Model-Based Imaginative Planning for Embodied Agents

Junru Song[1], Hengzhe Jin[1], Yucong Huang[1], Tingsong Jiang[2], Weien Zhou[2], Feifei Wang[3], Yang Yang[2]*, Ying Wen[1,4]*, Wen Yao[2]*

[1]Shanghai Jiao Tong University, [2]Intelligent Game and Decision Laboratory
[3]Renmin University of China, [4]Shanghai Innovation Institute
{songjunru,ying.wen}@sjtu.edu.cn, bigyangy@gmail.com, wendy0782@126.com

Abstract

Reasoning and planning critically rely on a predictive dynamics model. In symbolic domains such as mathematics and code, large language models (LLMs) internalize transition rules during pretraining, allowing reinforcement learning or test-time scaling to effectively elicit and generalize their reasoning ability. Embodied decision making is fundamentally different: agents must reason from sparse visual evidence under partial observability, while coping with environment-specific dynamics and affordances not captured by language priors. Here we propose **IMPLEMENT**, a model-based reasoning framework that enables frozen LLMs to perform *imaginative planning*. A lightweight world model converts raw pixels into object-centric symbolic states amenable to language-based reasoning, and predicts their evolution under hypothetical actions. To address epistemic uncertainty stemming from partial observability, we perform Monte Carlo state prediction via temperature sampling, enabling decision evaluation over multiple plausible futures. To support adaptation to unseen environments, we integrate Meta In-Context Learning, conditioning the world model on interaction history to continually refine its predictions. At inference time, the LLM and world model form a tight co-reasoning loop: the LLM proposes candidate actions, the world model simulates future trajectories, and the LLM refines its decisions, effectively inducing an online *policy iteration* scheme. Extensive experiments in ALFWorld demonstrate consistent advantages over finetuning-based and strong test-time scaling approaches, validating **IMPLEMENT** as an effective framework for grounding language agents in visual embodied environments.

1 Introduction

Reasoning and planning presuppose an accurate model of environment dynamics. In symbolic domains (*e.g.,* mathematics, code, and logical reasoning), transition rules are explicit, stable, and well represented in pretraining corpora, enabling large language models (LLMs) to internalize such dynamics during large-scale pretraining (Zhao et al., 2023; Naveed et al., 2025). Consequently, reinforcement learning or test-time scaling can directly elicit and generalize these latent reasoning capabilities to solve novel problems (Zhang et al., 2025b; Wang et al., 2024a; Guo et al., 2025). This paradigm, however, does not readily extend to embodied scenarios, where decision making is governed by noisy visual observations, implicit physical affordances, and environment-specific dynamics that deviate substantially from language priors.

Embodied environments introduce three fundamental challenges largely absent from symbolic reasoning. **First**, egocentric visual observations are high-dimensional yet contain sparse semantic signals; pixel-level reasoning is therefore brittle and noise-sensitive (Stone et al., 2021). **Second**, partial observability induces epistemic uncertainty: agents perceive only local regions and must gradually acquire information through interaction, causing uncertainty to compound over multi-step planning (Wang et al., 2023a). **Third**, embodied tasks exhibit persistent distribution shifts, as each instance presents novel spatial layouts, object configurations, and affordance structures, demanding rapid online adaptation from limited experience (Liu et al., 2023a; Yang et al., 2024c). Together, these factors render pretrained language priors insufficient for reliable long-horizon embodied planning (Chow et al., 2025; Liang et al., 2025).

Existing approaches pursue two primary strategies, neither fully addressing these challenges. Finetuning-based methods align vision-language models (VLMs) with embodied tasks via imitation or reinforcement learning, yet without explicit dynamics modeling, they fail to elicit generalizable embodied knowledge; moreover, they are computa-

* Corresponding Authors.

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 18125–18147

tionally expensive, risk catastrophic forgetting of general capabilities, and preclude deployment of frontier models (Lin et al., 2025; Hancock et al., 2025; Luo et al., 2025). Test-time scaling techniques (Wei et al., 2022; Wang et al., 2022; Silver et al., 2016) enhance reasoning coherence but still operate without explicit dynamics models, struggling to bootstrap reliable embodied planning. Recent work explores using pretrained LLMs themselves as world models for planning, yet such approaches rely heavily on internalized language priors that cannot accurately capture environment-specific dynamics or systematically represent uncertainty distributions (Deng et al., 2025; Yang et al., 2024a; Zhou et al., 2025). Consequently, they remain confined to simplified text-based environments rather than visually grounded physical settings. Critically, these methods treat the policy and world model as static, isolated modules, contrasting with human cognition where internal models co-evolve with decision-making through interaction (Ho et al., 2022; Qureshi et al., 2025).

In this work, we propose **IMPLEMENT**, a model-based reasoning framework driven by IMaginative Planning, empowering frozen Language models to become adaptive EMbodied agENTs. At its core lies a lightweight world model trained via self-supervised learning, which perceives visually but predicts symbolically. We adopt an object-centric state representation that extracts structured semantics from high-dimensional visual inputs, constructing a compact reasoning space. To handle partial observability and ambiguous outcomes, we perform Monte Carlo rollouts via temperature sampling, enabling agents to reason over diverse plausible futures. To support instance-specific adaptation, we integrate Meta In-Context Learning into world-model training, allowing predictions to be refined online using accumulated observations and failed interactions. Any frozen LLM can then leverage this model as an "imaginative mental world" to optimize decision-making.

At inference time, the world model and language agent form a *mutually reinforcing loop*: the agent utilizes the world model for informed, dynamics-aware decisions, while the world model continuously refines its predictive fidelity using the agent's interaction history. This iterative process naturally instantiates *generalized policy iteration*, where both components co-evolve in-context without parameter updates, delivering the promise to adapt any pretrained LLM into an optimal embodied policy. Extensive experiments in ALFWorld (Shridhar et al., 2020)–a challenging long-horizon embodied benchmark–demonstrate consistent advantage of **IMPLEMENT** over vanilla VLMs, finetuning-based methods, and recent test-time scaling approaches. Notably, **IMPLEMENT** exhibits superior sample efficiency under limited interaction budgets, highlighting its ability to rapidly adapt from sparse experience. These results validate **IMPLEMENT** as a general framework for grounding LLM reasoning in visual embodied environments while preserving the flexibility of pretrained models. Our code and data are available at `https://github.com/WoodySJR/IMPLEMENT`.

