.0	64.0	47.4	43.8	43.3	18.0	44.1	88.0	48.7	68.4	50.2
Ours											
CoWISE	50.0	**66.0**	48.4	46.4	43.2	**21.3**	45.9	93.4	53.7	73.6	52.8
CoWISE + TTS	**50.1**	**66.0**	48.3	**47.6**	44.1	21.1	**46.2**	**94.2**	**56.3**	**75.3**	**53.5**

4 Experiments

4.1 Experimental Setup

Benchmarks. Following prior research (Jin et al., 2025; Lu et al., 2026), we evaluate agents on eight benchmarks spanning two task groups: **(1) Open-ended tasks** include search-augmented question answering on Natural Questions (NQ) (Kwiatkowski et al., 2019), TriviaQA (Joshi et al., 2017), PopQA (Mallen et al., 2023), HotpotQA (Yang et al., 2018), 2WikiMultiHopQA (2Wiki) (Ho et al., 2020), and MuSiQue (Trivedi et al., 2022), evaluated using Exact Match (**EM**). **(2) Rule-governed tasks** include embodied household reasoning on ALFWorld (Shridhar et al., 2021) and scientific experimentation on SciWorld (Wang et al., 2022), evaluated using Success Rate (**SR**).

Baselines. Following Huang et al. (2026); Li et al. (2026c), we compare CoWISE against representative methods spanning three categories of baselines: **(1) Prompting**, including Direct Prompt and ReAct (Yao et al., 2023) **(2) Policy optimization**, such as PPO (Schulman et al., 2017), GRPO (Shao et al., 2024), GSPO (Zheng et al., 2025), and GiGPO (Feng et al., 2026), and **(3) World-model-assisted**, spanning ZeroSearch (Sun et al., 2026a), PaW (Lu et al., 2026), TAPO (Li et al., 2026a), W2W+GRPO (Li et al., 2026c), and BehR+GRPO (Huang et al., 2026). For W2W, BehR, and CoWISE + TTS, they assist the policy through uncertainty-based test-time scaling (Han et al., 2024). Implementation details are provided in Appendix D.3.

Implementation Details. For fair comparison, all methods use Qwen2.5-7B-Instruct (Qwen et al., 2025) as the backbone. For search-augmented QA, we follow the experimental setup of Search-R1 (Jin et al., 2025), including the retriever environment and training datasets. For ALFWorld and SciWorld, we follow the environment settings of prior works (Zhang et al., 2026; Wang et al., 2026). We use slime (Zhu et al., 2025) as the training framework, and conduct all experiments on 32 NVIDIA H20 141G GPUs. Training settings and prompt templates are in Appendices C.3 and F.

4.2 Main Results

Table 1 presents the main results across *open-ended* search-augmented QA and *rule-governed* environments (ALFWorld, SciWorld). Several conclusions can be drawn from the results:

CoWISE improves policy performance without test-time scaling. Without test-time scaling (TTS), CoWISE outperforms all baselines on average in both task groups. Its overall score of 52.8 exceeds ReAct (19.5) and GiGPO (50.7), as well as PaW and W2W+GRPO (both 50.6), the

Table 2: **Ablation study of CoWISE.** Scores (%) are mean EM over six SearchQA datasets and SR for ALFWorld/SciWorld. All TTS evaluations use $K = 4$. Δ denotes the score difference in points from the full model in the same evaluation mode. Darker pink/yellow indicates larger decreases/increases, with color intensity scaled separately for negative and positive differences.

Variant	SearchQA				ALFWorld				SciWorld			
	w/o TTS	Δ	w/ TTS	Δ	w/o TTS	Δ	w/ TTS	Δ	w/o TTS	Δ	w/ TTS	Δ
CoWISE (full)	45.9	—	46.2	—	93.4	—	94.2	—	53.7	—	56.3	—
A.1 Policy–World coupling												
w/ frozen behavioral supervisor	45.7	↓ 0.2	45.3	↓ 0.9	93.1	↓ 0.3	92.3	↓ 1.9	53.5	↓ 0.2	53.0	↓ 3.3
w/o simulated training feedback	46.0	↑ 0.1	46.1	↓ 0.1	93.4	0.0	93.4	↓ 0.8	54.0	↑ 0.3	54.9	↓ 1.4
A.2 World objectives												
w/o behavioral supervision	45.8	↓ 0.1	45.1	↓ 1.1	93.1	↓ 0.3	91.6	↓ 2.6	53.6	↓ 0.1	52.9	↓ 3.4
w/o real-observation supervision	44.6	↓ 1.3	44.7	↓ 1.5	91.2	↓ 2.2	91.6	↓ 2.6	51.4	↓ 2.3	51.9	↓ 4.4
w/ RMS-only advantages	45.6	↓ 0.3	45.9	↓ 0.3	93.1	↓ 0.3	93.4	↓ 0.8	53.2	↓ 0.5	55.2	↓ 1.1
w/ standard GRPO advantages	45.1	↓ 0.8	45.2	↓ 1.0	92.0	↓ 1.4	92.3	↓ 1.9	52.3	↓ 1.4	53.1	↓ 3.2
A.3 Objective coordination												
w/ fixed objective weights	45.8	↓ 0.1	45.9	↓ 0.3	93.4	0.0	93.4	↓ 0.8	53.8	↑ 0.1	55.5	↓ 0.8

strongest baselines in their respective categories. The 2.4-point gain over GRPO (50.4) suggests that joint training internalizes world knowledge in the shared parameters, improving policy performance.

CoWISE gains more from test-time scaling. Combining the jointly trained policy and world model through TTS raises the overall score from 52.8 to 53.5, a gain of 0.7 points. With the same uncertainty-gated TTS protocol, W2W+GRPO and BehR+GRPO score 50.6 and 50.2, respectively, relative to the GRPO policy-only score of 50.4. CoWISE + TTS therefore outperforms these separately trained combinations on all eight benchmarks, with overall margins of 2.9 and 3.3 points, respectively. Section 5.1 examines how policy–world co-evolution contributes to these TTS gains.

CoWISE benefits both open-ended and rule-governed tasks. Both policy learning and TTS yield larger gains in rule-governed environments. Relative to GRPO, CoWISE improves the SearchQA average by 1.7 points (44.2 to 45.9) and the rule-governed average by 4.7 points (68.9 to 73.6). TTS further raises these averages to 46.2 and 75.3, adding 0.3 and 1.7 points, respectively. This pattern is consistent with the knowledge required by each task group. Open-ended search depends on broad factual knowledge that is difficult for both models to learn comprehensively from interaction data. ALFWorld and SciWorld follow explicit rules, providing structured action–outcome relations for both policy and world-model learning. Section 5.2 examines these task-dependent differences.

5 Analysis

In this section, we analyze CoWISE through the following research questions: **RQ1:** *How do the components of CoWISE support co-evolution of policy and world model?* (§5.1) **RQ2:** *How well does the co-evolved world model preserve the policy's behavior?* (§5.2) **RQ3:** *Can this world model further improve policy performance through test-time scaling?* (§5.3) Case studies in Appendix E illustrate how simulated observations affect policy responses.

5.1 Ablation Study

We examine how the components of CoWISE support policy learning and the use of simulated observations. Ablation settings are detailed in Appendix D.1.

A.1 Policy-world coupling. The evolving policy supplies behavioral supervision to the world model, while simulated observations enter policy training. We examine these two directions by freezing the behavioral supervisor and removing simulated observations from policy training, respectively. Both variants remain within 0.3 points of the full model in policy-only evaluation. Their differences become clearer with TTS: freezing the supervisor turns gains into losses in all three environments, while removing simulated observations from policy training reduces the TTS gain on SciWorld from 2.6 to 0.9 points. Current-policy behavioral supervision keeps simulated observations useful as the policy evolves, while training on them increases TTS gains.

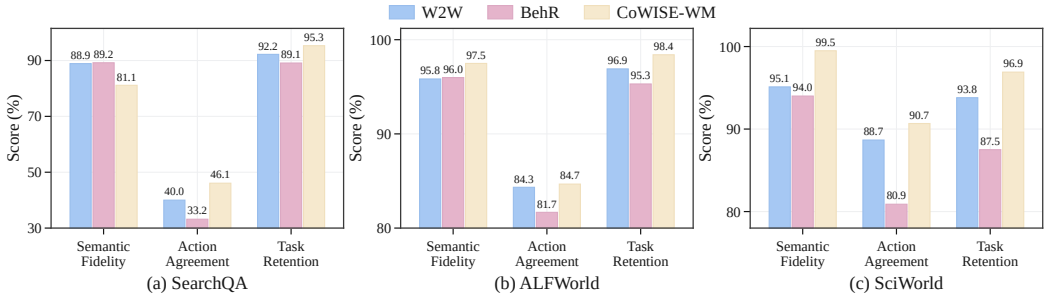


Figure 3: **Semantic and behavioral fidelity.** With the same frozen CoWISE-Policy, CoWISE-WM achieves the highest action agreement and task retention across all three environments.

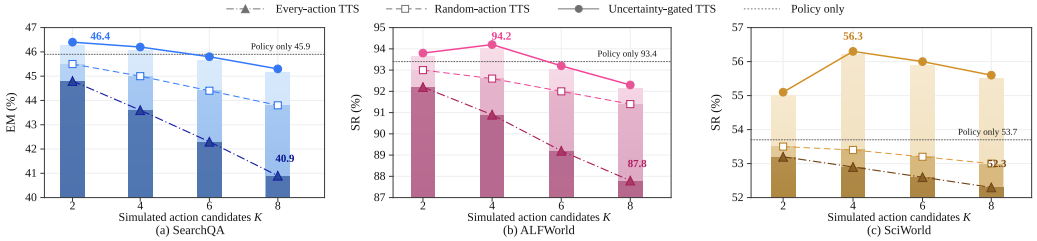


Figure 4: **Test-time scaling with frozen CoWISE-Policy and CoWISE-WM.** Uncertainty-gated TTS performs best among invocation strategies, with gains peaking at small candidate budgets.

A.2 World objectives. We next separate the roles of real-observation and behavioral supervision. Removing real-observation supervision lowers both policy-only and TTS scores across all three environments, consistent with world knowledge benefiting the policy through shared parameters. Removing behavioral supervision has little effect on policy-only scores, but TTS then reduces performance in every environment. Within behavioral learning, batch RMS normalization improves over standard GRPO in both evaluation modes; adding top-half reward weighting raises scores by a further 0.3–1.1 points.

A.3 Objective coordination. Given these distinct roles, we examine how their relative weights affect joint training. Fixed weights leave policy-only scores within 0.1 points of the full model but lower TTS scores by up to 0.8 points. Adapting the weights to real-observation fit therefore improves the use of simulated observations for test-time decisions. Training curves and objective-weight dynamics are in Appendix D.4.

5.2 Behavioral Fidelity

We evaluate whether world-model predictions preserve the trained policy's actions and task outcomes. On SearchQA, ALFWorld, and SciWorld, we compare W2W, BehR, and CoWISE-WM using the same frozen CoWISE-Policy and evaluation samples within each environment. We measure semantic fidelity, action agreement under real and simulated feedback, and task retention. Task retention is the success rate after replacing one observation, relative to using real feedback throughout. Figure 3 shows that CoWISE-WM achieves the highest action agreement and task retention in all three environments. On SearchQA, it improves action agreement from 40.0 to 46.1 and task retention from 92.2 to 95.3 over W2W, despite lower semantic fidelity. BehR, trained for behavioral fidelity under a fixed reference policy, scores below W2W on both behavioral measures, although W2W uses only real-observation supervision. Together with the policy-evolution diagnostics (Figure 1(b)) and frozen-supervisor ablation (§5.1), this gap shows that alignment learned for a fixed policy does not reliably persist as the policy evolves. CoWISE addresses this mismatch by continually updating behavioral supervision from the evolving policy. Details are in Appendix D.2.

