

Beyond Next-Observation Prediction: Agent-Authored World Modeling for Sequential Decision Making

Guangfeng Cai¹, Kaibing Yang¹, Shuo He², Yu Li¹,
Shengtian Yang¹, Jiaqi Lv¹, Lei Feng^{1*}

¹Southeast University, Nanjing, China

²Meituan, Shanghai, China

{cgfeng, fenglei}@seu.edu.cn

Abstract

Recent studies on world modeling for Large Language Model (LLM) agents typically formulate the learning objective as next-observation prediction. However, this objective ties supervision to what a transition happens to reveal, which may omit the dynamics most relevant to the agent’s current decision. To bridge this gap, we propose Agent-Authored World Modeling (AAWM), a training procedure that constructs supervision from the policy’s own decision needs. Specifically, at each state, the agent identifies what it needs to understand about the environment before acting. These needs drive the retrieval of relevant transition evidence across trajectories, which is then synthesized into training targets that capture decision-oriented dynamics instead of reconstructing the next observation. This aligns the training objective with the dynamics the policy needs before acting, not with the contents of the next observation. Experimental results validate the effectiveness of AAWM across multiple environments and training settings. These results show that decision-aware world-model targets provide a more effective learning signal than next-observation prediction.

1 Introduction

LLM agents (Achiam et al., 2023; Kimi Team et al., 2025; GLM Team et al., 2024; Liu et al., 2024) have increasingly operated in partially observed environments that require multi-turn interactions to gather information and complete tasks. In such settings, success depends not only on choosing fluent actions but also on representing environment dynamics: which actions are valid, which state variables persist, and which observations matter for the next decision. Recent work therefore fine-tunes large language models to predict the environment’s

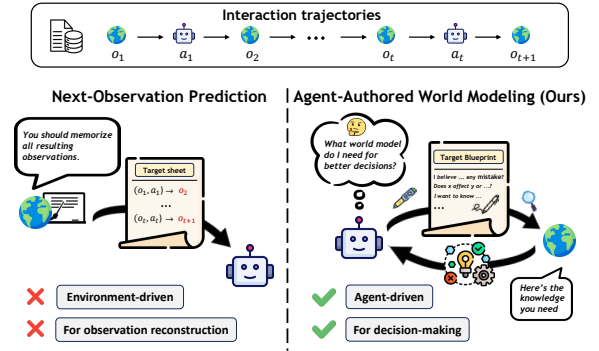


Figure 1: Comparison between next observation prediction and AAWM. Next-observation prediction trains on the observation returned by each action, so the target is determined by what the environment happens to reveal. AAWM asks the policy what it needs to know before acting and writes targets that describe the dynamics needed for decision making.

next response, treating this objective as world modeling (Zhang et al., 2026; Li et al., 2026a). The underlying assumption is that *better prediction of the next environment output will produce better action selection*.

This assumption overlooks a key difference between prediction and decision:

the policy often needs the state information that affects its next action rather than a full reconstruction of the next observation (Nair et al., 2020). Moreover, Li et al. (2024) observed an empirical trade-off between next-token prediction and goal-directed agent behavior, suggesting that prediction quality and decision quality are not the same objective. This raises a core target selection problem: *what environment dynamics does the policy need in the world modeling target before acting?*

Prior work has mostly addressed this problem by changing the form of the target. Some methods compress observations into semantic future repre-

*Corresponding author.

sentations (Berg et al., 2025), summarize transition-level state changes (Chae et al., 2025), or add task information for downstream planning (Qiao et al., 2024). These targets are often more compact or more planning-friendly than raw next observations, but their content is still determined by the transition exposed in the dataset.

As a result, the training target remains environment-driven rather than decision-driven: *it reflects what the environment happened to reveal, instead of what the policy needs to understand before acting.*

To address this problem, we propose AAWM (Agent-Authored World Modeling), a training procedure that constructs training targets from the policy’s current decision needs. Figure 1 illustrates the difference from next observation prediction. At each state, the policy first articulates its beliefs about environment dynamics and open questions whose answers could change the next action. These statements serve as retrieval queries over a pool of transition records, gathering evidence about the queried dynamics from other trajectories. The retrieved evidence, together with the current transition and the policy’s statements, is then synthesized into a natural-language target describing decision-oriented dynamics. Fine-tuning on these targets encourages the policy to represent the dynamics that matter for action selection instead of only reconstructing the next observation.

We evaluate AAWM on ALFWorld and WebShop at two model scales. With the same imitation learning and reinforcement learning setup, AAWM consistently outperforms next-observation world modeling, with gains of up to 6.3 and 6.2 success-rate points on the two environments. A separate AgentGym evaluation (Xi et al., 2025) across four environments confirms that AAWM is the only world modeling initialization that improves over imitation learning in every setting. Component ablations show that Self-Probing, Transition Retrieval, and world modeling fine-tuning each contribute to the final gain. Training analysis further shows that AAWM-initialized policies sustain broader decision-oriented reasoning during reinforcement learning while task success continues to rise.

Our contributions are summarized as follows:

- We identify target selection as a central problem in world modeling for language agents: training targets should capture the dynamics that affect action choice, not only reconstruct

environment responses.

- We instantiate this principle as AAWM, a world modeling procedure in which the policy’s own beliefs and uncertainties determine what the training target addresses.
- We demonstrate that AAWM outperforms next-observation world modeling across two environments, two model scales, and both supervised and reinforcement learning settings.

2 Related Work

LLMs as decision-making agents. LLM agents have been widely studied for sequential decision making in software engineering (Yang et al., 2024), embodied interaction (Wang et al., 2024), web navigation (Gur et al., 2024), and GUI operation (Zhang and Zhang, 2024; Hong et al., 2024). Prompting methods such as chain-of-thought reasoning and ReAct (Yao et al., 2023) structure intermediate reasoning traces to decompose complex tasks, while self-reflection mechanisms such as Reflexion (Shinn et al., 2023) incorporate verbal feedback from prior failures to revise action selection. Memory (Park et al., 2023), retrieval (Nakano et al., 2021), and tool-use (Gou et al., 2024; Schick et al., 2023) mechanisms further extend agent capabilities by supplying external context or executable operations during interaction. Recent training methods also reuse successful tool transitions, route different behavioral phases to specialized experts, and extract learning signals from rollout groups in which every attempt fails (Li et al., 2026b; Yang et al., 2026b,a). These approaches improve policy learning for long-horizon tasks. AAWM complements this line of work by constructing world modeling targets from the policy’s decision needs.

World models for LLM agents. World models have been widely used to provide future information and auxiliary supervision for sequential decision making. Recent work across language and vision-language decision-making systems has instantiated this idea through transition-level state change summarization (Chae et al., 2025), semantic compression (Berg et al., 2025), result state prediction (Guo et al., 2025), action simulation (Yu et al., 2025), and knowledge injection (Qiao et al., 2024).

A related line of work involves the agent more directly in world-model construction. Chen et al. (2025) internalizes environment dynamics from

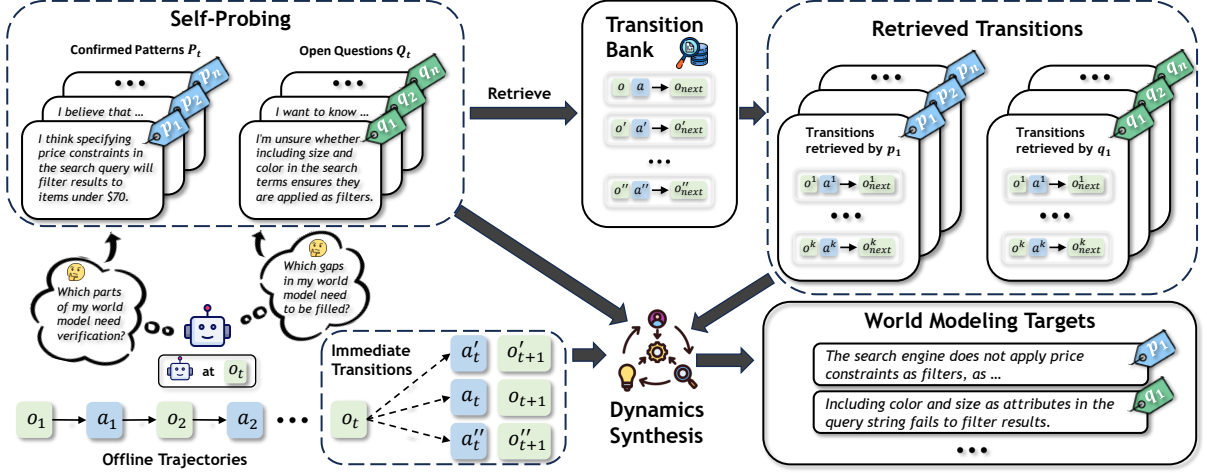


Figure 2: AAWM target construction pipeline. At a decision context o_t , the policy produces confirmed patterns P_t and open questions Q_t through Self-Probing. Each proposition retrieves evidence from the transition bank $\mathcal{T}$, and the immediate transitions $\mathcal{I}_t$ provide local evidence from the same context. Dynamics Synthesis combines these inputs into world modeling targets that correct mistaken beliefs and answer open questions when evidence permits.

the agent’s own rollouts; Wang et al. (2025) reinforces explicit prediction of environment states during multi-turn interactions; Hu et al. (2025) reconstructs the thinking process with world-model simulation; and Zhang et al. (2026) generate additional interaction data from the agent’s own actions and train on the resulting outcomes.

Despite this progress, the content of the training target is still determined by what each transition happens to reveal, which may omit the dynamics most relevant to the agent’s current decision. Our work therefore aims to dynamically adjust the policy’s target to its current beliefs and open questions.

