Beyond Prediction: Steering VLM Agents with Retrospective World Modeling

Yongjiang Liu[1], Jie Zhang[1]*, Haoyue Zhang[1], Jingcai Guo[2], Deze Zeng[3], Song Guo[1]
[1]The Hong Kong University of Science and Technology
[2]The Hong Kong Polytechnic University [3]China University of Geosciences (Wuhan)
yliunr@connect.ust.hk, csejzhang@ust.hk

Abstract

Equipping VLM agents with world modeling capabilities has shown strong potential for complex reasoning and long-horizon planning, while reducing the dependence of policy learning on costly real-world interactions. Existing methods mainly rely on prospective simulation to predict the consequences of candidate actions. However, this forward-only paradigm focuses on "*what will happen next*" and provides limited constraints for verifying whether an action is causally consistent with the observed state transition, which can lead to plausible-looking but physically incoherent behaviors. In this paper, we challenge the view of world modeling as only prospective prediction and introduce Retrospective World Modeling, a new agent learning paradigm that enables agents to reason backward by estimating the retrospective attribution distribution $P(\hat{a}_t|s_t, s_{t+1})$ for the action that most likely caused a given transition. Based on this capability, we formulate the Self-Consistency Reward (SCR), an intrinsic signal that measures the probabilistic consistency between the policy action and the retrospective explanation. Integrating SCR into reinforcement learning provides dense transition-level feedback and steers agents toward behaviors that are both task-effective and physically grounded. Extensive experiments across diverse agentic tasks show that our method substantially improves policy robustness and generalization over prospective-only world modeling baselines.

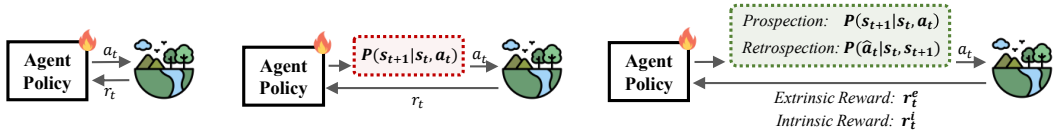
1 Introduction

Reinforcement Learning (RL) has recently emerged as a pivotal paradigm for advancing Vision–Language Model (VLM) agents, allowing them to transcend static reasoning and engage in complex, multi-turn interactions within dynamic environments [1–3], ranging from embodied exploration [4, 5] to open-ended digital tasks [6, 7]. Unlike single-turn tasks, success in these scenarios demands not just abstract planning but precise long-horizon decision-making, where every action cascades into future states. To navigate such dynamics, world modeling [8–11] has been introduced as a critical cognitive scaffold for autonomous agents. As illustrated in Fig. 1(b), unlike the reactive "*Reason-Act*" loop [12] (Fig. 1(a)), contemporary methods [13, 14] adopt a prospective "*Observe-Predict-Act*" regime. By internalizing world knowledge to simulate "*what will happen next*", these agents can mentally rehearse actions and anticipate consequences before real-world execution, significantly enhancing policy robustness and effectiveness.

While existing methods have made remarkable progress by internalizing world modeling capabilities directly into the policy to predict future states, we identify two critical limitations in this prospective-only paradigm. First, the reasoning process remains essentially open-loop [15, 16]. The agent

*Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).



(a) Conventional Agent Learning

(b) Agent Learning with World Modeling

(c) Agent Learning with Retrospective World Modeling (Ours)

Figure 1: Comparison between our retrospective world modeling framework and conventional agent learning frameworks.

inherently assumes its predictive foresight is accurate, lacking a mechanism to verify whether its planned actions causally align with the actual environmental dynamics. For instance, in a partially observable environment like *Sokoban*, an agent might hallucinate that an <Up> action will move a box upward, ignoring a blocking wall. Without real-time verification, the agent continues planning based on this flawed premise, leading to inevitable plan collapse in a dead end [17]. Second, this lack of verification represents a missed opportunity for intrinsic feedback. By overlooking the causal consistency between action and the resulting state transition, current methods discard a rich source of dense supervision. This forces the agent to rely solely on sparse, delayed extrinsic rewards upon task completion, making credit assignment extremely difficult in long-horizon tasks. This compels us to rethink: *Is a forward-only world modeling sufficient to equip an agent with a robust internal belief that is both causally consistent and self-correcting?*

In this paper, we posit that endowing agents with a bidirectional understanding of environmental dynamics can achieve more robust reasoning and decision-making. Beyond merely functioning as a forward simulator $P(s_{t+1}|s_t, a_t)$[2] that predicts future outcomes, an internal world model should also serve as a retrospective analyst, capable of deducing the underlying causality $P(\hat{a}_t|s_t, s_{t+1})$ behind the observed transitions. This retrospective capability is pivotal: it complements the agent to verify the logical soundness of its actions against actual environmental dynamics, rather than relying solely on plausible-looking predictions. To this end, we propose Retrospective World Modeling (RWM), a novel agent learning paradigm that augments agents with the ability to "*Reason backward*" (i.e., inferring the action responsible for the observed state change) and further used for policy optimization. Building upon RWM, we introduce the Self-Consistency Reward (SCR), a self-supervised signal that regularizes policy optimization towards actions whose effect can be consistently explained by the observed changes. Integrating SCR into RL establishes a closed-loop learning process that complements prospective planning with retrospective verification, shifting intrinsic supervision from curiosity-driven exploration toward exploitation-oriented consistency regularization.

We empirically evaluate our method across diverse agentic tasks, lifting the average success rate from 8% to 81% with strong generalization. Leveraging the granularity of retrospective feedback, our method exhibits higher action effectiveness and rapid convergence compared to baselines. Subsequent analyses validate the necessity of internalizing this intrinsic signal and assess various reward formulations and context regulation strategies to mitigate reward hacking.

2 Related Work

RL for LLM and VLM Agents. Recent studies have explored reinforcement learning (RL) to train long-horizon multi-turn LLM and VLM agents. Diverse capabilities have been boosted, such as planning [18–20], tool use [21–23], memory [24, 25], self-improvement [26–28], reasoning [29, 30], perception [31, 32]. RL algorithms, such as PPO [33], GRPO [34] and DAPO [35], etc., form a spectrum from general policy gradients to specialized preference learning. Recent work also studies training optimization for complex reasoning in agentic RAG, including dynamic batch-aware advantage weighting [36]. At the broader system level, recent work has surveyed protocols and multimodal coordination for AI-agent-native networks [37].

World Modeling. As a cornerstone of model-based RL [38, 39, 8], world models enable predictive planning and foresight [40, 41], achieving remarkable decision-making success via latent dynamics in

[2]Here, P is just a conceptual notation for the agent's internal world model of environmental dynamics. In our framework, this capability is instantiated by the unified policy network π_θ (Sec. 4).

systems like MuZero [42] and the Dreamer family [43, 44]. Recently adapted for LLM/VLM agents, this paradigm explicitly models state transitions (e.g., VAGEN [14]) or simulates complex domains ranging from web browsing [45, 46, 7, 47], mathematical reasoning [48], embodied control [49, 9], and games [50] to high-fidelity video prediction [51–53]. Beyond explicit modeling, some methods internalize strategies [54, 55], leveraging early exploration to align internal planning with environmental feedback.

Inverse Dynamics for Curiosity and Action Grounding. Inverse dynamics models infer actions from state transitions, and have been used in two closely related lines of work. In classical RL, curiosity-driven methods use inverse dynamics to learn action-relevant representations and compute forward prediction errors as intrinsic rewards for exploration [56–58]. In embodied and robotic learning, recent world-action models and video-based policy learning frameworks use inverse dynamics or action-conditioned world modeling to recover executable actions from visual transitions or generated future rollouts, such as PIDM [59], Motus [60], DreamZero [61], GigaWorld-0/Policy [62, 63]. These methods mainly employ inverse dynamics as an auxiliary representation-learning objective, an exploration signal, or an action decoding module. In contrast, RWM repurposes inverse action inference as a retrospective consistency validator for VLM-agent policy learning. Rather than relying on an external inverse model or rewarding prediction error to encourage exploration, RWM performs cross-turn retrospective attribution within the VLM reasoning process and converts discriminative action-transition consistency into an exploration-oriented dense intrinsic reward for directly regularizing RL policy optimization.

Summary. Despite these variations, internal world modeling in agents remains overwhelmingly prospective, tasked exclusively with answering "*what will happen next?*". We argue that this unidirectional perspective is fundamentally incomplete. Instead, we propose RWM, endowing agents with a bidirectional understanding of dynamics that unifies forward planning with retrospective causal verification for robust decision-making.

3 Preliminary

3.1 Problem formulation

We formulate the multi-turn interaction of VLM agents as a Partially Observable Markov Decision Process (POMDP), defined by the tuple $(\mathcal{S}, \mathcal{O}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \gamma)$. Here, $\mathcal{S}$ denotes the latent state space of the environment, which is typically complex and not fully observable. $\mathcal{O}$ represents the observation space (e.g., visual inputs and textual feedback), and $\mathcal{A}$ is the action space (e.g., API calls or high-level commands). The transition function $\mathcal{T} : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$ governs the environment's dynamics, while the reward function $\mathcal{R}$ evaluates task progress given a specific goal g.

At each turn t, the agent cannot access the true state $s_t \in \mathcal{S}$. Instead, it receives a partial observation $o_t \sim \mathcal{O}(s_t)$ and selects an action a_t based on the interaction history $h_t = (o_1, a_1, \ldots, o_{t-1}, a_{t-1}, o_t)$. This decision process is governed by a policy π_θ parameterized by a VLM, i.e., $a_t \sim \pi_\theta(\cdot|h_t)$. Upon executing a_t, the environment transitions to a new state s_{t+1} and emits a scalar reward r_t. The agent's objective is to learn an optimal policy π_θ that maximizes the expected cumulative return:

$$\max_\theta \ \mathbb{E}_{\pi_\theta} \left[\sum_{t=1}^{T} \gamma^{t-1} r_t \right], \tag{1}$$

where γ denotes the discount factor. The interaction continues until a terminal condition is met or a maximum horizon is reached, yielding a sequential decision-making loop.

3.2 Prospective Internal World Modeling in VLM Agents

While the POMDP formulation defines the learning objective for multi-turn interaction, a purely reactive policy $\pi_\theta(a_t|h_t)$ directly maps the interaction history to an executable action and may overlook explicit reasoning about latent states and environment dynamics. Recent VLM agent frameworks therefore incorporate prospective world modeling into the reasoning process, allowing the policy model to internally simulate the consequence of a candidate action before interacting with the environment [14, 13]. Specifically, at turn t, the agent first summarizes the partial observation o_t and history h_t into a textual belief state $\hat{s}_t$, which serves as an internal approximation of the

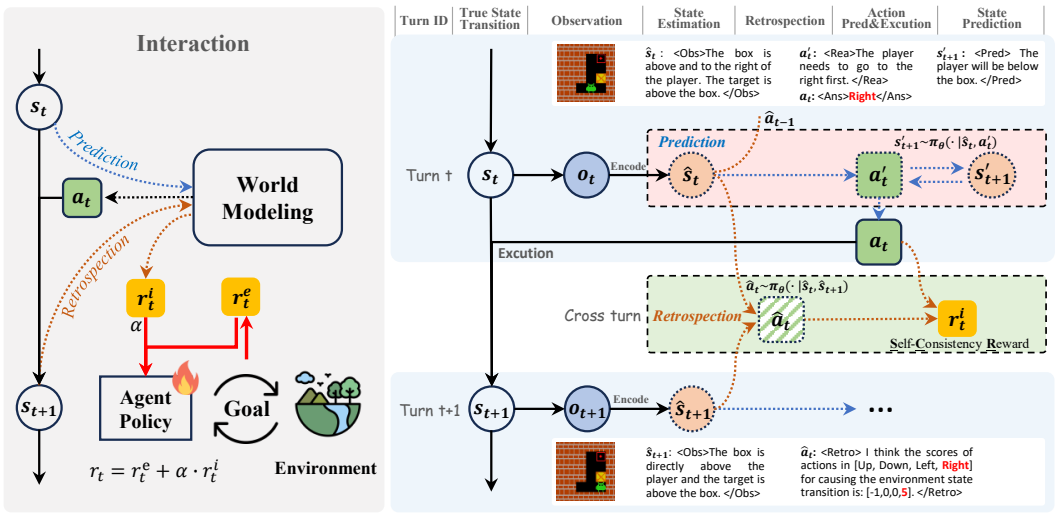


Figure 2: Overview of the RWM framework. At turn t, the agent first infers the current belief state $\hat{s}_t$ from the observation and performs retrospection over the previous transition $(\hat{s}_{t-1}, \hat{s}_t)$ and get attributed action $\hat{a}_{t-1}$. It then conducts prospective reasoning to propose a candidate action a'_t and predict an imagined next state s'_{t+1} before producing the executable action a_t. After observing the next belief state $\hat{s}_{t+1}$, RWM performs a cross-turn attribution step, inferring $\hat{a}_t$ from the realized transition $(\hat{s}_t, \hat{s}_{t+1})$, and evaluates its consistency with the executed action a_t. The resulting self-consistency reward is then assigned back to turn t for policy optimization.