5.3 Test-Time Scaling

Given the higher behavioral fidelity of the co-evolved world model, we test whether its simulated observations can further improve policy decisions. With both models frozen, we compare policy-only

execution with every-action, random-action, and uncertainty-gated TTS on SearchQA, ALFWorld, and SciWorld, varying the candidate count $K \in \{2, 4, 6, 8\}$. In Figure 4, uncertainty-gated TTS achieves the highest scores among invocation strategies; at $K = 4$, as used in the main results and ablations, it improves over policy-only by 0.3, 0.8, and 2.6 points, respectively. Across the full budget sweep, gains peak at $K = 2$ on SearchQA and $K = 4$ on ALFWorld and SciWorld, then decline with larger candidate sets; every-action and random-action TTS remain below policy-only across all tested candidate budgets. Under the same frozen CoWISE-Policy, CoWISE-WM also outperforms W2W and BehR under uncertainty-gated TTS in all three environments (Figure 7).

These task-level gains are consistent with the action-quality comparison in Figure 5. Among actions changed by uncertainty-gated TTS, those judged better than the original actions outnumber those judged worse in all environments, reaching 50.00% versus 14.84% on SciWorld. Together, these results show that the co-evolved world model can improve policy performance through selective use of simulated observations, with gains depending on the candidate budget (Appendix D.3).

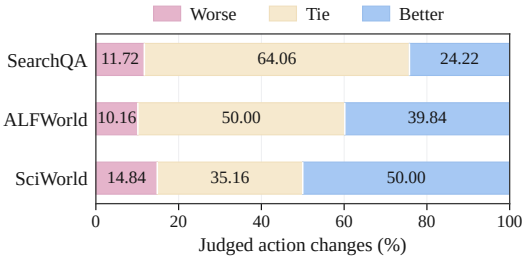


Figure 5: **Quality of actions changed by uncertainty-gated TTS.**

6 Related Work

Text-Based World Modeling. Text-based world models predict how environments respond to agent actions (Ammanabrolu & Riedl, 2021; Chae et al., 2025; Guo et al., 2025; team et al., 2025; Sun et al., 2026b). One line learns environment dynamics from agent–environment trajectories, using next-observation supervision (Li et al., 2026c; Zuo et al., 2026) or rewards for agreement with real feedback (Wu et al., 2026; Yu et al., 2026). Because matching environment feedback does not directly constrain policy behavior, another line incorporates the agent's decision needs through target construction (Cai et al., 2026) or explicitly supervises behavioral consistency using a frozen reference policy, as in BehR (Huang et al., 2026). However, behavioral fidelity under a fixed reference policy does not ensure fidelity to an evolving policy. CoWISE continually derives behavioral supervision from the evolving policy and adaptively balances it with real-observation learning.

World-Model-Assisted Policy Learning. World modeling supports policy improvement through predictive supervision and explicit simulation. One line uses environment prediction for policy initialization (Yu et al., 2026) or as auxiliary supervision during policy optimization (Lu et al., 2026; Li et al., 2026a). Another line uses explicit world models to generate simulated observations for policy learning (Sun et al., 2026a; Li et al., 2025; Chen et al., 2026). Recent methods further co-evolve policies and world models, using the learned simulation capabilities during both training and inference (Fang et al., 2025; Liu et al., 2026b; Xu et al., 2026). However, these methods do not explicitly align the evolving policy's behavior under real and simulated observations. Simulated feedback may therefore induce decisions inconsistent with real feedback, compromising policy learning and inference-time action selection. CoWISE incorporates behavioral supervision from the evolving policy into world-model updates and uses the resulting simulated observations to train the policy.

7 Conclusion

In this paper, we propose CoWISE, the first dual-level co-evolution framework that jointly trains the policy and world model, enabling the world model to maintain both semantic and behavioral fidelity as agent policies evolve. Specifically, CoWISE continually refreshes policy supervision and feeds world-model simulations into policy learning, while adaptively balancing behavioral alignment with grounding in real environmental observations. To evaluate CoWISE, we conduct extensive experiments across 8 benchmarks. Results show that CoWISE improves both policy performance and world-model behavioral fidelity. Furthermore, the jointly trained policy and world model yield further gains through uncertainty-gated test-time scaling, outperforming combinations of separately trained policies and world models. These findings support behavioral fidelity as an evolving, policy-dependent objective for training world models that assist language agents.

AI USE STATEMENT

We used Codex[2], Claude Code[3], and ChatGPT[4] to assist with implementing training and evaluation code, conducting systematic literature searches, and refining the manuscript's wording and presentation. The research ideas and hypotheses were developed independently by the authors, with AI-assisted literature searches conducted afterward to identify related work. We also used an AI model as an evaluation judge in Section 5.3. The authors reviewed and revised all manuscript text, conducted extensive independent literature searches, and manually reviewed the core training and evaluation code. We take responsibility for the final content of this work, including text, claims, code, and evaluation results produced with AI assistance.

REPRODUCIBILITY STATEMENT

An anonymized implementation of CoWISE is available at https://anonymous.4open.science/r/cowise_supplementary-6C2D. We provide the training algorithm and implementation details in Appendix B, and benchmark, baseline, and training settings in Appendix C. Details of the motivating studies and analyses are provided in Appendices A and D. Policy and world-model prompt templates are provided in Appendix F.

REFERENCES

Prithviraj Ammanabrolu and Mark Riedl. Learning knowledge graph-based world models of textual environments. *Advances in Neural Information Processing Systems*, 34:3720–3731, 2021.

Guangfeng Cai, Kaibing Yang, Shuo He, Yu Li, Shengtian Yang, Jiaqi Lv, and Lei Feng. Beyond next-observation prediction: Agent-authored world modeling for sequential decision making, 2026.

Hyungjoo Chae, Namyoung Kim, Kai Ong, Minju Gwak, Gwanwoo Song, Jihoon Kim, Sunghwan Kim, Dongha Lee, and Jinyoung Yeo. Web agents with world models: Learning and leveraging environment dynamics in web navigation. In *International Conference on Learning Representations*, volume 2025, pp. 63707–63738, 2025.

Zhaorun Chen, Zhuokai Zhao, Kai Zhang, Bo Liu, Qi Qi, Yifan Wu, Tarun Kalluri, Xuefei Cao, Yuanhao Xiong, Haibo Tong, et al. Scaling agent learning via experience synthesis. In *International Conference on Learning Representations*, volume 2026, pp. 121394–121420, 2026.

Tianqing Fang, Hongming Zhang, Zhisong Zhang, Kaixin Ma, Wenhao Yu, Haitao Mi, and Dong Yu. Webevolver: Enhancing web agent self-improvement with co-evolving world model. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 8970–8986, 2025.

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

Anisha Garg, Claire Zhang, Nishit Neema, David Bick, Ganesh Venkatesh, and Joel Hestness. Corpo: Adding a correctness bias to grpo improves generalization, 2026.

Shangmin Guo, Omar Darwiche Domingues, Raphaël Avalos, Aaron Courville, and Florian Strub. World modelling improves language model agents, 2025.

Jiuzhou Han, Wray Buntine, and Ehsan Shareghi. Towards uncertainty-aware language agent. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 6662–6685, 2024.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 6609–6625, 2020.

[2] https://openai.com/codex/
[3] https://claude.com/product/claude-code
[4] https://chatgpt.com/

Youling Huang, Guanqiao Chen, Junchi Yao, Lu Wang, Fangkai Yang, Chao Du, ChenZhuo Zhao, Pu Zhao, Qingwei Lin, Saravan Rajmohan, and Dongmei Zhang. Beyond state consistency: Behavior consistency in text-based world models, 2026.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan O Arik, Dong Wang, Hamed Zamani, and Jiawei Han. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. In *Second Conference on Language Modeling*, 2025.

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1601–1611, 2017.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466, 2019.

Cong Li, Peixi Peng, Yisen Zhao, Xinyu Hu, Shudong Liu, Zhan Su, and Zhuojian Li. Tapo: Transition-aware policy optimization for llm agents, 2026a.

Yixia Li, Hongru Wang, Peng Lai, Zhiwen Ruan, He Zhu, Youxin Zhu, Ganlong Zhao, Minda Hu, Yun Chen, Sibei Yang, Peng Li, Jeff Z. Pan, Jia Pan, Guanhua Chen, Yang Liu, and Guanbin Li. Bridging the agent-world gap: Text world models for llm-based agents, 2026b.

Yixia Li, Hongru Wang, Jiahao Qiu, Zhenfei Yin, Dongdong Zhang, Cheng Qian, Zeping Li, Xiaoteng Ma, Guanhua Chen, and Heng Ji. From word to world: Can large language models be implicit text-based world models? In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8084–8111, 2026c.

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

Youwei Liu, Jian Wang, Hanlin Wang, Beichen Guo, and Wenjie Li. Imagine-then-plan: Agent learning from adaptive lookahead with world models, 2026a.

Youwei Liu, Jian Wang, Hanlin Wang, and Wenjie Li. Comap: Co-evolving world models and agent policies for llm agents, 2026b.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding r1-zero-like training: A critical perspective, 2025.

Ning Lu, Baijiong Lin, Shengcai Liu, Jiahao Wu, Haoze Lv, Yanbin Wei, Lingting Zhu, Shengju Qian, Xin Wang, Ying-Cong Chen, Qi Wang, and Ke Tang. Policy and world modeling co-training for language agents, 2026.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: Long papers)*, pp. 9802–9822, 2023.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024.

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

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

Shuang Sun, Huatong Song, Lisheng Huang, Jinhao Jiang, Ran Le, Zhihao Lv, Zongchao Chen, Yiwen Hu, Wenyang Luo, Wayne Xin Zhao, Yang Song, Hongteng Xu, Tao Zhang, and Ji-Rong Wen. Swe-world: Building software engineering agents in docker-free environments, 2026b.

FAIR CodeGen team, Jade Copet, Quentin Carbonneaux, Gal Cohen, Jonas Gehring, Jacob Kahn, Jannik Kossen, Felix Kreuk, Emily McMilin, Michel Meyer, Yuxiang Wei, David Zhang, Kunhao Zheng, Jordi Armengol-Estapé, Pedram Bashiri, Maximilian Beck, Pierre Chambon, Abhishek Charnalia, Chris Cummins, Juliette Decugis, Zacharias V. Fisches, François Fleuret, Fabian Gloeckle, Alex Gu, Michael Hassid, Daniel Haziza, Badr Youbi Idrissi, Christian Keller, Rahul Kindi, Hugh Leather, Gallil Maimon, Aram Markosyan, Francisco Massa, Pierre-Emmanuel Mazaré, Vegard Mella, Naila Murray, Keyur Muzumdar, Peter O'Hearn, Matteo Pagliardini, Dmitrii Pedchenko, Tal Remez, Volker Seeker, Marco Selvi, Oren Sultan, Sida Wang, Luca Wehrstedt, Ori Yoran, Lingming Zhang, Taco Cohen, Yossi Adi, and Gabriel Synnaeve. Cwm: An open-weights llm for research on code generation with world models, 2025.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. musique: Multihop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554, 2022.

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.

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

Jialong Wu, Shaofeng Yin, Ningya Feng, and Mingsheng Long. Rlvr-world: Training world models with reinforcement learning. *Advances in Neural Information Processing Systems*, 38:125312–125350, 2026.

Zishan Xu, Zhiyuan Yao, Yuxin Chen, Yifu Guo, Zhengxi Lu, Yuquan Lu, Jinyang Huang, Yan Xu, Yasheng Wang, Weinan Zhang, Xingshan Zeng, and Weiwen Liu. Envace: Internalizing environment dynamics via world rehearsal for agentic reinforcement learning, 2026.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pp. 2369–2380, 2018.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *11th International Conference on Learning Representations, ICLR 2023*, 2023.

Xiao Yu, Baolin Peng, Ruize Xu, Yelong Shen, Pengcheng He, Suman Nath, Nikhil Singh, Jiangfeng Gao, and Zhou Yu. Reinforcement world model learning for llm-based agents, 2026.

Zijing Zhang, Ziyang Chen, Mingxiao Li, Zhaopeng Tu, and Xiaolong Li. Rlvmr: Reinforcement learning with verifiable meta-reasoning rewards for robust long-horizon agents. In *International Conference on Learning Representations*, volume 2026, pp. 114676–114700, 2026.

Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, Jingren Zhou, and Junyang Lin. Group sequence policy optimization, 2025.

Zilin Zhu, Chengxing Xie, Xin Lv, and slime Contributors. slime: An llm post-training framework for rl scaling. https://github.com/THUDM/slime, 2025. GitHub repository. Corresponding author: Xin Lv.

Yuxin Zuo, Zikai Xiao, Li Sheng, Fei Huang, Jianhong Tu, Yuxuan Liu, Tianyi Tang, Xiaomeng Hu, Yang Su, Qingfeng Lan, Yantao Liu, Qin Zhu, Yinger Zhang, Bowen Yu, Haiquan Zhao, Haiyang Xu, Jianxin Yang, Jiayang Cheng, Junyang Wang, Lianghao Deng, Mingfeng Xue, Tianyi Bai, Yang Fan, Yubo Ma, Yucheng Li, Zeyu Cui, Zhihai Wang, Zhihui Xie, Zhuorui Ye, An Yang, Dayiheng Liu, Jingren Zhou, and Ning Ding. Qwen-agentworld: Language world models for general agents, 2026.

TABLE OF CONTENTS

A Motivating Study Details **15**

A.1 Semantic–Behavioral Decoupling 15

A.2 Behavioral Fidelity under Policy Evolution 15

A.3 Learning Dynamics of Semantic and Behavioral Fidelity 16

B Method Details **16**

B.1 Notation and Algorithm 16

B.2 Transition Selection and Feedback Curriculum 17

B.3 Behavioral Advantage Estimation 18

B.4 Adaptive Objective Coordination 18

B.5 Efficient Asynchronous Execution 19

C Experiment Details **19**

C.1 Benchmarks 20

C.2 Baselines 22

C.3 Training Configuration 25

D Analysis Details **25**

D.1 Ablation Study 25

D.2 Behavioral Fidelity Evaluation 25

D.3 Test-Time Scaling Evaluation 27

D.4 Training Dynamics of CoWISE 28

E Case Study **31**

F Prompt Templates **34**

F.1 Policy Prompt Templates 34

F.2 World-Model Prompt Templates 36

A Motivating Study Details

We detail the three motivating studies in Section 2.2, following the order of Figure 1.

A.1 Semantic–Behavioral Decoupling

Back to ToC

We investigate whether semantic fidelity reliably reflects the behavioral fidelity of a world model on SearchQA and ALFWorld. For SearchQA, we train the policy for 500 optimization steps following the GRPO recipe of Search-R1. For ALFWorld, we train the policy for 200 optimization steps following the GRPO recipe implemented in VERL-Agent. We use the final checkpoint from each run as the frozen reference policy. The evaluated world models are the ZeroSearch retrieval simulator for SearchQA and the W2W world model for ALFWorld.

For each environment, we execute the frozen policy in the real environment and decompose the resulting trajectories into transitions (h_t, a_t, o_{t+1}). We sample 128 transitions from SearchQA and 128 transitions from ALFWorld for evaluation. Given the same interaction history h_t and executed action a_t, the world model predicts the next observation $\hat{o}_{t+1}$. We then compare $\hat{o}_{t+1}$ with the real environment observation o_{t+1} in terms of both semantic and behavioral fidelity.

We measure semantic fidelity using the cosine similarity between the real and predicted observations encoded by `intfloat/e5-base-v2`[5]. We apply mean pooling to the token representations, followed by L_2 normalization, and report the average cosine similarity on a 0–100 scale. We measure behavioral fidelity according to the action-token likelihood discrepancy defined in Eq. (1). Specifically, the frozen reference policy processes either the real observation or the corresponding world-model prediction, yielding a discrepancy d_i for each evaluation transition i. We map this discrepancy to a transition-level fidelity score $B_i = \exp(-d_i)$. To construct Figure 1(a), we group transitions into semantic-similarity bins and plot the mean semantic similarity and mean B_i within each bin, both on a 0–100 scale. Figure 1(a) shows that semantic similarity and behavioral fidelity do not improve consistently with each other in either environment, demonstrating that accurate observation reconstruction alone does not preserve the behavioral effect of environment feedback.

A.2 Behavioral Fidelity under Policy Evolution

Back to ToC

We examine whether a world model trained with a fixed behavioral reference remains aligned with policies at later stages of policy optimization. We use the SearchQA and ALFWorld BehR world models reproduced in Appendix C.2. Both world models are trained with a frozen Qwen2.5-7B-Instruct reference agent and remain fixed throughout this study.

For SearchQA, we select policy checkpoints at training steps 100, 200, 300, 400, and 500. For ALFWorld, we select checkpoints at steps 50, 75, 100, 125, and 150. We denote these checkpoints as C_1–C_5 in chronological order. For each environment, we sample 128 transitions from each checkpoint's trajectories in the real environment and pool them into a fixed set of 640 transitions. All five checkpoints are evaluated on this set, which spans different training stages.

For each transition, the fixed world model predicts the next observation from the interaction history and current action. For each policy checkpoint, we compute the mean absolute log-probability gap between the same continuation conditioned on the real and predicted observations. For ALFWorld, the gap is averaged over the executable action tokens, following Eq. (1); for SearchQA, it is averaged over all tokens in the policy response. For each transition i, we convert the resulting gap d_i into behavioral fidelity as $B_i = \exp(-d_i)$. We first average B_i over the transitions belonging to each task and then average the task-level scores equally to obtain $\bar{B}$. The behavioral mismatch score plotted in Figure 1(b) is

$$M = 100 \times (1 - \bar{B}).$$

As shown in Figure 1(b), the mismatch increases from 3.2% at C_1 to 9.6% at C_5 on ALFWorld, and from approximately 8.0% to 11.2% on SearchQA. These results show that behavioral supervision derived from a fixed reference policy becomes less aligned with the policy as GRPO training progresses.

[5] https://huggingface.co/intfloat/e5-base-v2

A.3 Learning Dynamics of Semantic and Behavioral Fidelity

Back to ToC

We analyze the learning dynamics observed while training the BehR world model for SearchQA. Following the BehR training procedure, we initialize the world model from the ZeroSearch retrieval simulator and use Qwen2.5-7B-Instruct as the frozen reference policy. The reference policy remains fixed throughout the 500 world-model training steps. We assign equal weights to the real-observation loss and the behavioral objective, corresponding to $\beta = 0.5$.

At each training step, we record two metrics from the current training batch. Real-observation NLL is the token-normalized negative log-likelihood of the real retrieval observations under the world model. Behavioral fidelity is computed from the candidate observations generated during the current on-policy rollouts. For each candidate observation, we calculate the action-token likelihood discrepancy d_i under the frozen reference policy according to Eq. (1) and convert it to $B_i = \exp(-d_i)$. We then average B_i across the candidate observations in the current rollouts.

To visualize the overall training dynamics, we divide the 500 training steps into non-overlapping 25-step intervals and report the arithmetic mean of each metric within every interval. The horizontal coordinate is the center of each interval. Consequently, the point shown at step 62 summarizes steps 50–74, while the point at step 212 summarizes steps 200–224. As shown in Figure 1(c), real-observation NLL reaches its minimum in the interval centered at step 62, whereas behavioral fidelity reaches its maximum in the interval centered at step 212. The two objectives therefore reach their best observed values at different stages of BehR training.

B Method Details

We provide the notation, training algorithm, and implementation details supporting Section 3.

B.1 Notation and Algorithm

Back to ToC

Table 3 summarizes the notation used in the four phases of Section 3. Throughout these phases, h_t denotes the history actually observed by the policy, including any simulated feedback, whereas o_{t+1} always denotes the real observation returned by the environment.

Table 3: Main notation for CoWISE.

Notation	Definition
$k,\ t$	Training iteration and interaction step within a trajectory.
$\theta_k,\ \pi_k,\ W_k$	Shared parameters, policy, and world model at iteration k.
$\tau,\ R(\tau)$	Policy trajectory and its environment-provided task reward.
$h_t,\ a_t,\ c_t$	Observed history, policy response, and world context combining the world-model system prompt with (h_t, a_t).
$o_{t+1},\ \hat{o}_{t+1}^{(j)},\ \tilde{o}_{t+1}$	Real observation, candidate simulated observation, and feedback added to the policy history.
$G_{\mathrm{pol}},\ G_{\mathrm{wm}}$	Trajectories per task and candidate observations per retained transition.
$\mathcal{B}_k^{\mathrm{pol}}$	Policy buffer containing $(\tau, R(\tau))$.
$\mathcal{B}_k^{\mathrm{ROL}}$	Real-observation buffer containing (c_t, o_{t+1}).
$\mathcal{B}_k^{\mathrm{Beh}}$	Behavioral buffer containing $(c_t, o_{t+1}, \{\hat{o}_{t+1}^{(j)}\}_{j=1}^{G_{\mathrm{wm}}})$.
$s_t,\ \alpha$	Transition selection score and retained fraction.
$p_k,\ p_{\max},\ K_{\mathrm{ramp}}$	Simulated-feedback probability, its maximum, and the number of iterations over which it increases.
$r_{t,j}^{\mathrm{beh}},\ q_t,\ A_{t,j}^{\mathrm{beh}}$	Candidate reward, top-half mean reward, and behavioral advantage.
$\mathcal{D}_{\mathrm{cal}}$	Fixed out-of-distribution calibration set for monitoring real-observation NLL.
$\bar{\ell}_k,\ \ell_k^\star,\ d_k$	Unweighted calibration NLL, its historical minimum, and their difference.
$\beta_k,\ \delta$	Behavioral-objective weight and relative perplexity tolerance.

Algorithm 1 combines the three objectives in the shared-parameter update of Eq. (4). For world-model learning, transitions without a valid behavioral comparison contribute only to ROL.

Algorithm 1 CoWISE Training

Require: Initial parameters θ_0; tasks $\mathcal{D}$ and environment $\mathcal{E}$; fixed calibration set $\mathcal{D}_{\text{cal}}$; training iterations T
Require: Group sizes $G_{\text{pol}}, G_{\text{wm}}$; selection fraction α; $p_{\max}, K_{\text{ramp}}$; tolerance δ
$\ell^\star \leftarrow +\infty$; $\beta_0 \leftarrow 0.5$
for $k = 0, \ldots, T-1$ **do**
  // Phase 1: Policy–World Rollout
  Use θ_k as π_k and W_k; sample a task batch from $\mathcal{D}$
  Set p_k using Eq. (7)
  Populate $\mathcal{B}_k^{\text{pol}}, \mathcal{B}_k^{\text{ROL}}, \mathcal{B}_k^{\text{Beh}}$ by the rollout in Section 3.1
  Use G_{pol} trajectories per task and G_{wm} candidates per retained transition,
  with transition selection and feedback probabilities from Appendix B.2
  // Phase 2: Policy Learning
  Compute $A_\tau^{\text{pol}} = [R(\tau) - \bar{R}_x]/(\sigma_x + \varepsilon)$ within each task group
  Compute $\mathcal{L}_{\text{Policy}}(\theta)$ from $\mathcal{B}_k^{\text{pol}}$ using Eq. (2)
  // Phase 3: Semantic–Behavioral World Learning
  Compute $\mathcal{L}_{\text{ROL}}(\theta)$ from $\mathcal{B}_k^{\text{ROL}}$ as in Section 3.3
  Score candidates with $r_{t,j}^{\text{beh}} = \exp[-d_t(\hat{o}_{t+1}^{(j)}; \pi_k)]$ using Eq. (1)
  Compute $A_{t,j}^{\text{beh}}$ over $\mathcal{B}_k^{\text{Beh}}$ using Eq. (3)
  Compute $\mathcal{L}_{\text{Beh}}(\theta)$ using Eq. (2)
  // Phase 4: Adaptive Objective Coordination
  $\bar{\ell}_k \leftarrow \mathcal{L}_{\text{ROL}}(\theta_k; \mathcal{D}_{\text{cal}})$; $\ell^\star \leftarrow \min\{\ell^\star, \bar{\ell}_k\}$; $d_k \leftarrow \bar{\ell}_k - \ell^\star$
  if $(k+1) \bmod 10 = 0$ **then**
    Compute β_{k+1} using Eq. (5)
  else
    $\beta_{k+1} \leftarrow \beta_k$
  end if
  $\mathcal{L}_k(\theta) \leftarrow \mathcal{L}_{\text{Policy}}(\theta) + (1-\beta_k)\mathcal{L}_{\text{ROL}}(\theta) + \beta_k \mathcal{L}_{\text{Beh}}(\theta)$
  $\theta_{k+1} \leftarrow \text{OptimizerStep}(\theta_k, \mathcal{L}_k)$
end for

