THINKING BY DOING: BUILDING EFFICIENT WORLD MODEL REASONING IN LLMS VIA MULTI-TURN INTERACTION

Anonymous authors
Paper under double-blind review

ABSTRACT

A central challenge for LLM agents is modeling how environment states evolve under actions, which calls for robust world modeling to improve foresight and enhance decision-making under uncertainty. Multi-turn RL provides direct feedback for learning these transitional dynamics, yet current approaches impose fixed reasoning structures that limit models to surface-level heuristics, often at the cost of inefficient interaction or over-dependence on external signals. Whether these interactive experiences truly lead to internalized knowledge of state transitions, however, remains unclear. To this end, we explore **w**orld-**m**odel internalization through efficient interaction and **act**ive reasoning (WMAct), which uses two mechanisms to convert interaction into internalized transition knowledge without prescribing a reasoning template: (1) reward rescaling that weights the outcome reward by the proportion of state-changing actions, discouraging ineffective interaction; (2) interaction-frequency annealing that gradually tightens the turn budget, encouraging less reliance on online feedback and more on internal transition prediction. Across Sokoban, Maze, and Taxi, our results reveal that WMAct enables effective single-turn solving of tasks that previously demanded multi-turn interaction. This performance gain, achieved without recourse to multi-turn feedback, is indicative of genuine planning from internalized state transitions rather than reactive pattern matching. Moreover, the learned capability transfers to general benchmarks (e.g., +5.05 on HMMT25, +2.24 on GPQA-Diamond).

1 INTRODUCTION

Recent advancements have been marked by the development of sophisticated reasoning large language models (LLMs), such as OpenAI-o1 (Jaech et al., 2024) and Deepseek-R1 (Guo et al., 2025a), distinguished by a protracted reasoning process before delivering a final answer. Largely unlocked by Reinforcement Learning (RL), this capability has driven substantial advances in demanding domains such as mathematics (Zeng et al., 2025; Hu et al., 2025; Yu et al., 2025c; He et al., 2025; Wei et al., 2024), coding (Pourreza et al., 2025; Wei et al., 2025b; Qwen Team, 2025), logic (Xie et al., 2025; Sprague et al., 2025; Cheng et al., 2025; Wei et al., 2025a), and multimodal (Yu et al., 2025a;b; Zhu et al., 2025).

Building on these capabilities, research increasingly aims to empower LLMs as autonomous agents. Such agentic tasks require models not only to deduce solutions but to formulate sequential plans, generate executable actions, and interact dynamically, challenging their grasp of world knowledge such as environmental properties and state transitions. Much recent effort therefore designs explicitly structured world model reasoning processes, such as task planning (Zhang et al., 2025b), environment perception (Chen et al., 2025; Wang et al., 2025b; Zhang et al., 2025b; Liao et al., 2025), and state change prediction (Wang et al., 2025b), and uses RL to reinforce these patterns. While effective in the short term, such human-designed patterns ultimately become a bottleneck: forcing the model to follow a fixed reasoning template (e.g., always emit a perception step, then a transition-prediction step) regardless of task or state limits its ability to spontaneously learn strategies from interaction (Silver et al., 2018; 2017; Sutton, 2019). To this end, we investigate a key question: **How can we build effective and efficient world model reasoning without constraining the model's cognitive flexibility?** Throughout this paper, we use *world model reasoning* to denote a model's

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.

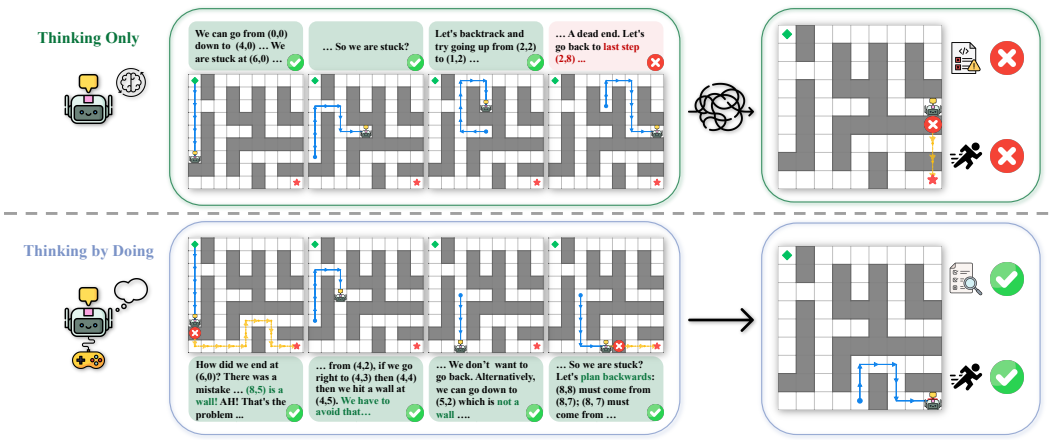


Figure 1: Illustration of why interactive reasoning (Thinking by Doing) can be more reliable than monolithic reasoning (Thinking Only). Top: A thinking-only agent internally predicts a complete trajectory before acting. Without intermediate observations to ground these predictions, an erroneous transition can corrupt the remainder of the plan and go undetected until execution, leading to failure. Bottom: A thinking-by-doing agent alternates reasoning with action. Each resulting observation checks the predicted transition, enabling the agent to detect discrepancies and replan before errors propagate. Thus, Thinking Only relies on delayed, outcome-level verification, whereas Thinking by Doing provides stepwise, transition-level grounding.

ability to leverage environment dynamics for three concrete purposes: (i) tracking the current state, (ii) predicting the next state that results from an action (transition prediction), and (iii) goal-directed planning over such predicted transitions.

Our investigation contrasts two RL training paradigms: *monolithic reasoning*, which requires the model to generate an entire plan for a task in a single turn, and *multi-turn interaction*, in which the model obtains intermediate feedback by interacting with the environment and actively responds to state transitions. We study both paradigms in Sokoban, Maze, and Taxi for a specific reason: these environments are *fully observable*, so a task can in principle be solved in a single turn by reasoning alone, with no hidden state to excuse a failure. Single-turn accuracy is therefore a clean measure—well defined and identical in meaning for both paradigms—of how much of the environment dynamics the model has internalized, which makes the two paradigms directly comparable on the same testbed. We stress that single-turn success alone does *not* prove a fully internalized world model; rather, under full observability it *upper-bounds* what internalized reasoning can achieve, and we treat it as one signal that we further corroborate with direct probes of state tracking, transition prediction, and planning (Section 4).

Through this analysis, we identified the following three challenges: (1) *Substantial cognitive burden without interaction.* Monolithic reasoning suffers from the burden of internally simulating continuous state transitions; without external interaction, this simulation risks reinforcing self-generated, erroneous knowledge, culminating in spurious success—strong performance in the training environment but failure to generalize. Multi-turn interaction instead lets the model leverage environmental feedback to self-evolve and internalize it into world model reasoning. (2) *Inefficient exploration leading to redundant world knowledge.* Models adopt brute-force strategies, enumerating numerous actions rather than analyzing the optimal path, which impedes acquiring higher-quality environmental knowledge from feedback. (3) *Excessive interaction dependency causes inefficient internalization.* LLMs tend to exploit interaction feedback solely for task completion, lacking the incentive to internalize complex environmental dynamics into their own reasoning, which hinders efficient long-horizon world model reasoning.

In this work, we formally present **w**orld-**m**odel internalization through efficient interaction and **act**ive reasoning (WMAct), which helps models learn world model reasoning from multi-turn interaction and addresses the above problems through two mechanisms. To reduce redundant actions and encourage purposeful interaction, we introduce a reward rescaling mechanism that scales the outcome reward by the proportion of effective actions. In addition, an interaction frequency annealing strategy progressively adjusts the allowed interaction turns, encouraging initial exploration while later

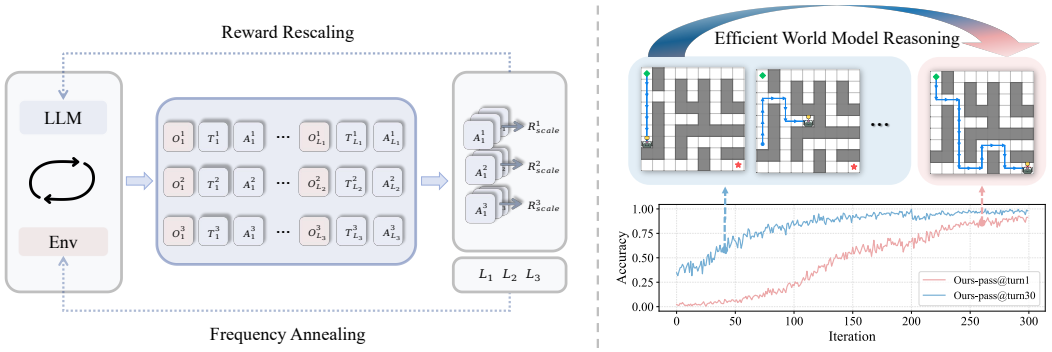


Figure 2: Illustration of the behavioral evolution of WMAct during training on a Maze example. As training progresses, the model's single-turn accuracy continually improves, ultimately matching multi-turn accuracy. Tasks that previously necessitated multi-turn interactive trial-and-error can now be effectively solved within a single turn. This shift from feedback-dependent execution to feedback-free planning provides behavioral evidence that the model has internalized environmental transition dynamics, reducing inference-time interaction without sacrificing task success.

compelling the model to distill and internalize environmental dynamics. Together, these mechanisms yield a more efficient world model reasoning process that resolves in a single turn tasks that previously required multiple interactions (Figure 2), as validated on Sokoban, Maze, and Taxi.

2 Related Work

2.1 World Model Reasoning

Empowering LLMs as autonomous agents requires world knowledge such as environment perception and state transition prediction. A foundational paradigm is ReAct (Yao et al., 2023), which interleaves reasoning and acting so that the model grounds each reasoning step in environmental feedback at inference time. Building on this idea, one line of work injects such knowledge through auxiliary tasks during pre-training or fine-tuning (Su et al., 2025; Liu et al., 2025; Zhang et al., 2025a). Another line enforces predefined structured reasoning during interaction. G1 (Chen et al., 2025) forces models to first output perceptual information for subsequent reasoning. RLVMR (Zhang et al., 2025b) provides dense process-level rewards for explicit meta-reasoning steps. VAGEN (Wang et al., 2025b) explicitly requires StateEstimation and TransitionModeling with specialized world-modeling rewards. However, these rigid patterns prevent models from learning flexible strategies through interaction. In contrast, WMAct focuses on enabling models to develop reasoning capabilities more autonomously.

2.2 Multi-turn Reinforcement Learning

Recent advances in LLMs have enabled autonomous agents for complex tasks (Team et al., 2025; Guo et al., 2025b; Wang et al., 2025c; Hong et al., 2025; Wang et al., 2025a; Qwen Team, 2025), motivating multi-turn RL algorithms. ArCHer (Zhou et al., 2024) improves sample efficiency via hierarchical policies. StarPO-S (Wang et al., 2025d) optimizes at trajectory level with variance-based filtering. SimpleTIR (Xue et al., 2025) addresses instability from distributional drift in tool-integrated reasoning. GRPO-RoC (Shang et al., 2025) resamples correct trajectories to reduce noise from code execution. While these methods focus on RL stability and sample efficiency, our goal is to compel models to internalize environmental dynamics for improved long-horizon capability. This also distinguishes our work from classical RL, which optimizes a task-specific executor tied to a single environment (Dietterich, 2000; Sutton & Barto, 2018). Instead, we study whether a general-purpose LLM can turn interaction into transferable world model reasoning. Several works have explored intrinsic motivation and curiosity-driven exploration (Pathak et al., 2017; Bellemare et al., 2016; Savinov et al., 2019), which encourage agents to seek novel states. In contrast, our reward rescaling does not incentivize state novelty; instead, it penalizes state-invariant actions to promote efficient goal-reaching. Further discussion is deferred to the Appendix A.2.