underlying latent state. Conditioned on $\hat{s}_t$ and the task goal g, the agent proposes a candidate action a'_t for mental rehearsal and predicts the corresponding future belief state, i.e., $s'_{t+1} \sim \pi_\theta(\cdot|\hat{s}_t, a'_t)$. Here, s'_{t+1}[3] denotes the imagined next state predicted before execution. This prospective simulation forms an internal reasoning trace:

$$\texttt{<Obs>}\hat{s}_t\texttt{</Obs><Res>}a'_t\texttt{</Res><Pred>}s'_{t+1}\texttt{</Pred>}.$$

The final executable action a_t (<Ans>a_t</Ans>) is then sampled by conditioning the policy on both the interaction history and the above prospective reasoning trace.

4 Method

Prospective world modeling enables VLM agents to imagine future states before execution, but it does not directly verify whether an executed action is consistent with the observed state transition. For each transition from turn t to turn $t+1$, after the agent executes a_t and receives the next observation, the resulting belief transition $(\hat{s}_t, \hat{s}_{t+1})$ provides a natural opportunity to evaluate whether a_t is a plausible cause of the observed change. Equivalently, at the beginning of turn $t+1$, this verification appears as a retrospective attribution step over the previous transition. Motivated by this observation, we propose Retrospective World Modeling (RWM), a new agent learning framework that uses retrospective action attribution as a consistency verification mechanism for policy learning. As shown in Fig. 2, RWM first performs retrospective attribution over the observed transition, then converts the attribution result into a margin-based intrinsic reward, and finally optimizes the policy with temporally aligned RL.

4.1 Retrospective World Modeling

RWM extends the prospective reasoning loop with a retrospective verification step. At the beginning of turn t, the agent receives a new observation o_t and summarizes the interaction context into a current belief state $\hat{s}_t$. Since the belief state from the previous turn, $\hat{s}_{t-1}$, is already available, the

[3]For simplicity, at the turn of $t+1$, we re-generate state belief $\hat{s}_{t+1}$ given the parallel observation o_{t+1}, rather than using the predicted state belief in the t-th turn.

agent can reason backward over the observed transition $(\hat{s}_{t-1}, \hat{s}_t)$ and infer which action could have caused this change. Formally, the retrospectively attributed action is sampled as:

$$\hat{a}_{t-1} \sim \pi_\theta(\cdot \mid \hat{s}_{t-1}, \hat{s}_t). \tag{2}$$

Here, $\hat{a}_{t-1}$ is a model-internal explanation of the previous transition rather than a ground-truth label. It is used to verify the consistency between the observed transition and the action actually executed in the previous turn. To avoid trivial copying from the interaction history, retrospective attribution is conditioned only on the two belief states involved in the transition, rather than on the full action history. For the first turn, where no previous transition is available, the retrospective field is filled with a `null` token and is excluded from reward computation.

After the retrospective step, the agent proceeds to prospective reasoning for the current turn. Conditioned on the current belief state $\hat{s}_t$, the task goal g, and the retrospective context $\hat{a}_{t-1}$, the policy proposes a candidate action a'_t for mental rehearsal and predicts the imagined next belief state s'_{t+1}. Together, the insights from both the retrospective grounding and prospective simulation yield a new trajectory before the agent outputs the final action a_t for the environment to execute:

$$\texttt{<Obs>}\hat{s}_t\texttt{</Obs><Retro>}\hat{a}_{t-1}\texttt{</Retro><Res>}a'_t\texttt{</Res><Pred>}s'_{t+1}\texttt{</Pred>}. \tag{3}$$

In this trace, the `<Retro>` field explains the transition that has already occurred, while the `<Rea>` and `<Pred>` fields support planning for the current action. This "Observe-Retrospect-Predict-Act" loop makes retrospective verification an explicit part of multi-turn decision-making, while preserving prospective simulation for planning.

4.2 Self-Consistency as Intrinsic Reward

The retrospective attribution in Eq. 2 provides a qualitative explanation of the previous transition, but using the sampled action token directly as a reward signal is unreliable. A direct token match between the sampled attribution and the executed action gives a sparse signal, and it is also vulnerable to shortcut learning when the previous action appears in the interaction history. To obtain a denser and more robust learning signal, we formulate retrospective verification as a discriminative scoring problem over the finite action space $\mathcal{A}$.

Discriminative Action Scoring. After the transition from turn t to turn $t+1$ is observed, RWM evaluates the realized transition $(\hat{s}_t, \hat{s}_{t+1})$ with a history-agnostic scoring interface implemented by the same VLM policy. Specifically, the model scores each candidate action according to how plausible it is as the cause of the observed transition:

$$\mathbf{v}_t = F_\theta(\hat{s}_t, \hat{s}_{t+1}; \mathcal{A}) \in \mathbb{R}^{|\mathcal{A}|}, \tag{4}$$

where F_θ denotes the retrospective scoring form of the policy model, and each entry $\mathbf{v}_t^{(a)}$ measures the plausibility of action $a \in \mathcal{A}$ as the cause of the transition from $\hat{s}_t$ to $\hat{s}_{t+1}$. In practice, these scores are derived from the logits assigned to candidate action tokens under a prompt that exposes only the state transition and the candidate action set.

Self-Consistency Reward (SCR). Given the score vector, we define the self-consistency reward as the normalized margin between the score of the executed action a_t and the strongest alternative action:

$$r_t^i = \frac{\mathbf{v}_t^{(a_t)} - \max_{a \in \mathcal{A} \setminus \{a_t\}} \mathbf{v}_t^{(a)}}{S_{\max} - S_{\min}}. \tag{5}$$

Here, $S_{\max}$ and $S_{\min}$ denote the upper and lower bounds of the scoring scale and are used to normalize the reward magnitude. A positive margin indicates that the executed action is scored as the most plausible explanation of the observed transition, while a negative margin indicates that another candidate action better explains the transition. Compared with direct action reconstruction, this margin-based formulation provides a dense signal and explicitly contrasts the executed action with the hardest distractor, encouraging sharper action-transition consistency.

Total Reward. The total reward for policy optimization is defined as a composite sum:

$$r_t = r_t^e + \alpha \cdot r_t^i, \tag{6}$$

where r_t^e denotes the extrinsic reward, including task success, format compliance, and step penalties, and α controls the strength of retrospective consistency regularization. In Eq. 6, the extrinsic reward provides task-level supervision, while `SCR` provides transition-level feedback that regularizes the policy toward actions whose effects can be consistently explained by the observed state change.

4.3 Policy Optimization

We optimize the policy π_θ using Proximal Policy Optimization (PPO [33]). During training, the critic V_ϕ is implemented as a scalar value head on the final hidden state of the VLM. Since each trajectory contains both environmental prompts and model-generated tokens, we apply a binary mask $\mathcal{M}$ so that policy gradients are computed only on generated reasoning and action tokens, while environmental prompts are excluded from the policy update.

Temporal Reward Assignment. The self-consistency reward is delayed by one turn because it requires the next belief state $\hat{s}_{t+1}$ to be observed. Therefore, although the score vector in Eq. 4 is computed at the beginning of turn $t+1$, the resulting reward r_t^i is assigned back to the action a_t executed at turn t: $r_t^i \leftarrow \mathrm{SCR}(\hat{s}_t, \hat{s}_{t+1}, a_t)$. This temporal assignment aligns the retrospective verification signal with the transition that generated it and preserves the standard turn-level reward structure of the underlying POMDP.

Dense Reward Integration and Configurations. Given the structured reasoning trajectory established earlier, training the agent with standard PPO relies solely on sparse task and format rewards. In this standard setup, credit assignment is handled by the conventional Generalized Advantage Estimation (GAE) [64], which propagates sparse rewards backward across the established trajectory (Eq. 3) without explicit turn boundaries. Combining our Self-Consistency Reward, we define this sparse-reward configuration as **RWM-Base** setting.

However, multi-turn VLM agents face a fundamental granularity gap: our composite reward r_t is defined at the macro-level of environmental turns, whereas PPO requires micro-level token advantages. To seamlessly backpropagate turn-wise semantic feedback, we go a step further. Following VAGEN [14], we adopt a Bi-Level GAE protocol (detailed formulations are provided in Appendix. B). This mechanism bridges the gap by first computing turn-level advantages and then injecting them into the micro-scale token generation, allowing for dense reward assignment. By employing this Bi-Level GAE for turn-aware credit assignment, we can effectively inject dense supervision into the reasoning steps. Alongside external LLM-as-a-Judge [14] supervision for visual state evaluation, we critically integrate our intrinsic Self-Consistency Reward (r_t^i) to explicitly penalize causal hallucinations at each turn. This comprehensive, dense-reward configuration establishes our **RWM-Full** setting.

Optimization Objective. Let τ denote the trajectory and $\mathcal{M}$ be the mask that isolates generated action tokens (excluding environmental prompts). The above process yields the specific token advantage A_i (Eq. 14) used in the PPO clipping objective $\mathcal{L}(\theta)$:

$$\mathbb{E}_{(\tau)\sim\pi_{\text{old}}}\left[\frac{\sum_i \mathcal{M}_i \cdot \min\left(\rho_i(\theta)A_i, \text{clip}(\rho_i(\theta), 1-\epsilon, 1+\epsilon)A_i\right)}{\sum_k \mathcal{M}_k}\right] \tag{7}$$

where $\rho_i(\theta) = \frac{\pi_\theta(\tau_i|\tau_{<i})}{\pi_{\text{old}}(\tau_i|\tau_{<i})}$ is the probability ratio and ϵ is the clipping parameter. This optimization ensures the agent's policy is updated stably, driven by both the intrinsic consistency verification and extrinsic task goals.

5 Experiments

5.1 Experimental Settings

Environments. To analyze the visual reasoning capabilities of VLM agents, we evaluate on four distinct agentic tasks. These benchmarks cover challenges including diverse visual state representations and action spaces: 2D grid puzzles (*Sokoban* [65] and *FrozenLake* [66]), embodied 3D navigation (*Navigation* [67]) and embodied object manipulation (*ManiSkill* [68, 69]). Sokoban requires pushing boxes to targets while avoiding deadlocks, and FrozenLake involves navigating to a goal under hazardous conditions (with deterministic dynamics). Navigation is a first-person indoor search task that demands spatial reasoning from partial observations, while ManiSkill focuses on robotic manipulation with a Panda arm using a hybrid action space (e.g., `pick(x,y,z)`), requiring precise visual grounding from 3D scenes to actionable coordinates. For Navigation, we assess performance on both *Base* and *Common Sense* evaluation sets. For Sokoban, we employ the standard 6×6 grid and further expand to a *Hard* setting (scaled to large dimensions) to probe generalization under increased complexity. More details are provided in Appendix. E.

Table 1: Main results on the 4 general agentic benchmarks. We report average task success rates across Sokoban, Navigation, ManiSkill, and FrozenLake benchmarks. We adopt Qwen2.5-VL-3B as the backbone for the trained models. The best and second-best performance among open-source models/methods are highlighted in **bold** and underline, respectively.