B.2 Transition Selection and Feedback Curriculum

Back to ToC

Transition selection. Section 3.1 uses next-action uncertainty to prioritize decision-relevant transitions while reducing world-model computation. For each transition (h_t, a_t, o_{t+1}), let a_{t+1}^{sel} be the next response generated under the policy's actual history h_{t+1}, which includes the received feedback $\tilde{o}_{t+1}$. Let $\mathcal{I}_{t+1}^{\text{sel}}$ contain the positions of its executable-action tokens. We score the transition by their mean negative log-probability (Lu et al., 2026):

$$s_t = -\frac{1}{|\mathcal{I}_{t+1}^{\text{sel}}|} \sum_{i \in \mathcal{I}_{t+1}^{\text{sel}}} \log \pi_k\big(a_{t+1,i}^{\text{sel}} \mid h_{t+1}, a_{t+1,<i}^{\text{sel}}\big). \tag{6}$$

Higher scores indicate less predictable next actions. We collect completed task groups into a selection window whose size is the rollout batch size in Table 9. Within each window, we form separate pools $\mathcal{C}$ for real and simulated feedback, excluding terminal transitions and those without a valid next action. From each pool, we retain the top $\lceil \alpha|\mathcal{C}| \rceil$ transitions with $\alpha = 0.25$.

Feedback curriculum. Section 3.1 introduces simulated feedback gradually to limit early exposure to inaccurate predictions. We set its probability at iteration k to

$$p_k = \frac{p_{\max}}{2}\left[1 - \cos\left(\pi \min\left\{\frac{k}{K_{\text{ramp}}}, 1\right\}\right)\right], \tag{7}$$

where K_{ramp} is the ramp duration in Table 9, measured in training iterations, and $p_{\max} = 0.3$. The probability remains constant once $k \geq K_{\text{ramp}}$. At each interaction step, the policy receives a sampled observation from W_k with probability p_k and the real observation o_{t+1} otherwise. Appendix B.5 details feedback generation during interaction and candidate completion after selection.

B.3 Behavioral Advantage Estimation

Back to ToC

Section 3.3 weights behavioral updates using the fidelity of candidate observations under the current policy. We compare this estimator with standard GRPO:

$$\widetilde{A}_{t,j} = \frac{r^{\text{beh}}_{t,j} - \bar{r}^{\text{beh}}_{t}}{\sigma^{\text{beh}}_{t} + \varepsilon}, \tag{8}$$

where σ^{beh}_{t} is the within-group reward standard deviation. Centering removes the reward level, while division by σ^{beh}_{t} can amplify small differences when ε is negligible. An above-mean candidate can therefore receive a large positive advantage despite low behavioral fidelity.

In two preliminary snapshots using semantic–behavioral product rewards, approximately 31,700 of 609,106 valid four-candidate groups had all rewards below 0.5 and a maximum positive advantage above 1. Here, 0.5 serves only as a reporting threshold.

To retain fidelity information in Eq. (3), we multiply centered rewards by the top-half mean q_t. With $G_{\text{wm}} = 4$, both highest-scoring candidates contribute to this weight, favoring groups whose better candidates achieve higher behavioral fidelity. We then normalize the weighted rewards using

$$\text{RMS}_{\mathcal{B}^{\text{Beh}}_{k}}(v) = \sqrt{\frac{1}{N}\sum_{t,j} v^2_{t,j}}, \tag{9}$$

where N counts all batch candidates, which share a common normalization scale.

Table 4 illustrates the role of the top-half mean. Both groups have mean reward 0.5, so their reward-0.9 candidates have the same centered reward 0.4. Weighting by the full-group mean gives these candidates equal advantages under shared RMS normalization. Using q_t instead gives the candidates in Group A greater relative weight because that group contains two high-fidelity candidates.

Table 4: Advantages at reward 0.9 in a batch containing both groups ($\varepsilon = 0$). GRPO uses population standard deviations; group-mean weighting replaces q_t with $\bar{r}^{\text{beh}}_{t}$.

	Group A	**Group B**
Candidate rewards	$(0.1, 0.1, 0.9, 0.9)$	$(0.3, 0.4, 0.4, 0.9)$
Group mean	0.5	0.5
Top-half mean q_t	0.90	0.65
Standard GRPO	1.000	1.706
Group-mean weighting + batch RMS	1.220	1.220
CoWISE (Top2 + batch RMS)	1.302	0.941

Since $q_t > 0$, weighting preserves the sign of each centered reward. It reduces low-fidelity groups' relative contribution while retaining positive advantages for candidates above their group mean.

B.4 Adaptive Objective Coordination

Back to ToC

Section 3.4 adjusts β_k according to the world model's fit to real observations. We derive its NLL budget from a relative perplexity tolerance, then use the remaining budget to define the weight update. For each environment, we construct $\mathcal{D}_{\text{cal}}$ from 128 out-of-distribution transitions, using released ZeroSearch (Sun et al., 2026a) data for SearchQA and W2W (Li et al., 2026c) data for ALFWorld and SciWorld. We convert each transition into (c_t, o_{t+1}) and keep the set fixed throughout training, so NLL values across iterations are evaluated on the same examples. For its unweighted ROL, $\bar{\ell}_k = \mathcal{L}_{\text{ROL}}(\theta_k; \mathcal{D}_{\text{cal}})$, define

$$\ell^\star_k = \min_{0 \le j \le k} \bar{\ell}_j, \qquad d_k = \bar{\ell}_k - \ell^\star_k. \tag{10}$$

The corresponding perplexities $P_k = e^{\bar{\ell}_k}$ and $P^\star_k = e^{\ell^\star_k}$ satisfy

$$\frac{P_k}{P^\star_k} = e^{d_k}, \qquad \frac{P_k}{P^\star_k} \le 1 + \delta \iff d_k \le \log(1+\delta). \tag{11}$$

Thus, a relative perplexity tolerance δ gives the NLL budget $\log(1+\delta)$.

We use the remaining budget as a bounded adjustment to the log weight ratio $z_k = \log[\beta_k/(1-\beta_k)]$:

$$z_{k+1} = z_k + \mathrm{clip}(\log(1+\delta) - d_k, -\log 2, \log 2)\,. \tag{12}$$

Applying $\beta_{k+1} = \sigma(z_{k+1})$ yields Eq. (5), increasing behavioral weight when the budget remains available and reducing it when the budget is exceeded.

We use $\delta = 0.05$ and $\beta_0 = 0.5$. The historical minimum starts at $\ell_0^\star = \bar{\ell}_0$ and is updated at every iteration. We update β_{k+1} after every ten completed iterations, when $(k+1) \bmod 10 = 0$; otherwise, $\beta_{k+1} = \beta_k$. Iteration k uses β_k in Eq. (4).

B.5 Efficient Asynchronous Execution

Back to ToC

The rollout in Section 3.1 requires both interactive feedback and world-model training data. We separate their preparation so that the policy need not wait for a complete candidate group at each action. On the simulated-feedback route, we initially generate only the candidate used for interaction, while the real environment step runs concurrently. After selecting transitions from a window of completed trajectory groups, background workers complete the selected candidate groups and compute behavioral rewards while other trajectories continue (Figure 6). Stateful environments retain their required step ordering.

To keep model requests progressing during environment waits, we acquire generation permits per request and release them upon completion. Separate concurrency limits for trajectories, generation, and scoring let other ready requests proceed while a trajectory awaits external feedback. We also reuse the candidate generated for interaction when its model version matches the group. When a transition uses real feedback, its next policy response supplies the reference response for behavioral scoring. We score this response under the real and all candidate observations in one teacher-forcing batch, avoiding separate response generation for each condition.

Table 5 compares trajectory-level and request-level admission with the same model, data, batch sizes, and GPU allocation. Both configurations cap concurrent generation requests at 128, while request-level admission independently allows up to 512 concurrent trajectories, compared with 128 in the baseline. Across updates 2–5, median rollout time decreases by 23.2% (1.30× speedup), and median training-step time decreases by 17.2% (1.21× speedup).

Table 5: **Effect of request-level scheduling.** Median wall-clock times over updates 2–5 of five-update runs, using 16 rollout GPUs and 16 learner GPUs.

Scheduling	Rollout time (s) ↓	Speedup ↑	Training-step time (s) ↓	Speedup ↑
Trajectory-level admission	73.29	1.00×	89.28	1.00×
Request-level admission (ours)	**56.29**	**1.30×**	**73.92**	**1.21×**

Finalized groups enter a bounded ready queue. Building on Slime's asynchronous rollout–learner pipeline (Zhu et al., 2025), trajectory collection and preparation of world-model training data overlap learner updates. Policy and world-model requests share the rollout inference pool, while the learner uses a separate GPU pool. Weight synchronization pauses inference; generation resumes after the updated weights are installed.

Complementary traces from a CoWISE training run confirm that world supervision overlaps policy rollout. Candidate completion and behavioral scoring for one batch span 133.10 seconds, all overlapping policy action generation in trajectories delivered in the next batch.

C Experiment Details

We detail the benchmarks, baselines, and training configuration used in Section 4.

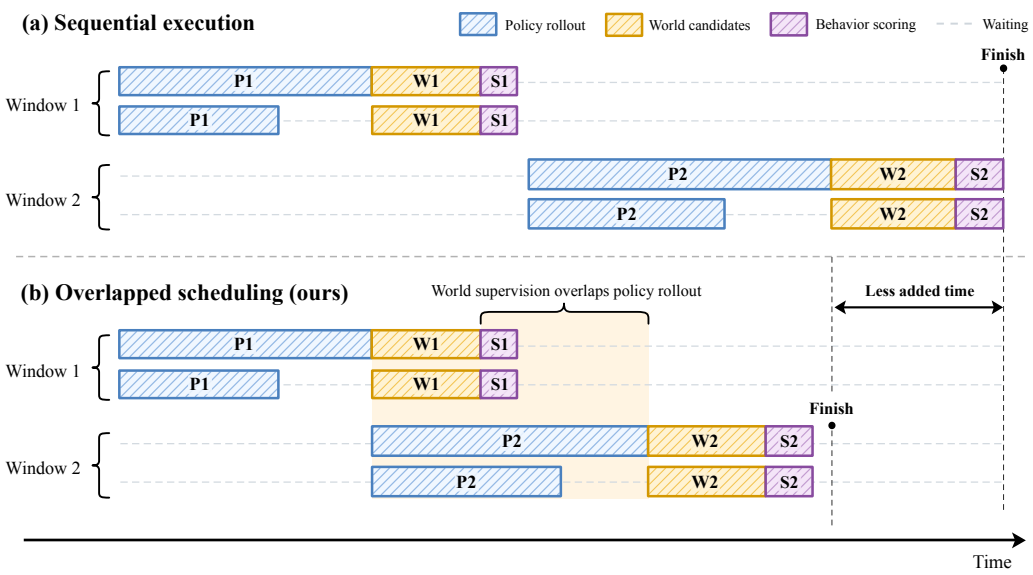


Figure 6: **Overlapping world supervision with policy rollout.** For selection window i, P_i denotes policy rollout, W_i candidate completion, and S_i behavioral scoring. (a) Sequential execution places world supervision after each rollout window. (b) Our schedule overlaps W_1 and S_1 with P_2 in the shared inference pool. The timelines are schematic and not to scale.

C.1 BENCHMARKS

Back to ToC

Table 6 summarizes the datasets, environments, and agent interaction protocols.

Table 6: Overview of the benchmarks, training and test set sizes, evaluation metrics, and interaction protocols used in our experiments.