3 Preliminaries

Problem setup. We consider a partially observed text environment $\mathcal{E}$ in which a language-model policy π_θ interacts through textual observations and actions. At step t , the policy receives the decision context o_t and samples an action $a_t \sim \pi_\theta(\cdot | o_t)$. The environment returns the next observation o_{t+1} , producing trajectories $\mathcal{D} = \{\tau_i\}$ with $\tau_i = (o_0, a_0, o_1, \dots, o_T)$, and we write $\mathcal{U}(\mathcal{D}) = \{(o_t, a_t, o_{t+1})\}$ for the set of all one-step transition records. From these records we maintain two collections. The *global transition pool* $\mathcal{T}$ contains records from all trajectories as a single retrievable collection. The *immediate transition set* $\mathcal{I}_t \subseteq \mathcal{T}$ contains records originating from the same context o_t , including the logged transition and additional transitions obtained by executing alternative actions from that context.

World modeling objective. A standard text world model predicts the environment response conditioned on the current context and action. For each transition record, a textual outcome target $\bar{o}_{t+1} = g(o_{t+1})$ is constructed from the next observation, where g may be the identity map, a state change extractor, or a summarizer. The next response prediction objective is to minimize

$$\mathcal{L}_{\text{obs}}(\theta) = - \sum_{\substack{(o_t, a_t, o_{t+1}) \\ \in \mathcal{U}(\mathcal{D})}} \log p_\theta(\bar{o}_{t+1} | o_t, a_t) \quad (1)$$

This objective is useful when the next observation contains the dynamics needed for control, but the dynamics that determine a good action can be sparse, implicit, or distributed across earlier transitions. Before acting, the policy may need to verify whether its current understanding of the environment is correct and to resolve specific gaps in that understanding. These needs reflect the state of the policy’s own world model, not the content of the next observation. AAWM addresses this by constructing world modeling targets from the policy’s decision-oriented modeling needs. These targets provide an auxiliary training objective for the same model that is subsequently trained and evaluated on downstream agentic tasks.

4 Agent-Authored World Modeling

To align the world modeling objective with what the policy needs before acting, AAWM constructs targets from the policy’s current decision context, with an overview of the pipeline shown in Figure 2.

Self-Probing elicits the policy’s current beliefs and open questions. *Transition Retrieval* selects supporting records from $\mathcal{T}$. *Dynamics Synthesis* combines the probes, the immediate transitions, and the retrieved transitions into a fine-tuning target. The following subsections describe these stages in detail.

4.1 Self-Probing

Instead of defining target content from the next observation, AAWM elicits the policy’s modeling needs at context o_t . Specifically, the policy is prompted to produce two sets of propositions:

$$(P_t, Q_t) \sim \pi_\theta(\cdot \mid o_t), \quad (2)$$

where $P_t = \{p_t^1, \dots, p_t^n\}$ are *confirmed patterns* and $Q_t = \{q_t^1, \dots, q_t^n\}$ are *open questions*.

Confirmed patterns specify parts of the policy’s current understanding that may require verification or correction against retrieved transitions.

Open questions identify dynamics that the policy has not yet resolved and whose answers could affect the next action.

Together, (P_t, Q_t) determine what the training target should contain and drive the subsequent retrieval and synthesis steps. The prompt template is in Appendix F.

4.2 Transition Retrieval

Since the required dynamics may be absent from the immediate transition set $\mathcal{I}_t$, we retrieve relevant transitions from the global transition pool $\mathcal{T}$.

For each proposition $r \in P_t \cup Q_t$, AAWM retrieves K transitions from $\mathcal{T}$ using maximal marginal relevance (Carbonell and Goldstein, 1998), which balances relevance to r with diversity across the selected records, ensuring the evidence covers different aspects of the queried dynamics instead of redundant records. The retrieved set is

$$\mathcal{R}_t = \bigcup_{r \in P_t \cup Q_t} \text{Retrieve}(r, \mathcal{T}; K). \quad (3)$$

Implementation details are in Appendix G.

4.3 Dynamics Synthesis

Given the policy’s acting needs specified by the probes and the transition evidence grounded by retrieval, the final stage synthesizes an agent-authored world-modeling target.

Specifically, an external instruction-following model f_{syn} receives the current context o_t , confirmed patterns P_t , open questions Q_t , the immediate transitions $\mathcal{I}_t$ from the same context, and the retrieved transitions $\mathcal{R}_t$ from the global pool:

$$y_t = f_{\text{syn}}(o_t, P_t, Q_t, \mathcal{I}_t, \mathcal{R}_t). \quad (4)$$

The model is instructed to correct any unsupported beliefs, resolve open questions the evidence permits, and summarize the dynamics most relevant to action selection. The resulting y_t thus describes environment dynamics shaped by what the policy needs before acting. The synthesis prompt is in Appendix F, and the synthesis model is described in Section 5.1.

4.4 World Modeling

Let $\mathcal{D}_A = \{(o_t, y_t)\}$ denote the dataset produced by Dynamics Synthesis. AAWM fine-tunes the policy with the agent-authored world-modeling objective

$$\mathcal{L}_A(\theta) = - \sum_{(o_t, y_t) \in \mathcal{D}_A} \log p_\theta(y_t \mid o_t), \quad (5)$$

where the subscript A denotes AAWM. The target therefore updates the policy toward the dynamics specified by its own decision needs. The resulting parameters initialize the subsequent imitation learning and reinforcement learning stages.

5 Experiments

5.1 Setup

Environments and evaluation. We evaluate on ALFWorld (Shridhar et al., 2021) and WebShop (Yao et al., 2022). ALFWorld is a textual embodied environment with six household task types, Pick, Clean, Cool, Look, Heat, and Pick2. WebShop is an HTML shopping environment where the agent searches, inspects, and purchases products according to user instructions. ALFWorld reports success rate for each task type and the aggregate **All**, while WebShop reports success rate **Succ.** and reward score **Score**. We follow the environment configuration and evaluation protocol from the open-source repository (Feng et al., 2025).

Baseline. We use Implicit World Modeling (IWM) as the world modeling baseline, a representative next observation prediction method for language agents (Zhang et al., 2026). Given the

current observation and action (o_t, a_t), IWM trains the model to predict the next observation. Following the original IWM setting, the target is the raw textual observation on ALFWorld and an offline summary of the next observation on WebShop.

World modeling data. We draw world modeling data from AgentTraj-L (Xi et al., 2025) and keep all successful trajectories executable in the open-source repository (Feng et al., 2025) as $\mathcal{D}$. At every action step, we execute the logged action and three additional actions from the same state, producing four transitions that form $\mathcal{I}_t$ and populate $\mathcal{T}$. Both IWM and AAWM share the same $\mathcal{D}$ and $\mathcal{T}$ within each backbone scale. We construct both sets only from training trajectories, so evaluation trajectories never enter target construction. Retrieval also excludes records from the query episode. Trajectory counts and action-sampling details are provided in Appendix A.1.

Models and target construction. We train the policy at two scales, Qwen2.5-1.5B-Instruct and Qwen2.5-7B-Instruct (Qwen Team, 2024). Self-Probing uses the policy backbone at the corresponding scale; in the main experiments, it produces $n = 3$ confirmed patterns and $n = 3$ open questions per state. Transition Retrieval embeds records with Qwen3-Embedding-0.6B (Zhang et al., 2025). We use Qwen3-30B-A3B-Instruct-2507 (Yang et al., 2025) only for target construction, where it produces WebShop observation summaries for IWM and synthesized targets for AAWM. This keeps the target construction model shared between the two world modeling methods.

Training protocol. Both IWM and AAWM first initialize the policy by supervised fine-tuning on their respective world modeling targets, followed by a lightweight imitation learning stage on 32 trajectories from $\mathcal{D}$ for one epoch. All RL rows then run Group Relative Policy Optimization (GRPO; Shao et al., 2024).

Table 1 reports prompting baselines and trained-policy results at each backbone scale. Within each backbone, all trained rows share the same trajectory source, transition pool, imitation data, and GRPO budget.

Base denotes downstream training without world modeling initialization. We report mean and standard deviation over three seeds. More details are in Appendix A.1.

5.2 Main Results

Table 1 shows that AAWM consistently improves over the world-modeling baseline on both agentic benchmarks and at both backbone scales. Although closed-source prompting models remain competitive, with Gemini-2.5-Pro (Gemini Team, Google, 2025) and GPT-4o (OpenAI, 2024) reaching 60.3% and 48.0% success on ALFWorld and 35.9% and 23.7% on WebShop, post-training narrows and then reverses this gap for the Qwen-family models. Among the trained methods, AAWM achieves the strongest overall performance.

AAWM provides a stronger initialization than next observation prediction. At the 1.5B scale, Base reaches 74.0% success on ALFWorld, while IWM improves it to 76.8% and AAWM further improves it to 83.1%. WebShop shows the same ordering, with success increasing from 59.4% (Base) to 67.2% (IWM) and 73.4% (AAWM). At the 7B scale, AAWM again achieves the strongest aggregate performance, reaching 90.1% on ALFWorld and 76.6% on WebShop.

AAWM consistently outperforms on the harder tasks. The ALFWorld breakdown shows that the aggregate gain is concentrated on the more difficult categories. Pick and Clean are easier for most methods, and they are the only categories where IWM slightly exceeds AAWM: Pick at 1.5B and Clean at 7B. On the harder categories, the ordering reverses and the margins become larger. At 1.5B, AAWM improves over IWM by 11.0 points on Cool, 10.6 on Pick2, and 7.5 on Look. At 7B, the gains are 9.0 points on Look, 7.4 on Cool, and 6.0 on Pick2. These tasks require reasoning over multi-step pre-conditions and persistent object states, which are difficult to recover from a single observation. This is where Self-Probing and cross-trajectory retrieval are expected to be most useful, because they expose decision-relevant dynamics beyond the immediate observation.

AAWM produces a more decision-oriented initialization. A further question is whether this advantage comes from the world modeling targets themselves or from their interaction with the imitation learning stage. To isolate this, we remove the imitation learning stage so that world modeling alone carries the full initialization budget before GRPO (Appendix D). IWM then drops sharply on ALFWorld, while AAWM stays close to GRPO from the backbone and still improves WebShop.