Models	Sokoban			Navigation			ManiSkill					Frozen	Overall
	Standard	Hard	**Average**	Base	Common	**Average**	Place	Stack	Drawer	Align	**Average**		
Open-Source Models													
Qwen2.5-VL-72B [70]	0.18	-	0.18	**0.72**	**0.75**	**0.74**	**1.00**	0.50	0.00	**1.00**	0.63	0.44	0.51
Qwen2.5-VL-7B [70]	0.13	-	0.13	0.28	0.39	0.34	0.00	0.00	0.00	0.75	0.19	0.14	0.27
VLM-R1-3B [71]	0.13	-	0.13	0.31	0.34	0.33	0.00	0.00	0.00	0.00	0.00	0.13	0.23
Beyond Prediction: Steering VLM Agents with Retrospective World Modeling (Backbone: Qwen2.5-VL-3B)													
Qwen2.5-VL-3B [70]	0.06	0.05	0.06	0.22	0.27	0.25	0.00	0.00	0.00	0.00	0.00	0.09	0.08
+ Vanilla-PPO [72]	0.18	0.13	0.16	0.32	0.25	0.29	0.00	0.00	0.00	0.00	0.00	0.21	0.12
+ ReAct-RL [12]	0.27	0.20	0.24	0.39	0.37	0.38	0.00	0.00	0.00	0.00	0.00	0.30	0.17
+ VAGEN-Base [14]	0.41	0.33	0.37	0.47	0.51	0.49	0.88	0.63	0.63	0.75	0.72	0.43	0.56
+ RWM-Base (Ours)	0.58	0.46	0.52	0.67	0.64	0.66	**1.00**	0.88	**0.88**	**1.00**	0.94	0.75	0.76
World Modeling with Bi-Level GAE and LLM-as-a-Judge (Using GPT-4.1 nano)													
+ VAGEN-Full	0.52	0.46	0.48	0.54	0.56	0.55	**1.00**	0.88	**0.88**	**1.00**	0.94	0.54	0.71
+ RWM-Full (Ours)	**0.69**	**0.63**	**0.66**	0.67	0.69	0.68	**1.00**	**1.00**	**0.88**	**1.00**	**0.97**	**0.77**	**0.81**
Proprietary Models													
GPT-4o [73]	0.43	0.38	0.41	0.75	0.69	0.72	0.50	0.63	0.00	0.88	0.50	0.54	0.53
GPT-5 [74]	0.70	0.62	0.66	0.75	0.81	0.78	1.00	0.63	0.00	1.00	0.66	0.77	0.70
o4-mini [75]	0.44	0.40	0.42	0.75	0.75	0.75	1.00	0.50	0.00	0.75	0.56	0.82	0.60
Claude 4.5 Sonnet [76]	0.31	0.26	0.29	0.67	0.67	0.67	0.63	0.50	0.00	1.00	0.53	0.80	0.54
Claude 3.7 Sonnet [77]	0.25	0.18	0.22	0.48	0.47	0.48	0.63	0.13	0.00	1.00	0.44	0.69	0.43
Gemini 2.5 Pro [78]	0.58	0.42	0.50	0.63	0.63	0.63	0.63	0.63	0.00	0.75	0.50	0.78	0.56

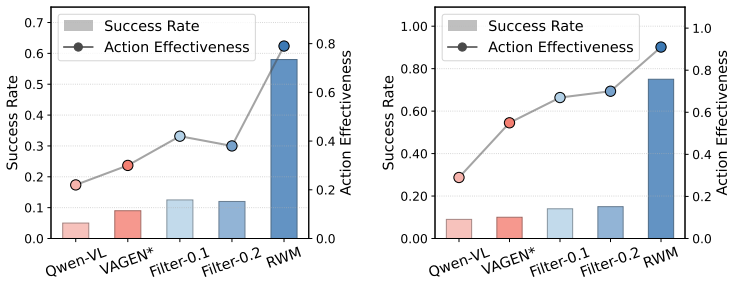
Baselines. We adopt Qwen2.5-VL-3B [70] as the VLM backbone. We compare our approach against two categories of baselines. (1) Open-source methods: **Vanilla-PPO** [72] directly fine-tunes the base model using PPO; **ReAct-RL** follows the ReAct [12] paradigm of interleaved reasoning traces ("observe-reason-act") optimized via PPO; **VAGEN-Base** [14] structures reasoning into explicit *prospective* world modeling (state estimation and future transition prediction) but relies on standard token-level GAE and sparse task rewards; **VAGEN-Full** [14] enhances this prospective paradigm by employing Bi-Level GAE for credit assignment and dense reasoning supervision provided by an external LLM-as-a-Judge. (2) Proprietary Models: We also report zero-shot results from state-of-the-art proprietary models, including `GPT-5` [74], `GPT-4o` [73], `o4-mini` [75], `Claude 4.5 Sonnet` [76], `Claude 3.7 Sonnet` [77], and `Gemini 2.5 Pro` [78]. More details are provided in Appendix. C.1.

Evaluation Metrics. Success rate is the primary metric. To evaluate how effectively our retrospective constraints filter out causally implausible hallucinations, we additionally report action validness (actions within the defined space) and action effectiveness (actions inducing meaningful state changes toward the goal, e.g., avoiding blocking walls), alongside standard rewards.

Implementation Details. We use `Qwen2.5-VL-3B` [70] as the backbone. Global training batch size is set to 128, with a learning rate of 1×10^{-6} for the actor and 1×10^{-5} for the critic. For evaluation, we set the generation temperature to 1.0. Results are averaged over 3 runs on 128 diverse test cases to ensure statistical significance. All experiments were conducted on $4\times$ NVIDIA H800 (80GB) GPUs.

5.2 Main Results

Main Performance. Tab. 1 reports success rates across four diverse environments, revealing a consistent performance hierarchy. RWM achieves substantial overall gains over model-free approaches (0.76 vs. 0.12 for Vanilla-PPO), and significantly outperforms ReAct-RL (0.17), confirming that explicit world modeling provides superior guidance over interleaved reasoning traces alone. Crucially,



(a) Sokoban (b) FrozenLake

Figure 3: Ablation on signal internalization. RWM-Filter applies the intrinsic reward r_t^i purely as a training-free rejection sampling threshold ($\delta = 0.1, 0.2$) to filter inconsistent actions. Results show that explicit RL internalization yields substantially higher success rates and action effectiveness. (*: val before train)

we compare against the prospective-only VAGEN across two configurations. In the Base setting, RWM-Base demonstrates a remarkable advantage, boosting the overall average success rate from 0.56 (VAGEN-Base) to 0.76. This superiority persists even with dense supervision: RWM-Full lifts the performance from 0.71 (VAGEN-Full) to an impressive 0.81. This sharp contrast suggests that our retrospective constraint effectively filters out causally implausible transitions often hallucinated by purely prospective models. Furthermore, despite utilizing a significantly smaller backbone (`Qwen2.5-VL-3B`), RWM surpasses several powerful proprietary models like GPT-4o (0.53) and Claude 4.5 Sonnet (0.54), and even outperforms GPT-5 (0.70), highlighting the potential of specialized agents to unlock robust reasoning capabilities through self-supervised grounding.

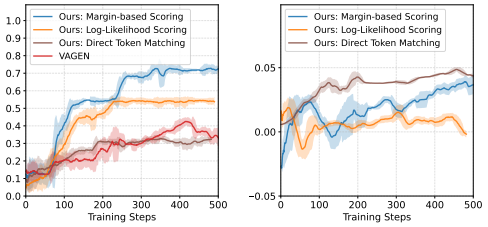
Generalization on Challenging Tasks. To probe robustness, we evaluate agents trained exclusively on standard maps on the unseen *Sokoban-Hard* setting. While baselines suffer severe degradation under this distribution shift, RWM exhibits superior generalization with a 0.46 success rate, significantly outperforming VAGEN (0.33). This resilience confirms that retrospective attribution forces the agent to internalize scalable physical rules rather than merely memorizing training layouts.

5.3 Ablation Study

In this section, we conduct extensive ablations to dissect our framework, aiming to validate retrospective signal internalization, justify our reward design, and show how to prevent reward hacking.

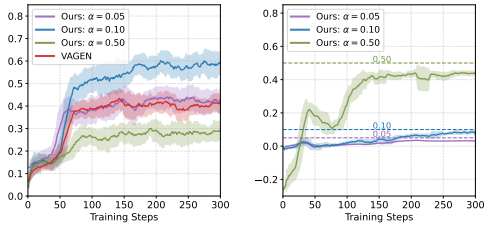
Is the RL-based "Internalization" Necessary? A core premise of our method is that the retrospective signal should be internalized into policy parameters. We investigate this via **RWM-Filter**, a training-free baseline using our intrinsic reward as a hard constraint for rejection sampling during inference. Fig. 3 reveals two critical insights. First, RWM-Filter consistently outperforms the vanilla VAGEN (e.g., +5% success rate), validating our consistency signal as a reliable indicator of physical grounding. Second, the RL-internalized RWM drastically outperforms RWM-Filter (over 40% gains). This confirms that while post-hoc filtering helps, embedding this causal constraint directly into policy parameters via RL is crucial to improve policy exploitation, transforming "*checking for errors*" into "*learning not to make them*".

Intrinsic Reward Design. To balance reward density with robustness against shortcut learning, we compare against two alternatives in Fig. 5. **1) Direct Token Matching** assigns a binary reward if the sampled token $\hat{a}_t$ matches a_t. It yields poor success rates ($\approx$0.30) and the reward saturate quickly (near the 0.05 upper bound), as sparse binary feedback hinders optimization and encourages shortcut learning, simply copying action history. **2) Log-Likelihood Scoring** (normalized log-probability of the ground-truth action) improves performance ($\approx$0.53) but suffers from high variance, as massive penalties for low-probability events destabilize training. **3) Margin-based Scoring** (Ours) avoids these pitfalls by penalizing the gap between the ground-truth action and the hardest distractor (Eq. 5). This provides a dense, bounded, and stable signal that effectively curbs shortcut learning and maximizes causal confidence, yielding the fastest convergence and highest success rate. More details are in Appendix. A.



(a) Success Rate (b) Self-Consistency Reward

Figure 5: Ablation on reward design on FrozenLake benchmark. We show the training success rate and intrinsic reward curves, comparing RWM with three reward designs and VAGEN.



(a) Success Rate (b) Self-Consistency Reward

Figure 6: Ablation on reward weight α on Sokoban benchmark. We show the training success rate and intrinsic reward curves. A large α tends to overfit, while a small α limits gains.

Mitigating Causal Hallucinations. To validate our core motivation that retrospective verification reduces hallucinations, we compare the training dynamics of action validness and effectiveness on Sokoban (Fig. 7). While the prospective-only baseline (VAGEN) gradually learns valid action formats, it severely struggles with effectiveness, often executing physically meaningless moves like pushing against blocking walls (peaking at only 0.8). In contrast, RWM rapidly achieves near-perfect validness and consistently maintains superior effectiveness (>0.95). This massive margin directly demonstrates that our retrospective constraint provides robust physical grounding, explicitly steering the agent away from causally implausible hallucinations toward purposeful and executable actions.

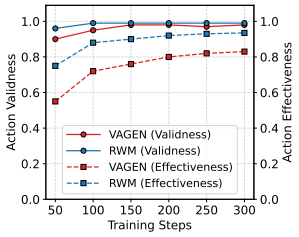


Figure 7: Ablation on action validness and effectiveness.

Sensitivity to Retrospection Weight. The parameter α (Eq. 6) balances retrospective grounding with prospective planning. As Fig. 6 shows, a minimal weight ($\alpha = 0.05$) provides insufficient gradients, yielding negligible gains over VAGEN. Conversely, an excessive weight ($\alpha = 0.50$) halves the success rate (from 0.58 to 0.29). In this regime, the intrinsic reward saturates early (Fig. 6(b)), indicating a form of reward hacking where the agent overfits to explaining transitions at the expense of task completion. This confirms the retrospective signal functions best as a regularizer.