3 METHOD

3.1 BACKGROUND

RL provides a formal framework for an agent to learn an optimal policy π_θ through environmental interaction, with the objective of maximizing the expected cumulative reward. For language models, a trajectory typically involves generating a response y from a given prompt x. The policy gradient theorem (Sutton et al., 1999; Sutton & Barto, 1998) offers a way to optimize this objective by directly computing the gradient of $J(\theta)$ with respect to the policy parameters θ. The gradient is given by:

$$\nabla_\theta J(\theta) = \mathbb{E}_{x\sim\mathcal{D}, y\sim\pi_\theta}[(R(x,y) - b)\nabla_\theta \log \pi_\theta(y|x)], \tag{1}$$

where $A(x,y) = R(x,y) - b$ is the advantage, typically estimated via Generalized Advantage Estimation (GAE) for stable estimation:

$$\hat{A}_t^{\text{GAE}} = \sum_{l=0}^{\infty} (\gamma\lambda)^l \delta_{t+l}, \tag{2}$$

with $\delta_t = r_t + \gamma V_\phi(s_{t+1}) - V_\phi(s_t)$. Building on GAE, Proximal Policy Optimization (PPO) improves sample efficiency through importance sampling ratio $\rho_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}$ and a clipped surrogate objective:

$$\mathcal{L}^{\text{CLIP}}(\theta) = \mathbb{E}_t\left[\min\left(\rho_t(\theta)\hat{A}_t, \text{clip}(\rho_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t\right)\right]. \tag{3}$$

While the clip scheme enhances stability, the threshold ϵ risks over-restricting exploration and can still leave PPO unstable on certain tasks. Crucially, our goal is to isolate the effect of WMAct from any confound introduced by the training algorithm itself. We therefore adopt an on-policy PPO—data is freshly collected with the latest policy before each optimization step, with no off-policy corrections, experience replay, or other algorithmic accelerations—so that any observed improvement is attributable to our interactive paradigm rather than to optimizer-side techniques, at the cost of some sample efficiency.

3.2 THE POTENTIAL OF LEARNING FROM INTERACTION

The dominant paradigm for complex reasoning samples $y \sim \pi_\theta(\cdot|x)$ as an entire plan and final answer, producing the whole trajectory without external feedback. This compels the model to internally simulate a vast solution space via continuous enumeration, hypothesis generation, and validation. While effective on tasks such as mathematics, code, and logic, it is inefficient, often involving substantial ineffective exploration during the extended reasoning process.

To address this limitation, we shift from generating an entire plan y in one step to producing a sequence of thinking steps T_t and action sets A_t over multiple turns, where $A_t = \{a_t^1, a_t^2, \ldots, a_t^{k_t}\}$ may contain multiple individual actions to lower cognitive burden. Each action set A_t is followed by an environmental observation o_t. The maximum number of interaction turns is constrained by a predefined limit L_{max}. At each turn t, the model generates a thinking step T_t and an action set $A_t \sim \pi_\theta(\cdot|x, H_{<t})$, where $H_{<t} = (o_1, T_1, A_1, o_2, \ldots, o_{t-1}, T_{t-1}, A_{t-1})$ represents interaction history. The final response is the concatenation of all thinking steps, action sets, and observations, forming the complete trajectory $y = (o_1, T_1, A_1, o_2 \ldots, o_L, T_L, A_L)$ and the optimization is to maximize the expected reward $J(\theta)$, implemented using the policy gradient framework defined in Equation (1). This interactive loop offers several key advantages. First, environmental feedback provides an immediate corrective signal, letting the model rectify erroneous assumptions and conserve computation otherwise spent on flawed reasoning paths. Second, learning from a series of feedback-driven interactions rather than a single lengthy, noisy trajectory improves training efficiency. Third, interaction facilitates superior internalization, letting the model more effectively simulate environmental dynamics and construct an internal world model.

3.3 WMACT

We formally present **w**orld-**m**odel internalization through efficient interaction and **act**ive reasoning (WMAct), designed to build robust world model reasoning via multi-turn interaction without

constraining the model's cognitive flexibility, through two mechanisms: (1) a reward scaling mechanism that ensures the model capitalizes on immediate feedback; and (2) an interaction frequency annealing strategy that balances reasoning and interaction, preventing the model from tending toward excessive interaction.

Reward rescaling. Outcome-based rewards are effective in the math and code domains, but in an interactive setting, relying solely on sparse outcome rewards yields excessively large exploration spaces: agents often resort to brute-force strategies, enumerating many actions to complete a task (Figure 3) rather than learning an efficient interactive strategy, hurting sample efficiency and generalization. To encourage more effective strategies, we scale the outcome reward by the proportion of effective actions within an episode, where an action is effective if it leads to a state different from the previous one. Formally, if an episode contains N actions and N_{eff} of them are effective, the scaled reward R_{scaled} is:

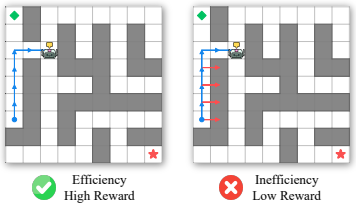


Figure 3: Comparison between different interactive strategies.

$$R_{\text{scaled}} = R_{\text{outcome}} \times \frac{N_{\text{eff}}}{N}. \tag{4}$$

This scaling incentivizes the agent to minimize redundant or ineffective actions and prioritize meaningful progress. We use environment state change as the effectiveness criterion because it is a deliberately minimal, pattern-agnostic signal: rather than hand-crafting the reward around specific failure patterns (e.g., loops)—which couples it to error modes that shift during training—we simply reward actions that move the environment to a new state and penalize state-invariant ones. A robustness analysis over alternative efficacy definitions is provided in the Appendix C.2.

Interaction Frequency Annealing. While multi-turn interaction provides rich feedback, it risks the model over-relying on environmental cues at the expense of internal reasoning, yielding shallow policies that fail to build a robust internal belief. To balance immediate feedback against the long-term need for deliberate world model reasoning, we introduce an interaction frequency annealing strategy. For every τ training iterations, we compute two statistics over recent episodes—the average interaction turns per prompt $\bar{L}$ and the maximum observed L'_{max}—and set the maximum allowed turns for the next phase as:

$$L_{\text{max}} = \frac{\bar{L} + L'_{\text{max}}}{2}. \tag{5}$$

This adaptive limit implements a curriculum: it permits more turns early in training and imposes a tighter cap later. Its intended effect is to retain task performance while reducing the opportunities for online feedback; we evaluate this effect using single-turn test-time execution.

4 Experiments

4.1 Settings

We validate WMAct on three distinct environments that stress complementary skills: Taxi (instruction following and sequential pickup/dropoff dependencies), Maze (spatial navigation and pathfinding), and Sokoban (irreversible symbolic planning and long-horizon foresight). Full environment details are provided in the Appendix B.1.

Training settings. We use an on-policy PPO that omits the KL divergence loss and entropy bonus. Experiments primarily use Qwen3-8B-SFT—supervised fine-tuned on our proprietary general and reasoning datasets—with Qwen2.5-7B-Instruct (Yang et al., 2024) included to isolate the effect of base model capabilities. We use a global batch size of 256, sample 8 responses per prompt at a decoding temperature of 1.0, and set the maximum completion length to 16k tokens with an initial limit of 30 interaction turns.

Evaluation settings. We use two evaluation protocols: *Standard* (in-domain tasks at the training difficulty level) and *Hard* (increased difficulty to test robustness and generalization). For Sokoban, Hard-1 increases the map size by half at a fixed box count and Hard-2 increases the box count at a

Table 1: Evaluation results on 3 tasks. **Standard** and **Hard** share the same environment type: **Standard** uses in-distribution environmental parameters identical to the training, while **Hard** uses more challenging ones. **PPO-EntirePlan** outputs the entire solution plan in a single turn, as is standard in RLVR for LLMs. **PPO-Interactive** allows environment interaction during training. Numbers are averaged over 8 evaluation runs on a fixed 256-problem held-out set (subscripts: sample standard deviation with $n-1$ divisor). Overall is the unweighted mean of the three per-task scores.

Model / Method	Sokoban			Maze		Taxi	Overall
	Standard	Hard-1	Hard-2	Standard	Hard	Standard	
Open-Source Models							
Qwen3-14B (Qwen Team, 2025)	$18.75_{\pm 2.61}$	$06.64_{\pm 1.67}$	$05.08_{\pm 1.46}$	$72.27_{\pm 2.99}$	$28.52_{\pm 3.01}$	$06.64_{\pm 1.67}$	22.40
Qwen2.5-32B-Instruct (Yang et al., 2024)	$00.00_{\pm 0.00}$	$00.00_{\pm 0.00}$	$00.00_{\pm 0.00}$	$00.68_{\pm 0.54}$	$00.39_{\pm 0.42}$	$02.34_{\pm 1.00}$	00.96
Qwen2.5-7B-Instruct (Yang et al., 2024)	$00.00_{\pm 0.00}$	$00.00_{\pm 0.00}$	$00.00_{\pm 0.00}$	$00.00_{\pm 0.00}$	$00.00_{\pm 0.00}$	$01.17_{\pm 0.71}$	00.39
Qwen3-8B (Qwen Team, 2025)	$16.41_{\pm 2.47}$	$04.59_{\pm 1.40}$	$01.17_{\pm 0.71}$	$63.67_{\pm 3.22}$	$17.77_{\pm 2.56}$	$09.38_{\pm 1.95}$	19.16
Backbone: Qwen3-8B							
PPO - EntirePlan	$73.54_{\pm 0.35}$	$03.91_{\pm 0.78}$	$00.39_{\pm 0.21}$	$84.52_{\pm 2.06}$	$40.72_{\pm 1.73}$	$06.15_{\pm 0.58}$	31.57
StarPO-S (Wang et al., 2025d)	$73.19_{\pm 0.99}$	$24.12_{\pm 1.56}$	$24.41_{\pm 1.68}$	$86.72_{\pm 2.63}$	$48.63_{\pm 0.80}$	$11.33_{\pm 1.88}$	39.86
WMAct*	$\mathbf{74.80}_{\pm 0.30}$	$\mathbf{34.96}_{\pm 1.67}$	$\mathbf{32.57}_{\pm 1.30}$	$\mathbf{90.58}_{\pm 0.39}$	$\mathbf{51.66}_{\pm 1.76}$	$\mathbf{54.59}_{\pm 1.45}$	**57.72**
Backbone: Qwen3-8B-SFT							
Qwen3-8B-SFT	$03.27_{\pm 1.19}$	$01.37_{\pm 0.78}$	$00.83_{\pm 0.61}$	$01.95_{\pm 3.77}$	$00.20_{\pm 0.59}$	$05.62_{\pm 0.64}$	02.84
PPO - EntirePlan	$49.12_{\pm 2.50}$	$02.34_{\pm 1.80}$	$00.34_{\pm 0.46}$	$75.05_{\pm 3.05}$	$26.51_{\pm 3.05}$	$38.92_{\pm 2.12}$	35.66
PPO - Interactive	$64.21_{\pm 2.15}$	$46.83_{\pm 2.23}$	$41.26_{\pm 3.15}$	$83.74_{\pm 2.16}$	$36.52_{\pm 3.87}$	$39.16_{\pm 2.56}$	50.02
StarPO-S (Wang et al., 2025d)	$68.85_{\pm 1.69}$	$40.38_{\pm 1.54}$	$36.52_{\pm 1.17}$	$86.62_{\pm 0.90}$	$48.44_{\pm 0.87}$	$53.03_{\pm 1.80}$	56.38
WMAct*	$\mathbf{78.56}_{\pm 2.32}$	$\mathbf{52.69}_{\pm 1.67}$	$\mathbf{49.90}_{\pm 1.96}$	$\mathbf{88.13}_{\pm 2.14}$	$\mathbf{50.59}_{\pm 3.06}$	$\mathbf{62.16}_{\pm 3.26}$	**63.97**
Proprietary Models							
GPT-4o (Hurst et al., 2024)	$01.95_{\pm 0.92}$	$00.00_{\pm 0.00}$	$00.00_{\pm 0.00}$	$04.69_{\pm 1.41}$	$01.56_{\pm 0.81}$	$06.25_{\pm 1.61}$	03.34
GPT-5 (OpenAI, 2025a)	$96.88_{\pm 1.17}$	$91.41_{\pm 1.87}$	$87.89_{\pm 2.18}$	$99.61_{\pm 0.42}$	$97.66_{\pm 1.00}$	$36.72_{\pm 3.22}$	75.81
OpenAI o3 (OpenAI, 2025c)	$98.83_{\pm 0.71}$	$90.23_{\pm 1.99}$	$85.16_{\pm 2.37}$	$95.70_{\pm 1.36}$	$93.36_{\pm 1.67}$	$72.66_{\pm 2.98}$	86.20
OpenAI o4-mini (OpenAI, 2025c)	$83.20_{\pm 2.50}$	$69.14_{\pm 3.09}$	$62.89_{\pm 3.23}$	$93.75_{\pm 1.61}$	$78.52_{\pm 2.75}$	$83.98_{\pm 2.45}$	80.62
Gemini 2.5 Pro (Comanici et al., 2025)	$36.72_{\pm 3.22}$	$17.19_{\pm 2.52}$	$20.31_{\pm 2.68}$	$83.59_{\pm 2.47}$	$63.28_{\pm 3.22}$	$17.97_{\pm 2.57}$	38.72
Claude 4.5 Sonnet (Anthropic, 2025)	$60.94_{\pm 3.26}$	$34.77_{\pm 3.19}$	$29.30_{\pm 3.04}$	$94.14_{\pm 1.57}$	$68.36_{\pm 3.11}$	$66.41_{\pm 3.15}$	63.11

Table 2: Performance across general benchmarks. Results marked with † are directly cited from the original paper. **Bold** indicates the best result and underline the second-best.