Backbone	Method	ALFWorld							WebShop	
		Pick	Look	Clean	Heat	Cool	Pick2	All	Score	Succ.
Prompting										
Closed-source Models	GPT-4o	75.3	60.8	31.2	56.7	21.6	49.8	48.0	31.8	23.7
	Gemini-2.5-Pro	92.8	63.3	62.1	69.0	26.6	58.7	60.3	42.5	35.9
Qwen2.5-1.5B-Instruct	Qwen2.5	5.9	5.5	3.3	9.7	4.2	0.0	4.1	23.1	5.2
	ReAct	17.4	20.5	15.7	6.2	7.7	2.0	12.8	40.1	11.3
	Reflexion	35.3	22.2	21.7	13.6	19.4	3.7	21.8	55.8	21.9
Qwen2.5-7B-Instruct	Qwen2.5	33.4	21.6	19.3	6.9	2.8	3.2	14.8	26.4	7.8
	ReAct	48.5	35.4	34.3	13.2	18.2	17.6	31.2	46.2	19.5
	Reflexion	62.0	41.6	44.9	30.9	36.3	23.8	42.7	58.1	28.8
Imitation Learning initialized by World Modeling										
Qwen2.5-1.5B-Instruct	Base	18.4 $\pm$ 3.6	6.8 $\pm$ 2.4	13.7 $\pm$ 5.8	11.6 $\pm$ 3.8	8.8 $\pm$ 4.8	3.7 $\pm$ 2.1	12.0 $\pm$ 2.6	26.6 $\pm$ 4.9	4.2 $\pm$ 1.2
	IWM	5.9 $\pm$ 3.4	1.6 $\pm$ 1.1	3.2 $\pm$ 1.8	2.4 $\pm$ 1.5	1.5 $\pm$ 1.0	0.7 $\pm$ 0.5	3.1 $\pm$ 1.2	21.6 $\pm$ 3.3	2.1 $\pm$ 0.5
	AAWM (Ours)	31.4 $\pm$ 4.0	17.6 $\pm$ 5.0	28.7 $\pm$ 6.6	27.3 $\pm$ 4.8	24.2 $\pm$ 5.8	4.8 $\pm$ 3.0	24.7 $\pm$ 2.6	25.7 $\pm$ 2.5	10.7 $\pm$ 1.2
Qwen2.5-7B-Instruct	Base	50.3 $\pm$ 3.2	11.4 $\pm$ 2.4	28.6 $\pm$ 3.6	27.3 $\pm$ 3.1	18.2 $\pm$ 2.8	9.7 $\pm$ 2.2	28.4 $\pm$ 2.8	19.9 $\pm$ 2.2	11.3 $\pm$ 2.1
	IWM	69.4 $\pm$ 5.2	31.7 $\pm$ 5.8	55.3 $\pm$ 6.4	51.6 $\pm$ 5.6	40.5 $\pm$ 6.1	19.6 $\pm$ 4.6	49.6 $\pm$ 4.8	27.4 $\pm$ 5.3	9.0 $\pm$ 3.5
	AAWM (Ours)	70.5 $\pm$ 5.6	35.7 $\pm$ 6.2	60.4 $\pm$ 6.8	55.6 $\pm$ 5.9	52.9 $\pm$ 6.4	14.7 $\pm$ 5.0	53.4 $\pm$ 5.0	37.0 $\pm$ 2.9	16.3 $\pm$ 2.3
Imitation Learning then Reinforcement Learning initialized by World Modeling										
Qwen2.5-1.5B-Instruct	Base	82.1 $\pm$ 7.3	61.5 $\pm$ 2.7	81.9 $\pm$ 7.5	85.1 $\pm$ 6.3	58.9 $\pm$ 1.6	59.0 $\pm$ 7.7	74.0 $\pm$ 4.6	71.1 $\pm$ 4.7	59.4 $\pm$ 2.7
	IWM	94.4 $\pm$ 1.5	64.5 $\pm$ 7.7	78.6 $\pm$ 4.6	80.2 $\pm$ 4.5	60.7 $\pm$ 3.8	61.3 $\pm$ 7.0	76.8 $\pm$ 2.8	77.0 $\pm$ 3.0	67.2 $\pm$ 4.1
	AAWM (Ours)	93.3 $\pm$ 4.6	72.0 $\pm$ 7.7	86.9 $\pm$ 5.2	88.7 $\pm$ 2.2	71.7 $\pm$ 3.1	71.9 $\pm$ 2.8	83.1 $\pm$ 1.7	81.5 $\pm$ 1.1	73.4 $\pm$ 3.6
Qwen2.5-7B-Instruct	Base	93.0 $\pm$ 4.2	66.7 $\pm$ 3.7	92.8 $\pm$ 6.0	91.2 $\pm$ 1.7	79.3 $\pm$ 5.4	68.6 $\pm$ 2.4	84.6 $\pm$ 1.3	80.8 $\pm$ 4.9	70.6 $\pm$ 4.8
	IWM	92.6 $\pm$ 2.2	76.3 $\pm$ 2.2	94.4 $\pm$ 5.2	93.7 $\pm$ 5.5	74.1 $\pm$ 5.2	77.4 $\pm$ 7.7	86.7 $\pm$ 2.0	82.2 $\pm$ 1.9	73.2 $\pm$ 2.5
	AAWM (Ours)	95.2 $\pm$ 5.6	85.3 $\pm$ 2.9	93.1 $\pm$ 2.5	94.1 $\pm$ 5.5	81.5 $\pm$ 7.6	83.4 $\pm$ 7.6	90.1 $\pm$ 2.0	84.2 $\pm$ 0.9	76.6 $\pm$ 2.1

Table 1: Main results on ALFWorld and WebShop. In trained panels, Base denotes no world modeling initialization, IWM denotes next-observation world modeling, and AAWM denotes Agent-Authored World Modeling. Trained rows report mean $\pm$ std over three seeds; bold marks the best within each backbone and condition.

Since both methods share the same Qwen3-30B-A3B-Instruct-2507 construction model on WebShop, this comparison controls for constructor capacity, indicating that the gap is not explained by construction-model scale. AAWM depends far less on action imitation because its targets already describe the dynamics the policy needs when selecting actions.

5.3 Ablation Studies

Table 2 separates two questions about the source of AAWM’s gains. The component ablations remove *Self-Probing* or *Transition Retrieval*, while the target comparisons replace AAWM with generic reasoning or hindsight supervision. All rows use the 1.5B backbone and share the same imitation learning stage. Full AAWM improves ALFWorld from 12.0% to 24.7% and WebShop from 4.2% to 10.7% over Base.

Retrieved evidence supplies dynamics beyond the current state. Removing Transition Retrieval lowers the results to 19.3% on ALFWorld and 7.5% on WebShop. Without retrieved evidence, *Dynamics Synthesis* sees only the immediate transi-

Method	ALFWorld	WebShop
Imitation Learning initialized by World Modeling		
Base	12.0 $\pm$ 2.6	4.2 $\pm$ 1.2
AAWM (Ours)	24.7$\pm$2.6	10.7$\pm$1.2
w/o Transition Retrieval	19.3 $\pm$ 2.4	7.5 $\pm$ 1.6
w/o Self-Probing	18.0 $\pm$ 1.6	7.0 $\pm$ 3.1
CoT-Only	14.8 $\pm$ 2.1	5.2 $\pm$ 1.6
Hindsight Reflection	20.8 $\pm$ 1.6	6.2 $\pm$ 1.4

Table 2: Ablation at Qwen2.5-1.5B-Instruct. All rows use the same lightweight imitation learning stage and are evaluated without GRPO. Base denotes imitation learning without world modeling initialization. ALFWorld reports aggregate success rate, and WebShop reports success rate. All cells report mean $\pm$ std over three seeds.

tions at the current state, which often do not contain enough information to resolve the policy’s open questions.

Self-Probing tells synthesis which dynamics to address. Removing Self-Probing lowers the results further to 18.0% and 7.0%. Without the policy’s confirmed patterns and open questions,

the synthesis model has no explicit signal about what the target should cover. The two components together outperform either alone, showing that both contribute to shifting the training target from environment-driven content to decision-driven content.

AAWM outperforms alternative supervision targets. We next test whether generic reasoning or reflection-style supervision can reproduce the benefit of AAWM’s target construction. CoT-Only uses Qwen3-30B-A3B-Thinking-2507 to generate reasoning from (o_t, a_t) without transition evidence, while Hindsight Reflection uses the same transition evidence and synthesis model as AAWM to write a hindsight lesson directly. CoT-Only gives a modest improvement over Base, and Hindsight Reflection is stronger, but both remain below full AAWM on the two environments. A separate State-Change control uses the same construction model to describe only the task-relevant difference between o_t and o_{t+1} rather than reconstructing the full next observation. This single-run control reaches 9.4% on ALFWorld and 5.7% on WebShop (Appendix B.1). Appendix C provides direct qualitative comparisons between next-observation targets and AAWM targets for matched decision contexts. Overall, the advantage of AAWM is not explained by generic reasoning, hindsight supervision, or transition abstraction alone.

5.4 Evaluation on More Environments

The main experiments use two environments with GRPO. To test whether decision-oriented targets remain beneficial across a broader set of environments under supervised training alone, we evaluate on four environments from AgentGym (Xi et al., 2025). All methods train Qwen2.5-1.5B-Instruct on the same mixture of AgentTraj-L trajectories from TextCraft, SciWorld, WebShop, and ALFWorld, and use the same imitation learning stage. We also include IWM-Summary, a variant that uses the same Qwen3-30B-A3B model as AAWM to summarize observations before prediction, which controls for the capacity of the construction model. Data counts and evaluation settings are in Appendix A.2.