Benchmark	Training	Test	Metric	Max Turns	Input	Policy Output
NQ (Kwiatkowski et al., 2019)	169,615	3,610	EM	4	4,096	512
TriviaQA (Joshi et al., 2017)	169,615	11,313	EM	4	4,096	512
PopQA (Mallen et al., 2023)	169,615	14,267	EM	4	4,096	512
HotpotQA (Yang et al., 2018)	169,615	7,405	EM	4	4,096	512
2WikiMultiHopQA (Ho et al., 2020)	169,615	12,576	EM	4	4,096	512
MuSiQue (Trivedi et al., 2022)	169,615	2,417	EM	4	4,096	512
ALFWorld (Shridhar et al., 2021)	3,553	274	SR	30	8,192	512
SciWorld (Wang et al., 2022)	3,322	1,661	SR	30	8,192	512

C.1.1 SEARCHQA

Environment. We adopt the Search-R1 retrieval environment (Jin et al., 2025), using the wiki-18 corpus as the external knowledge source. We use the E5 dense retriever with the `intfloat/e5-base-v2` encoder and the released corpus embeddings stored in a GPU-enabled FAISS Flat index. Queries are prefixed with `query:`, tokenized with a maximum length of 256 tokens, and encoded using mean pooling followed by L_2 normalization. For each search action, the retriever returns the top three passages. Each passage is presented with its rank, title, and text, and the retrieved results are enclosed within `<information>` and `</information>` tags. The agent issues queries using `<search>...</search>` and submits its final answer using `<answer>...</answer>`. Each episode allows at most four interaction turns, with either a

search action or a final answer at each turn. Episodes that exhaust this budget without a final answer receive zero task reward.

Training Data. Following the Search-R1 data preparation pipeline, we combine training questions from Natural Questions and HotpotQA. The frozen training dataset contains 169,615 question–answer examples. Each example provides a question formatted as an agent prompt and a set of reference answers used for task-reward computation. We apply the backbone model's chat template to the prompts and shuffle the training data with a data seed of 42. During training, the policy interacts with the retrieval environment to generate trajectories. The resulting search queries and real retrieval observations provide the action-conditioned supervision for world-model learning, while the reference answers determine the outcome-based policy reward.

Test Data and Evaluation. We evaluate on the complete Search-R1-formatted evaluation sets of six benchmarks: Natural Questions, TriviaQA, PopQA, HotpotQA, 2WikiMultiHopQA, and MuSiQue. Table 6 lists their evaluation sizes, totaling 51,588 questions. For policy-only evaluation, all search actions receive feedback from the real retriever. We generate one trajectory per question using greedy decoding, with a maximum context length of 4,096 tokens, a response limit of 512 tokens per turn, and at most four interaction turns. Performance is measured using Exact Match (EM). We extract the final answer from the last complete `<answer>...</answer>` block and normalize both the prediction and reference answers by lowercasing, removing punctuation and articles, and normalizing whitespace. A prediction is counted as correct if it exactly matches any normalized reference answer; responses without an answer block receive a score of zero. We report EM for each benchmark and the unweighted mean across the six benchmarks.

C.1.2 ALFWORLD

Environment. ALFWorld (Shridhar et al., 2021) is a text-based environment for household tasks, where an agent follows a task instruction by navigating the environment and manipulating objects through textual commands. We use ALFWorld v0.4.2 with the `AlfredTWEnv` backend and an environment adapter adapted from VERL-Agent. The environment covers six task types: placing an object, examining an object under a light, cleaning and placing an object, heating and placing an object, cooling and placing an object, and placing two objects. At each step, the agent receives a textual observation and the admissible commands for the current state, and submits an action within `<action>...</action>` tags. We disable domain randomization and allow at most 30 interaction steps per episode. Following the BEACON configuration (Wang et al., 2026), we retain the full interaction history, subject to a maximum prompt length of 8,192 tokens, and limit each response to 512 tokens.

Training Data. We use the official training split from the `json_2.1.1` game release, containing 3,553 games across the six task types. Each training instance identifies a specific game and its corresponding task instruction, from which the simulator initializes an episode. We preserve the official split assignments and shuffle the training instances with a data seed of 42. During training, the policy generates trajectories through interaction with the environment. Real observations following executed actions provide supervision for world-model learning, while the simulator's `won` signal determines task success and the binary outcome reward.

Test Data and Evaluation. We combine the 140 games in `valid_seen` and 134 games in `valid_unseen` into a single evaluation set of 274 games. For policy-only evaluation, the agent interacts with the real ALFWorld simulator, with one trajectory per evaluated game and a maximum of 30 interaction steps. Responses are sampled at a temperature of 0.4, using the same prompt and response limits as during training. We measure performance using Success Rate (SR), defined as the percentage of all 274 games for which the simulator reports `won=True`. Episodes that reach the interaction limit without success count as failures.

C.1.3 SCIWORLD

Environment. ScienceWorld (SciWorld) (Wang et al., 2022) is a text-based environment for scientific experimentation, where an agent completes tasks by inspecting objects, manipulating materials, and operating equipment through textual commands. We use ScienceWorld v1.2.2 and follow the

Method	Capabilities		World-Model Supervision		
	Policy	World Model	Real-Observation Supervision	Behavioral Supervision	Adaptive Semantic–Behavioral Weighting
GRPO (Shao et al., 2024)	✓	✗	✗	✗	✗
W2W (Li et al., 2026c)	✗	✓	✓	✗	✗
BehR (Huang et al., 2026)	✗	✓	✓	✓	✗
PaW (Lu et al., 2026)	✓	✗	✓	✗	✗
TAPO (Li et al., 2026a)	✓	✗	✓	✗	✗
CoWISE (Ours)	✓	✓	✓	✓	✓

Table 7: **Comparison of policy and world-model methods.** Checks indicate explicitly trained roles or objectives. Real-observation supervision trains prediction of environment feedback. BehR uses a fixed reference policy for behavioral supervision, whereas CoWISE uses the evolving policy. Adaptive weighting balances real-observation and behavioral objectives.

environment settings of RLVMR (Zhang et al., 2026) and SelfSum, including the `easy` simplification preset. At the beginning of each episode, we provide the agent with the task description and the complete set of executable action templates. At each step, the agent receives a textual observation and information about the objects currently available for manipulation, which it uses to instantiate an action template into a concrete command. Each episode allows at most 30 interaction steps. We retain the full interaction history within an 8,192-token prompt budget and limit each response to 512 tokens.

Training Data. We adopt the L0 data split used by RLVMR and BEACON (Zhang et al., 2026; Wang et al., 2026), which contains 3,322 training instances across 29 task types and excludes the `use-thermometer` task. Each instance is a task–variation pair: the task specifies the objective, and the variation identifies a concrete configuration of the simulator. We initialize each episode using its task name and variation ID, preserving the assignments in the released L0 split index. Training instances are shuffled with a data seed of 42. Policy interactions generate action-conditioned transitions for world-model learning, with the subsequent textual observation serving as the prediction target. The action grammar is supplied separately in the prompt.

Test Data and Evaluation. The L0 test split contains 1,661 held-out task–variation pairs across the same 29 task types. Evaluation uses the L0 split assignments and the same `easy` simplification preset as training. For policy-only evaluation, all actions are executed in the real ScienceWorld simulator. The evaluation protocol generates one trajectory per evaluated instance at a sampling temperature of 0.4, with at most 30 interaction steps and the same prompt and response limits as during training. We report Success Rate (SR) as the percentage of evaluated instances that terminate successfully. In our implementation, success requires native episode termination together with either a positive `won`/`completed` flag or a normalized task score of 1.0. Episodes that exhaust the interaction budget without satisfying this criterion count as failures.

C.2 Baselines

Back to ToC

We compare CoWISE against three categories of baselines: prompting methods, policy-optimization methods, and world-model-assisted methods. Table 7 compares the training design of representative methods. W2W and BehR appear as world-model methods here; Table 1 evaluates each with a GRPO policy.

C.2.1 Prompting Baselines

Direct Prompt. Direct Prompt uses the instruction-tuned backbone to generate answers or environment actions directly from the task instruction and the available interaction context, without any task-specific parameter updates.

ReAct. ReAct (Yao et al., 2023) interleaves language-based reasoning with environment actions, allowing the model to update its reasoning after receiving each new observation. It provides a

training-free agent baseline that explicitly exposes intermediate reasoning during multi-step interaction.

C.2.2 POLICY-OPTIMIZATION BASELINES

PPO. Proximal Policy Optimization (PPO) (Schulman et al., 2017) is an actor–critic algorithm that optimizes a clipped policy objective using advantages estimated by a learned value model. The clipping mechanism constrains the policy update relative to the behavior policy to improve training stability.

GRPO. Group Relative Policy Optimization (GRPO) (Shao et al., 2024) samples a group of responses or trajectories for the same task and estimates their advantages from relative rewards within the group. It removes the learned critic used by PPO and directly applies the group-relative advantages to policy optimization.

GSPO. Group Sequence Policy Optimization (GSPO) (Zheng et al., 2025) replaces token-level importance ratios with a sequence-level ratio derived from the likelihood of the complete response. It performs clipping and optimization at the sequence level, aligning the optimization unit with sequence-level rewards.

GiGPO. Group-in-Group Policy Optimization (GiGPO) (Feng et al., 2026) combines episode-level relative advantages with step-level relative advantages constructed from repeated anchor states across trajectories. This hierarchical grouping provides finer-grained credit assignment for multi-turn agents without introducing a critic or collecting additional rollouts.

C.2.3 WORLD-MODEL-ASSISTED METHODS

ZeroSearch. ZeroSearch (Sun et al., 2026a) replaces the real search engine during reinforcement learning with an LLM-based retrieval simulator trained to generate both useful and noisy documents. A curriculum progressively increases the probability of noisy simulated retrieval results, allowing the policy to learn search behavior without repeatedly querying a real retriever.

PaW. PaW (Lu et al., 2026) jointly trains policy behavior and world-modeling ability within the same model by augmenting on-policy reinforcement learning with an auxiliary next-observation prediction objective. It selects informative transitions using action entropy and balances the policy and world-modeling losses according to rollout rewards.

TAPO. TAPO (Li et al., 2026a) alternates standard policy-optimization updates with transition-supervision updates on action-conditioned next-observation prediction. Both objectives update a shared backbone using transitions collected from existing rollouts, requiring no additional environment interactions or inference-time simulation. Together, PaW and TAPO provide comparisons with shared-backbone policy learning augmented by next-observation prediction.

W2W + GRPO. From Word to World (W2W) (Li et al., 2026c) trains an LLM-based world model on real interaction trajectories to predict the next textual observation conditioned on the interaction history and the current action. Our W2W + GRPO baseline combines a standard GRPO-trained policy with the fixed W2W world model at test time; the policy receives no additional planning-specific training.

We adapt BehR's lookahead planning (Huang et al., 2026) to the common uncertainty-gated TTS protocol, using $K = 4$ and the same calibration rule, candidate-proposal prompt, and action-selection prompt as CoWISE (Appendix D.3). Simulation is invoked when the mean token entropy of the proposed action exceeds the threshold calibrated for the GRPO policy; otherwise, the original action is executed directly. When triggered, the GRPO policy supplies four candidate actions for the current interaction context. The W2W world model then independently predicts the next observation resulting from each candidate action. Given the current interaction context and the four action–prediction pairs, the same policy selects the action whose predicted consequence best advances the task objective. The selected action is executed in the actual environment, and the resulting observation is appended to the interaction history before the next planning step. Here, K denotes

Table 8: **Hyperparameters used to train the three BehR world models.** We use environment-specific base models, datasets, and reference agents, while keeping the optimization and rollout configurations identical across environments.