Model	AVG@64				AVG@16		AVG@4	AVG@1
	AIME24	AIME25	B-AIME	HMMT25	GPQA-D	LCB-v5	LiveBench	MMLU-P
OpenAI o1† (OpenAI, 2025b)	74.30	79.20	-	-	**78.00**	63.90	**75.70**	**89.30**
OpenAI o3-mini† (OpenAI, 2025c)	79.60	74.80	**63.60**	-	76.80	66.30	70.00	78.70
GPT-4o† (Hurst et al., 2024)	11.10	7.60	-	-	46.00	32.70	52.20	72.55
Qwen3-14B (Thinking)† (Qwen Team, 2025)	79.30	70.40	-	-	64.00	63.50	71.30	-
Qwen3-8B-SFT (Ours-Base)	85.10	77.92	52.88	63.44	59.91	65.32	67.93	72.35
WMAct-Sokoban	**86.56**	**79.48**	55.14	**68.49**	62.15	**67.14**	69.60	73.14
Δ_{Base}	▲1.46	▲1.56	▲2.26	▲5.05	▲2.24	▲1.82	▲1.67	▲0.79

fixed map size; for Maze, Hard-1 increases the map size by half. We average over eight evaluations on a fixed set of 256 held-out problems per environment to mitigate randomness, and evaluate all results in a single-turn manner for fair comparison. Additionally, we conduct a comprehensive evaluation of our paradigm on several representative benchmarks spanning three categories: General (MMLU-Pro (Wang et al., 2024), GPQA-Diamond (Rein et al., 2024), LiveBench (White et al., 2024)), Math (AIME24 (Art of Problem Solving, 2025), AIME25 (Art of Problem Solving, 2025), BeyondAIME (ByteDance-Seed, 2025), HMMT25 (HMMT Staff, 2025)), and Code (LiveCodeBench v5 (Jain et al., 2025)). We also present experimental results for larger-scale models in the Appendix C.3, along with results for ReAct.

4.2 Main Results

Performance on standard tasks. On standard tasks matching the training distribution, WMAct outperforms all baselines across Sokoban, Maze, and Taxi (78.56 / 88.13 / 62.16; Table 1), substantially exceeding both PPO-EntirePlan and multi-turn PPO, and even surpassing much larger proprietary models such as GPT-4o and Claude 4.5 Sonnet—demonstrating the effectiveness of our interactive paradigm coupled with efficacy-driven exploration and frequency annealing.

Table 3: **(a)** Zero-shot probes of world model reasoning on Sokoban. Action validity (one-step legality), Solvability (multi-step reachability), State prediction (full simulation). Δ vs. the stronger PPO baseline. **(b)** Reward rescaling and frequency annealing provide substantial and complementary improvements over the PPO-Interactive baseline across all difficulty levels.

(a) Quantitative probes of world model reasoning.

	Action validity		Solvability		State pred	
Method	Acc.	Δ	Acc.	Δ	Acc.	Δ
PPO-EntirePlan	58.0	–	34.0	↓20.0	16.0	↓2.0
PPO-Interactive	54.0	↓4.0	54.0	–	18.0	–
WMAct⋆	**62.0**	↑4.0	**78.0**	↑24.0	**40.0**	↑22.0

(b) Component ablation on Sokoban.

	Standard		Hard-1		Hard-2	
Method	Acc.	Δ	Acc.	Δ	Acc.	Δ
PPO-EntirePlan	49.12	↓15.09	02.34	↓44.49	00.34	↓40.92
PPO-Interactive	64.21	–	46.83	–	41.26	–
+ reward rescaling	73.68	↑09.47	50.78	↑03.95	48.05	↑06.79
+ freq. annealing	**78.56**	↑14.35	**52.69**	↑05.86	**49.90**	↑08.64

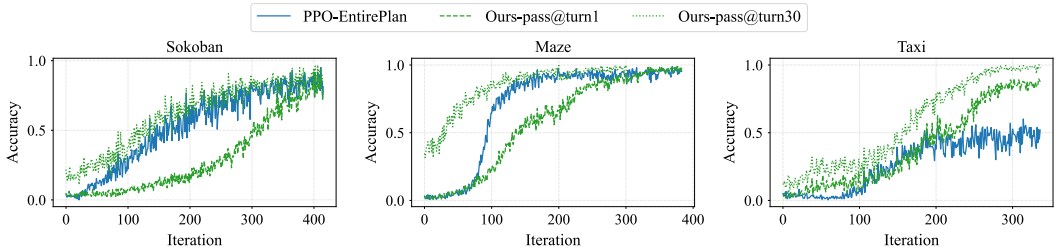


Figure 4: Training accuracy progress comparing WMAct with PPO-EntirePlan. WMAct's single-turn performance is not only comparable to PPO-EntirePlan but also gradually approaches its multi-turn counterpart across all environments. In the Taxi environment, WMAct clearly breaks through the performance ceiling of PPO-EntirePlan.

Generalization on challenging tasks. On harder variants with increased complexity, WMAct maintains strong performance (52.69 / 49.90 on Sokoban Hard-1/Hard-2, 50.59 on Maze Hard; Table 1), whereas PPO-EntirePlan and PPO-Interactive drop considerably. This indicates WMAct internalizes environmental dynamics well enough to generalize to unseen, more complex scenarios without overfitting the training distribution, condensing learned knowledge into a robust internal world model.

Performance on General Benchmarks. To exclude the possibility that the model overfits to the environment and to assess its transferability, we evaluate the WMAct model trained on Sokoban across general benchmarks. It improves comprehensively over Qwen3-8B-SFT (Table 2), with the largest gains on complex reasoning (+5.05 on HMMT25, +2.24 on GPQA-Diamond) and steady gains on general tasks (+1.67 on LiveBench). This indicates the thinking-by-doing paradigm enhances reasoning and planning capabilities that generalize far beyond the original interactive environment.

Direct probes of world model reasoning. Because task success reflects internalization only indirectly, we probe the three abilities central to our definition of world model reasoning more directly. On gold question–answer pairs generated from the Sokoban environment, we conduct zero-shot evaluation on the same models along three axes: *action validity*, whether a proposed action is legal in a given state; *solvability*, whether a state can still reach a solution; and *state prediction*, whether the model can generate the state resulting from an action sequence. Across all three in Table 1, WMAct consistently outperforms both PPO-EntirePlan and PPO-Interactive (Table 3a), with the largest margins on solvability (78.0 vs. 54.0) and state prediction (40.0 vs. 18.0). Since these probes demand reasoning about future consequences rather than merely selecting a locally valid action, they offer direct evidence that WMAct strengthens world model reasoning beyond what task success alone can reveal. A natural concern here is that our parallel improvements on general benchmarks instead stem from a generic reinforcement-learning-with-verifiable-rewards (RLVR) alignment effect. Yet two observations tie the gains back to world model reasoning: they concentrate precisely on the probes that require modeling environment dynamics—solvability and state prediction (Table 3a)—which alignment on unrelated data would not selectively improve, and reasoning traces on out-of-domain questions reveal a qualitative shift toward the constraint-driven, hypothesis-consistent behavior rewarded during interaction. We provide the full trace analysis and a discussion of matched PPO baselines in the Appendix C.6.

Efficient world model reasoning. Figure 4 presents training process for Sokoban, Maze, and Taxi under PPO-EntirePlan and WMAct's single-turn and multi-turn accuracy. Critically, WMAct's

Table 4: Comparison of different strategies. Δ denotes the percentage-point difference relative to the Qwen3-8B-SFT baseline. We compare our interaction-frequency annealing with reward-variance filtering from StarPO-S and fixed step-penalty regularization, and ablate the annealing interval τ and alternative definitions of action efficacy. Bold indicates the best result in each column.

Method	Setting	Standard		Hard-1		Hard-2	
		Avg.	Δ	Avg.	Δ	Avg.	Δ
Qwen3-8B-SFT	–	3.27	–	1.37	–	0.83	–
StarPO-S (Wang et al., 2025d)							
reward-variance filtering	p=50	68.85	↑65.58	40.38	↑39.01	36.52	↑35.69
WMAct							
step penalty	η=0.1	72.43	↑69.16	49.32	↑47.95	45.46	↑44.63
frequency annealing	τ=50	77.73	↑74.46	**52.93**	↑51.56	47.56	↑46.73
	τ=100	**78.56**	↑75.29	52.69	↑51.32	**49.90**	↑49.07
	τ=150	74.71	↑71.44	49.95	↑48.58	44.24	↑43.41
action-efficacy	recent-window	64.89	↑61.62	33.98	↑32.61	25.73	↑24.90
	count-based soft	57.32	↑54.05	26.76	↑25.39	27.10	↑26.27
	binary state change	64.84	↑61.57	31.69	↑30.32	33.50	↑32.67

single-turn performance is not only comparable to PPO-EntirePlan but gradually approaches its own multi-turn performance across all environments—strong evidence that knowledge acquired through interaction is effectively compressed and internalized, enabling robust performance even without environmental feedback. Moreover, in Taxi both accuracies clearly break through the PPO-EntirePlan ceiling, showing our interactive paradigm yields a more competent and generalizable final policy, not merely more efficient training-time exploration.

4.3 Ablation Study

The effects of each component. We ablate the contribution of each component (Table 3b). Interactive training (PPO-Interactive) substantially improves over the non-interactive baseline, especially on harder variants, confirming the fundamental benefit of multi-turn interaction. Adding effective-action reward scaling boosts results further, most notably on Hard-2 (41.26→48.05), showing the reward incentivizes efficient exploration by penalizing redundant actions. Adding interaction frequency annealing yields the best overall performance (78.56 / 52.69 / 49.90 across the three difficulty levels), confirming it effectively balances immediate feedback against gradually internalizing environmental dynamics.

Comparison between step penalty, StarPO-S, and frequency annealing. To mitigate inefficient trajectories, we compare frequency annealing against two representative regularizers motivated by distinct intuitions. The *step penalty* is a classic RL technique that compresses trajectories at the step level via a constant negative reward per step, serving as a natural baseline for trajectory efficiency. *StarPO-S* (Wang et al., 2025d) performs variance-based prompt filtering, retaining the top-$p\%$ of prompt groups by reward variance; like our method it embodies a curriculum philosophy, but it uses reward variance as a proxy for learning potential whereas we operate directly on the interaction turn budget. As shown in Table 4, the step penalty encourages myopic strategies that harm generalization and StarPO-S is suboptimal, while frequency annealing achieves the highest Standard success rate (78.56) with strong generalization on harder levels.