Table 3 reports success rate in each environment and the weighted aggregate All. Base reaches an All of 61.1%. The two next observation variants do not give a consistent improvement. IWM falls to 59.6%, because predicting raw observations

helps SciWorld but hurts WebShop and ALFWorld. IWM-Summary recovers WebShop and improves SciWorld for an All of 61.4%, yet it drops sharply on ALFWorld to 20.0%. AAWM instead reaches the best result in all four environments, with an All of 69.3% and the largest single gain of 21.0 points on SciWorld. It is the only world modeling initialization that improves over Base in every environment. Because all methods share the backbone, the trajectory mixture, and the imitation learning stage, this pattern across the four evaluated environment families supports targets shaped by the policy’s decision needs as a more reliable learning signal than targets derived from the next observation.

6 Analysis

The experiments establish the performance gains of AAWM over next-observation prediction. We next examine how its initialization affects early GRPO behavior and whether *Self-Probing* and *Dynamics Synthesis* form a useful, evidence-grounded target construction process. We then study the number of propositions and control for supervision budget and synthesizer scale.

Training dynamics. A policy that converges faster could be collapsing to a narrow output distribution or exploring more effectively. We first track success rate and response entropy during GRPO, alongside response length and rollout content. We compute response entropy as the token-level entropy of the policy’s next-token distribution, averaged over generated response tokens. Figure 3 shows the WebShop dynamics, and Appendix E reports the matching ALFWorld curves. The success curves show that AAWM learns faster. On WebShop all three methods stay low during the first ten steps, after which AAWM rises much more rapidly than Base and IWM and keeps the highest success rate for most of training, and ALFWorld follows the same trend. The entropy curves show that this faster convergence does not coincide with an early collapse in the response distribution.

To complement the entropy curves with direct behavioral evidence, we pool equal-budget rollouts from GRPO steps 0 and 50. We measure how broadly the policies visit environment states, how many distinct executable actions they take, and how many distinct state-action pairs occur in successful rollouts. Table 4 reports these three aspects of early exploration.

Method	TextCraft	SciWorld	WebShop	ALFWorld	All
Prompting					
Qwen2.5	18.3 $\pm$ 1.5	11.7 $\pm$ 3.1	21.3 $\pm$ 0.8	3.5 $\pm$ 1.8	13.0 $\pm$ 0.5
Imitation Learning initialized by World Modeling					
Base	31.7 $\pm$ 1.5	70.2 $\pm$ 2.8	94.5 $\pm$ 0.9	33.3 $\pm$ 2.0	61.1 $\pm$ 1.8
IWM	31.3 $\pm$ 1.5	79.8 $\pm$ 2.8	83.0 $\pm$ 2.5	30.2 $\pm$ 2.3	59.6 $\pm$ 2.4
IWM-Summary	32.3 $\pm$ 2.1	84.2 $\pm$ 2.8	94.5 $\pm$ 2.3	20.0 $\pm$ 2.2	61.4 $\pm$ 2.0
AAWM (Ours)	35.7$\pm$1.2	91.2$\pm$1.8	97.5$\pm$0.5	36.0$\pm$1.8	69.3$\pm$1.3

Table 3: Results on four AgentGym environments at Qwen2.5-1.5B-Instruct. All trained rows share the same AgentTraj-L trajectory mixture and imitation learning stage. All is weighted by the number of evaluation tasks per environment. Cells report success rate (%), mean $\pm$ std over three seeds.

Environment	Method	States	Valid actions	Success pairs
ALFWorld	IWM	932	375	629
	AAWM	887	342	683
WebShop	IWM	376	181	141
	AAWM	468	218	199

Table 4: Early GRPO behavior under equal rollout budgets, pooling steps 0 and 50. States, Valid actions, and Success pairs count distinct visited states, distinct executable actions taken, and distinct state-action pairs in successful rollouts, respectively.

On ALFWorld, AAWM visits slightly fewer raw states and actions but reaches more successful state-action pairs. On WebShop, it improves both raw coverage and successful behavior diversity. Its outputs are also longer on average, with 167.6 tokens on WebShop and 98.7 on ALFWorld against 121.4 and 57.8 for IWM, and the extra tokens appear mainly in the reasoning trace. Representative rollouts in Appendix I confirm that the AAWM initialized policy reasons about environment mechanisms before acting, including object persistence, action preconditions, and product attributes. Together, the behavioral measures and training dynamics support more success-oriented exploration after AAWM initialization, rather than uniformly greater raw coverage.

Alignment of Self-Probing and Synthesis. We evaluate whether the two stages of target construction operate as intended. A useful process should identify propositions that matter for action choice, resolve open questions when the evidence permits, correct mistaken beliefs, and keep synthesized claims grounded in the supplied transitions. For each environment, we randomly sample 100 confirmed patterns and 100 open questions.

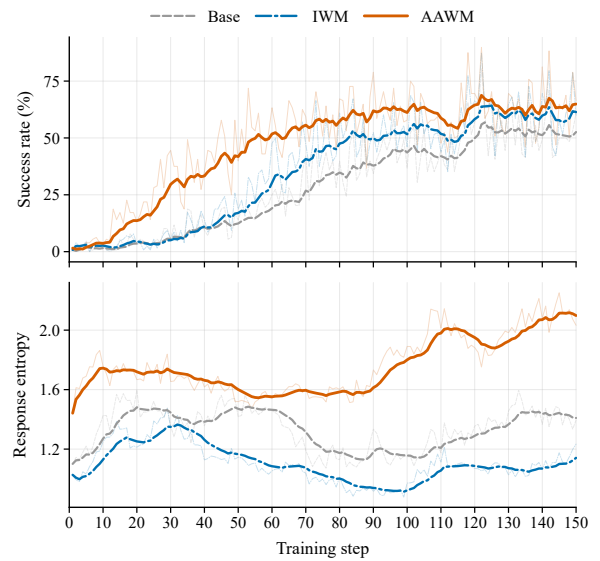


Figure 3: GRPO training dynamics on WebShop under the imitation learning then reinforcement learning setting initialized by world modeling. The backbone is Qwen2.5-1.5B-Instruct. The top panel shows success rate, and the bottom panel shows response entropy. Bold curves show time weighted EMA smoothing with coefficient 0.95. AAWM converges faster while maintaining higher response entropy during training.

Three independent LLM judges, Qwen3.6-Flash, DeepSeek-V4-Flash (DeepSeek-AI, 2026), and GLM-5 (GLM-5-Team, 2026), evaluate each example, and their majority vote gives the model score in Table 5. We also obtain human labels for 10 randomly sampled examples of each proposition type in each environment. *Decision helpfulness* measures whether a proposition could inform action choice. *Question resolution* and *pattern correction* measure how synthesis handles open questions and mistaken beliefs. *Factual correctness* measures whether the synthesized claims are supported by

Criterion	Majority	Human	Agreement / κ
Decision helpfulness	96.5%	95%	95% / .643
Question resolution	85%	80%	90% / .737
Pattern correction	17%	30%	70% / .348
Factual correctness	80%	70%	80% / .524

Table 5: Independent evaluation of *Self-Probing* and *Dynamics Synthesis*. Majority is the positive rate under majority vote from the three LLM judges. Agreement and Cohen’s κ compare the majority vote with human labels.

the supplied transitions.

Decision helpfulness and question resolution are both high, indicating that Self-Probing focuses on information useful for action choice and that synthesis usually answers the resulting open questions. Pattern correction is lower, though synthesis still revises some confirmed patterns. This rate is naturally lower because a positive case requires the original pattern to be mistaken and the synthesis output to correct it. Factual correctness further shows that most synthesized claims are supported by the transition evidence. Together, these results support the decision relevance and evidence grounding of the constructed targets. Representative triples are in Appendix H.

Sensitivity to the number of propositions. We vary the number of confirmed patterns and open questions n produced at each state (Appendix A.3). Increasing n from 1 to 3 clearly improves both environments, suggesting that a single proposition may not cover the dynamics needed for action selection, while moving from 3 to 5 adds little. We therefore use $n = 3$ in all main experiments as a compact setting that captures most of the benefit.

Supervision and synthesizer controls. Finally, we test whether the gains can be attributed to a larger supervision budget or reliance on the 30B synthesis model. Appendix B reports target token counts, matched-budget comparisons, construction cost, and synthesizer scale. Although each AAWM target is longer on average, using one target per state gives 46% fewer assistant target tokens on ALFWorld and 71% fewer on WebShop over two world modeling SFT epochs. AAWM remains stronger than IWM at every matched token budget, and the advantage remains when the 7B policy itself serves as the synthesizer. These controls show that the gains do not depend on more supervised tokens or a separate 30B construction model.

7 Conclusion

World modeling for language agents has typically predicted the next environment output, but prediction accuracy and decision quality serve different goals. AAWM addresses this gap by letting the policy state what it understands and what remains uncertain, then constructing training targets from these statements. Across model scales, training settings, and environment families, AAWM consistently outperforms next-observation world modeling. A natural extension is to repeat *Self-Probing* after each training round so that targets evolve with the policy’s understanding. More broadly, these results suggest that world modeling targets should be evaluated by the decisions they help the policy make.

Limitations

AAWM runs *Self-Probing* once during data construction, so the training target reflects the policy’s beliefs at that point rather than its beliefs as training proceeds. Repeating *Self-Probing* after each training round could turn the target into a curriculum that follows the policy’s changing understanding, at the cost of additional construction compute, and we leave this to future work.

Our experiments also cover text-only environments. Extending AAWM to multimodal settings, where a vision language policy must model dynamics from both images and text, is a natural next step that we have not yet explored.

All trained backbones in our experiments come from the Qwen model family. We report three-seed means for the main comparisons and mark single-run controls explicitly. A broader statistical study across more model families remains future work.