Historical Context Leakage. In principle, RWM requires that the agent deduce $\hat{a}_t$ solely from the state transition $(\hat{s}_t, \hat{s}_{t+1})$. However, in multi-turn RL, full history access risks historical context leakage: the agent trivially retrieves the previous action a_t from memory rather than inferring it via world dynamics, essentially reward hacking. To enforce genuine causal reasoning, we evaluate three regulation strategies in Tab. 2. **1) Regulation-free** allows unrestricted history access. It performs comparably to VAGEN, confirming that memory shortcuts render the retrospective signal ineffective. **2) Action Dropout** randomly occludes action-related tokens in the history with probability β (e.g., replacing them as `...<Rea>[MASKED]</Rea>...<Ans>[MASKED]</Ans>`). While mild dropout ($\beta = 0.1, 0.2$) yields modest gains, aggressive masking ($\beta = 0.5$) severely disrupts the semantic coherence required for long-horizon planning. **3) History-agnostic Inference (Ours)** strictly isolates the retrospective prompt, conditioning deduction solely on the observed state change. Synergizing with our Margin-based Scoring, which resists simple token copying by requiring evaluation over the full action space,

Table 2: Ablation on history leakage under 3 context regulation strategies.

Method	Sokoban	FrozenLake
Qwen2.5-VL-3B	0.06	0.09
VAGEN	0.41	0.43
RWM		
- *w/ Regulation-free*	0.39	0.40
- *w/ Dropout* ($\beta = 0.10$)	0.44	0.52
- *w/ Dropout* ($\beta = 0.20$)	0.50	0.59
- *w/ Dropout* ($\beta = 0.50$)	0.33	0.38
- History-agnostic	**0.58**	**0.75**

this strict information bottleneck completely cuts off memory exploitation, achieving the highest success rate. We adopt this setting for all RWM experiments.

6 Conclusion

In this paper, we challenge the conventional prospective-only world modeling, arguing that a robust agent requires not only predicting future outcomes but also serving as a retrospective analyst. To this end, we propose Retrospective World Modeling (RWM), a novel cross-turn framework that equips VLM agents with a bidirectional understanding of environmental dynamics and the capability of deducing underlying causality behind the observed transitions. By formulating the Self-Consistency Reward (SCR), we convert the retrospective causal attribution into a dense intrinsic signal, effectively steering the agent towards physically grounded and logically consistent behaviors. Our extensive experiments and ablation studies demonstrate the effectiveness of our method and the necessity of internalizing this intrinsic signal into policy. Furthermore, we assess various reward formulations and context regulation strategies to prevent history leakage, ensuring the agent learns genuine physical rules rather than exploiting memory shortcuts. Future work will extend this paradigm to address the challenge of causal ambiguity over longer horizons.

7 Acknowledgment

This research was supported by fundings from the Hong Kong RGC General Research Fund (152228/23E, 162161/24E, 162116/25E, 162180/25E), National Natural Science Foundation of China (NSFC) Key Program (No.62532005), Collaborative Research Fund (No. C1042-23GF, No. 5097-25G), NSFC/RGC Collaborative Research Scheme (Grant No. 62461160332 & CRS_HKUST602/24), Research Impact Fund (No. R5011-23), and Areas of Excellence Scheme (AoE/E-601/22-R), and ZTE Industry-University-Institute Cooperation Funds (Grant No. IA20260813004-V1).

References

[1] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, and Zhen Zhang. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638, September 2025.

[2] OpenAI, :, Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, and et al. Openai o1 system card, 2024.

[3] Zhiheng Xi, Wenxiang Chen, Xin Guo, Wei He, Yiwen Ding, Boyang Hong, Ming Zhang, Junzhe Wang, Senjie Jin, Enyu Zhou, et al. The rise and potential of large language model based agents: A survey. *Science China Information Sciences*, 68(2):121101, 2025.

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

[5] Rui Yang, Hanyang Chen, Junyu Zhang, Mark Zhao, Cheng Qian, Kangrui Wang, Qineng Wang, Teja Venkat Koripella, Marziyeh Movahedi, Manling Li, Heng Ji, Huan Zhang, and Tong Zhang. Embodiedbench: Comprehensive benchmarking multi-modal large language models for vision-driven embodied agents. In *Forty-second International Conference on Machine Learning*, 2025.

[6] Shuyan Zhou, Frank F. Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, Uri Alon, and Graham Neubig. Webarena: A realistic web environment for building autonomous agents. In *The Twelfth International Conference on Learning Representations*, 2024.

[7] Hyungjoo Chae, Namyoung Kim, Kai Tzu iunn Ong, Minju Gwak, Gwanwoo Song, Jihoon Kim, Sunghwan Kim, Dongha Lee, and Jinyoung Yeo. Web agents with world models: Learning

and leveraging environment dynamics in web navigation. In *The Thirteenth International Conference on Learning Representations*, 2025.

[8] Yann LeCun. A path towards autonomous machine intelligence version 0.9. 2, 2022-06-27. *Open Review*, 62(1):1–62, 2022.

[9] Siyin Wang, Zhaoye Fei, Qinyuan Cheng, Shiduo Zhang, Panpan Cai, Jinlan Fu, and Xipeng Qiu. World modeling makes a better planner: Dual preference optimization for embodied task planning. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 21518–21537. Association for Computational Linguistics, 2025.

[10] Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*, 2023.

[11] Jonathan Richens, Tom Everitt, and David Abel. General agents need world models. In *Forty-second International Conference on Machine Learning*, 2025.

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

[13] Eric Xing, Mingkai Deng, Jinyu Hou, and Zhiting Hu. Critiques of world models, 2025.

[14] 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*, 2025.

[15] Jiahan Zhang, Muqing Jiang, Nanru Dai, Taiming Lu, Arda Uzunoglu, Shunchi Zhang, Yana Wei, Jiahao Wang, Vishal M. Patel, Paul Pu Liang, Daniel Khashabi, Cheng Peng, Rama Chellappa, Tianmin Shu, Alan Yuille, Yilun Du, and Jieneng Chen. World-in-world: World models in a closed-loop world, 2025.

[16] Yang Ye, Tianyu He, Shuo Yang, and Jiang Bian. Reinforcement learning with inverse rewards for world model post-training, 2025.

[17] Jeff Pflueger and Michael Everett. Safety assessment in reinforcement learning via model predictive control, 2025.

[18] Siyu Zhu, Yanbin Jiang, Hejian Sang, Shao Tang, Qingquan Song, Biao He, Rohit Jain, Zhipeng Wang, and Alborz Geramifard. Planner-r1: Reward shaping enables efficient agentic rl with smaller llms, 2025.

[19] Davide Paglieri, Bartłomiej Cupiał, Jonathan Cook, Ulyana Piterbarg, Jens Tuyls, Edward Grefenstette, Jakob Nicolaus Foerster, Jack Parker-Holder, and Tim Rocktäschel. Learning when to plan: Efficiently allocating test-time compute for llm agents, 2025.

[20] Merve Atasever, Matthew Hong, Mihir Nitin Kulkarni, Qingpei Li, and Jyotirmoy V. Deshmukh. Multi-agent path finding via offline rl and llm collaboration, 2025.

[21] Joykirat Singh, Raghav Magazine, Yash Pandya, and Akshay Nambi. Agentic reasoning and tool integration for llms via reinforcement learning, 2025.

[22] Ziwei Zheng, Michael Yang, Jack Hong, Chenxiao Zhao, Guohai Xu, Le Yang, Chao Shen, and Xing Yu. Deepeyes: Incentivizing "thinking with images" via reinforcement learning, 2025.

[23] Mingyuan Wu, Jingcheng Yang, Jize Jiang, Meitang Li, Kaizhuo Yan, Hanchao Yu, Minjia Zhang, Chengxiang Zhai, and Klara Nahrstedt. Vtool-r1: Vlms learn to think with images via reinforcement learning on multimodal tool use, 2025.

[24] Wanjun Zhong, Lianghong Guo, Qiqi Gao, He Ye, and Yanlin Wang. Memorybank: Enhancing large language models with long-term memory. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 19724–19731, 2024.

[25] Sikuan Yan, Xiufeng Yang, Zuchao Huang, Ercong Nie, Zifeng Ding, Zonggen Li, Xiaowen Ma, Jinhe Bi, Kristian Kersting, Jeff Z. Pan, Hinrich Schütze, Volker Tresp, and Yunpu Ma. Memory-r1: Enhancing large language model agents to manage and utilize memories via reinforcement learning, 2026.

[26] Yuxin Zuo, Kaiyan Zhang, Li Sheng, Shang Qu, Ganqu Cui, Xuekai Zhu, Haozhan Li, Yuchen Zhang, Xinwei Long, Ermo Hua, Biqing Qi, Youbang Sun, Zhiyuan Ma, Lifan Yuan, Ning Ding, and Bowen Zhou. TTRL: Test-time reinforcement learning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[27] Chengsong Huang, Wenhao Yu, Xiaoyang Wang, Hongming Zhang, Zongxia Li, Ruosen Li, Jiaxin Huang, Haitao Mi, and Dong Yu. R-zero: Self-evolving reasoning llm from zero data, 2026.

[28] Yifei Zhou, Song Jiang, Yuandong Tian, Jason Weston, Sergey Levine, Sainbayar Sukhbaatar, and Xian Li. Sweet-rl: Training multi-turn llm agents on collaborative reasoning tasks, 2025.

[29] Shenzhi Wang, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xiong-Hui Chen, Jianxin Yang, Zhenru Zhang, Yuqiong Liu, An Yang, Andrew Zhao, Yang Yue, Shiji Song, Bowen Yu, Gao Huang, and Junyang Lin. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for LLM reasoning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[30] Wenkai Yang, Shuming Ma, Yankai Lin, and Furu Wei. Towards thinking-optimal scaling of test-time compute for LLM reasoning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[31] Huajie Tan, Yuheng Ji, Xiaoshuai Hao, Xiansheng Chen, Pengwei Wang, Zhongyuan Wang, and Shanghang Zhang. Reason-RFT: Reinforcement fine-tuning for visual reasoning of vision language models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[32] Linghao Zhu, Yiran Guan, Dingkang Liang, Jianzhong Ju, Zhenbo Luo, Bin Qin, Jian Luan, Yuliang Liu, and Xiang Bai. Shuffle-r1: Efficient rl framework for multimodal large language models via data-centric dynamic shuffle, 2025.

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

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

[35] Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan 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, Mu Qiao, Yonghui Wu, and Mingxuan Wang. Dapo: An open-source llm reinforcement learning system at scale, 2025.

[36] Y. Chen, F. Li, J. Wu, et al. Training optimization for complex reasoning tasks in ACN: dynamic batch-aware advantage weighting for agentic RAG. *ZTE Communications*, 24(2):26–32, June 2026.

[37] X. T. Zhang, H. Xiao, D. Wang, et al. Toward AI-agent-native 6G networks: a survey on protocols, multimodal coordination, and ISCC-driven dynamic networking. *ZTE Communications*, 24(2):16–25, June 2026.

[38] Richard S Sutton. Dyna, an integrated architecture for learning, planning, and reacting. *ACM Sigart Bulletin*, 2(4):160–163, 1991.

[39] David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. *Advances in neural information processing systems*, 31, 2018.

[40] Lin Guan, Karthik Valmeekam, Sarath Sreedharan, and Subbarao Kambhampati. Leveraging pre-trained large language models to construct and utilize world models for model-based task planning. *Advances in Neural Information Processing Systems*, 36:79081–79094, 2023.

[41] Shuofei Qiao, Runnan Fang, Ningyu Zhang, Yuqi Zhu, Xiang Chen, Shumin Deng, Yong Jiang, Pengjun Xie, Fei Huang, and Huajun Chen. Agent planning with world knowledge model. *Advances in Neural Information Processing Systems*, 37:114843–114871, 2024.

[42] Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, et al. Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, 588(7839):604–609, 2020.

[43] Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations*, 2020.

[44] Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse control tasks through world models. *Nature*, pages 1–7, 2025.

[45] Yu Gu, Kai Zhang, Yuting Ning, Boyuan Zheng, Boyu Gou, Tianci Xue, Cheng Chang, Sanjari Srivastava, Yanan Xie, Peng Qi, Huan Sun, and Yu Su. Is your LLM secretly a world model of the internet? model-based planning for web agents. *Transactions on Machine Learning Research*, 2025.