2 Related Work

LLMs for embodied sequential decision-making. LLMs have demonstrated remarkable prior knowledge and general-purpose reasoning, achieving human-level performances in complex question answering, code generation and multimodal understanding (Naveed et al., 2025). These advances have sparked growing interests in adapting LLMs to long-horizon, embodied sequential decision-making tasks, where agents must reason over extended action sequences to accomplish physically grounded objectives (Liang et al., 2025). To date, interactive text environments (Shridhar et al., 2020; Jansen, 2022; Wang et al., 2025a)) have served as one of the primary testbeds for building and benchmarking embodied agents, due to their simplicity, ease of development and fast simulation (Wang et al., 2023b,c; Liu et al., 2023b; Zhou et al., 2025). Nevertheless, agents developed in these abstract settings face substantial gaps when transferred to real-world embodied environments with noisy visual observations and nuanced environment dynamics. Fine-tuning approaches attempt to bridge this gap, yet they become increasingly costly with the scaling of language models and are brittle under distribution shifts (Shridhar et al., 2020; Yang et al., 2024b; Zhao et al., 2024). These limitations highlight the need for test-time scaling methods that empower frozen LLMs to adapt to varied embodied scenarios, grounding their decisions in environment dynamics while preserving their inherent generalization and reasoning abilities.

LLMs and world models. Recent works have started to explore the integration of LLMs with world models to compensate for the missing embodied knowledge. A "world model" generally

refers to a computational representation of the physical world capable of predicting state transitions given various actions (Ha and Schmidhuber, 2018). While Wang et al. (2025b); Chen et al. (2025), among others, explore model-based reinforcement learning for training embodied agents, these methods inherit the limitations of finetuning-based methods and necessitate meticulous reward design and state representation alignment between the policy and world model. More of recent works instead leverage zero-shot LLMs themselves as world models, combining them with planning algorithms such as Model Predictive Control and Monte-Carlo Tree Search to optimize trajectories (Deng et al., 2025; Yang et al., 2024a; Liu et al., 2023b; Hao et al., 2023; Zhou et al., 2024, 2025). While promising, these approaches are restricted to domains with clear semantics, such as math reasoning, web navigation and text games, and struggle to extend to visual embodied tasks. Meanwhile, in these systems, the LLMs used for policy generation and world modeling operate as decoupled modules, without co-evolving through iterative interaction.

3 Method

3.1 Problem Formulation

We consider a visually grounded embodied environment modeled as a partially observable Markov decision process (POMDP):

$$\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{G}, P, r, \gamma, T, \mathcal{O}, O), \quad (1)$$

where $s_t \in \mathcal{S}$ denotes the latent world state, $a_t \in \mathcal{A}$ an action, $o_t \in \mathcal{O}$ a high-dimensional observation (*e.g.*, a first-person RGB image) sampled from the observation kernel $O(o_t|s_t)$. $g \in \mathcal{G}$ represents the task goal, often specified in natural language. The environment dynamics are governed by the transition kernel $P(s_{t+1} \mid s_t, a_t)$ and reward function $r(s_t, a_t, s_{t+1} \mid g)$, with discount factor $\gamma \in (0, 1)$ and task horizon T.

The agent interacts with the environment through a history-dependent, goal-oriented policy:

$$\pi_\theta(a_t \mid h_t, g), \quad h_t = (o_0, a_0, \cdots, o_t) \quad (2)$$

to maximize the expected discounted return:

$$J(\pi) = \mathbb{E}\left[\sum_{t=0}^{T} \gamma^t r(s_t, a_t, s_{t+1} \mid g)\right]. \quad (3)$$

We focus on *high-level planning* where states can be expressed symbolically and actions correspond to semantically meaningful primitives (*e.g.*, "go to table 1", "pick up pen 1 from table 1"). Nevertheless, such decisions are still constrained by nuanced physical affordances and partial observability (see Appendix J for examples), where even strong LLMs can fall short. Meanwhile, low-level motor control is delegated to specialized controllers (*e.g.*, vision-language-action models). This is consistent with the widely adopted dual-process embodied architectures (Driess et al., 2023; Zhang et al., 2025a,d), where planning and execution are decoupled for tractability and modularity.

3.2 Symbolic World Model for Visual Embodied Environments

3.2.1 Object-Centric State Representation

We define a structured state representation as a dictionary comprising: (i) the agent's location, (ii) its inventory (currently held object), and (iii) receptacle- and object-level attributes and relations (*e.g.*, containment, open/closed, heated/cool). This symbolic representation follows the object-centric design of Chen et al. (2023), which has been shown to generalize well across a broad range of simulated and real-world planning scenarios. Importantly, it enables scalable state tracking for long-horizon tasks, abstracting away pixel-level noise while retaining the semantic structure required for LLM-based decision making. See Figure 1 and Appendix D for concrete examples.

To address partial observability, we explicitly seperate *perception* (extracting currently observable facts) from *prediction* (forecasting action outcomes). Concretely, a single world model M_ϕ operates in two modes:

- **Perception mode**, which incorporates new observation o_t and updates the symbolic state with deterministic, currently observable information.
- **Prediction mode**, which, given a candidate action (or action sequence), predicts *action-contingent* state updates (*e.g.*, newly revealed objects after navigation; attribute changes after object manipulation), without speculating unrelated parts of the world.

This yields a compact, decision-relevant predictive interface easier for an LLM planner to reason over, and mirrors human cognition: we reason over abstracted states rather than raw sensor streams, and simulate only what matters under a proposed intent.

Several design choices jointly promote robustness and scalability. First, perception is re-invoked

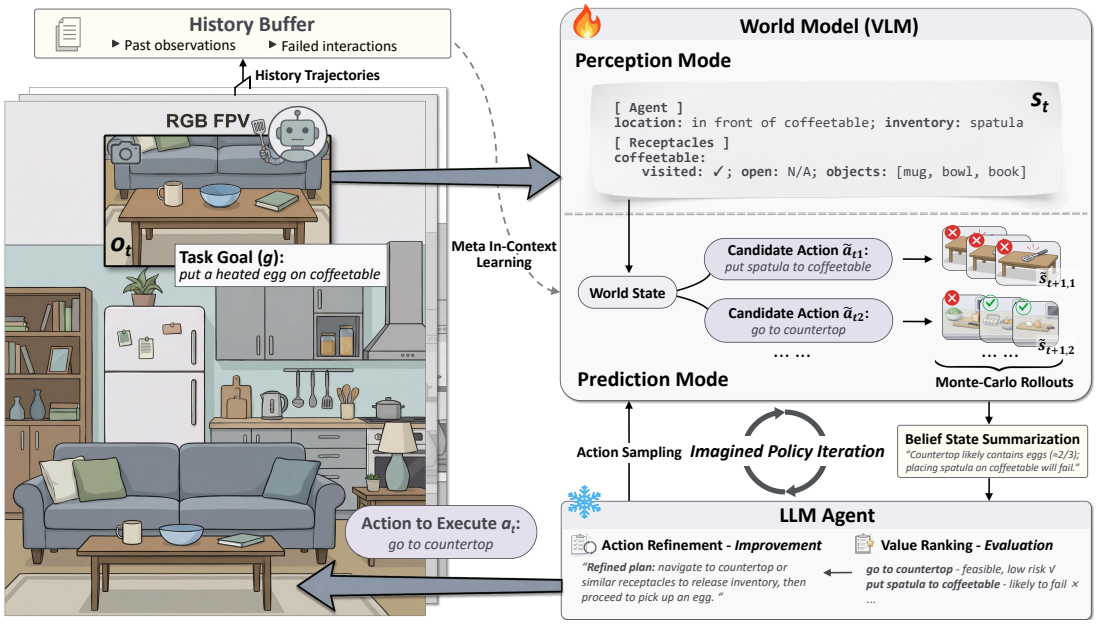


Figure 1: Overview of **IMPLEMENT**, an online model-based policy iteration framework. The agent grounds visual observations into a symbolic state and imagines action outcomes using a world model. The simulated Monte-Carlo rollouts are summarized into a belief state that guides iterative policy evaluation and refinement based on a frozen language model. A trajectory-level history buffer supports Meta In-Context Learning for test-time adaptation.