Ethical Considerations

This work studies world modeling for language agents in sandboxed benchmark environments. Because the proposed method improves sequential decision-making ability, similar techniques could potentially be misused in open-ended web, GUI, or tool-use settings where autonomous actions may have external consequences.

Our experiments do not deploy agents in real-world systems, and practical deployment would require additional safeguards, monitoring, and validation of generated targets.

Acknowledgments

This research is supported by the Big Data Computing Center of Southeast University.

References

- Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. [GPT-4 technical report](#). *arXiv preprint arXiv:2303.08774*.
- Jacob Berg, Chuning Zhu, Yanda Bao, Ishan Durugkar, and Abhishek Gupta. 2025. Semantic world models. *arXiv preprint arXiv:2510.19818*.
- Jaime Carbonell and Jade Goldstein. 1998. [The use of MMR, diversity-based reranking for reordering documents and producing summaries](#). In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 335–336.
- Hyungjoo Chae, Namyoun Kim, Kai Tzu-iunn Ong, Minju Gwak, Gwanwoo Song, Jihoon Kim, Sunghwan Kim, Dongha Lee, and Jinyoung Yeo. 2025. [Web agents with world models: Learning and leveraging environment dynamics in web navigation](#). In *The Thirteenth International Conference on Learning Representations*.
- Shiqi Chen, Tongyao Zhu, Zian Wang, Jinghan Zhang, Kangrui Wang, Siyang Gao, Teng Xiao, Yee Whye Teh, Junxian He, and Manling Li. 2025. Internalizing world models via self-play finetuning for agentic RL. *arXiv preprint arXiv:2510.15047*.
- DeepSeek-AI. 2026. DeepSeek-V4: Towards highly efficient million-token context intelligence.
- Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. 2025. [Group-in-group policy optimization for LLM agent training](#). *Advances in Neural Information Processing Systems*, 38:46375–46408.
- Gemini Team, Google. 2025. [Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities](#). Technical report, Google DeepMind.
- GLM-5-Team. 2026. [GLM-5: from vibe coding to agentic engineering](#). *Preprint*, arXiv:2602.15763.
- GLM Team, Aohan Zeng, Bin Xu, Bowen Wang, Chenhui Zhang, Da Yin, Dan Zhang, Diego Rojas, Guanyu Feng, Hanlin Zhao, and 1 others. 2024. [ChatGLM: A family of large language models from GLM-130B to GLM-4 all tools](#). *arXiv preprint arXiv:2406.12793*.
- Zhibin Gou, Zhihong Shao, Yeyun Gong, Yelong Shen, Yujia Yang, Minlie Huang, Nan Duan, and Weizhu Chen. 2024. [ToRA: A tool-integrated reasoning agent for mathematical problem solving](#). In *The Twelfth International Conference on Learning Representations*.
- Shangmin Guo, Omar Darwiche Domingues, Raphaël Avalos, Aaron Courville, and Florian Strub. 2025. World modelling improves language model agents. *arXiv preprint arXiv:2506.02918*.
- Izzeddin Gur, Hiroki Furuta, Austin Huang, Mustafa Safdari, Yutaka Matsuo, Douglas Eck, and Aleksandra Faust. 2024. [A real-world webagent with planning, long context understanding, and program synthesis](#). In *The Twelfth International Conference on Learning Representations*.
- Wenyi Hong, Weihang Wang, Qingsong Lv, Jiazheng Xu, Wenmeng Yu, Junhui Ji, Yan Wang, Zihan Wang, Yuxiao Dong, Ming Ding, and 1 others. 2024. [CogA: A visual language model for GUI agents](#). In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14281–14290.
- Mengkang Hu, Bowei Xia, Yuran Wu, Ailing Yu, Yude Zou, Qiguang Chen, Shijian Wang, Jiarui Jin, Kexin Li, Wenxiang Jiao, Yuan Lu, and Ping Luo. 2025. [Agent2World: Learning to generate symbolic world models via adaptive multi-agent feedback](#). *arXiv preprint arXiv:2512.22336*.
- Kimi Team, Yifan Bai, Yiping Bao, Y Charles, Cheng Chen, Guanduo Chen, Haiting Chen, Huarong Chen, Jiahao Chen, Ningxin Chen, and 1 others. 2025. [Kimi K2: Open agentic intelligence](#). *arXiv preprint arXiv:2507.20534*.
- Margaret Li, Weijia Shi, Artidoro Pagnoni, Peter West, and Ari Holtzman. 2024. Predicting vs. acting: A trade-off between world modeling & agent modeling. *arXiv preprint arXiv:2407.02446*.
- Yixia Li, Hongru Wang, Jiahao Qiu, Zhenfei Yin, Dongdong Zhang, Cheng Qian, Zeping Li, Xiaoteng Ma, Guanhua Chen, and Heng Ji. 2026a. [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)*, pages 8084–8111, San Diego, California, United States. Association for Computational Linguistics.
- Yu Li, Guangfeng Cai, Shengtian Yang, Han Luo, Shuo Han, Xu He, Dong Li, and Lei Feng. 2026b. [PhGPO: Pheromone-guided policy optimization for long-horizon tool planning](#). *arXiv preprint arXiv:2602.13691*.
- Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, and 1 others. 2024. [DeepSeek-V3 technical report](#). *arXiv preprint arXiv:2412.19437*.
- Suraj Nair, Silvio Savarese, and Chelsea Finn. 2020. [Goal-aware prediction: Learning to model what matters](#). In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 7207–7219. PMLR.

- Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, and 1 others. 2021. WebGPT: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332*.
- OpenAI. 2024. [GPT-4o system card](#).
- Joon Sung Park, Joseph O’Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. 2023. [Generative agents: Interactive simulacra of human behavior](#). In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, pages 1–22.
- Shuofei Qiao, Runnan Fang, Ningyu Zhang, Yuqi Zhu, Xiang Chen, Shumin Deng, Yong Jiang, Pengjun Xie, Fei Huang, and Huajun Chen. 2024. [Agent planning with world knowledge model](#). In *Advances in Neural Information Processing Systems*, volume 37, pages 114843–114871.
- Qwen Team. 2024. [Qwen2.5 technical report](#). *arXiv preprint arXiv:2412.15115*.
- Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language models can teach themselves to use tools. *Advances in neural information processing systems*, 36:68539–68551.
- Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, and 1 others. 2024. [DeepSeekMath: Pushing the limits of mathematical reasoning in open language models](#). *arXiv preprint arXiv:2402.03300*.
- Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning. *Advances in neural information processing systems*, 36:8634–8652.
- Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. 2021. [ALFWorld: Aligning text and embodied environments for interactive learning](#). In *International Conference on Learning Representations*.
- Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. 2024. [Voyager: An open-ended embodied agent with large language models](#). *Transactions on Machine Learning Research*.
- Kangrui Wang, Pingyue Zhang, Zihan Wang, Yaning Gao, Linjie Li, Qineng Wang, Hanyang Chen, Chi Wan, Yiping Lu, Zhengyuan Yang, Lijuan Wang, Ranjay Krishna, Jiajun Wu, Fei-Fei Li, Yejin Choi, and Manling Li. 2025. VAGEN: Reinforcing world model reasoning for multi-turn VLM agents. In *Advances in Neural Information Processing Systems*, volume 38, pages 172871–172933.
- Zhiheng Xi, Yiwen Ding, Wenxiang Chen, Boyang Hong, Honglin Guo, Junzhe Wang, Xin Guo, Dingwen Yang, Chenyang Liao, Wei He, Songyang Gao, Lu Chen, Rui Zheng, Yicheng Zou, Tao Gui, Qi Zhang, Xipeng Qiu, Xuanjing Huang, Zuxuan Wu, and Yu-Gang Jiang. 2025. [AgentGym: Evaluating and training large language model-based agents across diverse environments](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 27914–27961, Vienna, Austria. Association for Computational Linguistics.
- An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.
- John Yang, Carlos E Jimenez, Alexander Wettig, Kilian Lieret, Shunyu Yao, Karthik Narasimhan, and Ofir Press. 2024. [SWE-agent: Agent-computer interfaces enable automated software engineering](#). *Advances in Neural Information Processing Systems*, 37:50528–50652.
- Kaibing Yang, Guangfeng Cai, Shengtian Yang, Shuo He, Yu Li, Mengyi Liu, Pengwei Chen, Jun Xu, and Lei Feng. 2026a. [Progress-conditioned group policy optimization for long-horizon agentic tasks](#). *arXiv preprint arXiv:2607.22724*.
- Shengtian Yang, Yu Li, Shuo He, Yewen Li, Qingpeng Cai, Peng Jiang, and Lei Feng. 2026b. [Phase-aware mixture of experts for agentic reinforcement learning](#). In *Forty-third International Conference on Machine Learning*.
- Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. 2022. WebShop: Towards scalable real-world web interaction with grounded language agents. *Advances in Neural Information Processing Systems*, 35:20744–20757.
- Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. [ReAct: Synergizing reasoning and acting in language models](#). In *The Eleventh International Conference on Learning Representations*.
- Xiao Yu, Baolin Peng, Ruize Xu, Michel Galley, Hao Cheng, Suman Nath, Jianfeng Gao, and Zhou Yu. 2025. Dyna-Think: Synergizing reasoning, acting, and world model simulation in AI agents. *arXiv preprint arXiv:2506.00320*.
- Kai Zhang, Xiangchao Chen, Bo Liu, Tianci Xue, Zeyi Liao, Zhihan Liu, Xiyao Wang, Yuting Ning, Zhaorun Chen, Xiaohan Fu, Jian Xie, Yuxuan Sun, Boyu Gou, Qi Qi, Zihang Meng, Jianwei Yang, Ning Zhang, Xian Li, Ashish Shah, and 11 others. 2026. [Agent learning via early experience](#). In *Forty-third International Conference on Machine Learning*.
- Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang,

Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. 2025. Qwen3 Embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*.

Zhuosheng Zhang and Aston Zhang. 2024. [You only look at screens: Multimodal chain-of-action agents](#). In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 3132–3149, Bangkok, Thailand. Association for Computational Linguistics.