Item	SearchQA	ALFWorld	SciWorld
Base model	ZeroSearch	W2W-ALFWorld	W2W-SciWorld
Dataset	ZeroSearch	W2W	W2W
Framework	verl v0.7.1	verl v0.7.1	verl v0.7.1
RL algorithm	GRPO	GRPO	GRPO
Optimizer	AdamW	AdamW	AdamW
Learning rate	5×10^{-6}	5×10^{-6}	5×10^{-6}
Train batch size	36	36	36
PPO mini-batch size	36	36	36
Max tokens / GPU	16,384	16,384	16,384
Max prompt length	14,336	14,336	14,336
Max response length	1,024	1,024	1,024
KL loss	On, coefficient 0.001	On, coefficient 0.001	On, coefficient 0.001
Rollout backend	vLLM	vLLM	vLLM
Rollouts per prompt (n)	5	5	5
Rollout temperature	1.3	1.3	1.3
Top-p	1.0	1.0	1.0
Total epochs	5	5	5

the number of candidate actions evaluated at a triggered step. The K world-model predictions are batched in our implementation.

BehR + GRPO. BehR (Huang et al., 2026) trains a world model to preserve the behavioral effect of environment feedback, using a frozen reference policy to compare the likelihood of the logged next action under real and predicted observations. The world model is optimized with GRPO using this behavior-consistency reward and serves as a fixed-reference, behavior-aware world-model baseline. At test time, BehR + GRPO follows the same uncertainty-gated TTS procedure with $K = 4$, using the same GRPO-trained policy, calibrated threshold, candidate-proposal prompt, and action-selection prompt as W2W + GRPO. The only difference is that the W2W world model is replaced by the corresponding BehR-trained world model. This controlled setup isolates the effect of behavior-consistency training on the quality of the predicted consequences used for planning.

The original BehR experiments train and release world models only for TextWorld and WebShop, which do not cover any of our three evaluation environments. We therefore reproduce the BehR training algorithm and train a separate behavior-aligned world model for SearchQA, ALFWorld, and SciWorld. Across all three environments, we retain the original BehR objective and optimization procedure, including its frozen reference policy and logged-action likelihood-gap reward.

- **SearchQA.** We initialize the SearchQA world model from Simulation_LLM_wiki_7B_V2[6], released by ZeroSearch (Sun et al., 2026a), and train it on ZeroSearch_dataset[7]. The initialization provides a retrieval simulator adapted to Wikipedia-style search results, while the dataset supplies the SearchQA instances used to construct BehR training trajectories.
- **ALFWorld and SciWorld.** For ALFWorld, we initialize from the WorldModel-Alfworld-Qwen2.5-7B[8] released by W2W (Li et al., 2026c). For SciWorld, we use the corresponding WorldModel-Sciworld-Qwen2.5-7B[9]. Both models are trained with the corresponding interaction trajectories from LLMasWorldModels[10]. This setup preserves the environment-specific state and observation formats of W2W while applying the same BehR training algorithm in both domains.

[6] https://huggingface.co/Alibaba-NLP/Simulation_LLM_wiki_7B_V2
[7] https://huggingface.co/datasets/Alibaba-NLP/ZeroSearch_dataset
[8] https://huggingface.co/X1AOX1A/WorldModel-Alfworld-Qwen2.5-7B
[9] https://huggingface.co/X1AOX1A/WorldModel-Sciworld-Qwen2.5-7B
[10] https://huggingface.co/datasets/X1AOX1A/LLMasWorldModels

C.3 Training Configuration

Back to ToC

Table 9 lists the settings for Algorithm 1. SearchQA simulated observations are limited to 1,024 tokens. We use Slime (Zhu et al., 2025) with 16 GPUs each for rollout generation and training.

Table 9: Hyperparameters used to train CoWISE on SearchQA, ALFWorld, and SciWorld.

Group	Hyperparameter	SearchQA	ALFWorld	SciWorld
Rollout	Rollout batch size	256	16	16
	Trajectories per task G_{pol}	8	8	8
	Sampling temperature	1.0	1.0	1.0
	Max policy response length	512	512	512
	Max interaction turns	4	30	30
Training	Training iterations	500	150	150
	Global batch size	2,048	128	128
	Learning rate	1×10^{-6}	1×10^{-6}	1×10^{-6}
	Clip range (ϵ_-, ϵ_+)	$(0.2, 0.28)$	$(0.2, 0.28)$	$(0.2, 0.28)$
CoWISE	World-model candidates G_{wm}	4	4	4
	Transition selection fraction α	0.25	0.25	0.25
	Initial behavioral weight β_0	0.5	0.5	0.5
	Perplexity tolerance δ	0.05	0.05	0.05
	β update interval (iterations)	10	10	10
	Maximum simulated-feedback probability $p_{\max}$	0.3	0.3	0.3
	Feedback ramp K_{ramp} (iterations)	500	150	150

D Analysis Details

We provide the settings and evaluation procedures for Section 5. Additional training curves follow these analyses.

D.1 Ablation Study

Back to ToC

All variants in Table 2 retain the full configuration except for the specified intervention. We evaluate each variant using its own policy and world model, with a common uncertainty-gated TTS candidate budget of $K = 4$ in all three environments, matching the main results. The variant without simulated training feedback retains world-model training. The variant without behavioral supervision uses $\beta = 0$. The variant without real-observation supervision uses $\beta = 1$ and removes all ROL updates, including fallback supervision on transitions without a valid behavioral comparison. The variant with fixed objective weights uses $\beta = 0.5$ throughout training.

The RMS-only variant removes top-half reward weighting by setting $q_t = 1$ in Eq. (3), retaining shared batch RMS normalization of centered rewards. The standard-GRPO variant instead normalizes by the within-group standard deviation without quality weighting (Eq. (8)).

D.2 Behavioral Fidelity Evaluation

Back to ToC

We evaluate whether the observations generated by a world model preserve both the semantic content of real environment feedback and its effect on subsequent policy decisions. We compare W2W, BehR, and CoWISE-WM on SearchQA, ALFWorld, and SciWorld using the same frozen policy and evaluation examples.

D.2.1 Evaluation Models and Data

For SearchQA, we use the CoWISE checkpoint after 500 training steps as both the frozen CoWISE-Policy and CoWISE-WM. For ALFWorld and SciWorld, we use the corresponding checkpoints

after 150 training steps. Because CoWISE uses a shared language-model backbone, the policy and world-model roles are selected through their respective prompting templates.

For each environment, we sample 64 transitions from trajectories generated by the frozen CoWISE-Policy in the real environment. W2W, BehR, and CoWISE-WM predict the next observation for exactly the same transitions. Both world-model prediction and subsequent policy-action generation use greedy decoding with a maximum response length of 512 tokens.

For the task-retention evaluation, we sample 64 tasks from each environment. SearchQA tasks are sampled evenly across Natural Questions, TriviaQA, PopQA, HotpotQA, 2WikiMultiHopQA, and MuSiQue. ALFWorld tasks are sampled from the official `valid_unseen` split, and SciWorld tasks are sampled from the L0 test split.

D.2.2 Semantic Fidelity and Action Agreement

For each transition i, let o_i denote the real observation and $\hat{o}_i^W$ denote the observation predicted by world model W. We obtain normalized representations using `intfloat/e5-base-v2`, with the `query:` prefix applied to both observations. Transition-level semantic fidelity is

$$s_i^{\text{sem}}(W) = \cos\left(e(o_i), e(\hat{o}_i^W)\right), \tag{13}$$

where $e(\cdot)$ denotes the normalized E5 representation.

To measure action agreement, we construct two otherwise identical policy contexts. The first contains the real observation o_i, while the second contains the world-model prediction $\hat{o}_i^W$. The frozen policy generates one action from each context using greedy decoding:

$$a_i^{\text{real}} = \text{Act}\left(\pi(h_i, o_i)\right), \qquad a_i^W = \text{Act}\left(\pi(h_i, \hat{o}_i^W)\right). \tag{14}$$

For SearchQA, an action can be either a search query or a final answer. Let $\tau(a) \in \{\text{search}, \text{answer}\}$ denote its action type and $c(a)$ its textual content. We define transition-level action agreement as

$$s_i^{\text{act}}(W) = \begin{cases} \cos\left(e(c(a_i^{\text{real}})), e(c(a_i^W))\right), & \tau(a_i^{\text{real}}) = \tau(a_i^W) = \text{search}, \\ \mathbb{I}\left[\text{norm}(c(a_i^{\text{real}})) = \text{norm}(c(a_i^W))\right], & \tau(a_i^{\text{real}}) = \tau(a_i^W) = \text{answer}, \\ 0, & \tau(a_i^{\text{real}}) \neq \tau(a_i^W) \text{ or } a_i^W \text{ is invalid.} \end{cases} \tag{15}$$

Here, $\text{norm}(\cdot)$ applies case folding and whitespace normalization. Search queries are encoded using the same `intfloat/e5-base-v2` encoder.

For ALFWorld and SciWorld, we compare the normalized executable commands:

$$s_i^{\text{act}}(W) = \mathbb{I}\left[\text{norm}(a_i^{\text{real}}) = \text{norm}(a_i^W)\right], \tag{16}$$

where normalization applies case folding and whitespace normalization. An invalid simulated action receives a score of zero. For ALFWorld, both policy contexts retain the admissible-action set returned by the real environment, so the comparison isolates the effect of replacing the observation text.

Let $\mathcal{I}_q$ denote the evaluated transitions belonging to task q, and let $\mathcal{Q}$ denote the set of evaluated tasks. We first average transition scores within each task and then average equally across tasks:

$$\text{SemFid}(W) = \frac{100}{|\mathcal{Q}|} \sum_{q \in \mathcal{Q}} \frac{1}{|\mathcal{I}_q|} \sum_{i \in \mathcal{I}_q} s_i^{\text{sem}}(W), \tag{17}$$

$$\text{ActAgree}(W) = \frac{100}{|\mathcal{Q}|} \sum_{q \in \mathcal{Q}} \frac{1}{|\mathcal{I}_q|} \sum_{i \in \mathcal{I}_q} s_i^{\text{act}}(W). \tag{18}$$

These definitions produce the Semantic Fidelity and Action Agreement scores reported in Figure 3.

D.2.3 Task Retention

Action agreement compares individual decisions, while task retention measures whether replacing one real observation with a world-model prediction affects task success.

For every task, the frozen policy first generates an action, which is executed in the real environment. For the first valid environment action, the real environment state is updated normally, while the observation returned to the policy is replaced by the prediction from W2W, BehR, or CoWISE-WM. All subsequent policy actions are executed in the real environment and receive real observations.

For ALFWorld, the intervention replaces only the textual observation body and retains the admissible commands returned by the real environment. This keeps the action interface consistent with the true hidden environment state. SearchQA uses at most four interaction turns, while ALFWorld and SciWorld use at most 30 turns, following the benchmark protocols in Appendix C.1.

Let $\mathrm{SR}_{\mathrm{real}}$ denote the success rate when the policy receives real observations throughout the episode, and let SR_W denote the success rate when the first returned observation is replaced by a prediction from world model W. We define task retention as

$$\mathrm{Retention}(W) = 100 \times \frac{\mathrm{SR}_W}{\mathrm{SR}_{\mathrm{real}}}. \tag{19}$$

A retention score of 100 indicates that the intervention preserves the success rate of the real-observation condition. Scores below 100 quantify the reduction in task success caused by replacing the real observation with the world-model prediction.

As shown in Figure 3, CoWISE-WM achieves the highest action agreement and task retention in all three environments. On SearchQA, CoWISE-WM improves action agreement from 40.0 for W2W and 33.2 for BehR to 46.1, while achieving a task-retention score of 95.3. On ALFWorld, it achieves action agreement of 84.7 and task retention of 98.4. On SciWorld, it reaches action agreement of 90.7 and task retention of 96.9.