Different efficacy definition for action-efficacy-based reward rescaling. We define an action as effective if it changes the state, using this minimal shared signal across Maze, Sokoban, and Taxi to avoid environment-specific heuristics. To assess robustness, we introduce two variants: (1) recent-window efficacy, which considers an action effective only if the next state has not appeared in the last k states, targeting short cycles and ping-pong; and (2) count-based soft efficacy, which replaces the binary signal with a decay weight based on the next state's visitation count, providing a smoother signal that does not treat all revisits as equally ineffective. Table 4 shows directly penalizing redundancy or cyclic behavior yields stronger gains, while the softer count-based variant gives a weaker signal and underperforms, especially on harder settings. The recent-window variant remains competitive with WMAct on Standard and Hard-1, confirming that the benefit stems from discouraging ineffective interaction rather than from a specific heuristic.

The effects of τ. We investigate the impact of the interaction frequency annealing interval τ. As shown in Table 4, a medium interval $\tau = 100$ is optimal across all Sokoban variants (78.56, 52.69, 49.90 on Standard, Hard-1, Hard-2). A shorter $\tau = 50$ degrades most on Hard-2, whose additional boxes demand more complex planning, suggesting overly frequent annealing disrupts multi-step reasoning; a longer $\tau = 150$ hurts all tasks, as the model fails to reduce its interaction dependency. The schedule must therefore balance sufficient interaction turns for skill acquisition against progressive internalization, and $\tau = 100$ strikes this balance.

The effects of model prior. Internalization is not automatic: it emerges only when the base model already possesses advanced reasoning behaviors. Repeating our training on Qwen2.5-7B-Instruct, we find it fails to internalize—its single-turn accuracy does not track its multi-turn accuracy, so the gap never closes (in contrast to the stronger model in Figure 4). We attribute this to the model's lack of reasoning patterns such as reflection, self-correction, and strategic foresight, indicating that behavioral priors are as important as the interactive framework itself. Full training curves and analysis are provided in the Appendix C.4.

5 Discussion and Conclusion

Our study illustrates *how RL fundamentally shapes the development of world model reasoning, moving models beyond rigid, human-designed cognitive patterns.* We identify multi-turn interaction paradigm as a potent mechanism, while monolithic reasoning imposes a substantial cognitive burden. The thinking-by-doing paradigm bypasses this burden by allowing agents to validate solution paths through direct environmental feedback, which effectively prunes incorrect strategies and reinforces environmental dynamics into reasoning. However, we identify several critical constraints on this process. First, cognitive priors are consequential, as advanced reasoning mechanisms such as reflection and self-correction are essential for attributing environmental outcomes to preceding actions. Without these mechanisms, interactions remain isolated episodes rather than yielding evidence for refining a coherent world model. Second, interaction efficiency is critical, since models exhibit a tendency toward brute-force enumeration that impedes knowledge acquisition. Our reward rescaling counteracts this tendency by incentivizing purposeful interaction. Third, feedback dependence is influential, in that the richness of feedback can foster a myopic over-reliance on interaction. Our interaction frequency annealing mitigates this over-reliance to promote efficient long-horizon planning.

Limitations and Failure Modes. *(1) Environment scope.* We deliberately study fully observable, deterministic grid worlds as clean diagnostic testbeds rather than as a claim about all environments: full observability makes single-turn accuracy a clean internalization measure, and because these worlds stress distinct skills (topological navigation, irreversible manipulation, and state tracking with subgoal planning), consistent gains from one shared mechanism cannot be dismissed as an environment-specific artifact, while their explicit action-conditioned transitions let us directly inspect the learned dynamics. *(2) Effective action definition.* Our effective action definition is minimal and pattern-agnostic based on rule-based judgment to avoid the influence of human heuristics. For scenarios in which rule-based assessment is intractable, a model-based judge could be adopted; nevertheless, we defer this direction to future work.

Generative AI Disclosure

In this work, we requested generative AI tools to assist us in refining the presentation of our research methodology and experimental results, improving the clarity and readability of the manuscript. These tools were not used to generate or preprocess datasets, formulate or prove any mathematical assertions, implement the proposed methods, create or modify experimental figures, or perform qualitative data analysis. All suggestions provided by the AI were carefully reviewed, revised as needed, and validated by the authors. Specifically, we manually ensured that all methodological descriptions accurately reflect the actual experimental procedures, verified that every numerical result and comparative statement in the manuscript is consistent with the reported experimental results. We assume full responsibility for the final version of this work, including all text, data, claims, and other materials produced with the assistance of generative AI.

REFERENCES

Anthropic. Introducing claude sonnet 4.5. Blog post, 2025. URL https://www.anthropic.com/claude-sonnet-4-5-system-card. Accessed: September 2025.

Art of Problem Solving. AIME Problems and Solutions. Official Website, 2025. URL https://artofproblemsolving.com/wiki/index.php/AIMEProblemsandSolutions. Accessed: September 2025.

Marc Bellemare, Sriram Srinivasan, Georg Ostrovski, Tom Schaul, David Saxton, and Remi Munos. Unifying count-based exploration and intrinsic motivation. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. URL https://proceedings.neurips.cc/paper_files/paper/2016/file/afda332245e2af431fb7b672a68b659d-Paper.pdf.

ByteDance-Seed. BeyondAIME: Advancing Math Reasoning Evaluation Beyond High School Olympiads. Hugging Face repository, 2025. URL https://huggingface.co/datasets/ByteDance-Seed/BeyondAIME. Accessed: November 2025.

Liang Chen, Hongcheng Gao, Tianyu Liu, Zhiqi Huang, Flood Sung, Xinyu Zhou, Yuxin Wu, and Baobao Chang. G1: Bootstrapping perception and reasoning abilities of vision-language model via reinforcement learning. *arXiv preprint arXiv:2505.13426*, 2025.

Zhoujun Cheng, Shibo Hao, Tianyang Liu, Fan Zhou, Yutao Xie, Feng Yao, Yuexin Bian, Nilabjo Dey, Yonghao Zhuang, Yuheng Zha, Yi Gu, Kun Zhou, Yuqi Wang, Yuan Li, Richard Fan, Jianshu She, Chengqian Gao, Abulhair Saparov, Taylor W. Killian, Haonan Li, Mikhail Yurochkin, Eric P. Xing, Zhengzhong Liu, and Zhiting Hu. Revisiting reinforcement learning for LLM reasoning from a cross-domain perspective. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2025. URL https://openreview.net/forum?id=xUBgfvyip3.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

Thomas G. Dietterich. Hierarchical reinforcement learning with the MAXQ value function decomposition. *Journal of Artificial Intelligence Research*, 13:227–303, 2000.

D. Guo, D. Yang, H. Zhang, J. Song, P. Wang, Q. Zhu, R. Xu, R. Zhang, S. Ma, X. Bi, X. Zhang, X. Yu, Y. Wu, Z. F. Wu, Z. Gou, Z. Shao, Z. Li, Z. Gao, A. Liu, B. Xue, and Z. Zhang. DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning. *Nature*, 645(8081):633–638, 2025a. doi: 10.1038/s41586-025-09422-z.

Dong Guo, Faming Wu, Feida Zhu, Fuxing Leng, Guang Shi, Haobin Chen, Haoqi Fan, Jian Wang, Jianyu Jiang, Jiawei Wang, et al. Seed1. 5-vl technical report. *arXiv preprint arXiv:2505.07062*, 2025b.

Jujie He, Jiacai Liu, Chris Yuhao Liu, Rui Yan, Chaojie Wang, Peng Cheng, Xiaoyu Zhang, Fuxiang Zhang, Jiacheng Xu, Wei Shen, et al. Skywork open reasoner 1 technical report. *arXiv preprint arXiv:2505.22312*, 2025.

HMMT Staff. HMMT 2025 Problems and Solutions. Official Website, 2025. URL https://www.hmmt.org/. Accessed: November 2025.

Wenyi Hong, Wenmeng Yu, Xiaotao Gu, Guo Wang, Guobing Gan, Haomiao Tang, Jiale Cheng, Ji Qi, Junhui Ji, Lihang Pan, et al. Glm-4.1 v-thinking: Towards versatile multimodal reasoning with scalable reinforcement learning. *arXiv preprint arXiv:2507.01006*, 2025.

Jingcheng Hu, Yinmin Zhang, Qi Han, Daxin Jiang, Xiangyu Zhang, and Heung-Yeung Shum. Open-reasoner-zero: An open source approach to scaling up reinforcement learning on the base model. *arXiv preprint arXiv:2503.24290*, 2025.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024.

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. Livecodebench: Holistic and contamination free evaluation of large language models for code. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=chfJJYC3iL.

Yi Liao, Yu Gu, Yuan Sui, Zining Zhu, Yifan Lu, Guohua Tang, Zhongqian Sun, and Wei Yang. Think in games: Learning to reason in games via reinforcement learning with large language models. *arXiv preprint arXiv:2508.21365*, 2025.

Zhaoyang Liu, Jingjing Xie, Zichen Ding, Zehao Li, Bowen Yang, Zhenyu Wu, Xuehui Wang, Qiushi Sun, Shi Liu, Weiyun Wang, Shenglong Ye, Qingyun Li, Xuan Dong, Yue Yu, Chenyu Lu, YunXiang Mo, Yao Yan, Zeyue Tian, Xiao Zhang, Yuan Huang, Yiqian Liu, Weijie Su, Gen Luo, Xiangyu Yue, Biqing Qi, Kai Chen, Bowen Zhou, Yu Qiao, Qifeng Chen, and Wenhai Wang. Scalecua: Scaling open-source computer use agents with cross-platform data. *arXiv preprint arXiv:2509.15221*, 2025. URL https://github.com/OpenGVLab/ScaleCUA.

Shakir Mohamed and Danilo Jimenez Rezende. Variational information maximisation for intrinsically motivated reinforcement learning. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. URL https://proceedings.neurips.cc/paper_files/paper/2015/file/e00406144c1e7e35240afed70f34166a-Paper.pdf.

OpenAI. Gpt-5 system card. Blog post, 2025a. URL https://cdn.openai.com/gpt-5-system-card.pdf. Accessed: August 2025.

OpenAI. Openai o1 system card. Blog post, 2025b. URL https://cdn.openai.com/o1-system-card-20241205.pdf. Accessed: December 2025.

OpenAI. Openai o3 and o4-mini system card. Blog post, 2025c. URL https://cdn.openai.com/pdf/2221c875-02dc-4789-800b-e7758f3722c1/o3-and-o4-mini-system-card.pdf. Accessed: April 2025.

Deepak Pathak, Pulkit Agrawal, Alexei A. Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 2778–2787. PMLR, 06–11 Aug 2017. URL https://proceedings.mlr.press/v70/pathak17a.html.

Mohammadreza Pourreza, Shayan Talaei, Ruoxi Sun, Xingchen Wan, Hailong Li, Azalia Mirhoseini, Amin Saberi, and Sercan O Arik. Reasoning-SQL: Reinforcement learning with SQL tailored partial rewards for reasoning-enhanced text-to-SQL. In *The Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=HbwkIDWQgN.

Qwen Team. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. GPQA: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=Ti67584b98.

Nikolay Savinov, Anton Raichuk, Damien Vincent, Raphael Marinier, Marc Pollefeys, Timothy Lillicrap, and Sylvain Gelly. Episodic curiosity through reachability. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=SkeK3s0qKQ.

Ning Shang, Yifei Liu, Yi Zhu, Li Lyna Zhang, Weijiang Xu, Xinyu Guan, Buze Zhang, Bingcheng Dong, Xudong Zhou, Bowen Zhang, et al. rstar2-agent: Agentic reasoning technical report. *arXiv preprint arXiv:2508.20722*, 2025.

David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas baker, Matthew Lai, Adrian Bolton, Yutian Chen, Timothy P. Lillicrap, Fan Hui, L. Sifre, George van den Driessche, Thore Graepel, and Demis Hassabis. Mastering the game of go without human knowledge. *Nature*, 550:354–359, 2017. URL https://api.semanticscholar.org/CorpusID:205261034.

David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, et al. A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play. *Science*, 362(6419): 1140–1144, 2018. doi: 10.1126/science.aar6404.