A Experimental Details

A.1 Main Experiment Details

Trajectory selection. AgentTraj-L is a collection of high-quality agent trajectories released with AgentGym (Xi et al., 2025). For the main ALFWorld and WebShop experiments, we select complete trajectories whose action sequences can be executed from start to finish in the GiGPO open-source evaluation code and reach a successful outcome. The selected set $\mathcal{D}$ contains 1,655 trajectories with 19,039 action steps for ALFWorld, and 1,027 trajectories with 4,971 action steps for WebShop.

Transition collection. We build the transition bank from every action step in $\mathcal{D}$. At each step, we execute the action in the trajectory and three additional actions from the same state, so each step contributes exactly four one-step transitions. For ALFWorld, the three additional actions are sampled uniformly without replacement from the admissible action set after removing the trajectory action and help. For WebShop, two additional actions are sampled from the matched policy backbone with temperature 0.8, and one additional action is sampled uniformly from the admissible action set after removing the trajectory action. Sampled actions are canonicalized and deduplicated. If a policy sampled action cannot be executed or duplicates an existing action, we replace it with an unused admissible action. This procedure produces exactly three additional actions for every step. The final transition bank contains 76,156 one-step transitions for ALFWorld and 19,884 one-step transitions for WebShop.

Training and evaluation separation. All trajectories in $\mathcal{D}$ and all records in $\mathcal{T}$ come from the training split. Evaluation trajectories do not enter world modeling data construction, and retrieval excludes records from the same episode as the query. We find no identical task or goal text between the training and evaluation splits. As an additional check within the training data, we compare each query goal with the goal of every retrieved source trajectory. Table 6 shows that exact and near nonexact matches are rare.

Supervised fine-tuning settings. World model fine-tuning and imitation learning both use full-parameter supervised fine-tuning. The learning rate is 1×10^{-5} . The effective batch size is 16 for ALFWorld and 8 for WebShop. The world modeling stage uses two epochs for both IWM and AAWM.

Environment	Exact goal	Near nonexact goal
ALFWorld	0.9%	0.0%
WebShop	0.0%	0.3%

Table 6: Goal overlap between training queries and retrieved source trajectories. Near nonexact matches have token set Jaccard similarity $0.95 \leq J < 1$.

The imitation learning stage uses one epoch. The imitation learning subset contains 32 trajectories from $\mathcal{D}$, with 429 action steps for ALFWorld and 158 action steps for WebShop. For each action step, the model is supervised to generate the action content in the format `<action>...</action>`.

GRPO settings. We use GRPO for all reinforcement learning experiments and follow the public implementation of Feng et al. (2025). For ALFWorld, the maximum prompt length is 2048 tokens, the maximum response length is 512 tokens, and each episode allows up to 50 environment steps. For WebShop, the maximum prompt length is 4096 tokens, the maximum response length is 512 tokens, and each episode allows up to 15 environment steps. Both environments use a rule-based reward of 10 for success and 0 for failure, with a penalty of -0.1 for invalid actions. The actor learning rate is 1×10^{-6} . For group-based RL, the group size is 8 and each rollout samples 16 groups, giving 128 environments in total. The rollout temperature is 1.0, the validation temperature is 0.4, and the KL coefficient is 0.01. The mini batch size is 256 for ALFWorld and 64 for WebShop.

A.2 Multi-Environment Evaluation Details

The multi-environment experiment uses the AgentGym evaluation code and is separate from the main ALFWorld and WebShop setting. It trains on the full AgentTraj-L train split for TextCraft, SciWorld, WebShop, and ALFWorld, and evaluates on the full AgentTraj-L eval split for each environment. No additional branch transitions are collected in this experiment, because the mixed train split already provides broad transition coverage across environments. For AAWM, the immediate transition set $\mathcal{I}_t$ contains only the original one-step transition at the corresponding step.

The training data contain 2,420 trajectories for ALFWorld, 3,930 for WebShop, 374 for TextCraft, and 2,120 for SciWorld, for a total of 8,844 trajectories. Evaluation uses 200 tasks each for ALFWorld, WebShop, and SciWorld, and 100 tasks for TextCraft. The backbone is Qwen2.5-1.5B-Instruct.

n	ALFWorld	WebShop
1	10.7 $\pm$ 2.2	7.8 $\pm$ 0.8
3	24.7 $\pm$ 2.6	10.7 $\pm$ 1.2
5	26.0 $\pm$ 1.2	11.7 $\pm$ 0.8

Table 7: Sensitivity to the number of Self-Probing propositions n at Qwen2.5-1.5B-Instruct. Each setting uses the same world model fine-tuning and lightweight imitation learning protocol and is evaluated without GRPO. ALFWorld reports aggregate success rate, and WebShop reports success rate. Cells report mean $\pm$ std over three seeds.

All supervised runs use full fine-tuning with learning rate 1×10^{-5} and effective batch size 32. IWM, IWM-Summary, and AAWM use two epochs of world model fine-tuning followed by one epoch of imitation learning. Base only uses one epoch of imitation learning without world model fine-tuning. The IWM-Summary baseline summarizes observations in all four environments using Qwen3-30B-A3B-Instruct-2507 before next-observation prediction. Evaluation uses greedy decoding with temperature 0.0. The maximum numbers of interaction rounds are 20 for TextCraft, 30 for SciWorld, 6 for WebShop, and 30 for ALFWorld.

A.3 Sensitivity to Self-Probing Propositions

Table 7 varies the number of confirmed patterns and open questions n produced at each state. Increasing n from 1 to 3 gives a clear improvement on both environments, showing that a single proposition is insufficient to cover the dynamics needed for action selection. Increasing n from 3 to 5 yields smaller additional gains. We therefore use $n = 3$ in all main experiments as a compact setting that captures most of the benefit.

B Additional Control Analyses

B.1 State-Change Target

State-Change uses the same construction model and training protocol as the main 1.5B experiments, but its target describes only the task-relevant difference between o_t and o_{t+1} . In a single run, it reaches 9.4% success on ALFWorld and 5.7% on WebShop. This control remains below AAWM on both environments, suggesting that transition abstraction alone does not reproduce decision-oriented target construction.

Method	ALFWorld	WebShop
State-Change	9.4	5.7

Table 8: Single-run results for the State-Change target at Qwen2.5-1.5B-Instruct. Both environments report success rate.

B.2 Supervision Budgets

Table 9 reports the assistant target tokens used during two epochs of world modeling SFT. Each AAWM target is longer on average, but AAWM produces one target per state while IWM produces one target for each of four transitions. As a result, AAWM uses 46% fewer target tokens on ALFWorld and 71% fewer on WebShop.

Environment	Method	Targets / state	Avg. tokens	Total tokens
ALFWorld	IWM	4	44.5	6.780M
	AAWM	1	96.3	3.665M
WebShop	IWM	4	98.8	3.928M
	AAWM	1	113.2	1.125M

Table 9: Assistant target tokens over two world modeling SFT epochs.

We also train both methods at the original AAWM budget B_A , the mean budget B_M , and the original IWM budget B_I . Table 10 shows that AAWM remains stronger at every matched point. The benefit therefore does not come from receiving more supervised target tokens.

Environment	Method	B_A	B_M	B_I
ALFWorld	IWM	3.1	5.4	3.1
	AAWM	24.7	19.6	25.0
WebShop	IWM	1.6	2.3	2.1
	AAWM	10.7	9.4	4.7

Table 10: Success rates under matched assistant target token budgets. The budgets B_A , B_M , and B_I are 3.67M, 5.22M, and 6.78M tokens on ALFWorld and 1.13M, 2.53M, and 3.93M tokens on WebShop.

B.3 Construction Cost and Synthesizer Scale

Target construction is performed once before downstream training. Table 11 shows that most of its token use comes from Self-Probing and Dynamics Synthesis, while retrieval takes a small fraction of runtime. Each target uses three confirmed patterns and three open questions, with three retrieved records for each proposition. It therefore receives 18 retrieved records by design.

We further use Qwen2.5-7B-Instruct as both the policy and the AAWM synthesizer. The same 7B

Environment	Self-Probing input / output	Synthesis input / output	Retrieval time	Branch output
ALFWorld	12.70M / 2.05M	34.85M / 1.83M	0.3%	0
WebShop	4.70M / 0.45M	22.55M / 0.56M	0.1%	0.175M

Table 11: Offline construction accounting at the 1.5B policy scale. All token values are totals over the construction set. Retrieval time is its share of total construction runtime. Branch output counts tokens generated for alternative actions: ALFWorld samples all alternatives uniformly without an LLM call, whereas WebShop samples two of the three from the policy backbone.

Method	Constructor	ALFWorld	WebShop
IWM	raw / 7B summary	49.6 $\pm$ 4.8	9.1 $\pm$ 3.5
AAWM	7B	52.3 $\pm$ 2.1	13.0 $\pm$ 1.6
AAWM	30B	53.4$\pm$5.0	16.3$\pm$2.3

Table 12: Synthesizer scale at the Qwen2.5-7B-Instruct policy scale. Cells report mean $\pm$ std over three seeds.

model produces the WebShop summaries for IWM, while ALFWorld IWM continues to use raw observations. Table 12 shows that AAWM retains its advantage with this matched 7B constructor. The smaller synthesizer nearly preserves ALFWorld performance and gives a smaller result on WebShop, where synthesis must integrate more product information.

C Direct Comparison of World Modeling Targets

The following two examples compare the supervision generated by next-observation world modeling and AAWM for matched decision contexts.