at every environment step, and the symbolic state is updated by overwriting relevant fields with the latest visual evidence rather than accumulating stale information; this continual re-estimation mechanism mitigates error propagation across steps. Second, the state explicitly tracks only confirmed, observed information—receptacles that have not yet been visited are marked as *unvisited* rather than populated with speculative content, preventing hallucinated knowledge from distorting downstream planning and reducing context length. Third, predictions are restricted to action-contingent state updates relevant to the proposed intent, avoiding unnecessary speculation about unrelated parts of the world. Finally, Monte Carlo rollouts are further compressed into concise belief-state summaries before being passed to the planner, keeping the reasoning context compact. Together, these mechanisms enhance both the reliability and the scalability of the framework.

3.2.2 World Model Training via Self-labeling Transitions

A key benefit of restricting the world model to perception and prediction rather than any decision making is that data collection becomes substantially simpler: we can learn from self-labeled environment transitions without reward annotation or high-quality expert demonstrations. Specifically, we collect trajectories using a lightweight behavior policy dominated by random exploration, while injecting a small amount of goal-directed interactions to mitigate distributional bias. Please refer to Appendix A.1 for details.

At each step, let the environment transition from state s_{t-1} to s_t after executing action a_t, with corresponding observations o_{t-1} and o_t. We instantiate M_ϕ by supervised-fine-tuning a pre-trained vision-language model using token-level cross-entropy on serialized symbolic states. We unify perception and prediction training within one model by conditioning on special mode tokens "[PER]" and "[PRE]":

$$\mathcal{L} = \lambda_1 \cdot \mathcal{L}_{\text{per}} + \lambda_2 \cdot \mathcal{L}_{\text{pre}},$$

$$\mathcal{L}_{\text{per}} = \text{CE}(s_t, M_\phi(s_{t-1}, a_{t-1}, o_t; [\text{PER}])),$$

$$\mathcal{L}_{\text{pre}} = \text{CE}(s_t, M_\phi(s_{t-1}, a_{t-1}; [\text{PRE}])),$$

where s_t is overloaded to denote symbolic state, CE denotes the cross-entropy loss, and we weight the two training objectives equally in practice.

3.2.3 MetaICL for Test-time Adaptation

While the learned world model captures dynamics shared across a class of tasks, individual environments often exhibit instance-specific characteristics, such as object layouts and inaccessible receptacles. Such environment-specific knowledge is difficult to encode in a static model, yet crucial for

reliable planning. To address this challenge, we equip the world model with the ability to adapt at test time by leveraging limited interaction history though Meta In-Context Learning (MetaICL; Min et al., 2022). Concretely, during prediction training, we augment each input with a trajectory context τ consisting of historical observations and failed actions from the same environment:

$$\mathcal{L}_{\text{pre}} = \text{CE}\big(s_t, M_\phi(s_{t-1}, a_t, \tau; [\texttt{PRE}])\big). \quad (4)$$

This training scheme encourages the model to treat the trajectory context as implicit evidence about environment-specific dynamics and affordances. At test time, the same conditioning mechanism enables M_ϕ to adapt its predictions online, effectively yielding a *self-evolving* world model. The construction of the trajectory context τ is detailed in Appendix D.

3.3 IMPLEMENT: Model-based Online Policy Iteration

We now introduce **IMPLEMENT**, a test-time scaling framework where a frozen LLM planner interacts with M_ϕ to progressively refine its decisions.

At each environment step, the world model first performs perception to update the symbolic state:

$$s_t \leftarrow M_\phi(s_{t-1}, a_{t-1}, o_t; [\texttt{PER}]). \quad (5)$$

Then the LLM and world model engage in iterative interactions, forming an "imagine-evaluate-refine" loop.

3.3.1 Action Sampling

Given goal g and interaction history $h_t = (s_0, a_0, \cdots, s_t)$ (here we overload h_t to denote symbolic traces), the LLM samples a set of candidate action sequences and outputs a `help` flag indicating whether model-based evaluation is needed:

$$\begin{gathered}(\mathcal{A}_t, \texttt{help}) \sim \pi_\theta(\cdot \mid g, h_t, \mathcal{E}_t), \\ \texttt{help} \in \{\texttt{True}, \texttt{False}\}.\end{gathered} \quad (6)$$

Here $\mathcal{A}_t = \{\tilde{a}_{t1}, \cdots, \tilde{a}_{tn}\}$, and each $\tilde{a}_{ti} = (a^1_{ti}, \cdots, a^H_{ti})$ is a candidate action sequence with planning horizon H. $\mathcal{E}_t$ is a memory that stores previously evaluated candidates and their predicted outcomes, initialized as $\varnothing$. If $\texttt{help} = \texttt{False}$, the agent commits to execution directly (*e.g.*, execute the top-ranked candidate); otherwise, we invoke world-model rollouts for deliberation.

3.3.2 Monte-Carlo Rollouts

When the LLM requests assistance, indicating low confidence, we evaluate each candidate sequence $\tilde{a}_{ti}$ by rolling it out in the learned world model:

$$\hat{s}^{(m)}_{t+H,i} \leftarrow M_\phi(s_t, \tilde{a}_{ti}, \tau; [\texttt{PRE}]),\ m = 1, \cdots, M. \quad (7)$$

We sample with *non-zero* temperature to obtain M diverse plausible futures, which are further summarized into a textual *belief state* $b_{t+H,i}$ before being appended into the memory:

$$\mathcal{E}_t \leftarrow \mathcal{E}_t \bigcup \{(\tilde{a}_{ti}, b_{t+H,i})\}_{i=1}^n. \quad (8)$$

3.3.3 Action Refinement

Given the updated memory $\mathcal{E}_t$, the LLM assesses and ranks the utility of evaluated candidates, after which it decides whether to commit to one of them or to further refine the actions through another round of model-based imagination. This tightly coupled interaction yields a *mutually reinforcing loop*: the LLM leverages model feedback to perform "imaginative planning" and make informed decisions, while the world model benefits from richer interaction history to improve its prediction.

3.3.4 Action Execution

The proposed framework focuses on high-level planning, requiring an additional execution layer to translate action primitives into low-level control signals. Here we assume the availability of low-level controllers or skill libraries that map each action into environment-specific actuation commands, such as joint torques, end-effector motions, or navigation controllers. Importantly, **IMPLEMENT** is agnostic to the specific implementation of low-level control, enabling high modularity.