[46] Jichen Feng, Yifan Zhang, Chenggong Zhang, Yifu Lu, Shilong Liu, and Mengdi Wang. Web world models, 2025.

[47] Tianqing Fang, Hongming Zhang, Zhisong Zhang, Kaixin Ma, Wenhao Yu, Haitao Mi, and Dong Yu. WebEvolver: Enhancing web agent self-improvement with co-evolving world model. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng, editors, *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 8959–8975, Suzhou, China, November 2025. Association for Computational Linguistics.

[48] Shibo Hao, Yi Gu, Haodi Ma, Joshua Jiahua Hong, Zhen Wang, Daisy Zhe Wang, and Zhiting Hu. Reasoning with language model is planning with world model. In *The 2023 Conference on Empirical Methods in Natural Language Processing*, 2023.

[49] Jiannan Xiang, Tianhua Tao, Yi Gu, Tianmin Shu, Zirui Wang, Zichao Yang, and Zhiting Hu. Language models meet world models: Embodied experiences enhance language models. *Advances in neural information processing systems*, 36:75392–75412, 2023.

[50] Sai Wang, Yu Wu, and Zhongwen Xu. Cogito, ergo ludo: An agent that learns to play by reasoning and planning, 2025.

[51] Alexi Gladstone, Ganesh Nanduru, Md Mofijul Islam, Peixuan Han, Hyeonjeong Ha, Aman Chadha, Yilun Du, Heng Ji, Jundong Li, and Tariq Iqbal. Energy-based transformers are scalable learners and thinkers, 2025.

[52] Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, et al. Video generation models as world simulators. *OpenAI Blog*, 1(8):1, 2024.

[53] Xuanhua He, Tianyu Yang, Ke Cao, Ruiqi Wu, Cheng Meng, Yong Zhang, Zhuoliang Kang, Xiaoming Wei, and Qifeng Chen. Active intelligence in video avatars via closed-loop world modeling. *arXiv preprint arXiv:2512.20615*, 2025.

[54] 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*, 2025.

[55] Shiqi Chen, Tongyao Zhu, Zian Wang, Jinghan Zhang, Kangrui Wang, Siyang Gao, Teng Xiao, Yee Whye Teh, Junxian He, and Manling Li. Internalizing world models via self-play finetuning for agentic rl. *arXiv preprint arXiv:2510.15047*, 2025.

[56] Deepak Pathak, Pulkit Agrawal, Alexei A Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *International conference on machine learning*, pages 2778–2787. PMLR, 2017.

[57] Yuri Burda, Harri Edwards, Deepak Pathak, Amos Storkey, Trevor Darrell, and Alexei A. Efros. Large-scale study of curiosity-driven learning. In *International Conference on Learning Representations*, 2019.

[58] Zhang-Wei Hong, Tsu-Jui Fu, Tzu-Yun Shann, and Chun-Yi Lee. Adversarial active exploration for inverse dynamics model learning. In *Conference on Robot Learning*, pages 552–565. PMLR, 2020.

[59] Yang Tian, Sizhe Yang, Jia Zeng, Ping Wang, Dahua Lin, Hao Dong, and Jiangmiao Pang. Predictive inverse dynamics models are scalable learners for robotic manipulation. In *The Thirteenth International Conference on Learning Representations*, 2025.

[60] Hongzhe Bi, Hengkai Tan, Shenghao Xie, Zeyuan Wang, Shuhe Huang, Haitian Liu, Ruowen Zhao, Yao Feng, Chendong Xiang, Yinze Rong, et al. Motus: A unified latent action world model. *arXiv preprint arXiv:2512.13030*, 2025.

[61] Seonghyeon Ye, Yunhao Ge, Kaiyuan Zheng, Shenyuan Gao, Sihyun Yu, George Kurian, Suneel Indupuru, You Liang Tan, Chuning Zhu, Jiannan Xiang, et al. World action models are zero-shot policies. *arXiv preprint arXiv:2602.15922*, 2026.

[62] GigaWorld Team, Angen Ye, Boyuan Wang, Chaojun Ni, Guan Huang, Guosheng Zhao, Haoyun Li, Jiagang Zhu, Kerui Li, Mengyuan Xu, et al. Gigaworld-0: World models as data engine to empower embodied ai. *arXiv preprint arXiv:2511.19861*, 2025.

[63] Angen Ye, Boyuan Wang, Chaojun Ni, Guan Huang, Guosheng Zhao, Hao Li, Hengtao Li, Jie Li, Jindi Lv, Jingyu Liu, et al. Gigaworld-policy: An efficient action-centered world–action model. *arXiv preprint arXiv:2603.17240*, 2026.

[64] John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2015.

[65] Max-Philipp B. Schrader. gym-sokoban. `https://github.com/mpSchrader/gym-sokoban`, 2018.

[66] Mark Towers, Ariel Kwiatkowski, Jordan Terry, John U Balis, Gianluca De Cola, Tristan Deleu, Manuel Goulão, Andreas Kallinteris, Markus Krimmel, Arjun KG, et al. Gymnasium: A standard interface for reinforcement learning environments. *arXiv preprint arXiv:2407.17032*, 2024.

[67] Eric Kolve, Roozbeh Mottaghi, Winson Han, Eli VanderBilt, Luca Weihs, Alvaro Herrasti, Daniel Gordon, Yuke Zhu, Abhinav Gupta, and Ali Farhadi. AI2-THOR: An Interactive 3D Environment for Visual AI. *arXiv*, 2017.

[68] Stone Tao, Fanbo Xiang, Arth Shukla, Yuzhe Qin, Xander Hinrichsen, Xiaodi Yuan, Chen Bao, Xinsong Lin, Yulin Liu, Tse-Kai Chan, Yuan Gao, Xuanlin Li, Tongzhou Mu, Nan Xiao, Arnav Gurha, Viswesh N, Yong Woo Choi, Yen-Ru Chen, Zhiao Huang, Roberto Calandra, Rui Chen, Shan Luo, and Hao Su. Maniskill3: GPU parallelized robot simulation and rendering for generalizable embodied AI. In *7th Robot Learning Workshop: Towards Robots with Human-Level Abilities*, 2025.

[69] Ayano Hiranaka, Minjune Hwang, Sharon Lee, Chen Wang, Li Fei-Fei, Jiajun Wu, and Ruohan Zhang. Primitive skill-based robot learning from human evaluative feedback. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7817–7824. IEEE, 2023.

[70] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond, 2023.

[71] Haozhan Shen, Peng Liu, Jingcheng Li, Chunxin Fang, Yibo Ma, Jiajia Liao, Qiaoli Shen, Zilun Zhang, Kangjia Zhao, Qianqian Zhang, et al. Vlm-r1: A stable and generalizable r1-style large vision-language model. *arXiv preprint arXiv:2504.07615*, 2025.

[72] Zihan Wang, Kangrui Wang, Qineng Wang, Pingyue Zhang, Linjie Li, Zhengyuan Yang, Xing Jin, Kefan Yu, Minh Nhat Nguyen, Licheng Liu, Eli Gottlieb, Yiping Lu, Kyunghyun Cho, Jiajun Wu, Li Fei-Fei, Lijuan Wang, Yejin Choi, and Manling Li. Ragen: Understanding self-evolution in llm agents via multi-turn reinforcement learning, 2025.

[73] OpenAI. Gpt-4o system card. `https://arxiv.org/abs/2410.21276`, 2024.

[74] OpenAI. Introducing gpt-5, 2025. OpenAI blog.

[75] OpenAI. Introducing openai o3 and o4-mini, 2025. OpenAI blog.

[76] Anthropic. Introducing claude sonnet 4.5, 2025. Anthropic blog.

[77] The claude 3 model family: Opus, sonnet, haiku.

[78] Google. Gemini 2.5: Our most intelligent ai model, 2025. Google blog.

Appendix

A Details of Reward Design

In the main text (Section 4.2), we introduced Margin-based Scoring as our primary method for quantifying the self-consistency intrinsic reward. To validate the superiority of this design, our ablation study (Section 5.3) compares it against two alternative formulations: Direct Token Matching and Log-Likelihood Scoring. In this section, we provide the formal definitions and detailed implementations for these alternative designs.

Relationships Among Reward Designs. It should be noted that both Log-Likelihood Scoring and Margin-based Scoring are discriminative action scoring methods, meaning they use the same scoring mechanism. The difference lies in how the resulting score vector is converted into a reward. In contrast, Direct Token Matching does not require any scoring.

A.1 Direct Token Matching

In this variant, we simplify the retrospective process by enforcing a discrete decision rather than evaluating the full probability distribution. Instead of calculating a margin-based score over the entire action space, the agent is prompted to directly sample a single action token that explicitly identifies the cause of the transition.

Formally, given the observed transition tuple $(\hat{s}_t, \hat{s}_{t+1})$, the policy samples a predicted retrospective action $\hat{a}_t$:

$$\hat{a}_t \sim \pi_\theta(\cdot \mid \hat{s}_t, \hat{s}_{t+1}) \tag{8}$$

The intrinsic reward is then formulated as a sparse binary signal using the indicator function $\mathbb{I}(\cdot)$:

$$r_t^i = \mathbb{I}(\hat{a}_t = a_t) = \begin{cases} 1, & \text{if } \hat{a}_t = a_t \\ 0, & \text{otherwise} \end{cases} \tag{9}$$

where a_t denotes the ground-truth action actually executed by the agent in turn t. Similiar to Eq.6, r_t^i is balanced by weight α.

While straightforward, this approach suffers from two critical limitations during policy optimization:

- **Sparsity:** The binary nature of the signal provides zero-gradient information when the prediction is incorrect, which severely hinders efficient gradient estimation for policy optimization.
- **Shortcut Learning:** Since the ground-truth action a_t is inherently present in the agent's interaction history, the agent tends to exploit this information leakage by simply retrieving the action from memory (reward hacking), rather than explicitly reasoning about the underlying environmental dynamics.

A.2 Log-Likelihood Scoring

This strategy shares the same discriminative scoring mechanism as the Margin-based one, where the policy evaluates candidate actions and outputs a score vector $\mathbf{v} \in \mathbb{R}^{|\mathcal{A}|}$. For consistent implementation across methods, we instruct the model to assign scores within a discrete integer range of $[-5, 5]$. Note that this setting also determines the normalization constants in Eq. 5, where $S_{\text{max}} = 5$ and $S_{\text{min}} = -5$.

Unlike the margin-based formulation, this approach treats the scores as unnormalized logits. We first apply a Softmax function to convert $\mathbf{v}$ into a probability distribution p:

$$p(a) = \frac{\exp(\mathbf{v}^{(a)})}{\sum_{a' \in \mathcal{A}} \exp(\mathbf{v}^{(a')})} \tag{10}$$

The intrinsic reward is then calculated as the log-probability of the ground-truth action a_t, centered by the log-probability of a random policy (uniform distribution):

$$r_t^i = \log p(a_t) - \log\left(\frac{1}{|\mathcal{A}|}\right) \tag{11}$$

Subtracting $\log(1/|\mathcal{A}|)$ ensures that the reward is positive when the model's confidence exceeds random guessing and negative otherwise. Intuitively, this aligns with the expectation that the signal strength should scale with the extent to which the correct inference deviates from random chance.

While statistically sound, we observe that this formulation leads to training instability due to the **asymmetric sensitivity** of the logarithmic function. The reward magnitude explodes for low-probability events, overshadowing the signal from correct predictions. Consider a standard environment with an action space size of $|\mathcal{A}| = 4$. The random baseline is $\log(0.25) \approx -1.39$.

- If the agent is confident and correct ($p(a_t) \approx 0.999$), the reward is bounded: $\log(0.999) - \log(0.25) \approx +1.38$.
- However, if the agent assigns a low probability ($p(a_t) \approx 0.001$), the penalty becomes massive: $\log(0.001) - \log(0.25) \approx -5.62$.