ALFWorld case
Task. heat some cup and put it in cabinet. Current observation. You arrive at microwave 1. The microwave 1 is closed.
IWM target. (given the action “heat cup 1 with microwave 1”) You heat the cup 1 using the microwave 1.
AAWM target. The environment allows heating an item with the microwave without requiring the microwave to be open or the item to be placed inside; the heating action automatically processes the item while it is in hand, and the item remains in hand afterward. Consequently, the microwave’s state does not affect the heating action, and no retrieval from the microwave is needed after heating.
Interpretation. The IWM target states only the immediate outcome and misses the unusual action preconditions and postconditions. The AAWM target captures the decision-critical dynamics: the closed microwave does not block heating, and the cup remains in hand, so the agent can proceed directly to the cabinet without

opening or retrieving it.

WebShop variant-selection case

Task. Find me hand wash men’s sleep & lounge with long sleeve, elastic waistband for daily wear with color: multi 9, and size: medium, and price lower than 80.00 dollars.

Current observation. Instruction: [SEP] Find me hand wash men’s sleep & lounge with long sleeve, elastic waistband for daily wear with color: multi 9, and size: medium, and price lower than 80.00 dollars [SEP] Back to Search [SEP] < Prev [SEP] size [SEP] small [SEP] medium [SEP] large [SEP] x-large [SEP] xx-large [SEP] color [SEP] multi 1 [SEP] multi 10 [SEP] multi 2 [SEP] multi 3 [SEP] multi 4 [SEP] multi 5 [SEP] multi 6 [SEP] multi 7 [SEP] multi 8 [SEP] multi 9 [SEP] InterestPrint Deer and Stars Men’s Loungewear Pajama Sets - Long Sleeve Tee and Jogger Pant [SEP] Price: \$43.59 [SEP] Rating: N.A. [SEP] Description [SEP] Features [SEP] Reviews [SEP] Buy Now

IWM-Summary target. (given the action “click[medium]”)

This is a product search result page displaying a men’s loungewear item matching the specified criteria. The active filters show size “medium” and color “multi 9” selected, with the featured product being “Interest-Print Deer and Stars Men’s Loungewear Pajama Sets” priced at \$43.59. Key visible elements include the product name, price, and navigation options like “Back to Search” and “Prev”, alongside the structured size and color filter lists. The page confirms the search parameters are applied and a relevant item is listed within the price range.

AAWM target. The environment requires explicit selection of size and color on the product page to capture the desired specifications for purchase. The search query’s filters do not automatically apply to the product’s variant options; instead, the system only records selections made via the product page’s filters. Selecting size “medium” and color “multi 9” before purchasing ensures the correct variant is purchased.

Interpretation. The IWM-Summary target describes the next page and even suggests that both requested options are already applied, but it does not expose the selection mechanism. The AAWM target captures the key dynamics: query attributes do not set product variants.

Method	ALFWorld	WebShop
Reinforcement Learning initialized by World Modeling		
Base	71.6 ± 2.8	58.6 ± 3.8
IWM	13.2 ± 3.2	54.2 ± 2.7
AAWM (Ours)	70.6 ± 2.7	63.8 ± 2.7

Table 13: Direct reinforcement learning at Qwen2.5-1.5B-Instruct. All rows start GRPO without the lightweight imitation learning stage. Base denotes GRPO directly from the backbone. IWM and AAWM denote GRPO from the corresponding world modeling initialization. ALFWorld reports aggregate success rate, and WebShop reports success rate. Cells report mean $\pm$ std over three seeds.

D Reinforcement Learning without Imitation Learning

We isolate whether world modeling initialization remains useful when GRPO starts without the lightweight imitation learning stage. Table 13 compares direct GRPO from the backbone with GRPO from the two world modeling initializations at Qwen2.5-1.5B-Instruct.

Without imitation learning, IWM reaches only 13.2% on ALFWorld and 54.2% on WebShop. This is far below direct GRPO from the backbone on ALFWorld and also lower on WebShop. By contrast, AAWM reaches 70.6% on ALFWorld and 63.8% on WebShop. It therefore preserves the direct GRPO performance on ALFWorld and improves WebShop over the base row. This shows that AAWM relies less on imitation learning than next observation prediction. A likely reason is that AAWM trains on decision-oriented dynamics, so its world modeling target is closer to the information used by the policy during reinforcement learning.

E Additional Training Dynamics

Figure 4 shows the ALFWorld training dynamics under the same imitation learning then reinforcement learning setting as Figure 3. AAWM reaches higher success earlier than Base and IWM, while also maintaining higher response entropy through most of training. This matches the WebShop pattern and shows that the faster improvement of AAWM does not come from early output collapse. Together with the behavioral measures in Table 4, this pattern supports more success-oriented exploration during early GRPO training.

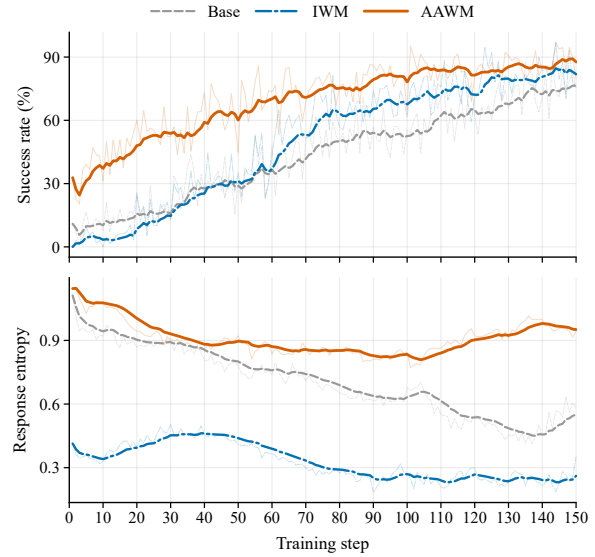


Figure 4: GRPO training dynamics on ALFWorld under the imitation learning then reinforcement learning setting initialized by world modeling. The backbone is Qwen2.5-1.5B-Instruct. The top panel shows success rate, and the bottom panel shows response entropy. Bold curves show time-weighted EMA smoothing with coefficient 0.95. AAWM reaches higher success earlier while maintaining higher response entropy.

F Prompt Templates

This appendix reproduces the prompt templates used in Self-Probing and Dynamics Synthesis (Sections 4.1 and 4.3) and evaluation (Section 6). Curly-brace tokens ($\{\dots\}$) denote runtime variables.

Self-Probing prompt

{decision_context}

You are an agent operating in an interactive environment. Before taking your next action, externalize your understanding — your internal model of how this environment operates.

Output exactly in this format — two categories, with the number of statements specified below:

[CONFIRMED PATTERNS]

Cause-and-effect relationships, requirements, or constraints you believe to be true about how this environment operates. State them as general rules you’ve inferred, not observations you currently see.

[OPEN QUESTIONS]

Specific uncertainties about environment mechanics whose answers would affect which action you choose next.

Format: Each statement on its own line. No numbering, no bullet points.

Your statements will be verified and answered, then provided back to you to inform your current decision.

For this environment, output exactly {max_beliefs} confirmed pattern(s) and exactly {max_questions} open question(s). Do not output extra statements.

The variables `max_beliefs` and `max_questions` are set to n , the per-state proposition count reported in [Section 5.1](#).

Dynamics Synthesis prompt

```
{decision_context}

— Agent’s Current Understanding —
[CONFIRMED PATTERNS]
{confirmed_patterns}

[OPEN QUESTIONS]
{open_questions}

— Retrieved Evidence —
{retrieved_transition_evidence}

— Current Step Outcome —
{immediate_transitions}

Internalize the retrieved evidence and the current step
outcomes shown above, then perform the following
task. Write as if stating knowledge directly — do not
reference or cite the evidence records.

Your task:
1. Correct any confirmed patterns that appear to be
wrong.
2. Answer the open questions where possible.
3. Synthesize the key environment dynamics most
relevant to the current task goal and action selection.

Requirements:
– Describe cause-effect relationships specific to this
environment — not general knowledge any agent would
already possess.
– Focus on dynamics bearing on the current task goal
and the situation at hand.
– DO NOT contradict the facts observed in the current
and prior steps.
– DO NOT include action suggestions or plans.
– Write a concise, coherent paragraph with no preamble
or closing remarks.
```

The `retrieved_transition_evidence` field groups K retrieved transitions under the proposition that retrieved them:

Retrieved transition formatting

```
For confirmed pattern 1:
Record 1:
  State: "{observation_t}"
  Action: "{action_t}"
  Result: "{observation_t1}"
...
Record K:
  State: "{observation_t}"
  Action: "{action_t}"
  Result: "{observation_t1}"
...

For open question n:
Record 1:
  State: "{observation_t}"
  Action: "{action_t}"
  Result: "{observation_t1}"
...
Record K:
```

```
State: "{observation_t}"
Action: "{action_t}"
Result: "{observation_t1}"
```

Probing effectiveness judge prompt

You are judging one proposition about an interactive text environment and one synthesis paragraph.

Input fields:

- `state_context`: task and current environment context.
- `type`: P or Q.
- `proposition`: a single confirmed pattern (P) or open question (Q).
- `synthesis_output`: a paragraph written after seeing transition evidence.

Definitions:

P means the agent states a dynamics pattern it believes to hold.

Q means the agent asks about a dynamics point it wants resolved.

```
<sample>
state_context:
{state_context}
```

```
type:
{type}
```

```
proposition:
{proposition}
```

```
synthesis_output:
{synthesis_output}
</sample>
```

Dimensions for judgment:

D1 Decision-Helpfulness: Output 1 if the proposition concerns an environment mechanism, precondition, dependency, constraint, or action effect that could meaningfully inform task completion or current/future action choices. Output 0 if it is only a surface description, tautology, generic observation, formatting detail, or irrelevant fact.

D2 Synthesis Contribution:

- If `type=P`, output 1 only when the proposition contains an error, inaccuracy, or over-confident unsupported claim, and `synthesis_output` explicitly corrects it. Otherwise output 0, including when a correct P is merely restated or extended.

- If `type=Q`, output 1 only when `synthesis_output` gives a definite answer to the open question. Output 0 if it is silent, hedged, partial, or only restates the question.