Zayne Rea Sprague, Fangcong Yin, Juan Diego Rodriguez, Dongwei Jiang, Manya Wadhwa, Prasann Singhal, Xinyu Zhao, Xi Ye, Kyle Mahowald, and Greg Durrett. To cot or not to cot? chain-of-thought helps mainly on math and symbolic reasoning. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=w6nlcS8Kkn.

Kaya Stechly, Karthik Valmeekam, Atharva Gundawar, Vardhan Palod, and Subbarao Kambhampati. Beyond semantics: The unreasonable effectiveness of reasonless intermediate tokens. In *First Workshop on Foundations of Reasoning in Language Models*, 2025.

Liangcai Su, Zhen Zhang, Guangyu Li, Zhuo Chen, Chenxi Wang, Maojia Song, Xinyu Wang, Kuan Li, Jialong Wu, Xuanzhong Chen, et al. Scaling agents via continual pre-training. *arXiv preprint arXiv:2509.13310*, 2025.

Richard Sutton. The bitter lesson. Blog post, 2019. URL http://www.incompleteideas.net/IncIdeas/BitterLesson.html. Accessed: August 2024.

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 2nd edition, 2018.

Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In S. Solla, T. Leen, and K. Müller (eds.), *Advances in Neural Information Processing Systems*, volume 12. MIT Press, 1999. URL https://proceedings.neurips.cc/paper_files/paper/1999/file/464d828b85b0bed98e80ade0a5c43b0f-Paper.pdf.

R.S. Sutton and A.G. Barto. Reinforcement learning: An introduction. *IEEE Transactions on Neural Networks*, 9(5):1054–1054, 1998. doi: 10.1109/TNN.1998.712192.

Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, et al. Kimi k1. 5: Scaling reinforcement learning with llms. *arXiv preprint arXiv:2501.12599*, 2025.

Bin Wang, Bojun Wang, Changyi Wan, Guanzhe Huang, Hanpeng Hu, Haonan Jia, Hao Nie, Mingliang Li, Nuo Chen, Siyu Chen, et al. Step-3 is large yet affordable: Model-system co-design for cost-effective decoding. *arXiv preprint arXiv:2507.19427*, 2025a.

Kangrui Wang, Pingyue Zhang, Zihan Wang, Yaning Gao, Linjie Li, Qineng Wang, Hanyang Chen, Yiping Lu, Zhengyuan Yang, Lijuan Wang, Ranjay Krishna, Jiajun Wu, Li Fei-Fei, Yejin Choi, and Manling Li. VAGEN: Reinforcing world model reasoning for multi-turn VLM agents. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025b. URL https://openreview.net/forum?id=xpjWEgf8zi.

Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, et al. Internvl3.5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*, 2025c.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. MMLU-pro: A more robust and challenging multi-task language understanding benchmark. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024. URL https://openreview.net/forum?id=y10DM6R2r3.

Zihan Wang, Kangrui Wang, Qineng Wang, Pingyue Zhang, Linjie Li, Zhengyuan Yang, Xing Jin, Kefan Yu, Minh Nhat Nguyen, Licheng Liu, et al. Ragen: Understanding self-evolution in llm agents via multi-turn reinforcement learning. *arXiv preprint arXiv:2504.20073*, 2025d.

Haoran Wei, Youyang Yin, Yumeng Li, Jia Wang, Liang Zhao, Jianjian Sun, Zheng Ge, Xiangyu Zhang, and Daxin Jiang. Slow perception: Let's perceive geometric figures step-by-step. *arXiv preprint arXiv:2412.20631*, 2024.

Yana Wei, Liang Zhao, Jianjian Sun, Kangheng Lin, jisheng yin, Jingcheng Hu, Yinmin Zhang, En Yu, Haoran Lv, Zejia Weng, Jia Wang, Qi Han, Zheng Ge, Xiangyu Zhang, Daxin Jiang, and Vishal M. Patel. Open vision reasoner: Transferring linguistic cognitive behavior for visual reasoning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025a. URL https://openreview.net/forum?id=oEgybA04dY.

Yuxiang Wei, Olivier Duchenne, Jade Copet, Quentin Carbonneaux, Lingming Zhang, Daniel Fried, Gabriel Synnaeve, Rishabh Singh, and Sida Wang. SWE-RL: Advancing LLM reasoning via reinforcement learning on open software evolution. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025b. URL https://openreview.net/forum?id=ULblO61XZ0.

Colin White, Samuel Dooley, Manley Roberts, Arka Pal, Ben Feuer, Siddhartha Jain, Ravid Shwartz-Ziv, Neel Jain, Khalid Saifullah, Siddartha Naidu, et al. Livebench: A challenging, contamination-free llm benchmark. *arXiv preprint arXiv:2406.19314*, 4, 2024.

Jialong Wu, Shaofeng Ma, Chengyue Wang, Beiya Mu, Zhehao Zhang, Mingsheng Long, and Yang Wang. Rlvr-world: Training world models with reinforcement learning. In *Advances in Neural Information Processing Systems*, 2025.

Tian Xie, Zitian Gao, Qingnan Ren, Haoming Luo, Yuqian Hong, Bryan Dai, Joey Zhou, Kai Qiu, Zhirong Wu, and Chong Luo. Logic-rl: Unleashing llm reasoning with rule-based reinforcement learning. *arXiv preprint arXiv:2502.14768*, 2025.

Zhenghai Xue, Longtao Zheng, Qian Liu, Yingru Li, Xiaosen Zheng, Zejun Ma, and Bo An. Simpletir: End-to-end reinforcement learning for multi-turn tool-integrated reasoning. *arXiv preprint arXiv:2509.02479*, 2025.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

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

En Yu, Kangheng Lin, Liang Zhao, jisheng yin, Yana Wei, Yuang Peng, Haoran Wei, Jianjian Sun, Chunrui Han, Zheng Ge, Xiangyu Zhang, Daxin Jiang, Jingyu Wang, and Wenbing Tao. Perception-r1: Pioneering perception policy with reinforcement learning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025a. URL https://openreview.net/forum?id=BeXcXrXetA.

En Yu, Kangheng Lin, Liang Zhao, Yana Wei, Zining Zhu, Haoran Wei, Jianjian Sun, Zheng Ge, Xiangyu Zhang, Jingyu Wang, and Wenbing Tao. Unhackable temporal reward for scalable video MLLMs. In *The Thirteenth International Conference on Learning Representations*, 2025b. URL https://openreview.net/forum?id=Gf1uBeuUJW.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, YuYue, Weinan Dai, Tiantian Fan, Gaohong Liu, Juncai Liu, LingJun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Ru Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Yuxuan Song, Xiangpeng Wei, Hao Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Yonghui Wu, and Mingxuan Wang. DAPO: An open-source LLM reinforcement learning system at scale. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025c. URL https://openreview.net/forum?id=2a36EMSSTp.

Weihao Zeng, Yuzhen Huang, Qian Liu, Wei Liu, Keqing He, Zejun Ma, and Junxian He. Simplerl-zoo: Investigating and taming zero reinforcement learning for open base models in the wild. In *The Second Conference on Language Modeling*, 2025.

Kai Zhang, Xiangchao Chen, Bo Liu, Tianci Xue, Zeyi Liao, Zhihan Liu, Xiyao Wang, Yuting Ning, Zhaorun Chen, Xiaohan Fu, et al. Agent learning via early experience. *arXiv preprint arXiv:2510.08558*, 2025a.

Zijing Zhang, Ziyang Chen, Mingxiao Li, Zhaopeng Tu, and Xiaolong Li. Rlvmr: Reinforcement learning with verifiable meta-reasoning rewards for robust long-horizon agents. *arXiv preprint arXiv:2507.22844*, 2025b.

Yifei Zhou, Andrea Zanette, Jiayi Pan, Sergey Levine, and Aviral Kumar. ArCHer: Training language model agents via hierarchical multi-turn RL. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=b6rA0kAHT1.

Zining Zhu, Liang Zhao, Kangheng Lin, Jinze Yang, En Yu, Chenglong Liu, Haoran Wei, Jianjian Sun, Zheng Ge, and Xiangyu Zhang. Perpo: Perceptual preference optimization via discriminative rewarding. *arXiv preprint arXiv:2502.04371*, 2025.

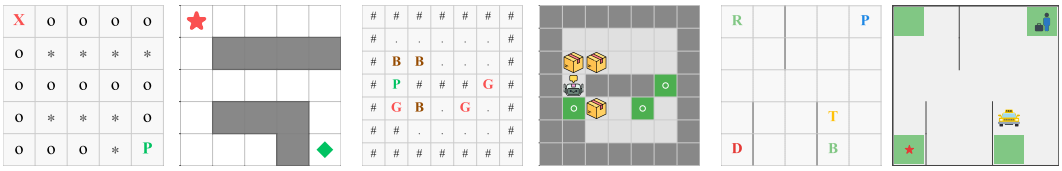
A Extended Related Work

A.1 World Model Reasoning

Empowering LLMs as autonomous agents requires world knowledge such as environment perception and state transition prediction. Approaches to building world models in LLMs are diverse. One line of work injects knowledge through auxiliary tasks during pre-training or fine-tuning. For instance, Agentic CPT (Su et al., 2025) builds an entity-anchored knowledge memory from diverse data sources to synthesize multi-style question-answers. ScaleCUA (Liu et al., 2025) introduces a suite of GUI-related tasks to comprehensively lay a foundation for the digital world. The early experience paradigm (Zhang et al., 2025a) constructs next-state prediction samples from agent trajectories. A different line of work attempts to improve reasoning by enforcing a predefined, structured reasoning process during interaction. These methods compel the model to follow a fixed cognitive pattern designed by humans. G1 (Chen et al., 2025) forces the model to first output perceptual information, providing a reliable state foundation for subsequent structured reasoning and achieving the mutual bootstrapping of perception and reasoning abilities. RLVMR (Zhang et al., 2025b) provides dense process-level rewards for explicit structured meta-reasoning steps, effectively enhancing the error recovery capabilities and generalization to unseen tasks. VAGEN (Wang et al., 2025b) explicitly forces agents to generate StateEstimation and TransitionModeling, and reinforces this process using a specialized world-modeling reward, leading to superior comprehension of environments. Although structured reasoning works, these rigid patterns prevent models from learning flexible, efficient strategies through interaction. Our research focuses on enabling models to develop reasoning capabilities more autonomously.

A.2 Multi-turn reinforcement learning

The rapid advancement of LLMs has facilitated the creation of autonomous agents capable of performing complex tasks (Team et al., 2025; Guo et al., 2025b; Wang et al., 2025c; Hong et al., 2025; Wang et al., 2025a; Qwen Team, 2025). These agents typically operate through multi-turn interactions, requiring dynamic reasoning, action execution, and adaptation based on environmental feedback.



(a) Maze Environment (b) Sokoban Environment (c) Taxi Environment

Figure 5: **Environment Visualization.** For each environment, the left panel shows the standardized character map for agent observation, and the right panel presents the visual illustration for human intuition and inspection.

This paradigm has motivated the development of multi-turn reinforcement learning algorithms aimed at training such agents effectively. For example, Zhou et al. (Zhou et al., 2024) propose ArCHer, a hierarchical RL method that improves sample efficiency by separating high-level and low-level policies. Wang et al. (Wang et al., 2025d) introduce StarPO, which optimizes at the trajectory level and mitigates a common failure mode called echo trap through variance-based filtering. Xue et al. (Xue et al., 2025) attribute instability in tool-integrated reasoning to distributional drift from tool feedback, leading to void turns and stabilizing training by excluding affected trajectories. Similarly, Shang et al. (Shang et al., 2025) present GRPO-RoC, a strategy that resamples correct trajectories to reduce noise from code execution environments during policy updates. While these methods focus on improving the stability and sample efficiency of the RL algorithm itself for multi-step task completion, our primary goal is not merely to complete the task but to compel the model to internalize environmental dynamics, ultimately improving long-horizon capability.