3.4 Connection to Policy Iteration

We interpret **IMPLEMENT** as an instance of *generalized policy iteration* (GPI; Sutton et al., 1998) carried out at test time. Although no explicit parametric value function is learned, the interplay between model-based rollouts and LLM-based reasoning induces an *implicit value landscape* over candidate actions. This enables an alternating process between implicit policy evaluation and in-context policy improvement.

3.4.1 Model-based Evaluation

At iteration k, the LLM samples a finite set of candidate action sequences $\mathcal{A}_t$ of horizon H from an implicit action distribution $\pi^{(k)}_\theta$. For each candidate $\tilde{a} \sim \mathcal{A}_t$, the world model M_ϕ approximates the distribution over future trajectories via model-based rollouts. Conditioned on the predicted futures, the

LLM performs goal-conditioned reasoning to assess how well the resulting state transitions align with task objective. We summarize this evaluation process using a value-like functional:

$$\begin{aligned}&\hat{Q}^{(k)}(s_t, \tilde{a}_{ti} \mid g)\\&= \mathbb{E}_{\{s_{t+h}\}_{h=1}^{H} \sim M_\phi}\Big[\sum_{h=1}^{H} \gamma^{h-1} \hat{r}(s_{t+h-1}, a_{ti}^h, s_{t+h} \mid g)\\&+ \gamma^H \hat{V}(s_{t+H} \mid g)\Big], \qquad \tilde{a}_{ti} \in \mathcal{A}_t,\end{aligned} \tag{9}$$

where $\hat{r}$ and $\hat{V}$ do not correspond to explicit scalar reward or value functions, but instead abstract the LLM's internal comparative reasoning over candidate actions. Concretely, $\hat{r}$ captures local, step-wise assessments of task progress (*e.g.*, satisfaction of subgoals or avoidance of failed actions), while $\hat{V}$ reflects a holistic judgment of the desirability of the predicted terminal belief state. This essentially induces a *preference ordering* over candidate actions that is sufficient to drive policy improvement.

3.4.2 In-context Policy Improvement

Given evaluated candidates stored in $\mathcal{E}_t$, the LLM updates its decision in-context, analogous to:

$$\pi_\theta^{(k+1)}(s_t) \leftarrow \arg\max_{\tilde{a}} \hat{Q}^{(k)}(s_t, \tilde{a} \mid g). \tag{10}$$

This improvement step may either result in committing to the highest-ranked candidate or proposing refined actions for further evaluation. The `help` flag in Eq.6 serves as an adaptive termination criterion: when the LLM judges additional evaluation is unlikely to alter its decision, the iteration halts and the agent commits to execution. From an optimization perspective, this corresponds to reaching an approximate *fixed point* of the implicit evaluation and improvement operators defined by LLM-world-model interaction. We summarize **`IMPLEMENT`** with pseudocode in Appendix L.

4 Experiment

4.1 Experimental Setup

Environments. We conduct our experiments in ALFWorld (Shridhar et al., 2020), a cross-modality embodied simulation platform for long-horizon household tasks. ALFWorld provides paired visual environments instantiated in AI2-THOR and parallel textual interface which offers precise symbolic state annotations. The benchmark includes six task categories (Pick & Place, Clean & Place, Heat & Place, Cool & Place, Look in Light, and Pick Two Objects & Place), each requiring navigation and object interaction to complete a predefined language instruction (see Appendix I for examples). Tasks could involve reasoning and interaction over more than ten objects and require over 30 steps for a human expert to complete, enabling comprehensive evaluation of long-horizon planning, instruction following, and world knowledge grounding. Importantly, ALFWorld's visual environments represent realistic embodied settings rather than purely PDDL-based textual counterparts, demanding precise reasoning about physical affordances and spatial constraints. Following prior works, we evaluate on the 134 out-of-distribution tasks (Shridhar et al., 2020; Liu et al., 2023b; Yang et al., 2024b).

Baselines. We compare our approach against three categories of baselines. (i) **Vanilla VLMs**, where pretrained VLMs directly perform decision making by conditioning on the goal, interaction history, and FPV images. (ii) **Finetuning-based methods**: MiniGPT-4 (Zhu et al., 2023), BLIP-2 (Li et al., 2023), LLaMA-Adapter (Gao et al., 2023), InstructBLIP (Dai et al., 2023) and EMMA (Yang et al., 2024b), which finetune VLMs on task demonstrations. (iii) **Test-time scaling methods**: including ReAct (Yao et al., 2022), Reflexion (Shinn et al., 2023), Self-consistency (Wang et al., 2022) and SimuRA (Deng et al., 2025), which enhance LLM reasoning through special prompting techniques. Please refer to Appendix B for details.

4.2 Implementation Details

We instantiate the world model using Qwen2.5-VL-7B (Bai et al., 2025) and train it via supervised fine-tuning on approximately 80K environment transitions sampled from ALFWorld. In the training data, each object instance carries a unique numeric identifier assigned by the simulator in order of appearance (*e.g.*, "apple 1", "apple 2"); the world model thus learns to generate and maintain consistent instance IDs during state tracking and prediction. To evaluate the adaptability of **`IMPLEMENT`**, we instantiate the planner with multiple policy LLMs spanning different scales and reasoning styles, including both proprietary and open-source models. Low-level execution is handled by ALFWorld's built-in oracle controllers for all methods to ensure a fair comparison. We report success rates over 1, 6, and 12 trials, which jointly reflect zero-shot performance and the agent's ability to improve through online interaction. All metrics are averaged over

134 held-out tasks across six task categories for stable and representative assessment. To ensure reproducibility, detailed prompt templates and experiment configurations are provided in Appendix A.2, G and H.

4.3 Comparative Studies

Table 1 compares **IMPLEMENT** with baseline methods across different policy backbones and evaluation budgets. We have the following observations:

First, vanilla VLM agents consistently underperform across task categories and trial budgets. Although these models possess strong vision-language alignment, they struggle to accumulate and organize long-term visual evidence into coherent, goal-oriented world knowledge, often leading to myopic or unstable behaviors.

Second, finetuning-based approaches exhibit limited generalization when evaluated on out-of-distribution tasks. This indicates that heavy reliance on demonstration data without explicit dynamics modeling tends to overfit and reduce robustness to unseen environments.

Third, test-time scaling methods, including ReAct, Reflexion, and Self-consistency, frequently lag behind. While these approaches can stabilize reasoning trajectories that are already supported by LLM's internal priors, they lack explicit mechanisms to acquire, verify and update world knowledge through interaction. SimuRA further leverages pretrained LLMs as world models; however, it still relies heavily on language priors and lacks principled mechanisms to handle uncertainty in state transitions. Consequently, its predictions often reflect inaccurate understanding of physical dynamics and produce misleading signals that distort downstream decision making.