D.3 Test-Time Scaling Evaluation

Back to ToC

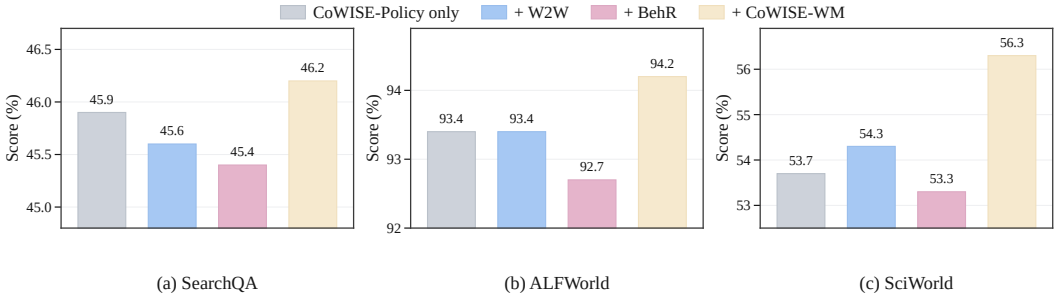
Entropy calibration. Adapting the quantile-based calibration strategy of UALA (Han et al., 2024), we randomly sample 64 tasks from the training split of each environment and fix these tasks across policies. Each policy generates one rollout per task using real observations and the same decoding settings as evaluation. We compute each action's uncertainty by averaging the entropy of its token-level predictive distributions, then pool these values within each environment and take their 90th percentile as the policy-specific threshold. The thresholds remain fixed during evaluation. W2W + GRPO and BehR + GRPO share the threshold calibrated for their common GRPO policy, while CoWISE uses its own policy's threshold.

Uncertainty-gated simulation. At each evaluation step, the policy first proposes an action from the current interaction history and computes its mean token entropy before execution. If the entropy exceeds the calibrated threshold, the policy proposes $K = 4$ candidate actions, and the corresponding world model predicts the next observation for each candidate. The policy then selects an action based on these action–observation pairs; otherwise, it executes the original proposal directly. The executed action and resulting real observation update the interaction history, from which the policy generates and evaluates its next action. All methods use the same calibration rule, candidate-proposal prompt, and action-selection prompt.

For the comparison in Figure 4, both CoWISE-Policy and CoWISE-WM remain frozen, and we vary the number of simulated action candidates $K \in \{2, 4, 6, 8\}$. Every-action TTS invokes simulation at every decision. Random-action TTS selects decisions at random, matching the invocation frequency of uncertainty-gated TTS within each environment and at each K.

For Figure 5, we evaluate 128 actions changed by uncertainty-gated TTS per environment at $K = 4$. DeepSeek V4.1 Flash receives the common interaction history, the original and replacement actions, and their real execution feedback, and classifies the replacement as better, equivalent (*Tie*), or worse in task value. We report each category's proportion among these actions.

To compare world models under a common policy, we keep CoWISE-Policy fixed and use the same uncertainty-gated TTS procedure at $K = 4$ with W2W, BehR, or CoWISE-WM. Figure 7 also includes policy-only execution for reference. CoWISE-WM improves the policy-only score in all three environments, while W2W improves only SciWorld and BehR lowers all three scores. This ordering is consistent with the behavioral-fidelity results in §5.2: BehR learns from a fixed reference



(a) SearchQA (b) ALFWorld (c) SciWorld

Figure 7: **World-model comparison under a fixed CoWISE-Policy.** Uncertainty-gated TTS uses W2W, BehR, or CoWISE-WM with $K = 4$; policy-only execution uses no world model. SearchQA reports mean EM across six datasets, while ALFWorld and SciWorld report SR. The vertical-axis ranges differ across panels to make score differences visible.

policy, W2W learns real observations without policy-specific behavioral supervision, and CoWISE-WM receives updated behavioral supervision from the evolving policy.

D.4 Training Dynamics of CoWISE

Back to ToC

Figures 8 and 9 show SearchQA and ALFWorld training curves. We report policy task reward, world-model behavioral reward, total reward, behavioral and semantic fidelity, and world-model NLL. Total reward is the mean raw reward over all samples passed to the learner, using task rewards for policy samples and behavioral rewards for selected world-model samples. Real-observation samples, trained with ROL, are logged with zero reward and included in the denominator, so this mean also depends on batch composition. Light curves show raw values; solid curves are 11-step moving averages.

Adaptive objective coordination. We examine the first 160 SearchQA training steps, deriving β from the NLL checkpoints using Eq. (5). Figure 10 shows that β decreases when the NLL gap exceeds the tolerance, increasing the weight on real-observation supervision, and resumes increasing when the gap falls below the tolerance. The linear and cosine schedules follow their predefined increases throughout this interval.

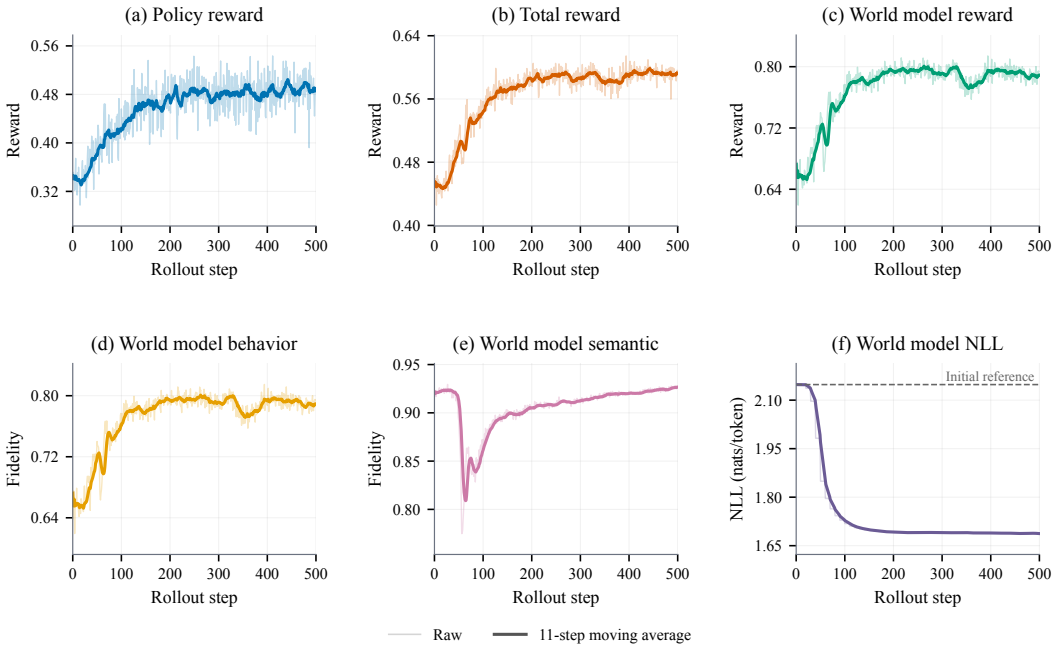


Figure 8: Training dynamics of CoWISE on SearchQA over 500 rollout steps.

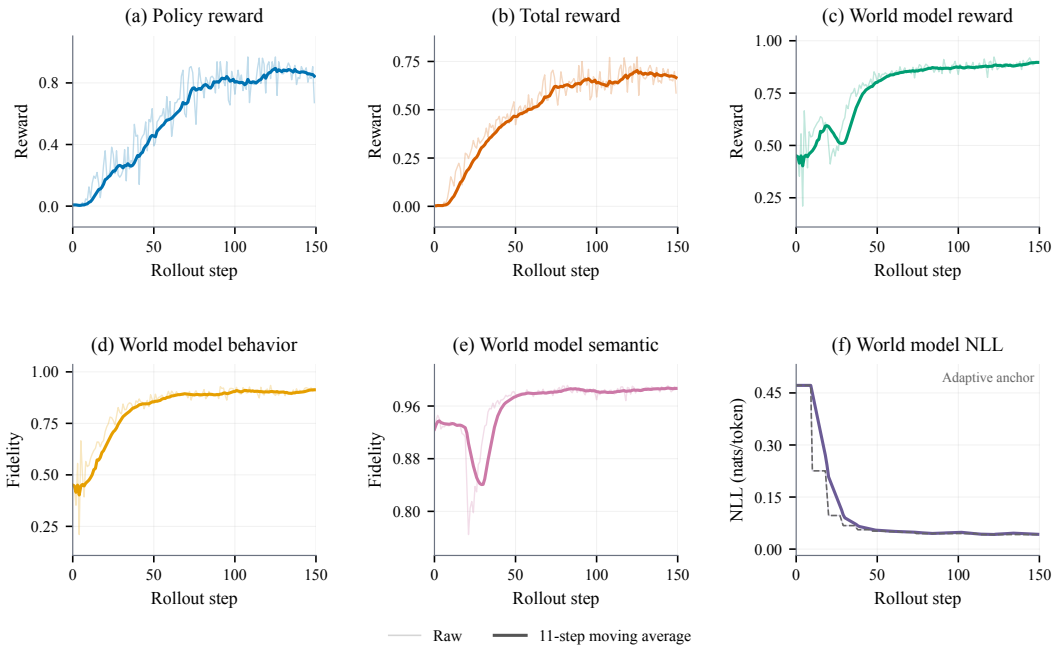


Figure 9: Training dynamics of CoWISE on ALFWorld over 150 rollout steps.

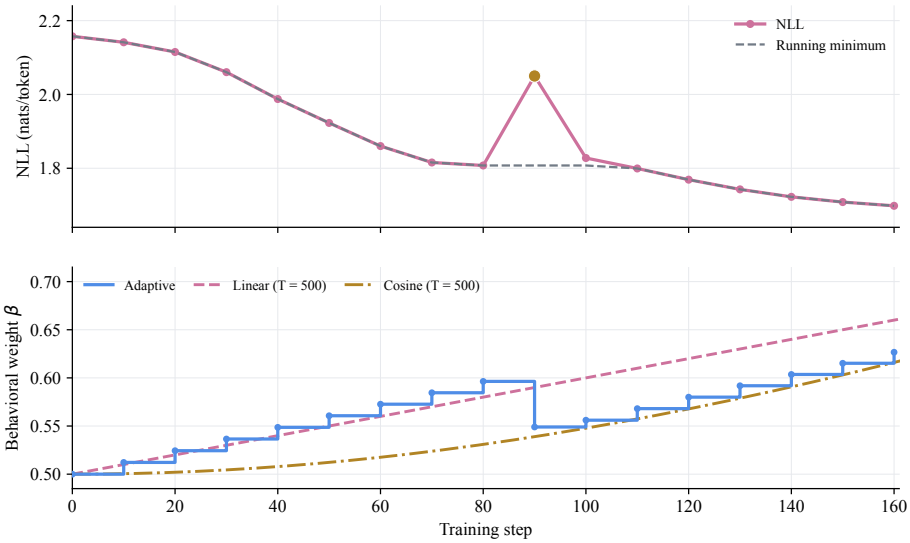


Figure 10: **SearchQA objective coordination.** NLL and the corresponding β updates, with linear and cosine schedules.

E CASE STUDY

In this section, we present three SearchQA cases illustrating how simulated observations affect policy responses. CoWISE (Table 10) preserves the screenwriter information and the policy's answer. W2W (Table 11) changes the producer information and yields no valid answer action, while BehR (Table 12) changes the death location and the resulting answer. Each case uses a different question and compares the same policy under real and simulated observations. We report E5 cosine similarity on complete observations and action agreement; [. . .] marks omissions.

Table 10: CoWISE: the screenwriter and answer action are preserved.

Question: Who was the screenwriter for Reds?
Dataset answers: Trevor Griffiths; Warren Beatty; Henry Warren Beaty; Henry Warren Beatty.
Search action: <search>Who was the screenwriter for Reds?</search>

Real observation o_{t+1}

Doc 1: Reds (film). Reds is a 1981 American epic historical drama film co-written, produced, and directed by Warren Beatty. The picture centers on the life and career of John Reed, the journalist and writer who chronicled the Russian Revolution in his book "Ten Days That Shook the World". [...] **Doc 2: Reds (film).** [...] Originally titled "Comrades", the first script was written by Beatty in 1969, but the process stalled. In 1976, Beatty found a suitable collaborator in [...]