Intrinsic Motivation and Exploration. To mitigate the challenge of sparse extrinsic rewards, prior research has introduced various forms of intrinsic motivation. These primarily focus on state novelty or predictability. For instance, curiosity can be formulated as the prediction error of environment dynamics (Pathak et al., 2017) or derived from state visitation pseudo-counts via density models (Bellemare et al., 2016). From an information-theoretic perspective, empowerment measures an agent's control over future states through mutual information (Mohamed & Jimenez Rezende, 2015). More recently, episodic curiosity has been proposed by measuring the reachability between current observations and past experiences (Savinov et al., 2019). While these methods effectively guide agents to discover new states, they often overlook the *efficiency* of the interaction policy, especially in the context of Large Language Models (LLMs). We observe that LLM-based agents frequently achieve goals through "brute-force enumeration"—generating excessive and redundant actions that lead to the same state. Unlike traditional curiosity mechanisms that prioritize state discovery (*what* is reached), our proposed reward rescaling focuses on the quality of the trajectory (*how* it is reached). By penalizing ineffective actions that fail to induce state transitions, we incentivize the agent to internalize concise and efficient interactive strategies.

B Experiment Setting

B.1 Environments

We evaluate our method on three distinct, challenging grid-world environments—Maze, Sokoban, and Taxi—all requiring agents to execute long-horizon sequential planning and meticulous state tracking. These environments are specifically adapted to interface with Language Models (LLMs) by converting the internal state into a standardized text-based ASCII map. The system also employs a textual action space with action parsing and validation, ensuring a predictable protocol for all environments. The complete prompt of the first turn provided to the LLM at each step follows a standardized structure: `{system prompt} {environment description} {environment action prompt}`, where specific examples are shown in Section E. Subsequent turns only contain the description of the environment feedback and the action prompt.

The Maze task (Figure 5a) focuses on topological navigation and pathfinding under structural constraints where complexity is intrinsically linked to the grid size. Our configuration specifically utilizes 11×11 and 15×15 boards to span Standard and Hard difficulty regimes across all stages of experimentation. Maze construction employs the PRIM algorithm, which is favored for its generation

of structurally intricate and highly varied topological layouts, an advantage over the DFS method's tendency to yield repetitive, long-corridor environments. Upon initialization, the agent begins at an available corner cell, and the destination is programmatically designated as the path point maximally distant from the starting location, thus guaranteeing a maximal planning distance for each instance. The environment state is then rendered to the LLM as the ASCII map employing dedicated symbols: P for the player, X for the goal, o for navigable paths, and * for walls. Interaction is mediated by five discrete actions—move up, down, left, right, and no operation—with movement strictly validated against structural constraints. The episode concludes either upon goal achievement or when the operational limit is reached.

The Sokoban environment (Figure 5b) is the primary testbed for object manipulation and deadlocks due to its inherent requirement for non-reversible action constraints, requiring the agent to push all boxes onto designated goal locations. Task complexity is precisely controlled by two orthogonal factors: the grid dimensionality and the number of boxes. Our configuration defines three distinct difficulty regimes: Standard (7×7 grid, 2 boxes), Hard-1 (10×10 grid, 2 boxes, increasing the spatial search space), and Hard-2 (7×7 grid, 3 boxes, increasing object coordination complexity). All generated puzzle instances are guaranteed to be solvable, achieved by utilizing a mechanism that generates the starting state through a controlled backtracking sequence from a solved state. The environment state is presented to the LLM as the ASCII map, leveraging clear symbols for all entities: P for the player, B for the box, G for the goal, * for the box on goal, # for walls, and . for empty spaces. Interaction is mediated by a comprehensive discrete action space that critically separates simple positional movement—move up, down, left, right—from the core object manipulation commands—push up, down, left, right—alongside a necessary no operation command.

The Taxi environment (Figure 5c) is a classic reinforcement learning domain designed to assess an agent's ability to perform sequential decision-making, meticulous state tracking, and multi-step sub-goal planning—pickup, navigation, and dropoff—within a discrete 5×5 grid world. The taxi agent is tasked with retrieving a passenger from one of four designated initial locations and delivering them to one of four specified destinations. The complexity is determined by the state space size and the strictness of the sequential rules, involving a fixed execution order: picking up the passenger first and then dropping them off at the correct destination. The environment state is presented to the LLM as the ASCII map, which visually renders the geometric layout and utilizes a detailed legend for all entities: T for the taxi, P for the passenger, D for the destination, and critical compounded symbols like T(P) for the taxi carrying the passenger and T/P for the taxi and unpicked passenger co-located. The action space comprises six fundamental commands: four directional navigation actions—move up, down, left, and right—and two interaction actions, including pickup and dropoff. These interaction actions are subject to strict validity checks (e.g., dropoff is only valid at the destination while carrying the passenger), with invalid attempts resulting in immediate negative feedback.

Table 5: More results across general benchmarks. **Bold** indicates the best and underline the second.

Model	AVG@64				AVG@16		AVG@4	AVG@1
	AIME24	AIME25	B-AIME	HMMT25	GPQA-D	LCB-v5	LiveBench	MMLU-P
Qwen3-8B-SFT(Ours-Base)	85.10	77.92	52.88	63.44	59.91	65.32	67.93	72.35
WMAct-Maze	85.21	78.07	53.75	65.10	61.68	67.56	68.54	72.64
WMAct-Taxi	85.00	78.80	53.02	65.57	61.24	**67.70**	69.14	72.71
WMAct-Sokoban	**86.56**	**79.48**	**55.14**	**68.49**	**62.15**	67.14	**69.60**	**73.14**

B.2 Training Setting

We conduct reinforcement learning on instances of our Qwen3-8B-SFT and Qwen-2.5-Instruct. We sample 256 trajectories from 32 distinct environment prompts, each replicated 8 times, to increase prompt efficiency. Specifically, we use a vLLM backend configured with a temperature of 1.0, top-p of 1.0, and top-k of 0. Maximum generation length is constrained within 16K tokens, with a maximum of 12K tokens per turn. Each turn, the generated response will contain a think section and several actions, wrapped in `<think></think>` and `<action></action>` tags, respectively. We will extract and sequentially execute all actions found within the `<action></action>` tags. The overall reward R_{overall} is determined by an outcome reward and a format reward, where an invalid action results in a small penalty, ensuring operational integrity. For WMAct, the outcome reward is proportionally scaled based on Eq 4. And then, we employ the on-policy Proximal Policy Optimization (PPO) algorithm on entire trajectories, where generalized Advantage Estimation (GAE)

Table 6: Per-iteration wall-clock time (seconds) for PPO-Interactive vs. WMAct with Qwen3-8B. WMAct's extra cost is limited to lightweight statistics collection and does not materially change per-iteration runtime.

Method	Maze	Sokoban	Taxi
PPO-Interactive	415	522	495
WMAct*	421	528	499

Table 7: ReAct-style multi-turn baseline on Sokoban. ReAct relies on multi-turn interaction at inference time. Numbers are averaged over 8 evaluation runs (subscripts: standard deviation). For reference, WMAct attains 78.56 / 52.69 / 49.90 on Standard / Hard-1 / Hard-2 using *single-turn* inference, i.e., competitive with or exceeding these ReAct scores despite acting without environmental feedback.

Method	Standard	Hard-1	Hard-2
GPT-4o (Hurst et al., 2024)	$04.88_{\pm 0.90}$	$00.68_{\pm 0.58}$	$00.20_{\pm 0.20}$
Claude 4.5 Sonnet (Anthropic, 2025)	$76.37_{\pm 1.01}$	$59.77_{\pm 2.58}$	$46.39_{\pm 1.55}$
Qwen3-8B (Qwen Team, 2025)	$36.23_{\pm 1.28}$	$13.87_{\pm 1.57}$	$13.35_{\pm 1.75}$
Qwen3-14B (Qwen Team, 2025)	$40.63_{\pm 0.84}$	$17.84_{\pm 1.51}$	$16.54_{\pm 1.76}$

with $\gamma = 1$ and $\lambda = 1$ is used to evaluate the advantage. The KL penalty and entropy regularization are disabled. The policy and critic networks are optimized using Adam with learning rates of 1×10^{-6} and 5×10^{-6}, respectively. These learning rates are linearly warmed up over the first 20 and 50 iterations. We perform one policy update for every 12 critic updates, with no initial critic warm-up phase.

Computational overhead. The additional cost of WMAct over standard multi-turn PPO is limited to lightweight per-episode statistics (counting state-changing actions and recent turn counts) and is negligible relative to policy/value optimization. Table 6 reports per-iteration wall-clock time for PPO-Interactive and WMAct with Qwen3-8B: the two are within about 1% across all three environments. We use a strict on-policy setup mainly to control confounds and obtain clean comparisons; this is not intrinsic to WMAct, which can in principle be combined with off-policy training.

B.3 Evaluation Setting

We held out 256 prompts as an evaluation set for each environment to ensure they would not appear in the training set. Our evaluation protocol mirrors the training configuration to ensure consistent assessment where the vLLM-based decoding stack (temperature=1.0, top-p=1.0, top-k=0) and enforces the same think-action response structure. An evaluation episode concludes upon the first of these conditions being met: (1) the environment signals task completion, (2) the generation length reaches a predefined maximum, or (3) the context token budget is nearly exhausted. **Statistical significance.** Every reported number is averaged over 8 independent evaluation runs on the same fixed held-out set of 256 problems (i.e., the "eight evaluations" refer to independent decoding runs, not different problem sets); the AVG@k columns similarly average k independent runs. Table 1 reports mean $\pm$ standard deviation for the main Sokoban comparison. The standard deviations (roughly ± 1.5 to ± 3 points) are small relative to the double-digit gaps between WMAct and the baselines, confirming the reported gains are stable across runs rather than within-run noise.

C More Evaluation Results

C.1 ReAct Baseline

Because ReAct (Yao et al., 2023) interleaves reasoning and acting at inference time, it inherently relies on multi-turn interaction, whereas our central evaluation is single-turn to measure how well the model has internalized environment dynamics. For completeness we still report a ReAct-style baseline on Sokoban across several proprietary and open models in Table 7. Notably, WMAct's *single-turn* scores (78.56 / 52.69 / 49.90) are competitive with or exceed the strongest ReAct-style *multi-turn* results, underscoring that learning from interaction can be compressed into single-pass reasoning rather than requiring repeated environmental feedback.

Table 8: Scalability to a larger backbone on Sokoban. The gains from WMAct not only persist but grow with scale, improving over PPO-Interactive by 37.74 / 34.71 / 31.50 absolute points on Standard / Hard-1 / Hard-2.

Method	Standard	Hard-1	Hard-2
Backbone: Qwen3-14B			
Qwen3-14B	12.94	03.47	02.93
PPO - EntirePlan	06.49	01.03	00.68
PPO - Interactive	32.57	08.06	07.81
WMAct*	**70.31**	**42.77**	**39.31**

C.2 Sensitivity to the Effective-Action Definition

Our default definition marks an action as effective if it changes the state, chosen because it is a minimal signal shared across Maze, Sokoban, and Taxi that avoids environment-specific heuristics. To test whether the benefit hinges on this specific rule, we evaluate two alternatives: a *recent-window* variant that judges effectiveness by whether the resulting state appears among the last-k visited states (targeting short cycles and ping-pong behavior), and a *count-based soft* variant that replaces the binary signal with a weight decaying in the visitation count of the next state. As shown in Table 4, the results show some sensitivity but not a change in the main conclusion: definitions that more directly penalize redundant or cyclic interaction remain strong, while the softer count-based variant gives a weaker signal and lower performance, especially on harder settings. This indicates the gain comes from discouraging ineffective interaction in general, rather than from one particular hand-crafted definition.

C.3 Scalability to Larger Models

To assess whether the benefits of WMAct persist at larger scale, we additionally train Qwen3-14B on Sokoban. As shown in Table 8, the gains not only persist but become larger: relative to PPO-Interactive, WMAct improves by 37.74, 34.71, and 31.50 absolute points on Standard, Hard-1, and Hard-2, respectively. This suggests that the benefits of WMAct do not diminish, and may even grow, with model scale.