In contrast, **IMPLEMENT** without imaginative planning, where the agent directly commits to execution without deliberation, already achieves strong performance. This proves the effectiveness of the symbolic state representation and perception module in grounding high-level decisions. The full **IMPLEMENT** framework yields further performance gains, validating the importance of model-based policy iteration. These improvements are particularly pronounced in low-trial regimes, indicating that imaginative planning, combined with MetaICL, enables more efficient learning from limited interaction experience. Importantly, the observed performance gains are consistent across diverse LLM policy backbones. This suggests that **IMPLEMENT** complements the underlying reasoning capacity of various policy models and provides a general and robust mechanism for grounding LLM decision making in visual embodied environments.

4.4 Ablation Studies

4.4.1 Effect of MetaICL and Imaginative Planning

Figure 2 (a) illustrates the success rate as a function of interaction trials. While all variants benefit from additional trials, the full **IMPLEMENT** consistently outperforms its counterparts without MetaICL (*i.e.,* $\tau = \varnothing$ in Eq.7) and without imaginative planning across all interaction budgets, with the performance gap most pronounced in the early stage ($k \leq 5$). This indicates that these components effectively enable the world model, and in turn, the LLM agent, to evolve through limited experience.

Figure 2 (b) and (c) further analyze the effects of imaginative planning and MetaICL on planning efficiency and reliability. Removing imaginative planning results in slower task completion and a substantially higher ratio of action failure, indicating that without model-based imagination the agent struggles to reason about action feasibility. Disabling MetaICL also degrades performance, albeit to a lesser extent, suggesting that online adaptation from past trajectories provides additional benefits beyond static world models.

4.4.2 Effect of Temperature Sampling

Figure 2 (d) analyzes the impact of world-model sampling temperature on task success. When the temperature is set to zero, corresponding to single-point state prediction, performance degrades significantly. This indicates that Monte Carlo-style rollouts are crucial for capturing uncertainty in embodied environments with partial observability. Increasing the temperature from 1 to 1.5 leads to a slight drop in success rate, although not statistically significant. This suggests that overly high stochasticity may introduce excessive noise into imagined futures and also impair planning performances.

4.4.3 Effectiveness of LLM-Guided Termination

Figure 2 (e) analyzes the effectiveness of the LLM-guided termination mechanism (*i.e.,* the `help` flag in Eq.6) for policy iteration. We compare this adaptive strategy with a fixed termination baseline that enforces a constant number of iteration rounds (set to 5). The results show no statistically significant

Table 1: Comparison studies. **Bold** and underlined numbers indicate the best and second-best performance within each LLM policy group, respectively. Results for finetuning-based methods are directly taken from Yang et al. (2024b), and "-" denotes results that were not reported. Complete success rate curves are reported in Appendix C.

	Success Rate @ 12							Success Rate @ 6	Success Rate @ 1
Agent	**Pick**	**Clean**	**Heat**	**Cool**	**Look**	**Pick2**	**Avg.**	**Avg.**	**Avg.**
Finetuning-based agents									
MiniGPT-4 (Zhu et al., 2023)	0.04	0.00	0.19	0.17	0.67	0.06	0.16	–	–
BLIP-2 (Li et al., 2023)	0.00	0.06	0.04	0.11	0.06	0.00	0.04	–	–
LLaMA-Adapter (Gao et al., 2023)	0.17	0.10	0.27	0.22	0.00	0.00	0.13	–	–
InstructBLIP (Dai et al., 2023)	0.50	0.26	0.23	0.06	0.17	0.00	0.22	–	–
EMMA (Yang et al., 2024b)	0.71	0.94	0.85	0.83	0.88	0.67	0.82	0.58	0.19
Frozen-LLM agents									
GPT-4.1-mini (Achiam et al., 2023)									
vanilla	0.63	0.52	0.70	0.48	0.56	0.29	0.55	0.55	0.31
ReAct (Yao et al., 2022)	0.81	0.93	0.74	0.81	0.72	**0.79**	0.81	0.72	0.35
Reflexion (Shinn et al., 2023)	**0.89**	0.89	0.83	0.57	0.83	0.57	0.78	0.74	0.46
Self-consistency (Wang et al., 2022)	0.78	0.93	0.74	0.81	0.83	0.57	0.79	0.71	0.48
SimuRA (Deng et al., 2025)	0.81	0.81	0.57	0.57	0.83	0.64	0.72	0.62	0.33
IMPLEMENT w/o imaginative planning	0.85	0.93	0.74	0.81	0.89	0.57	0.82	0.76	0.46
IMPLEMENT (ours)	**0.89**	**0.96**	**0.87**	**0.86**	**0.94**	0.71	**0.88**	**0.86**	**0.66**
GPT-4.1 (Achiam et al., 2023)									
vanilla	0.56	0.70	0.74	0.67	0.72	0.57	0.66	0.65	0.41
ReAct (Yao et al., 2022)	0.74	0.93	0.61	0.76	0.83	**0.79**	0.78	0.73	0.48
Reflexion (Shinn et al., 2023)	0.81	0.93	0.78	**0.86**	0.78	0.64	0.82	0.81	0.60
Self-consistency (Wang et al., 2022)	0.70	0.89	0.74	**0.86**	0.72	0.64	0.77	0.75	0.60
SimuRA (Deng et al., 2025)	0.81	0.85	0.83	0.76	0.83	**0.79**	0.82	0.78	0.52
IMPLEMENT w/o imaginative planning	0.81	0.89	0.78	**0.86**	0.83	**0.79**	0.83	0.79	0.60
IMPLEMENT (ours)	**0.85**	**1.0**	**0.91**	0.62	**0.89**	0.71	**0.85**	**0.82**	**0.68**
Gemini-2.5-flash (Comanici et al., 2025)									
vanilla	0.74	0.74	0.78	0.38	0.67	0.29	0.63	0.59	0.34
ReAct (Yao et al., 2022)	**0.81**	0.93	0.57	0.52	0.83	0.79	0.75	0.72	0.48
Reflexion (Shinn et al., 2023)	**0.81**	0.93	0.83	0.67	0.83	0.50	0.78	0.78	0.66
Self-consistency (Wang et al., 2022)	0.74	0.89	0.78	0.52	0.83	0.57	0.74	0.72	0.62
SimuRA (Deng et al., 2025)	**0.81**	0.85	0.65	0.67	0.78	**0.86**	0.77	0.68	0.53
IMPLEMENT w/o imaginative planning	0.78	0.93	0.83	0.67	0.94	0.79	0.82	0.82	0.66
IMPLEMENT (ours)	**0.81**	**0.96**	**0.91**	**0.81**	**1.0**	0.79	**0.88**	**0.86**	**0.75**
Qwen2.5-VL-72b (Bai et al., 2025)									
vanilla	0.41	0.52	0.78	0.67	0.44	0.14	0.52	0.45	0.19
ReAct (Yao et al., 2022)	0.78	0.85	0.70	0.81	0.72	0.71	0.77	0.72	0.52
Reflexion (Shinn et al., 2023)	**0.89**	0.93	0.74	0.81	0.78	0.43	0.79	0.76	0.52
Self-consistency (Wang et al., 2022)	0.81	0.85	0.83	**0.86**	0.72	0.57	0.79	0.75	0.45
SimuRA (Deng et al., 2025)	0.74	0.93	0.78	0.81	0.83	0.57	0.79	0.76	0.63
IMPLEMENT w/o imaginative planning	0.85	0.89	0.74	**0.86**	0.78	0.71	0.82	0.78	0.52
IMPLEMENT (ours)	**0.89**	**0.96**	**0.91**	**0.86**	**0.89**	**0.79**	**0.89**	**0.85**	**0.74**