This property induces large fluctuations with a highly asymmetric range ($r_t^i \in (-\infty, \log|\mathcal{A}|]$). As demonstrated in our ablation study (Fig. 6), the intrinsic signal functions best as a *curiosity regularizer* rather than a *dominator*. However, the log-likelihood formulation creates a fundamental dilemma for calibrating the weighting coefficient α. Consider the previous example where the positive reward is $+1.38$ but the penalty can easily reach -5.62 or worse.

- To prevent the massive penalty (dominator) from overriding the extrinsic task reward and causing overly conservative behavior, α must be set to a very small value.
- However, heavily scaling down the penalty simultaneously renders the positive signal negligible (e.g., shrinking $+1.38$ to near zero), thereby stripping the signal of its ability to guide the policy.

Consequently, it is difficult to find a single α that allows the intrinsic reward to consistently serve as a gentle regularizer across both success and failure cases.

A.3 Case Study: Comparison of Reward Calculations

To intuitively illustrate the behavioral differences among the three reward formulations, we provide a concrete case study in Tab. 3. Consider a scenario where the action space is $\mathcal{A} = [\text{Up}, \text{Down}, \text{Left}, \text{Right}]$, and the actual executed action (ground truth) is `Right`. The agent, however, hallucinates and infers the action was `Left`.

As shown in the calculations, **Direct Token Matching** yields a binary zero, providing no gradient information regarding how far off the prediction was. **Log-Likelihood Scoring** results in a massive penalty (-8.61), demonstrating the high-variance and instability issues discussed in Sec. A.2. In contrast, our **Margin-based Scoring** gracefully bounds the penalty to -1.0, providing a stable and dense learning signal.

B Details of Advantage Estimation

As discussed in Sec. 4.3, our composite reward r_t (which importantly encompasses the intrinsic Self-Consistency Reward r_t^i) is evaluated at the macro-level of environmental turns. However, optimizing the VLM policy via PPO requires fine-grained advantage estimates at the micro-level of individual tokens. To bridge this granularity gap, we adopt the Bi-Level General Advantage Estimation (GAE) protocol introduced by VAGEN [14].

For completeness and to clarify our specific implementation within the Retrospective World Modeling (RWM) framework, we formally describe this two-stage credit assignment process below:

Turn-Level Advantage Estimation. We first calculate the advantage for each environmental turn based on the macro-level reward r_t. Let $\bar{\tau}_{\leq a_t}$ denote the full token prefix of the trajectory up to and including the generated action at turn t. Using the critic network V_ϕ, we compute the turn-level TD-error δ_t^{turn} and subsequently the turn-level advantage A_t^{turn} via standard backward propagation across turns:

$$\delta_t^{\text{turn}} = r_t + \gamma_{\text{turn}} V_\phi(\bar{\tau}_{\leq a_{t+1}}) - V_\phi(\bar{\tau}_{\leq a_t}) \tag{12}$$

$$A_t^{\text{turn}} = \delta_t^{\text{turn}} + \gamma_{\text{turn}} \lambda_{\text{turn}} A_{t+1}^{\text{turn}} \tag{13}$$

Table 3: Case study comparing the three intrinsic reward calculations. The executed ground-truth action is `Right`, but the agent's retrospection leans towards `Left`.

Method	Agent Retrospection Output	Calculation Detail & Final Reward
Direct Token Matching	`<retrospection>I think the possible action for causing the environment state transition is: Left. </retrospection>`	$r^i = \mathbb{I}(\texttt{Left} == \texttt{Right})$ **Result: 0.00**
Log-Likelihood Scoring ($\mathbf{v} = [-1, -5, 5, -5]$)	`<retrospection>I think the scores of actions in [Up, Down, Left, Right] for causing the environment state transition is: [-1,-5,5,-5]. </retrospection>`	$p(\texttt{Right}) = \frac{e^{-5}}{e^{-1}+e^{-5}+e^{5}+e^{-5}}$ $p(\texttt{Right}) \approx 4.53 \times 10^{-5}$ $r^i = \log(p(\texttt{Right})) - \log(0.25)$ $r^i = -10.00 - (-1.39)$ **Result: -8.61** (Massive Penalty)
Margin-based Scoring (Ours) ($\mathbf{v} = [-1, -5, 5, -5]$)	`<retrospection>I think the scores of actions in [Up, Down, Left, Right] for causing the environment state transition is:[-1,-5,5,-5]. </retrospection>`	$v(\text{Right}) = -5$ $\max_{a \neq \texttt{Right}} v(a) = v(\texttt{Left}) = 5$ $S_{\max} = 5, S_{\min} = -5$ $r^i = \frac{-5-5}{5-(-5)} = \frac{-10}{10}$ **Result: -1.00** (Bounded Penalty)

where γ_{turn} and λ_{turn} are the discount factor and the GAE decay parameter at the turn level, respectively.

Token-Level Advantage Estimation. After deriving the turn-level advantages, we perform an inner GAE loop for the generated tokens within each specific action a_t. Let $\delta_{t,i}^{\text{token}}$ represent the intrinsic token-level TD-error for the i-th token, which is typically derived from the KL divergence penalty against the reference model to prevent policy degradation.

Crucially, to inject the episodic semantic feedback into token generation, the backward pass for the token-level advantages is anchored by the turn-level advantage. Specifically, the aggregated signal A_t^{turn} is incorporated into the advantage of the final generated token of action a_t. Standard GAE then propagates this fused signal backward through all preceding tokens within that specific turn:

$$A_{t,i}^{\text{token}} = \delta_{t,i}^{\text{token}} + \gamma_{\text{token}}\lambda_{\text{token}} A_{t,i+1}^{\text{token}} \tag{14}$$

Token advantage $A_{t,i+1}^{\text{token}}$ is used in the policy optimization, as illustrated in Eq. 7. By doing so, the dense self-consistency verification signal formulated at the turn level is successfully distributed to optimize the fine-grained token generation process.

C Experimental Details

We use `Qwen2.5-VL-3B` [70] as the backbone. During training, the global batch size is set to 128, with a learning rate of 1×10^{-6} for the actor and 1×10^{-5} for the critic. For evaluation, we set the generation temperature to 1.0. Results are averaged over 3 runs on 128 diverse test cases to ensure statistical significance. top-p 1.0, and max new tokens 1024. We repeat 3 evaluations on 128 test cases generated from the environment and calculate the average to mitigate randomness. For agent's interaction with environment, the maximum turn number is set to 4. To support retrospective reasoning, the interaction trajectory τ_t is structured as: `<think><Obs>`$\hat{s}_t$`</Obs><Retro>`$\hat{a}_{t-1}$`</Retro><Rea>`a'_t`</Rea><Pred>`s'_{t+1}`</Pred></think><Ans>`a_t`</Ans>`, where `<Obs>`, `<Retro>`, `<Rea>`, `<Pred>` and `<Ans>` are abbreviations for observation, retrospection, reasoning, prediction and final answer. All experiments were conducted on 4× NVIDIA H800 (80GB) GPUs.

C.1 Baselines

To comprehensively evaluate the effectiveness of our Retrospective World Modeling (RWM) paradigm, we compare it against a spectrum of reinforcement learning baselines. These baselines are carefully selected to isolate the contributions of intermediate reasoning, structured world modeling, and advantage estimation techniques.

Vanilla-PPO [72]. This is the most fundamental RL baseline, which directly maps observations to executable actions. The agent is trained purely on sparse environmental rewards at the end of the

trajectory using standard Token-Level GAE. It serves to demonstrate the baseline performance of the VLM when it acts purely reactively.

ReAct-RL [12]. Building upon the direct-action approach, this baseline incorporates the ReAct paradigm by prompting the agent to generate free-form, natural language reasoning traces ("observe-reason-act") before committing to an action. Like Vanilla-PPO, it is optimized using Token-Level GAE with sparse task rewards. This baseline tests whether unstructured chain-of-thought generation is sufficient for complex agentic tasks without explicit world modeling constraints.

VAGEN-Base [14]. As a more challenging exploration baseline, VAGEN-Base introduces structured *prospective* world modeling, prompting the agent to explicitly generate state estimations and predict future transitions. However, it strips away any dense semantic feedback, relying entirely on sparse task success and basic formatting rewards. For credit assignment, it falls back to standard Token-Level GAE, where the sparse reward is anchored to the final token of the trajectory and propagated backward across all action tokens without explicit turn boundaries. This setup rigorously tests whether prospective world modeling capabilities can naturally emerge from pure trial-and-error interactions without explicit guidance.

VAGEN-Full [14]. Representing the upper bound of the prospective reasoning paradigm, VAGEN-Full provides dense supervision to the agent's internal world model. It calculates a comprehensive reasoning reward using an external LLM-as-a-Judge to directly evaluate the factual correctness of the agent's prospective state estimations and predictions. Crucially, to resolve the temporal credit assignment bottleneck inherent in multi-turn RL, it implements the Bi-Level GAE mechanism. By computing a macro-scale turn-level advantage (A_t^{turn}) first and injecting it as a terminal target into the micro-scale token-level inner-GAE, this setup ensures that episodic success is explicitly credited to the specific reasoning tokens that justified the action.

Summary of Method Configurations. To provide a clear conceptual boundary between the prospective VAGEN framework and our proposed retrospective RWM framework, we summarize their structural and algorithmic differences. While both frameworks share the concept of explicitly structuring the VLM's internal thoughts, they fundamentally diverge in their reasoning direction (Prospective vs. Retrospective). These distinctions are detailed in Table 4. Both RWM-Base and RWM-Full use SCR in addition to task and format rewards. Within each configuration, VAGEN and RWM share the advantage-estimation scheme and external judge setting. The Base comparison therefore does not involve Bi-Level GAE or external judge supervision, while the Full comparison holds both components fixed. These comparisons isolate the addition of retrospective reasoning and SCR as a package, but do not by themselves separate their individual contributions.

Table 4: Comparison of training configurations across VAGEN and RWM variants.

Setting	Reasoning Direction	Credit Assignment	Dense Reward Components
VAGEN-Base	Prospective Only	Token-Level GAE	None (Sparse Task and Format Reward Only)
VAGEN-Full	Prospective Only	Bi-Level GAE	LLM-as-a-Judge Reward
RWM-Base	Prospective + Retrospective	Token-Level GAE	**Self-Consistency Reward**
RWM-Full	Prospective + Retrospective	Bi-Level GAE	LLM-as-a-Judge + **Self-Consistency Reward**

D Additional Analyses of Retrospective World Modeling

This section provides controlled ablations and additional analyses of the source of RWM's improvements, its sensitivity to stochastic transitions and belief-state errors, variation across independent training runs, and computational cost. We also clarify the relationship between retrospective attribution and inverse dynamics, and the scope of the consistency signal.

D.1 Disentangling Retrospective Reasoning and Reward Supervision

Matched controls. We evaluate three additional Base variants with the same backbone, Token-Level GAE, rollout budget, and other optimization settings:

- **Prompt-matched control:** retains the <Retro> field and approximately the same reasoning-token budget, but describes what changed between the two belief states instead of attributing the transition to an action. It does not use SCR.
- **RWM-Base without SCR:** retains genuine retrospective action attribution and the same reasoning structure, but removes the intrinsic consistency reward by setting $\alpha = 0$.
- **RWM-Base with FTC:** retains the retrospective reasoning structure but replaces SCR with a forward transition consistency reward comparing the imagined next state with the subsequently observed belief state:

$$r_t^{\mathrm{FTC}} = \cos\big(E(s'_{t+1}), E(\hat{s}_{t+1})\big)\,, \tag{15}$$

where E is the frozen Qwen3-Embedding-0.6B model. FTC is applied at the same transition timesteps and with the same intrinsic-reward coefficient α as SCR. It measures forward state agreement without contrasting the executed action against alternative actions.

Table 5: Controlled ablations of retrospective reasoning and intrinsic reward supervision.