C.4 Effect of Model Prior

We present the full training dynamics of Qwen2.5-7B-Instruct in Figure 6. A key observation is that, unlike more advanced models, Qwen2.5-7B-Instruct does not exhibit the trend of multi-turn interaction driving improved single-turn performance, indicating it fails to internalize environmental changes into its own belief state. We reiterate that our criterion for internalization is the *closing of the gap between single-turn and multi-turn accuracy during training*, not the raw multi-turn (or pass@k) score: a model whose multi-turn performance rises while its single-turn performance lags is still relying primarily on online feedback rather than internalizing dynamics. Thus a modest rise in Qwen2.5's multi-turn pass@k is not evidence of internalization, because its single-turn accuracy does not track it, in contrast to the stronger model in Figure 4. We attribute this primarily to the model's lack of advanced reasoning patterns such as reflection, self-correction, and strategic foresight. These patterns are crucial for internalization because they enable the model to abstract general principles from specific feedback, update its internal state meaningfully, and form compressed representations of environmental dynamics. Without these capabilities, the model treats each interaction as an isolated event rather than as evidence for refining a coherent world model. This highlights that architectural and behavioral patterns are as important as the interactive learning framework itself for achieving emergent world modeling capabilities.

C.5 Supplemental General Benchmarks

As a complementary analysis, Table 5 further compares the general capabilities of models trained across three environments of distinct complexity: Maze, Sokoban, and Taxi. The results clearly indicate that while all models trained with the interactive WMAct paradigm exhibit significant improvement over the base model Qwen3-8B-SFT, the inherent difficulty and challenge type of the training environment critically influence the generalization of the learned skills. Specifically,

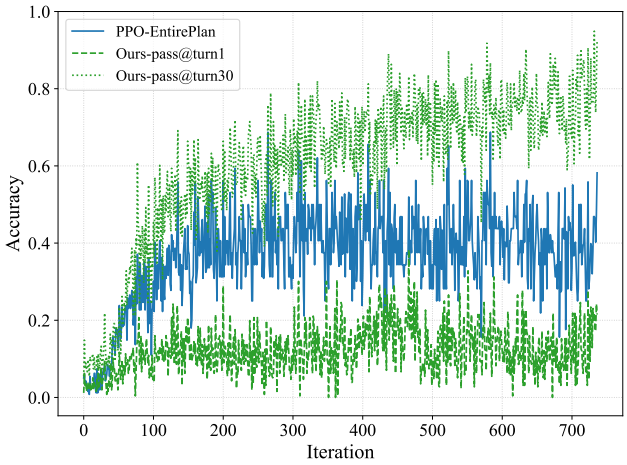


Figure 6: The training dynamic of Qwen2.5-7B-Instruct. Qwen2.5-7B-instruct failed to internalize knowledge from multi-turn interactions to improve its problem-solving efficiency, which underscored the importance of advanced reasoning patterns.

the WMAct-Sokoban maintains its lead across most metrics, demonstrating particular strength in complex reasoning and state tracking benchmarks such as HMMT25 (68.49) and GPQA-Diamond (62.15). This strongly supports our view that the requirements of Sokoban—specifically dealing with non-reversible actions and deadlock avoidance—compel the model to learn a more refined, abstract planning capability with long-term dependency tracking. Conversely, the performance of WMAct-Maze and WMAct-Taxi is generally strong but slightly inferior to the Sokoban-trained agent. As the Maze primarily focuses on topological pathfinding and the Taxi emphasizes multi-step sub-goal decomposition, the reasoning constraints in these two tasks are relatively less stringent. However, we observe a notable exception on the LiveCodeBench v5 metric, where WMAct-Taxi achieves a slight lead of 67.70 over WMAct-Sokoban (67.14). This marginal advantage may be attributable to the Taxi environment's core requirement of sequential sub-goal planning, a structure that closely mirrors the execution flow and step-by-step function calling often required in code generation and problem-solving benchmarks like LiveCodeBench v5.

Isolating the WMAct effect from RLVR alignment. To attribute the general-benchmark gains to WMAct specifically, rather than to reinforcement learning with verifiable rewards (RLVR) in general, the informative comparison is against the *same* RLVR training without our two mechanisms, i.e., PPO-EntirePlan and PPO-Interactive, evaluated on the identical general benchmark suite. If a generic alignment effect explained the gains, these baselines would improve comparably; if WMAct improves world model reasoning, its advantage should be concentrated on dynamics-sensitive reasoning while general scores remain at least non-degraded. Our world-model probes (Table 3a) already show this concentration in-domain. This is also consistent with the observation that intermediate reasoning tokens do not always causally drive task success (Stechly et al., 2025), so a gain concentrated on dynamics-sensitive probes is more diagnostic than aggregate benchmark movement.

Reasoning-trace analysis on out-of-domain questions. Beyond aggregate scores, we inspect reasoning traces on out-of-domain questions (e.g., a GPQA chemistry item). Whereas the base model starts from relevant cues but drifts, relaxes decisive constraints, and becomes self-inconsistent, the WMAct model stays constraint-driven and hypothesis-consistent, keeping the governing constraint central throughout the trace and aligning its final answer with its reasoning. This mirrors the state-tracking and constraint-checking behavior rewarded during interaction, indicating that WMAct refines the model's reasoning process itself rather than only aligning outputs (Wu et al., 2025).

C.6 Behavior Analysis

In this subsection, we investigate whether the model develops increasingly strong world-model-like reasoning behaviors over the course of training. Our analysis is grounded in observable behaviors during puzzle-solving interactions in a maze-based environment, avoiding assumptions about the

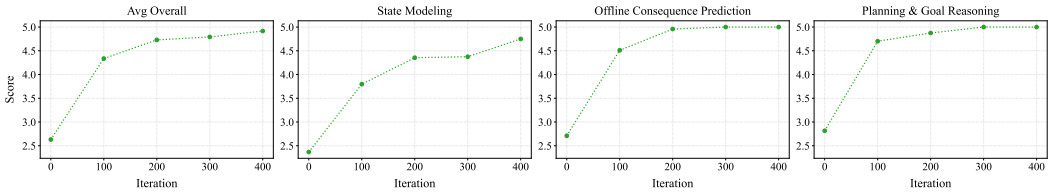


Figure 7: **Behavior Pattern Analysis in the Maze Environment.** Evolution of world-model-related reasoning patterns during training.

model's internal representations and focusing instead on whether its expressed reasoning reflects structured understanding and anticipation of environment dynamics.

To conduct this analysis, we evaluate model-generated interaction trajectories at different training iterations and employ GPT-5.2 as a judge to assess the quality of the model's reasoning behavior. The judge assigns scores based solely on the content explicitly expressed in the model's responses, ensuring a conservative and behavior-driven evaluation protocol.

We assess world-model-like reasoning along three complementary dimensions within a unified evaluation framework. Specifically, the evaluation examines whether the model can accurately identify relevant environment states, anticipate how states evolve under different actions, and maintain consistency in state updates; whether it can reason about the consequences of alternative actions through offline mental simulation rather than direct interaction; and whether it demonstrates goal-directed planning behaviors, such as reasoning about reachability or constraints prior to action selection instead of relying on reactive or trial-and-error strategies. The final overall score is computed by averaging these dimensions, serving as an aggregate indicator of world-model-like behavior.

As illustrated in Figure 7, all evaluation metrics improve consistently as training progresses. This pattern suggests that the model not only becomes more reliable in modeling environment states and predicting short-horizon dynamics but also gradually acquires stronger abilities in offline consequence reasoning and structured, goal-oriented planning. Together, these trends indicate the emergence of increasingly coherent and robust world-model-like reasoning behaviors throughout training.

System Prompt

You are an expert evaluator of sequential decision-making models in puzzle-solving environments. You will be given a single turn from an interaction trajectory consisting of environment states and a response. Your task is to evaluate whether the model's observable behavior exhibits reasoning patterns associated with world-model-like capabilities, based solely on the provided turn. Base your evaluation strictly on what is explicitly stated or strongly implied in the model's responses. Be conservative: do not assume capabilities without clear behavioral evidence. Be strict and selective: high scores must be earned with clear, explicit evidence. If evidence is weak or merely plausible, keep scores in the middle or low range.

User Prompt

You are asked to be a strict judge to evaluate a LLM's reasoning ability in a puzzle-solving environment based on a single interaction turn.
You will be provided with:

- A single turn interaction between a model and the environment;
- A set of evaluation dimensions with detailed scoring guidelines;
- A required output format.

Your task is to carefully analyze the turn and assign scores according to the evaluation dimensions below.
Base your evaluation strictly on the observable behavior in the turn.

Do not assume anything, including the model's internal representations or information from earlier or later turns.
Be strict and discriminative: avoid score inflation.
Full marks (5) must be earned with clear, explicit evidence.

Evaluation Dimensions

1. State Identification, Prediction, and Update
Does the model demonstrate an ability to:

- correctly identify relevant environment states;
- anticipate how the state may change after the action is taken;
- update its understanding of the state.

Score (1–5, integer):

- 1 = Severe errors or missing: clearly misreads or invents state, ignores key inputs, or contradicts itself within the turn (negative example: treats a wall as a path; states the goal is in the wrong location).
- 2 = Partial and unstable: mentions only a few elements, misses key constraints, or fails to update (negative example: only repeats the input without explaining state changes).
- 3 = Mostly correct but shallow: identifies the main state and makes limited prediction, with minor mistakes or omissions (positive example: notes current position and a key obstacle but gives a vague transition).
- 4 = Accurate with updates: most key information is correct, explicitly describes state change, with minor omissions allowed (positive example: explains position change after the action and relevant constraints).
- 5 = Precise and multi-faceted: explicitly tracks multiple state variables and constraints, clearly predicts transitions, and updates consistently without contradiction (positive example: multi-factor state plus an explicit transition chain).

2. Offline Action Consequence Prediction
Does the model reason about the consequences of possible actions without directly executing them in the environment?
This includes counterfactual reasoning, risk anticipation, or explicit discussion of what would happen if a different action were taken.
Score (1–5, integer):

- 1 = No offline reasoning: only executes an action or repeats the input, with no consequences discussed (negative example: outputs an action without any stated outcome).
- 2 = Vague or generic: expresses only a fuzzy preference without specific outcomes (negative example: "should help" or "maybe better").
- 3 = Short-horizon prediction: gives one-step or implicit consequences with limited detail (positive example: says the move reaches a specific cell or avoids a specific obstacle).
- 4 = Clear causal reasoning: for at least one alternative action, states explicit "if X then Y" outcomes and evaluates risk/benefit (positive example: compares two routes and their outcomes).
- 5 = High-bar offline reasoning: clearly shows at least one strong pattern, such as explicit counterfactuals with concrete outcomes, risk/benefit tradeoffs grounded in predicted consequences, or multi-option comparison with consistent selection.

3. Planning and Goal-Directed Reasoning
Does the LLM's behavior reflect planning toward goals, including both short-term decision making and long-term objective satisfaction?
Consider whether the LLM sets subgoals, reasons over multiple steps, or sacrifices immediate actions for future benefit.
Score (1–5, integer):

- 1 = Purely reactive: responds only to the immediate step, with no goal or plan (negative example: “I’ll just try this”).
- 2 = Limited planning or weak goal awareness: mentions a goal and/or proposes a 1–2 step plan, but the plan is shallow, incomplete, or partially inconsistent with the goal.
- 3 = Multi-step planning: states a sequence of steps or subgoals and considers mid-term effects.
- 4 = Long-horizon planning: explicit multi-step plan plus clear goals or subgoals and explicit short-vs-long-term tradeoffs.
- 5 = Near-perfect planning: sustained long-horizon plan with clear goals and subgoals, multiple advanced planning patterns, and consistent alignment between goals and actions.