difference in success rates. Note that although the maximum number of iterations in **IMPLEMENT** is also capped at 5, the LLM terminates early in most cases, resulting in an average of only 2.4 iterations per environment step. This indicates that the LLM can reliably identify fixed points of policy iteration, yielding meaningful computational savings.

Additional efficiency analysis and performance analysis of the world model is delegated to Appendix F and E, respectively.

5 Conclusion

We present **IMPLEMENT**, a model-based reasoning framework that enables pretrained LLMs to act as adaptive embodied agents. **IMPLEMENT** introduces a symbolic world model that transforms raw visual observations into compact states and predicts their transitions. By combining Monte Carlo state prediction to handle epistemic uncertainty and Meta In-Context Learning to enable rapid adaptation to novel environments, the world model provides a continually improving substrate for decision making. At test time, the LLM and world model co-evolve through interaction: imagined futures guide policy refinement, while accumulated experience refines the world model itself–together inducing a generalized policy iteration process. Empirical results in ALFWorld demonstrate robust gains over finetuning-based and test-time scaling methods. Overall, our findings highlight the complementary roles of explicit world modeling and general reasoning for embodied decision making, representing an important step to systematically extend language agents beyond symbolic domains.

Several directions merit further investigation. **Efficiency**: while model-based planning yields favorable overall compute when accounting for faster task-level convergence (Appendix F), the per-step

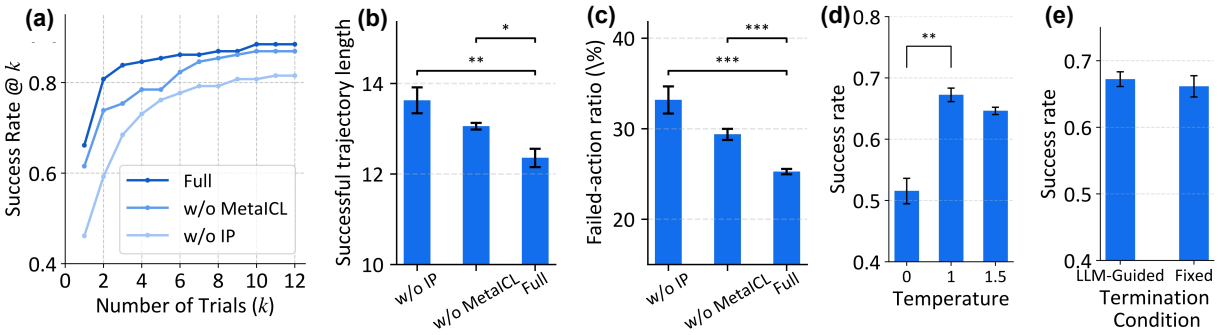


Figure 2: Ablation studies with GPT-4.1-mini as the policy. **(a)** Effect of MetaICL and imaginative planning (IP) on success rate across interaction trials. **(b)&(c)** Effect of MetaICL and IP on planning efficiency and reliability, respectively. **(d)** Effect of world model sampling temperature. **(e)** Effect of different termination criteria. The results in panel (b-e) are averaged over 12 repeated trials. Statistical significance is assessed using a two-tailed Mann-Whitney U test. Significance levels are denoted as $p < 0.05$ $(*)$, $p < 0.01$ $(**)$, and $p < 0.001$ $(***)$.

latency remains on the order of tens of seconds; system-level optimizations such as speculative decoding, model distillation, and adaptive rollout budgets are promising avenues. **Reducing privileged supervision**: equivalent state annotations could be obtained via pretrained open-vocabulary detectors, VLM-based captioning, or semi-automatic bootstrapping with limited human annotation, broadening applicability beyond environments with built-in symbolic engines (see Appendix N). **Perception robustness**: transferring to real-world environments with higher visual noise will require noise-robust state estimation techniques such as domain randomization (Tobin et al., 2017) and stronger object-centric visual pretraining. **Scalability of symbolic representations**: extending the fixed object-centric schema to open-vocabulary or dynamically constructed representations would enable handling more complex real-world tasks with unconstrained entities. **Broader evaluation**: validating `IMPLEMENT` on additional benchmarks with different action granularity and world dynamics would further strengthen the evidence for cross-environment generalization.

Limitations

Our work has several limitations. First, the world model is trained using supervised transitions, which constitutes a form of *privileged learning* commonly adopted in robotics (Pinto et al., 2017; Yamada et al., 2024; Wang et al., 2024b). While such a paradigm has been shown effective for sim-to-real transfer with limited real-world data, validating its effectiveness within our framework remains an important direction for future work. Meanwhile, imaginative planning incurs additional test-time computation due to model rollouts and interaction loops. In practice, we bound this overhead through explicit limits on interaction rounds and compact belief state summarization. Further improvements could be achieved through adaptive rollout allocation, early termination strategies, and careful consideration of performance-efficiency tradeoffs.

Acknowledgments

This work was supported by the National Key R&D Program of China (2024YFC3505402) and Intelligent Game and Decision Laboratory.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, and 1 others. 2025. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*.

Siwei Chen, Anxing Xiao, and David Hsu. 2023. Llm-state: Open world state representation for long-horizon task planning with large language model. *arXiv preprint arXiv:2311.17406*.

Zixuan Chen, Jing Huo, Yangtao Chen, and Yang Gao. 2025. Robohorizon: An llm-assisted multi-view world model for long-horizon robotic manipulation. *arXiv preprint arXiv:2501.06605*.

Wei Chow, Jiageng Mao, Boyi Li, Daniel Seita, Vitor Guizilini, and Yue Wang. 2025. Physbench: Benchmarking and enhancing vision-language models for physical world understanding. *arXiv preprint arXiv:2501.16411*.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, and 1 others. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*.

Wenliang Dai, Junnan Li, Dongxu Li, Anthony Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale N Fung, and Steven Hoi. 2023. Instructblip: Towards general-purpose vision-language models with instruction tuning. *Advances in neural information processing systems*, 36:49250–49267.

Mingkai Deng, Jinyu Hou, Zhiting Hu, and Eric Xing. 2025. Simura: A world-model-driven simulative reasoning architecture for general goal-oriented agents. *arXiv preprint arXiv:2507.23773*.

Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, and 1 others. 2023. Palm-e: an embodied multimodal language model. In *Proceedings of the 40th International Conference on Machine Learning*, pages 8469–8488.

Peng Gao, Jiaming Han, Renrui Zhang, Ziyi Lin, Shijie Geng, Aojun Zhou, Wei Zhang, Pan Lu, Conghui He, Xiangyu Yue, and 1 others. 2023. Llama-adapter v2: Parameter-efficient visual instruction model. *arXiv preprint arXiv:2304.15010*.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

David Ha and Jürgen Schmidhuber. 2018. World models. *arXiv preprint arXiv:1803.10122*, 2(3).

Asher J Hancock, Xindi Wu, Lihan Zha, Olga Russakovsky, and Anirudha Majumdar. 2025. Actions as language: Fine-tuning vlms into vlas without catastrophic forgetting. *arXiv preprint arXiv:2509.22195*.

Shibo Hao, Yi Gu, Haodi Ma, Joshua Hong, Zhen Wang, Daisy Wang, and Zhiting Hu. 2023. Reasoning with language model is planning with world model. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 8154–8173.

Mark K Ho, David Abel, Carlos G Correa, Michael L Littman, Jonathan D Cohen, and Thomas L Griffiths. 2022. People construct simplified mental representations to plan. *Nature*, 606(7912):129–136.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, and 1 others. 2022. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3.

Peter Jansen. 2022. A systematic survey of text worlds as embodied natural language environments. In *Proceedings of the 3rd Wordplay: When Language Meets Games Workshop (Wordplay 2022)*, pages 1–15.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR.

Wenlong Liang, Rui Zhou, Yang Ma, Bing Zhang, Songlin Li, Yijia Liao, and Ping Kuang. 2025. Large model empowered embodied ai: A survey on decision-making and embodied learning. *arXiv preprint arXiv:2508.10399*.

Bingqian Lin, Yunshuang Nie, Khun Loun Zai, Ziming Wei, Mingfei Han, Rongtao Xu, Minzhe Niu, Jianhua Han, Liang Lin, Cewu Lu, and 1 others. 2025. Evolvenav: Self-improving embodied reasoning for llm-based vision-language navigation. *arXiv e-prints*, pages arXiv–2506.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. 2023a. Libero: Benchmarking knowledge transfer for lifelong robot learning. *Advances in Neural Information Processing Systems*, 36:44776–44791.

Zhihan Liu, Hao Hu, Shenao Zhang, Hongyi Guo, Shuqi Ke, Boyi Liu, and Zhaoran Wang. 2023b. Reason for future, act for now: A principled framework for autonomous llm agents with provable sample efficiency. *arXiv preprint arXiv:2309.17382*.

Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. 2025. An empirical study of catastrophic forgetting in large language models during continual fine-tuning. *IEEE Transactions on Audio, Speech and Language Processing*.

Sewon Min, Mike Lewis, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2022. Metaicl: Learning to learn in context. In *Proceedings of the 2022 conference of the North American chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2791–2809.

Humza Naveed, Asad Ullah Khan, Shi Qiu, Muhammad Saqib, Saeed Anwar, Muhammad Usman, Naveed Akhtar, Nick Barnes, and Ajmal Mian. 2025. A comprehensive overview of large language models. *ACM Transactions on Intelligent Systems and Technology*, 16(5):1–72.

Lerrel Pinto, Marcin Andrychowicz, Peter Welinder, Wojciech Zaremba, and Pieter Abbeel. 2017. Asymmetric actor critic for image-based robot learning. *arXiv preprint arXiv:1710.06542*.

Rizwan Qureshi, Ranjan Sapkota, Abbas Shah, Amgad Muneer, Anas Zafar, Ashmal Vayani, Maged Shoman, Abdelrahman Eldaly, Kai Zhang, Ferhat Sadak, and 1 others. 2025. Thinking beyond tokens:

From brain-inspired intelligence to cognitive foundations for artificial general intelligence and its societal impact. *arXiv preprint arXiv:2507.00951*.

Neil Rabinowitz, Frank Perbet, Francis Song, Chiyuan Zhang, S. M. Ali Eslami, and Matthew Botvinick. 2018. Machine theory of mind. In *International Conference on Machine Learning*, pages 4218–4227. PMLR.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 36:8634–8652.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. 2020. Alfworld: Aligning text and embodied environments for interactive learning. *arXiv preprint arXiv:2010.03768*.

David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, and 1 others. 2016. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489.

Austin Stone, Oscar Ramirez, Kurt Konolige, and Rico Jonschkowski. 2021. The distracting control suite–a challenging benchmark for reinforcement learning from pixels. *arXiv preprint arXiv:2101.02722*.

Richard S Sutton, Andrew G Barto, and 1 others. 1998. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge.

Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. 2017. Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 23–30. IEEE.

Andrew Wang, Andrew C Li, Toryn Q Klassen, Rodrigo Toro Icarte, and Sheila A McIlraith. 2023a. Learning belief representations for partially observable deep rl. In *International Conference on Machine Learning*, pages 35970–35988. PMLR.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. 2023b. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291*.

Haonan Wang, Junfeng Sun, Xingdi Yuan, Ruoyao Wang, and Ziang Xiao. 2025a. Bytesized32refactored: Towards an extensible interactive text games corpus for llm world modeling and evaluation. *arXiv preprint arXiv:2509.23979*.

Jun Wang, Meng Fang, Ziyu Wan, Muning Wen, Jiachen Zhu, Anjie Liu, Ziqin Gong, Yan Song, Lei Chen, Lionel M Ni, and 1 others. 2024a. Openr: An open source framework for advanced reasoning with large language models. *arXiv preprint arXiv:2410.09671*.

Junqiao Wang, Zhongliang Yu, Dong Zhou, Jiaqi Shi, and Runran Deng. 2024b. Vision-based deep reinforcement learning of uav autonomous navigation using privileged information. *arXiv preprint arXiv:2412.06313*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

Yucen Wang, Rui Yu, Shenghua Wan, Le Gan, and De-Chuan Zhan. 2025b. Founder: Grounding foundation models in world models for open-ended embodied decision making. In *Forty-second International Conference on Machine Learning*.

Zihao Wang, Shaofei Cai, Guanzhou Chen, Anji Liu, Xiaojian Ma, and Yitao Liang. 2023c. Describe, explain, plan and select: Interactive planning with large language models enables open-world multi-task agents. *arXiv preprint arXiv:2302.01560*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Jun Yamada, Marc Rigter, Jack Collins, and Ingmar Posner. 2024. Twist: Teacher-student world model distillation for efficient sim-to-real transfer. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9190–9196. IEEE.

Chang Yang, Xinrun Wang, Junzhe Jiang, Qinggang Zhang, and Xiao Huang. 2024a. Evaluating world models with llm for decision making. *arXiv preprint arXiv:2411.08794*.