Instruction:

Return strict JSON only, with fields in this order. The two score fields must be integers, either 0 or 1:

```
{"rationale": "<=50
words", "d1_decision_helpfulness": 0,
"d2_synthesis_contribution": 1}
```

G Transition Retrieval Details

Each proposition $q \in P_t \cup Q_t$ from Self-Probing acts as an independent retrieval query against the transition pool $\mathcal{T}$ ([Section 4.2](#)). Transitions are embedded with Qwen3-Embedding-0.6B and indexed with FAISS inner-product search over normalized

Algorithm 1 Per-query MMR retrieval (run independently for each proposition).

Require: Proposition embedding q ; transition index $\mathcal{T}$; recall pool size R ; final hits K ; trade-off λ

- 1: $C \leftarrow$ top- R from $\mathcal{T}$ by $\langle q, c \rangle$, excluding same-episode hits
- 2: $S \leftarrow \emptyset$
- 3: **while** $|S| < K$ **do**
- 4: **for** each $c \in C \setminus S$ **do**
- 5: $d(c, S) \leftarrow 0$ if $S = \emptyset$; otherwise $d(c, S) \leftarrow \max_{s \in S} \langle c, s \rangle$
- 6: $\text{score}(c) \leftarrow \lambda \langle q, c \rangle - (1-\lambda)d(c, S)$
- 7: $c^* \leftarrow \arg \max_c \text{score}(c)$
- 8: $S \leftarrow S \cup \{c^*\}$
- 9: **return** S

Parameter	Value
Recall pool per query (R)	15
Final hits per query (K)	3
MMR trade-off (λ)	0.5
Embedding model	Qwen3-Embedding-0.6B
Index type	FAISS inner product

Table 14: Transition retrieval hyperparameters.

embeddings. Each indexed unit is a single transition record (o_t, a_t, o_{t+1}) .

Per-query MMR. For each proposition q , we first retrieve a recall pool of R candidates by embedding similarity, then apply maximal marginal relevance to select the final K transitions that balance relevance to q with diversity among the selected set. Algorithm 1 describes the procedure. Each proposition is processed independently; there is no cross-query deduplication, so the same transition may appear under different propositions. When retrieving for a state from trajectory τ_i , transitions originating from the same episode and overlapping step are excluded.

When S is empty, the redundancy term is zero, so the first record is selected by relevance alone.

Hyperparameters. Table 14 lists the retrieval configuration. The recall pool R is larger than the final K to give MMR room to diversify within each proposition’s candidate set.

H Self-Probing and Dynamics Synthesis Pairs

This appendix presents four representative triples produced during AAWM data construction. Each case displays the task, the current observation, one confirmed pattern and one open question elicited by Self-Probing, and the corresponding responses in the synthesis output. Where synthesis sentences unrelated to the displayed propositions are omitted,

the omission is marked with an ellipsis.

ALFWorld Case 1

Task. Clean some potato and put it in microwave.

Observation. You arrive at sinkbasin 1. On the sinkbasin 1, you see nothing.

Confirmed pattern. “I believe cleaning a potato with the sinkbasin 1 will remove dirt from it.”

Open question. “I don’t know if the sinkbasin 1 can clean the potato without it being on the sink.”

Synthesis. The environment permits cleaning an item while holding it at the sinkbasin without placing it on the sink first, as demonstrated by successful cleaning actions (e.g., clean potato 1 with sinkbasin 1) when the sinkbasin’s contents do not include the item being cleaned. The sinkbasin’s current empty state does not block the cleaning action, and the potato remains in inventory during cleaning, resulting in a clean potato ready for microwave use.

ALFWorld Case 2

Task. Clean some spoon and put it in drawer.

Observation. You open the drawer 3. The drawer 3 is open. In it, you see nothing.

Confirmed pattern. “I think the inventory action reveals items the agent is currently holding.”

Open question. “I’m unsure whether the examine action reveals hidden items in a drawer.”

Synthesis. The examine action does not reveal hidden items in a drawer, as it only confirms the visible state (e.g., “drawer 4 is open. In it, you see nothing”), which is already known after opening. ... The inventory action only displays items the agent is currently holding, not environmental objects like spoons on countertops or in drawers, so it cannot indicate the spoon’s location without prior pickup.

WebShop Case 1

Task. Find me machine wash men’s t-shirts with long sleeve with color: black, and size: xx-large, and price lower than \$70.00.

Observation. Search.

Confirmed pattern. “I think specifying price constraints in the search query will filter results to items under \$70.”

Open question. “I’m unsure whether including size and color in the search terms ensures they are applied as filters.”

Synthesis. The search engine does not apply price constraints (e.g., price < 70.00) as filters, as results include items exceeding the price limit. Including color and size as attributes (e.g., color:black, size:xx-large) in the query string fails to filter results, as non-matching items still appear. ...

WebShop Case 2

Task. Find me wide leg, slim fit, straight leg, loose fit women’s shorts with color: red, and size: xx-large, and price lower than \$40.00.

Observation. Back to Search | < Prev | color | black | blue | dark gray | gray | navy | red | silver | size | small | medium | large | x-large | xx-large | Burband Womens High Waist Bike Shorts Tummy Control Workout Yoga Pants Running Exercise Running Shorts with Side Pockets | Price: \$6.82 | Rating: N.A. | Description | Features | Reviews | Buy Now

Confirmed pattern. “I think color and size filters apply to the search results, not individual product pages.”

Open question. “I’m unsure whether changing color/size filters after clicking a product affects the current product’s display.”

Synthesis. The product detail page displays color and size filters that directly modify the current product’s available variants (e.g., clicking “red” selects the red color option for this specific product, not the search results), while “Back to Search” returns to the original filtered search results without altering the applied criteria. This confirms that color/size selections on product detail pages are product-specific, not global search filters, and “Back to Search” resets to the search context with the same filters active.

I RL Rollout Reasoning

During reinforcement learning, AAWM-initialized policies (Qwen2.5-7B-Instruct) frequently produce reasoning traces that reference specific environment mechanisms before selecting an action. This appendix presents four rollout excerpts illustrating this pattern, with the complete reasoning trace for each step.

ALFWorld Case 1

Task. Put a hot cup in cabinet.

Observation. You arrive at microwave 1. The microwave 1 is closed.

Reasoning. The goal is to put a hot cup in a cabinet, and we have the cup in our inventory. To achieve this, we need to heat the cup in the microwave. The current observation shows the microwave is closed, but the microwave can be used to heat the cup regardless of its state (closed or open) when the cup is in inventory. Heating the cup in the microwave will make it hot, and we can then move it to a cabinet. Since we have the cup and are at the microwave, the immediate next step is to heat the cup.

Action. heat cup 2 with microwave 1

ALFWorld Case 2

Task. Put a clean spatula in diningtable.

Observation. You arrive at sinkbasin 1. On the sinkbasin 1, you see nothing.

Reasoning. Cleaning the spatula 1 with the sinkbasin 1 is a valid action that directly transforms the spatula into a clean state. The environment validates that this action works as expected (e.g., spatulas are successfully cleaned when executed on a sinkbasin with them in inventory), and the current environment state shows the spatula is in inventory and the sinkbasin is empty (no evidence of prior cleaning attempts or water requiring preconditions), making cleaning the immediate step to achieve the goal.

Action. clean spatula 1 with sinkbasin 1

WebShop Case 1

Task. Find me gluten free, protein serving pantry staples with flavor name: flax seeds, and size: 4.25 ounce (pack of 12), and price lower than \$70.00.

Observation. Back to Search | < Prev | flavor name | country ranch | multi-seeds | original almond | ... | flax seeds | honey cinnamon | size | 4.25 ounce (pack of 1) | ... | 4.25 ounce (pack of 12) | 7.7 ounce (pack of 1) | Blue Diamond Almonds Nut Thins Gluten Free Cracker Crisps, Hint of Sea Salt, 4.25 Oz Boxes (Pack of 12) | Price: \$34.44 | Rating: N.A. | Description | Features | Reviews | Buy Now

Reasoning. The current product listing (Blue Diamond Almonds) already satisfies the size and price criteria (4.25 ounce pack of 12 for \$34.44), but the flavor is not “flax seeds” as specified in the goal. The action to correct this is to change the flavor filter to “flax seeds” by clicking on the “flax seeds” option in the flavor menu. This will update the product listing to show only items with the correct flavor, ensuring the goal’s flavor requirement is met.

Action. click[flax seeds]

WebShop Case 2

Task. Find me loose fit women’s tops, tees & blouses with short sleeve, long sleeve with color: c5-wine, and size: large, and price lower than \$40.00.

Observation. Back to Search | < Prev | size | small | medium | large | x-large | xx-large | color | a1-blue | a2-black | a2-wine | ... | c5-army green | c5-black | c5-blue | c5-gray | c5-green | c5-purple | c5-red | ...

Reasoning. The product “FABIURT Summer Tops for Women” has already been filtered by c5-wine color, so it remains to select the size large. The size option “large” is listed in the available size options and does not conflict with the color choice. Selecting size large will satisfy the goal of finding products in size large while maintaining the color c5-wine.

Action. click[large]

J Use of Scientific Artifacts

Our experiments use publicly available research artifacts. ALFWorld, WebShop, and the Agent-Gym codebase are released under MIT licenses. The Qwen2.5-1.5B-Instruct, Qwen2.5-7B-Instruct, Qwen3-Embedding-0.6B, and Qwen3-30B-A3B-

Instruct-2507 models are released under Apache-2.0 licenses. The verl-agent (GiGPO) codebase is also released under Apache-2.0. The AgentTraj-L dataset does not specify an explicit license on its public dataset page; we therefore use it only for research experiments and do not redistribute the original trajectories or any derivative training targets. Should we release code, prompts, or trained checkpoints, we will do so under terms compatible with the licenses of the underlying artifacts.