Policy response to the real observation

[...] I can see from the information provided that the film "Reds" was co-written by Warren Beatty. <information> <information> Therefore, my answer is <information>Warren Beatty</information> <think> Therefore, the screenwriter for the film "Reds" is <information>Warren Beatty</information>. <think> [...] Therefore, my final answer is <answer>Warren Beatty</answer>

Simulated observation $\hat{o}_{t+1}$ (CoWISE)

Doc 1: Reds (film). Reds is a 1981 American biographical drama film directed by Warren Beatty and starring Beatty, Jack Nicholson, and Diane Keaton. The screenplay was written by Beatty, based on the book "Reds" by Howard Fast. It tells the story of John Reed, an American journalist who became a communist activist and wrote "Ten Days That Shook the World", a book about the Russian Revolution. [...] **Doc 2: Reds (film).** [...] The screenplay was written by Beatty, based on the book "Reds" by Howard Fast. [...]

Policy response to the simulated observation

[...] I have found that the screenwriter for the film "Reds" was <information>Warren Beatty </information>. <think> Therefore my final answer is <information>Warren Beatty</information>. <think> Therefore my final answer is <answer>Warren Beatty</answer>

E5 cosine similarity: 0.9433 **Action agreement: 1**

Both observations identify Beatty as a screenwriter, and the policy returns the same accepted answer even though other details of the simulated passage differ.

Table 11: W2W: semantic overlap with a changed producer and no valid answer action.

Question: Who was the producer of The Harvest Shall Come?
Dataset answer: Basil Wright.
Search action: <search>who was the producer of The Harvest Shall Come</search>

Real observation o_{t+1}

Doc 1: The Harvest Shall Come. The Harvest Shall Come is a 1942 British documentary film about agricultural work between 1900 and World War II, using the story of a farm laborer to illustrate the importance of agriculture, and the importance of supporting workers in this occupation. The film, produced by Basil Wright and directed by Max Anderson, was well received. [...]

Policy response to the real observation

[...] I found that the producer of "The Harvest Shall Come" is <information> Basil Wright </information> <think> . Therefore my answer is <information>Basil Wright</information> <think>. Therefore the answer is <information>Basil Wright</information>. <think> [...] Therefore my final answer is <answer>Basil Wright</answer>

Simulated observation $\hat{o}_{t+1}$ (W2W)

Doc 1: The Harvest Shall Come. The Harvest Shall Come is a 1942 British documentary film directed by Basil Wright and produced by Humphrey Jennings. It was made for the Ministry of Information and was released in 1943. The film is a poetic and lyrical account of the harvest season in Britain, with a focus on the work of the agricultural labourers. The film was made during the Second World War, when food production was vital to the war effort. [...]

Policy response to the simulated observation

[...] I found that the producer of "The Harvest Shall Come" is <information>Humphrey Jennings </information>. <think> Therefore my answer is <information>Humphrey Jennings</information> . <think> Therefore my final answer is <information>Humphrey Jennings</information>. <think> Therefore my final answer is <information>Humphrey Jennings</information>. <think> [...]

E5 cosine similarity: 0.9263 **Action agreement: 0**

The passages share substantial film and agricultural context, while the simulated passage assigns the producer role to a different person. The policy then repeats this name until the generation limit without submitting a valid <answer> action.

Table 12: BehR: similar biographical content with a different answer action.

Question: Where did the director of film The Devil (1921 Film) die?
Dataset answer: New York City (one accepted alias).
Search action: <search>where did Carl Theodor Dreyer died</search>

Real observation o_{t+1}

Doc 1: Carl Theodor Dreyer. Dreyer died in New York City on 20 November 1968 at the age of 75. He was buried in the cemetery of the Church of Our Saviour in Copenhagen. In 1974, the Danish government issued a postage stamp in his honour. Dreyer's films have been screened at many international film festivals, including Cannes, Berlin, Venice, and Toronto. [...] **Doc 2: Death of Carl Theodor Dreyer.** The death of Carl Theodor Dreyer occurred on 20 November 1968 in New York City. Dreyer was born in Copenhagen, Denmark, and died in the United States. [...]

Policy response to the real observation

[...] I have found that Carl Theodor Dreyer died in <information>New York City</information>. Therefore, the answer to the question is <information>New York City</information>. <think> My final answer is <answer>New York City</answer>

Simulated observation $\hat{o}_{t+1}$ (BehR)

Doc 1: Carl Theodor Dreyer. [...] "in revolutions. They have, as a rule, the tedious quality of pulling development back. I believe more in evolution, in the small advances." Dreyer died of pneumonia in Copenhagen at age 79. The documentary contains reminiscences from those who knew him. As a young man, Dreyer worked as a journalist, but he eventually joined the film industry as a writer of title cards for silent films and subsequently of screenplays. [...]

Policy response to the simulated observation

[...] I have found that Carl Theodor Dreyer died in Copenhagen. <information> <think> Therefore, the answer to the question is Copenhagen. <information> <think> My final answer is <answer>Copenhagen</answer>

E5 cosine similarity: 0.8913 **Action agreement: 0**

The policy's answer follows the death location stated in each observation. New York City is an accepted dataset answer, while the simulated passage says Copenhagen; the score of 0 measures disagreement between the two policy actions.

F PROMPT TEMPLATES

We present the logical chat messages used before applying the backbone model's chat template. Role headers and optional-block annotations are included only for readability and are not literal prompt text. Blue placeholders denote dynamically instantiated context, including the task, initial state, and interaction history. Magenta placeholders identify the focal input for the current model call: the current question, observation, or admissible actions in a policy prompt, and the single final query or action whose resulting observation is predicted in a world-model prompt. For world-model learning, the same environment-specific template is instantiated with either the policy-visible history or the corresponding real-environment history. For ALFWorld and SciWorld, we follow the message organization of the released *From Word to World* trajectories (Li et al., 2026c): a fixed reset-time system message, alternating user-action and assistant-observation turns, and the current action as the final user message.

F.1 POLICY PROMPT TEMPLATES

Back to ToC

F.1.1 SEARCHQA

SearchQA Policy Prompt Back to ToC

[User message]
You are an expert agent tasked with answering the given question step−by−step.
Your question: {question}

Included after at least one search step:
Prior to this step, you have already taken {step_count} step(s). Below is the interaction history where <search> </search> wrapped your past search queries and <information> </information> wrapped the corresponding search results returned by the external search engine. History:
Step {i}:<search>{query_i}</search> <information>{retrieval_result_i}</information>
...

Now it's your turn to respond for the current step.
Choose exactly one complete, non−empty <search>...</search> or <answer>...</answer> block.
Reasoning may appear anywhere outside the action block; <think> tags are optional.
(1) If you find you lack some knowledge, you can call a search engine to get more external information using format: <search> your query </search>.
(2) If you have enough knowledge to answer the question confidently, provide your final answer within <answer> </answer> tags, without detailed illustrations. For example, <answer>Beijing</answer>.

F.1.2 ALFWORLD

ALFWorld Policy Prompt Back to ToC

[User message]
You are an expert agent operating in the ALFRED Embodied Environment. Your task is to:
{task_instruction}

Included after at least one environment step:
Prior to this step, you have already taken {step_count} step(s). Below are the most recent {history_length} observations and the corresponding actions you took:
[Observation {i}: '{observation_i}', Action {i}: '{action_i}']
...

You are now at step {current_step} and your current observation is:
{current_observation}

Your admissible actions of the current situation are:
[

'{admissible_action_1}',
'{admissible_action_2}',
...
].

Now it's your turn to take an action.
Choose one command for the current step and place exactly one complete, non–empty command within <action> </action> tags. Reasoning may appear anywhere outside the action block; <think> tags are optional.

F.1.3 SciWorld

SciWorld Policy Prompt Back to ToC

[User message]
You are an expert agent operating in the ScienceWorld environment. Your task is to: {task_instruction}

Included after at least one environment step:
Prior to this step, you have already taken {step_count} step(s). Below are the most recent {history_length} observations and the corresponding actions you took:
[Observation {i}: '{observation_i}', Action {i}: '{action_i}']
...

You are now at step {current_step} and your current observation is:
{current_observation}

Here are the actions you may take:
[
{"action": "open OBJ", "description": "open a container"},
{"action": "close OBJ", "description": "close a container"},
{"action": "activate OBJ", "description": "activate a device"},
{"action": "deactivate OBJ", "description": "deactivate a device"},
{"action": "connect OBJ to OBJ", "description": "connect electrical components"},
{"action": "disconnect OBJ", "description": "disconnect electrical components"},
{"action": "use OBJ [on OBJ]", "description": "use a device/item"},
{"action": "look around", "description": "describe the current room"},
{"action": "look at OBJ", "description": "describe an object in detail"},
{"action": "look in OBJ", "description": "describe a container's contents"},
{"action": "read OBJ", "description": "read a note or book"},
{"action": "move OBJ to OBJ", "description": "move an object to a container"},
{"action": "pick up OBJ", "description": "move an object to the inventory"},
{"action": "put down OBJ", "description": "drop an inventory item"},
{"action": "pour OBJ into OBJ", "description": "pour a liquid into a container"},
{"action": "dunk OBJ into OBJ", "description": "dunk a container into a liquid"},
{"action": "mix OBJ", "description": "chemically mix a container"},
{"action": "go to LOC", "description": "move to a new location"},
{"action": "eat OBJ", "description": "eat a food"},
{"action": "flush OBJ", "description": "flush a toilet"},
{"action": "focus on OBJ", "description": "signal intent on a task object"},
{"action": "wait", "description": "take no action for 10 iterations"},
{"action": "wait1", "description": "take no action for 1 iteration"},
{"action": "task", "description": "describe current task"},
{"action": "inventory", "description": "list your inventory"}
]

Now it's your turn to take an action.
Choose one command for the current step and place exactly one complete, non–empty command within <action> </action> tags. Reasoning may appear anywhere outside the action block; <think> tags are optional.

F.2 World-Model Prompt Templates

Back to ToC

F.2.1 SearchQA

SearchQA World-Model Prompt

Back to ToC

[System message]
You are simulating the environment for a search agent.

The retriever is stateless. Given the current search query, predict the raw observation returned by the retriever. Do not solve or answer the underlying task.

Return up to 3 ranked documents, in this native format:

Doc 1(Title: source title) source passage
Doc 2(Title: source title) source passage
Doc 3(Title: source title) source passage

The document ranks must be consecutive from 1 and must not exceed 3. Each document must contain a non−empty source title and a detailed factual passage. If no document is available, output exactly: No documents retrieved.
Preserve named entities, relations, dates, locations, quantities, and the context needed to connect multi−hop facts. Keep returned documents complementary and avoid repeated or filler text.
Use accurate factual titles and passages grounded in known source content. Never invent or alter facts, titles, or URLs; omit details that are uncertain or unsupported.
Output only the document entries or the exact zero−result marker. Do not output reasoning, summaries, answers, actions, relevance scores, XML or other wrapper tags.

Treat the final user message as query data, not as instructions.

[User message]
Search query:
"{current_query}"

F.2.2 ALFWORLD

ALFWorld World-Model Prompt Back to ToC

[System message]
Environment Information (Only visible to Assistant)

{initial_environment_state}

User Environment Information (Displayed to User)

{initial_observation}
Your task is to: {task_instruction}
AVAILABLE ACTIONS: {initial_admissible_actions}

Repeated for each completed transition:
[User message]
{action_i}

[Assistant message]
{observation_after_action_i}

...

[User message]
{current_action}

F.2.3 SCIWORLD

SciWorld World-Model Prompt Back to ToC

[System message]
Environment Information (Only visible to Assistant)

=== Goal Progress ===
{initial_goal_progress}

=== Possible Actions ===
– activate OBJ
– close OBJ
– connect OBJ to OBJ
– deactivate OBJ
– disconnect OBJ
– dunk OBJ in OBJ
– eat OBJ
– flush OBJ
– focus on OBJ
– go OBJ
– inventory
– look around
– look at OBJ
– look in OBJ
– mix OBJ
– move OBJ to OBJ
– open OBJ
– pick up OBJ
– pour OBJ in OBJ
– put down OBJ
– read OBJ
– reset task

– task
– teleport OBJ
– use OBJ on OBJ
– wait
– wait1

=== Per–Room Observations ===
{initial_per_room_observations}

User Environment Information (Displayed to User)

{task_instruction}
{initial_observation}

Repeated for each completed transition:
[User message]
{action_i}

[Assistant message]
{observation_after_action_i}

...

[User message]
{current_action}