Method	Sokoban	Navigation		FrozenLake
		Base	Common	
VAGEN-Base	0.41	0.47	0.51	0.43
Prompt-matched control	0.40	0.49	0.52	0.41
RWM-Base without SCR	0.43	0.48	0.54	0.41
RWM-Base with FTC	0.46	0.50	0.56	0.49
RWM-Base	**0.58**	**0.67**	**0.64**	**0.75**

Table 5 shows that the prompt-matched control and retrospective attribution without SCR remain close to VAGEN-Base. Their changes are small and inconsistent across settings, suggesting that the additional field, longer reasoning trace, or retrospective prompting alone does not explain the observed gains. Enabling SCR while retaining the attribution trace improves success over the no-SCR variant by 0.15 and 0.14 on Sokoban Standard and Hard, 0.19 and 0.10 on the two Navigation settings, and 0.34 on FrozenLake.

FTC provides smaller improvements over the no-SCR variant: 0.03 on Sokoban Standard, 0.02 on each Navigation setting, and 0.08 on FrozenLake. Under these controls, SCR yields stronger gains than forward state agreement alone. The results support the importance of converting retrospective attribution into action-relative optimization feedback, rather than attributing the improvement solely to additional reasoning or the presence of dense transition feedback. They do not establish superiority over arbitrary dense reward designs or all curiosity- and representation-based objectives.

D.2 Robustness to Stochastic Transitions

RWM evaluates adjacent one-step tuples $(\hat{s}_t, a_t, \hat{s}_{t+1})$: the agent issues one action and receives the next observation before acting again. This avoids ambiguity due specifically to unobserved intermediate action sequences, but does not eliminate many-to-one transitions, no-op ambiguity, partial observability, or stochastic aliasing.

For discrete candidate actions, SCR uses the normalized difference between the executed-action score and the strongest alternative score. When these two scores are close, the reward magnitude is small. This property can attenuate feedback on transitions for which the scorer assigns similar plausibility to the executed action and its competitors; it is not a guarantee of calibrated uncertainty, since an erroneous scorer may still produce a large margin. The bounded reward and small coefficient limit its contribution relative to the task objective.

Controlled slippery FrozenLake. To test stochastic inverse-attribution ambiguity, we introduce a slippery FrozenLake variant. The policy-issued action is realized with probability $1 - p_{\mathrm{slip}}$, and each perpendicular action is realized with probability $p_{\mathrm{slip}}/2$. SCR is computed with respect to the policy-issued action. The stochastic environment samples the realized movement, whose identity is not revealed to the retrospective scorer. The scorer therefore observes the resulting transition without access to the sampled perturbation.

Table 6: Success rates under controlled stochastic transitions in FrozenLake. p_{slip} is the probability that the policy-issued action is replaced by a perpendicular action.

p_{slip}	VAGEN-Base	RWM-Base	Δ
0.0	0.43	0.75	$+0.32$
0.1	0.40	0.69	$+0.29$
0.2	0.36	0.61	$+0.25$
0.3	0.31	0.52	$+0.21$

As shown in Table 6, performance decreases for both methods as stochasticity increases, while RWM retains an advantage at every tested level. At $p_{\text{slip}} = 0.3$, the improvement is 21 percentage points. This supports the usefulness of retrospective consistency under the tested moderate stochastic perturbations, while the decreasing advantage also indicates sensitivity to transition ambiguity. These results do not establish robustness to unrestricted or extreme stochastic dynamics.

D.3 Sensitivity to Belief-State Errors

We analyze held-out Sokoban and FrozenLake transitions using three belief inputs: simulator-derived *oracle beliefs*, policy-produced *generated beliefs*, and *corrupted beliefs*, in which each spatial relation is replaced by another valid relation with probability p. This analysis evaluates the retrospective scorer, rather than retraining or evaluating the full acting policy under corrupted observations.

We report attribution accuracy, the mean rank of the executed action among four candidates (rank 1 is best), and mean normalized SCR. Attribution accuracy measures recovery of the recorded executed action; in aliased transitions it should not be interpreted as recovery of a uniquely identifiable cause.

Table 7: Sensitivity of retrospective attribution to belief inputs. Higher accuracy and SCR, and lower executed-action rank, are better. Corruption replaces individual spatial relations with probability p.

Belief input	p	Sokoban			FrozenLake		
		Accuracy (%)	Rank	SCR	Accuracy (%)	Rank	SCR
Oracle	0	94.2	1.09	0.86	96.4	1.05	0.91
Generated	0	87.1	1.17	0.80	90.3	1.15	0.85
Corrupted	0.1	80.8	1.29	0.71	84.5	1.27	0.74
Corrupted	0.2	71.6	1.49	0.56	76.8	1.43	0.63
Corrupted	0.3	63.3	1.58	0.47	67.5	1.65	0.51

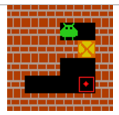
As shown in Table 7, generated beliefs achieve attribution accuracies of 87.1% on Sokoban and 90.3% on FrozenLake, compared with 94.2% and 96.4% for oracle beliefs. Increasing corruption reduces accuracy and SCR and worsens executed-action rank. At $p = 0.2$, accuracy remains 71.6% and 76.8%, respectively. These results show that the scorer retains useful attribution information under the tested moderate perturbations, while confirming that reward quality depends on belief quality. They do not establish end-to-end robustness to arbitrary perception errors, systematic misdescription, or missing task-relevant information.

E Environments

In this section, we introduce the information and the parameters of the environments that we use in the experiments. The selected tasks span a wide range of challenges, encompassing reasoning over both 2D symbolic layouts and 3D photorealistic scenes, and demonstrate adaptability to diverse embodiment settings, including household agents and game environments.

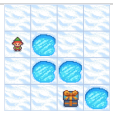
Sokoban is a classic grid-based puzzle in which the agent is required to push all boxes to target locations. The environment is a 2D grid with a discrete action space consisting of four movements (up, down, left, right). For Sokoban, we use a 6×6 grid as the standard setting for all models. Meanwhile, we construct a Hard setting of Sokoban to probe generalization on more demanding tasks, by increasing the map dimensions by 2. Detailed information is listed in Tab.8.

Table 8: Information about Sokoban Environment.

Name	Value
room size	6×6, 8×8
num of boxes	1
maximum turn number	4
action space	Up, Down, Left, Right
format reward	0.5
step penalty	-0.1
success reward	10
intrinsic reward weight α	0.1
example	

FrozenLake is a 2D grid-based map in which the agent need to reach a goal while avoiding holes. The visual state and discrete action space are similar to Sokoban. The default slippery dynamics are disabled to ensure deterministic transitions. For FrozenLake, we use 4×4 grid for all models. Detailed information is listed in Tab.9.

Table 9: Information about FrozenLake Environment.

Name	Value
room size	4×4
is slippery	False
maximum turn number	4
action space	Up, Down, Left, Right
format reward	0.5
step penalty	-0.1
success reward	10
intrinsic reward weight α	0.05
example	

Navigation is a 3D embodied navigation task in which the agent follows instructions to locate a target object. The agent perceives the environment from a first-person perspective and interacts through a discrete action space (e.g., `MoveAhead`). For Navigation, we evaluate on two eval sets: base and common sense. Detailed information is listed in Tab.10.

ManiSkill is also a 3D embodied manipulation benchmark where an agent controls a Panda robotic arm under stochastic dynamics. It features a hybrid action space that combines discrete action types with continuous parameters (e.g., `pick(x, y, z)`). Given third-person 3D observations, the agent must perform fine-grained visual grounding to map objects to precise coordinates, enabling accurate and physically executable interactions. Detailed information is listed in Tab.11. Since actions in this environment consist of two components, including a discrete action type and continuous spatial parameters, we adapt SCR to jointly infer both the action type $\hat{u}_t$ (i.e., pick, place, or push) and the continuous parameters $\hat{p}_t$ (i.e., spatial coordinates). The modified SCR enforces a strict type match via an indicator function $\mathbb{I}$, followed by a normalized spatial penalty (with $D_{\max}$ denoting the

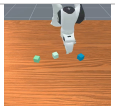
Table 10: Information about Navigation Environment.

Name	Value
resolution	255
success threshold	1.5
step distance	0.5
action space	MoveAhead, MoveBack, MoveRight, MoveLeft, RotateRight, RotateLeft, LookUp, LookDown
format reward	0.5
step penalty	-0.1
success reward	10
intrinsic reward weight α	0.05
example	

maximum distance):

$$r_t^i = \mathbb{I}(\hat{u}_t = u_t) \cdot \left(1 - \frac{\|\hat{p}_t - p_t\|_2}{D_{\max}}\right) \tag{16}$$

Table 11: Information about ManiSkill Environment.

Name	Value
action space	pick(x, y, z), place(x, y, z), push(x1, y1, z1, x2, y2, z2)
format reward	0.5
step penalty	-0.1
success reward	10
intrinsic reward weight α	0.05
example	

F Discussion and Future Direction

The Horizon of Causality. While RWM demonstrates strong efficacy in standard interactive settings, our analysis reveals a fundamental challenge of *causal ambiguity*. Our framework assumes that the transition (s_t, s_{t+1}) contains sufficient information to infer the underlying action. Importantly, we also account for more general settings where the temporal horizon between observations expands, such as multi-action turns in which an agent executes a sequence $\{a_1, \ldots, a_k\}$ before receiving o_{t+1}. In these cases, the correspondence between state changes and actions naturally becomes many-to-one. Consider a simple FrozenLake scenario in Fig. 8: the sequences `<Up, Left>` and `<Left, Up>` may lead to an identical final state configuration. If the agent executes the former while the retrospective world modeling infers the latter, the resulting intrinsic reward would incorrectly penalize a valid action, yielding a false negative signal, since the executed action is already fixed and unambiguous. This underscores a critical insight for the field of world modeling: *dense observation is a prerequisite for accurate causal grounding*. When observations are sparse, the ground truth becomes ambiguous.

To extend the horizon of causality, future work in world modeling must address this by moving beyond single-trajectory lookback to set-based retrospection, where the reward acknowledges the set of all physically valid causal paths rather than overfitting to a single deterministic history.

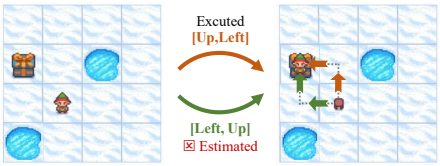


Figure 8: Case Study: Causal Ambiguity in Multi-Action Turns. Distinct action sequences can induce identical state transitions, posing challenges to the horizon of causality.

Limitation. While RWM significantly improves agent reasoning and decision-making, we acknowledge two primary limitations. First, in environments with *extreme stochastic dynamics*, the inverse transition mapping can become ambiguous (i.e., multiple distinct actions or random environmental perturbations may yield the exact same next observation). This causal ambiguity could introduce noise into the Self-Consistency Reward, which may require more advanced probabilistic modeling in future work. Second, scaling RWM to *extremely long-horizon tasks* currently poses practical hardware challenges. Training multi-turn VLM agents via RL is highly memory-intensive; extending the interaction to very long turns naturally encounters Out-Of-Memory (OOM) bottlenecks. Future engineering optimizations in memory-efficient RL (e.g., context compression or KV-cache offloading) should be explored to deploy this paradigm in longer, more complex scenarios.

G Case Study

We report some cases of the agent trained with our RWM, as shown in Tab.12 and Tab.13.

Table 12: Case study for our agent in a Sokoban environment.

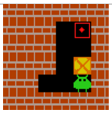
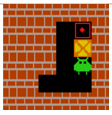
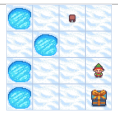
Name	Value
state_0	 (initial state)
agent	<think> <observation>The box is above and to the right of the player. The target is above the box. </observation> <reasoning>The player needs to go to the right first. </reasoning> <prediction>The player will be below the box. </prediction> </think> <answer>Right </answer>
state_1	
agent	<think> <observation>The box is directly above the player and the target is above the box. </observation> <retrospection>I think the scores of actions in [Up, Down, Left, Right] for causing the environment state transition is: [-1,0,0,5]. </retrospection> <reasoning>The player needs to go up to push the box. </reasoning> <prediction>The player will be below the box, the target will be above the box. </prediction> </think> <answer>Up </answer>
state_2	
agent	<think> <observation>The box is above the player and the target is directly above the box. </observation> <retrospection>I think the scores of actions in [Up, Down, Left, Right] for causing the environment state transition is: [5,0,-5,-5]. </retrospection> <reasoning>The player needs to go up to push the box up to the target. </reasoning> <prediction>The player will be below the box, the box will be at the same position of the target. </prediction> </think> <answer>Up </answer>
state_3	 (success)

Table 13: Case study for our agent in a FrozenLake environment.