Output Format
Output your evaluation strictly in the following JSON format:

```
{
  "state_modeling": {
    "score": <int>,
    "justification": "<brief justification>"
  },
  "offline_consequence_prediction": {
    "score": <int>,
    "justification": "<brief justification>"
  },
  "planning_and_goal_reasoning": {
    "score": <int>,
    "justification": "<brief justification>"
  }
}
```

Trajectory
Below is the single interaction turn to be evaluated:
`[turn]`

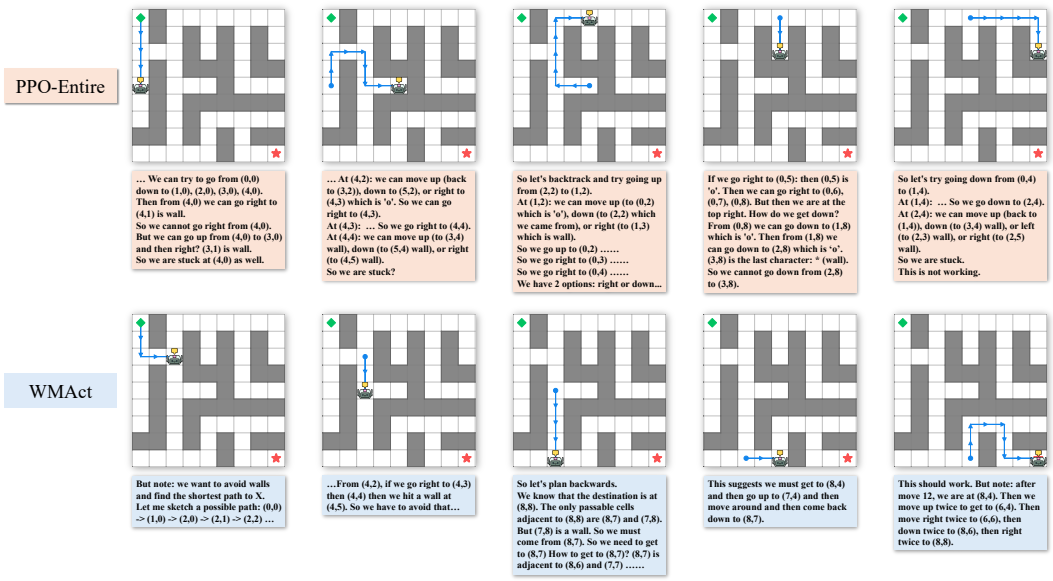


Figure 8: **Reasoning Trace Visualization: A qualitative comparison of PPO-Entire versus WMAct in a Maze task.** The top (PPO-Entire) demonstrates a more rigid, often enumeration-based or pre-computed planning approach, which frequently leads to suboptimal trajectories or impasses due to insufficient intermediate reflection and a lack of dynamic adaptation. In contrast, the bottom (WMAct) showcases the emergent interactive thinking and planning patterns of our model. Each step highlights WMAct's ability to analyze the current state, reflect on recent moves, and dynamically adjust its strategy, resulting in more robust self-correction mechanisms and efficient goal achievement.

D Visual Case Studies and Reasoning Patterns

To qualitatively assess the efficacy of World Model Reasoning, we present comparative reasoning trace visualizations across the Sokoban, Maze, and Taxi environments. These case studies collectively validate that WMAct's interactive thinking, driven by intermediate reflection, enables critical capabilities for solving complex, constraint-heavy tasks.

For the Maze (Figure 8), PPO-Entire adheres to a rigid, enumeration-based approach, frequently leading to suboptimal trajectories and impasses. Conversely, WMAct demonstrates superior long-term foresight, allowing it to quickly adapt to the complex topology and avoid unnecessary local detours, resulting in a significantly more efficient path.

For the Sokoban (Figure 9), PPO-Entire's greedy simulation leads the agent into an irreversible deadlock by failing to anticipate constraints. In contrast, WMAct showcases robust dynamic recovery and self-correction, swiftly identifying the flawed trajectory and executing a non-greedy, path-clearing maneuver to secure an efficient solution.

For the Taxi (Figure 10), PPO-Entire lacks the global view, often resorting to myopic local moves. WMAct, however, excels at systematic sub-task decomposition, enabling it to hierarchically structure the problem (e.g., pick up $\rightarrow$ drop off) and map out an optimal, wall-avoiding trajectory for the entire sequence. These results confirm that WMAct facilitates key capabilities—including robust recovery, environmental foresight, and hierarchical planning—that are essential for advanced sequential decision-making.

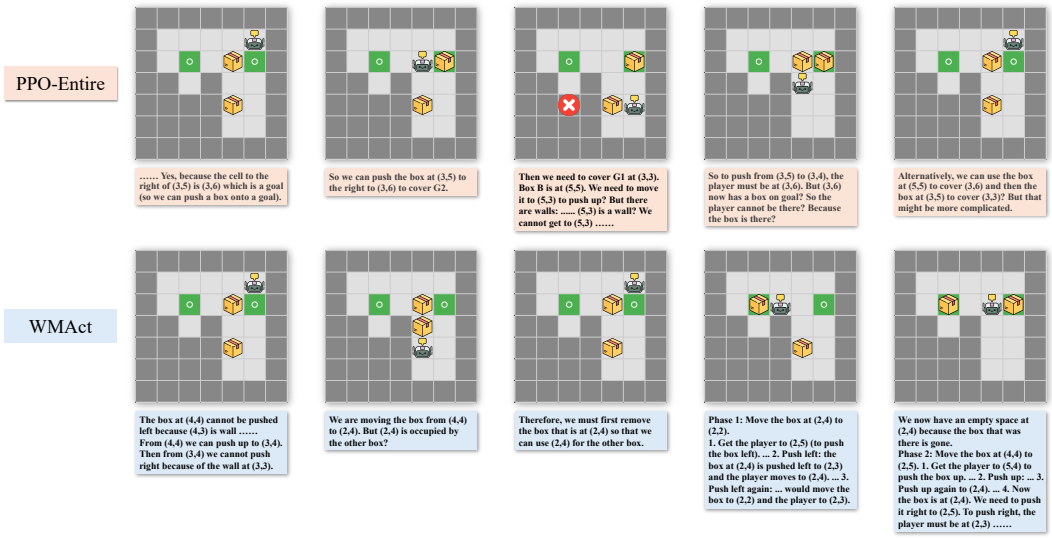


Figure 9: **Reasoning Trace Visualization: A qualitative comparison of PPO-Entire versus WMAct in a Sokoban task.** The top (PPO-Entire) demonstrates a common pitfall of model-based planning in Sokoban, where the agent, lacking robust foresight and intermediate reflection, gets trapped in an irreversible deadlock. In contrast, the bottom (WMAct) showcases the ability to systematically decompose the task and execute a non-greedy, long-term plan, demonstrating dynamic self-correction and resulting in an efficient, deadlock-free solution.

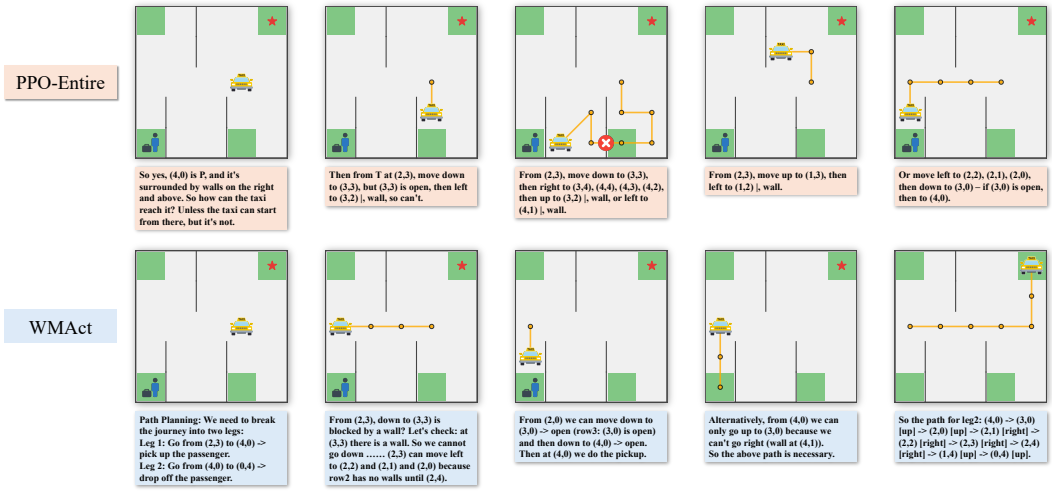


Figure 10: **Reasoning Trace Visualization: A qualitative comparison of PPO-Entire versus WMAct in a Taxi task.** The top (PPO-Entire) appears to lack robust global planning, often resorting to greedy local moves and struggling with dead-ends due to insufficient foresight regarding long-term path feasibility. In contrast, the bottom (WMAct) showcases superior global planning and sub-task decomposition, enabling it to systematically map out an efficient, wall-avoiding path.

E Prompt Details

At the initial turn, the LLM receives a comprehensive prompt composed of three distinct, functional components: the {`system prompt`} for defining the agent's role; the {`environment description`} for providing rules, map legend, and current state; and the {`action prompt`} for strictly mandating the output format. Crucially, the structural components of the prompt—the {`system prompt`} and the {`action prompt`}—are held strictly identical and un-engineered

across all three distinct environments. This uniformity demonstrates the generality and minimal engineering overhead of our approach, ensuring that the framework's effectiveness relies on the explicit reasoning structure rather than domain-specific prompt tuning.

System Prompt

You are a helpful assistant.

Environment Action Prompt

For each turn, your response must follow a strict two-part structure. First, you must conduct your detailed reasoning within <think> and </think> tags. Inside your thoughts, analyze the current state, reflect on your previous actions, and plan your next step. Second, immediately following your reasoning, you must provide your output within <action> and </action> tags. The content of the <action> tag is critical and depends on the task. For interactive tasks, generate one or more commands separated by a semicolon, for example, <action>move up; move right</action>. Your response for the turn concludes after you output the closing </action> tag. Now, generate your thought and action based on the current state.

Consequently, the diverse and sophisticated thinking patterns evidenced in Section D are not hard-coded but emerge as learned capabilities, directly elicited by the model from the unique constraints and demands of each environment.

Maze Environment Description Example

You are an agent tasked with navigating a maze. Your goal is to move the player (P) to the destination (X) through a series of turns.

Maze Board:

```
X o o o o o o o o
* * o * o * * * *
o o o * o o o o o
o * * * * * o * *
o o o * o o o o o
o * * * * * o * o
o o o * o o o * o
o * o * o * o * *
o * o * o * o o P
```

Legend:
'P' represents the player starting point
'X' represents the destination point
'o' represents empty space (passable)
'*' represents a wall (impassable)

Your available moves are:
'move up': move player one cell up.
'move down': move player one cell down
'move left': move player one cell to the left
'move right': move player one cell to the right

Sokoban Environment Description Example

You are playing Sokoban, a puzzle game where you need to push boxes onto goal locations.

Game state:

```
#######
#.P..##
#GB.#.#
#..GG.#
#.BB..#
#..####
#######
```

Legend:

- P: Player
- B: Box
- G: Goal location
- *: Box on goal location
- #: Wall
- .: Empty space

Progress: 0/3 boxes on goals.

Available actions:

- push up, push down, push left, push right (to push a box if adjacent)
- move up, move down, move left, move right (to move without pushing)
- no operation (stay in place)

Taxi Environment Description Example

You are an agent controlling a taxi in a 5x5 grid world. Your goal is to pick up the passenger and deliver them to the destination.

Taxi Map:

```
+-----------------------------+
|  P  :     |     :     :  G  |
|     :     |     :     :     |
|     :     :     :     :     |
|     |     :     |  T  :     |
|  Y  |     :     |  D  :     |
+-----------------------------+
```

‘|’ represents a wall (an impassable barrier).
‘:’ represents an open path (a tile you can move through).
‘T’ represents the taxi position.
‘P’ represents the passenger position (if not in the taxi).
‘D’ represents the destination.
‘R’, ’G’, ’Y’, ’B’ are special points.
‘T/P’ means the taxi and the passenger are in the same cell, but the passenger has NOT been picked up yet.
‘T(P)’ means the taxi is carrying the passenger (the passenger has been picked up).
‘T(P)/D’ means the taxi carrying the passenger is at the destination.
‘T/P/D’ means the taxi and the passenger all arrive at the destination cell.
‘P/D’ means the passenger and the destination are in the same cell (passenger not picked up).

Your available actions are:
‘move down’: move the taxi one cell down.
‘move up’: move the taxi one cell up.
‘move right’: move the taxi one cell right.
‘move left’: move the taxi one cell left.
‘pickup’: pick up the passenger. This action is only valid when the taxi is at the passenger’s location; otherwise, it will incur a penalty.
‘dropoff’: drop off the passenger. This action is only valid when the taxi is at the destination and carrying the passenger; otherwise, it will incur a penalty.