Name	Value
state_0	 (initial state)
agent	<think> <observation>The player is to the left of and above the goal. There is a hole below and to the left of the player. </observation> <reasoning>I should go right first. </reasoning> <prediction>The player will be above of the goal. </prediction> </think> <answer>Right </answer>
state_1	
agent	<think> <observation>The player is above the goal. </observation> <retrospection>I think the scores for actions [Up, Down, Left, Right] are: [0,-5,-5, 5]. </retrospection> <reasoning>The goal is below me, so I should go down first. </reasoning> <prediction>The player will be above of the goal. </prediction> </think> <answer>Down </answer>
state_2	
agent	<think> <observation>The player is above the goal. </observation> <retrospection>I think the scores for actions [Up, Down, Left, Right] are: [-5 ,5,-1, -5]. </retrospection> <reasoning>The goal is below me, so I should go down. </reasoning> <prediction>The player will be above of the goal. </prediction> </think> <answer>Down </answer>
state_3	
agent	<think> <observation>The goal is below the player. </observation> <retrospection>I think the scores for actions [Up, Down, Left, Right] are: [-5,5,-5,-5]. </retrospection> <reasoning>The goal is below me, so I should go down. </reasoning> <prediction>The player will be at the same position of the goal. </prediction> </think> <answer>Down </answer>
state_4	 (success)

H Prompts

We report the prompts used in the training process, as shown in Tab. 14, Tab. 15, and Tab. 16.

Table 14: Prompt for Sokoban.

```
Prompt for Sokoban
----------
(Initial Turn)
----------
You are a Sokoban solver.
Your goal is to push all boxes onto targets.

Rules:
1.  Push boxes (can't pull).
2.  Avoid walls.

Actions you can take:  Up, Down, Left, Right.
You should take 1 action at a time.

Initial Observation:
<image>

Decide your next action.
You should take 1 action at a time.

You should first give the description of your observation, then your reasoning,
then predict the next state, and finally your answer.

For the content you provide within the ‘<observation>‘ and ‘<prediction>‘ tags,
you must strictly describe the relative position of the ‘target‘ and any visible
‘box‘ objects **relative to the player**.  Your description/prediction must
include **both** a vertical and a horizontal directional relationship for each
object.  Use ONLY the terms ‘above‘, ‘below‘, ‘left‘, ‘right‘, or ‘same‘ for
describing these relationships.

Your response should be in the format of:
<think><observation>...</observation><reasoning>...</reasoning><prediction>...
</prediction></think><answer>...</answer>

e.g.  <think><observation>The box is below the player and the target is
below and to the left of the player.</observation><reasoning>I need to
go down to push the box down to the target.</reasoning><prediction>The
player will be above the box, the target will be to the left of the
box.</prediction></think><answer>Down</answer>
----------
(N-th Turn, N>0)
----------
The new observation is:
<image>

Decide your next action.
You should take 1 action at a time.

First, you should first describe the current observation.  Second, using
this information, infer which action could plausibly have led to the current
observation.  Ignore and do not rely on your previous answer or intended action.
Your inference must be based only on the observed state and the state transition
from the previous step.  Then provide your reasoning, predict the next state,
and finally give your answer.
```

```
1) For the content you provide within the ‘<observation>‘ and ‘<prediction>‘
tags, you must strictly describe the relative position of the ‘target‘ and any
visible ‘box‘ objects **relative to the player**.  Your description/prediction
must include **both** a vertical and a horizontal directional relationship for
each object.  Use ONLY the terms ‘above‘, ‘below‘, ‘left‘, ‘right‘, or ‘same‘
for describing these relationships.
2) For the content you provide within the ‘<retrospection>‘, you must strictly
assign a score to each action in [Up, Down, Left, Right].  Output a vector of
length 4, with integer scores ranging from -5 to 5.  A higher score indicates a
higher likelihood that the action was taken to reach the current observation.

Your response should be in the format of:
<think><observation>...</observation><retrospection>...</retrospection>
<reasoning>...</reasoning><prediction>...</prediction></think>
<answer>...</answer>

e.g.  <think><observation>The box is below the player and the target is below
and to the left of the player.</observation><retrospection>I think the scores of
actions in [Up, Down, Left, Right] for causing the environment state transition
is:  [0,5,-1,-5].</retrospection><reasoning>I need to go down to push the box
down to the target.</reasoning><prediction>The player will be above the box, the
target will be to the left of the box.</prediction></think><answer>Down</answer>
```

Table 15: Prompt for FrozenLake.

```
Prompt for FrozenLake
-----------
(Initial Turn)
-----------
You are a FrozenLake solver.
Your task is to reach the goal and avoid falling into holes.

Actions you can take:  Left, Down, Right, Up.
You should take 1 action at a time.

Initial Observation:
<image>

Decide your next action.
You should take 1 action at a time.

You should first describe the observation, then your reasoning, then predict the
next state, and finally your answer.

For the content you provide within the ‘<observation>‘ and ‘<prediction>‘ tags,
you must strictly describe the relative position of the ‘target‘ (the gift
box) and any visible ‘hole‘ (blue circles) objects **relative to the player**.
Your description/prediction must include **both** a vertical and a horizontal
directional relationship for each object.  Use ONLY the terms ‘above‘, ‘below‘,
‘left‘, ‘right‘, or ‘same‘ for describing these relationships.

Your response should be in the format of:
<think><observation>...</observation><reasoning>...</reasoning><prediction>...
</prediction></think><answer>...</answer>

e.g.  <think><observation>The player is above and on the right side of target.
There is a hole below and at the left of the player.</observation><reasoning>I
should go down first.</reasoning><prediction>The player will be above
and on the right side of target.  There is a hole at the left of the
player.</prediction></think><answer>Down</answer>
-----------
```

```
(N-th Turn, N>0)
-----------
The new observation is:
<image>

Decide your next action.
You should take 1 action at a time.

First, you should first describe the current observation.  Second, using
this information, infer which action could plausibly have led to the current
observation.  Ignore and do not rely on your previous answer or intended action.
Your inference must be based only on the observed state and the state transition
from the previous step.  Then provide your reasoning, predict the next state,
and finally give your answer.

1) For the content you provide within the ‘<observation>‘ and ‘<prediction>‘
tags, you must strictly describe the relative position of the ‘target‘ and any
visible ‘box‘ objects **relative to the player**.  Your description/prediction
must include **both** a vertical and a horizontal directional relationship for
each object.  Use ONLY the terms ‘above‘, ‘below‘, ‘left‘, ‘right‘, or ‘same‘
for describing these relationships.
2) For the content you provide within the ‘<retrospection>‘, you must strictly
assign a score to each action in [Left, Down, Right, Up].  Output a vector of
length 4, with integer scores ranging from -5 to 5.  A higher score indicates a
higher likelihood that the action was taken to reach the current observation.

Your response should be in the format of:
<think><observation>...</observation><retrospection>...</retrospection>
<reasoning>...</reasoning><prediction>...</prediction></think>
<answer>...</answer>

e.g.  <think><observation>The player is on the right side of the goal.
There is a hole at the left of the player.</observation><retrospection>I
think the scores of actions in [Left, Down, Right, Up] for causing the
environment state transition is:  [0,5,-1,-5].</retrospection><reasoning>I
should go down first.</reasoning><prediction>The player will be on
the right side of the goal.  There is a hole at the left of the
player.</prediction></think><answer>Down</answer>
```

Table 16: Prompt for Navigation.

```
Prompt for Navigation
-----------
(Initial Turn)
-----------
You are a home robot and perform navigation tasks according to instructions.

Actions you can take:  moveahead, moveback, moveright, moveleft, rotateright,
rotateleft, lookup, lookdown.

The instruction will be provided with each observation.  Look at the image
carefully and navigate to complete the instruction.

Rules:
1.  You should take 1 action at a time.
2.  If you find yourself seems to be stuck, you can lookdown or lookup to see if
there’s any object above or below you, you can also rotate to see if there’s any
object behind you.
```

```
Initial Observation:
<image>

Human Instruction:
I am looking for a garbage can.  Can you navigate to that object?

Decide your next action.
You can take 1 action at a time.

You should first give your thought process with the your observation, reasoning,
and prediction of next state, then your answer.  Both the observation and
prediction should describe what you see or expect to see in the environment.

Your response should be in the format of:
<think><observation>...</observation><reasoning>...</reasoning><prediction>
...</prediction></think><answer>...</answer>

e.g.  <think><observation>There is a garbage can in the upper left corner
of the image.</observation><reasoning>To move to the garbage can, we can go
forward-left, but since there's a kitchen counter directly ahead, we should
go left first.  </reasoning><prediction> I will be in front of the garbage
can.</prediction></think> <answer>moveleft</answer>
-----------
(N-th Turn, N>0)
-----------
The new observation is:
<image>

Decide your next action.
You should take 1 action at a time.

First, you should first describe the current observation.  Second, using
this information, infer which action could plausibly have led to the current
observation.  Ignore and do not rely on your previous answer or intended action.
Your inference must be based only on the observed state and the state transition
from the previous step.  Then provide your reasoning, predict the next state,
and finally give your answer.

1) Both the observation and prediction should describe what you see or expect to
see in the environment.
2) For the content you provide within the '<retrospection>', you must strictly
assign a score to each action in [moveahead, moveback, moveright, moveleft,
rotateright, rotateleft, lookup, lookdown].  Output a vector of length 8,
with integer scores ranging from -5 to 5.  A higher score indicates a higher
likelihood that the action was taken to reach the current observation.

Your response should be in the format of:
<think><observation>...</observation><retrospection>...</retrospection>
<reasoning>...</reasoning><prediction>...</prediction></think>
<answer>...</answer>

e.g.  <think><observation>There is a garbage can in the upper left
corner of the image.  </observation> <retrospection> I think the scores
of actions in [moveahead, moveback, moveright, moveleft, rotateright,
rotateleft, lookup, lookdown] for causing the environment state transition is:
[0,-5,-1,5,-5,-5,-5,-5].</retrospection><reasoning> To move to the garbage can,
we can go forward-left, but since there's a kitchen counter directly ahead, we
should go left first.</reasoning><prediction> I will be in front of the garbage
can.</prediction></think><answer>moveleft</answer>
```

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: See our abstract and introduction.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: See Section. F.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

   Justification: Our paper does not include theoretical results

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The experimental details can be found in Sec. 5.1 and Sec. C.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

Justification: The code and data are not publicly available at the time of submission. We plan to release the implementation upon acceptance.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The experimental details can be found in Sec. 5.1 and Sec. C.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: As illustrated in Sec. C, we repeat 3 evaluations on 128 test cases generated from the environment and calculate the average to mitigate randomness.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: All experiments were conducted on 4× NVIDIA H800 (80GB) GPUs. (Sec. 5.1 and Sec. C)

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research presented in this paper complies with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: This paper focuses on foundational algorithmic research in reinforcement learning and world modeling for autonomous agents. As a generic optimization method evaluated on standard closed-environment benchmarks, it does not have a direct, immediate path to specific negative societal applications, nor does it involve sensitive data, fairness issues, or human subjects.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The trained models and environments utilized in this work pose no high risk for malicious misuse or dual-use, and therefore do not require specific safety filters or safeguards.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All existing base models and benchmark environments are properly cited in the main text and Appendix. We adhere to their respective open-source licenses for academic use.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [No]

    Justification: The code and data are not publicly available at the time of submission. We plan to release the implementation upon acceptance.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: This paper strictly involves computational experiments and does not involve any crowdsourcing or research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The research does not involve human subjects, making IRB approval inapplicable.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: Large Vision-Language Models (used as the agent policy) and LLMs (used as the judge) are central to our core methodology. Their specific usage, including model backbones, prompt designs, and optimization configurations, are documented in Sec.4 and Appendix.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.