Yijun Yang, Tianyi Zhou, Kanxue Li, Dapeng Tao, Lusong Li, Li Shen, Xiaodong He, Jing Jiang, and Yuhui Shi. 2024b. Embodied multi-modal agent trained by an llm from a parallel textworld. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 26275–26285.

Yue Yang, Fan-Yun Sun, Luca Weihs, Eli VanderBilt, Alvaro Herrasti, Winson Han, Jiajun Wu, Nick Haber, Ranjay Krishna, Lingjie Liu, and 1 others. 2024c. Holodeck: Language guided generation of 3d embodied ai environments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16227–16237.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. 2022. React: Synergizing reasoning and acting in language models. In *The eleventh international conference on learning representations*.

Ning Zhang, Yongjia Zhao, Minghao Yang, and Shuling Dai. 2025a. Llms augmented hierarchical reinforcement learning with action primitives for long-horizon manipulation tasks. *Scientific Reports*, 15(1):36779.

Qiyuan Zhang, Fuyuan Lyu, Zexu Sun, Lei Wang, Weixu Zhang, Wenyue Hua, Haolun Wu, Zhihan Guo, Yufei Wang, Niklas Muennighoff, and 1 others. 2025b. A survey on test-time scaling in large language models: What, how, where, and how well? *arXiv preprint arXiv:2503.24235*.

Shao Zhang, Xihuai Jiang, Wenhao Zhao, Weinan Zhang, and Ying Wen. 2025c. Leveraging dual process theory in language agent framework for real-time simultaneous human-ai collaboration. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4081–4108.

Shao Zhang, Xihuai Wang, Wenhao Zhang, Chaoran Li, Junru Song, Tingyu Li, Lin Qiu, Xuezhi Cao, Xunliang Cai, Wen Yao, and 1 others. 2025d. Leveraging dual process theory in language agent framework for real-time simultaneous human-ai collaboration. *arXiv preprint arXiv:2502.11882*.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, and 1 others. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 1(2).

Zhonghan Zhao, Wenhao Chai, Xuan Wang, Boyi Li, Shengyu Hao, Shidong Cao, Tian Ye, and Gaoang Wang. 2024. See and think: Embodied agent in virtual environment. In *European Conference on Computer Vision*, pages 187–204. Springer.

Siyu Zhou, Tianyi Zhou, Yijun Yang, Guodong Long, Deheng Ye, Jing Jiang, and Chengqi Zhang. 2024. Wall-e: World alignment by rule learning improves world model-based llm agents. *arXiv preprint arXiv:2410.07484*.

Siyu Zhou, Tianyi Zhou, Yijun Yang, Guodong Long, Deheng Ye, Jing Jiang, and Chengqi Zhang. 2025. Wall-e 2.0: World alignment by neurosymbolic learning improves world model-based llm agents. *arXiv preprint arXiv:2504.15785*.

Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. 2023. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*.

A World Model Training Details

A.1 Data Collection

We adopt a mixed behavioral policy for data collection, which consists of the following types of actions:

- **Random admissible actions (π_{adm})**, which successfully transit the world into a different state but might not align with the task goal. These interactions are intended to equip the world model with knowledge of world dynamics.

- **Random inadmissible actions (π_{inadm})**, which violate physical affordances and thus bring no change to world states. These actions are sampled from admissible actions accumulated from *previous* interactions and include rich physical affordances for the world model to learn from.

- **Expert actions (π_{exp})**, which are obtained from rule-based policies. However, we note that these expert policies have not been specifically optimized and are reported to achieve a mere success rate of 60-70%. Meanwhile, these actions are not used for behavioral cloning, but to correct the biased data distribution from random interaction: some specific actions (*e.g.*, `heat`) must be preceded by a sequence of particular actions (*e.g.*, picking up a heatable object and then going to a microwave) in order to take effect, and are thus underrepresented by random policies.

During each step in data collection, the action is sampled from a mixture of these policies:

$$a_t \sim p_1 \cdot \pi_{\text{adm}}(\cdot \mid s_{t-1}) + p_2 \cdot \pi_{\text{inadm}}(\cdot \mid s_{t-1}) + p_3 \cdot \pi_{\text{exp}}(\cdot \mid s_{t-1}), \tag{11}$$

where $p_1 = p_2 = 0.4,\ p_3 = 0.2$.

The admissible and expert actions are both derived from hand-coded rules prebuilt in the ALFWorld simulation environment (Shridhar et al., 2020). These sampling heuristics can be further replaced with language agents for autonomous data collection procedure, thus forming a fully closed-loop system where LLMs actively construct a world model through exploration and rely on it for subsequent decision making. We leave this intriguing direction for future work.

Table 2: World Model Training Hyperparameters.

Hyperparameter	Value
Epochs	3
Learning rate	2×10^{-5}
Number of device	4×NVIDIA A800 80GB
Per-device train batch size	4
Gradient accumulation step	4
Max gradient norm	0.2
LoRA rank (r)	16
LoRA scaling (α)	16
LoRA dropout	0.05
LoRA target modules	all-linear
Optimizer	AdamW (torch_fused)
AdamW β_1	0.9
AdamW β_2	0.999
Weight decay	0.01

A.2 Training Hyperparameters

To ensure reproducibility, the hyperparameters of world model training are listed in Table 2.

B Detailed Introduction to Baselines

We compare our approach against three categories of baseline methods:

- **Vanilla VLMs** In this setting, pretrained vision-language models (GPT-4.1, GPT-4.1-mini, Gemini-2.5-Flash, and Qwen2.5-VL-72B) are directly prompted with the task goal, interaction history, and first-person-view (FPV) images to generation actions. To ensure context efficiency, FPV inputs are truncated to the most recent five frames.

- **Finetuning-based approaches** This category includes MiniGPT-4 (Zhu et al., 2023), BLIP-2 (Li et al., 2023), LLaMA-Adapter (Gao et al., 2023), InstructBLIP (Dai et al., 2023) and EMMA (Yang et al., 2024b). These methods fine-tune VLMs using task demonstrations collected either from human experts (the former four methods) or from powerful LLMs with access to privileged environmental information (EMMA). For these baselines, we directly report the experimental results from Yang et al. (2024b).

- **Test-time-scaling approaches** We consider ReAct (Yao et al., 2022), Reflexion (Shinn et al., 2023), Self-consistency (Wang et al., 2022), and SimuRA (Deng et al., 2025), which improve the reasoning ability of frozen LLM agents through test-time prompting and self-feedback mechanisms. For Reflexion, we use a memory buffer of 1 so that the agent reflects on its most immediate experience. For Self-consistency, we sample $n = 10$ responses and select actions based on majority voting. For Reflexion and Self-consistency, we provide the same perceived world state as `IMPLEMENT`. For ReAct, since observations are interleaved with actions and intermediate thoughts, to avoid excessively long contexts, we instead provide privileged textual observations from AI2-THOR. We note that SimuRA is also a world-model-based method, but relies entirely on pretrained LLMs for world modeling and performs planning via depth-first search, without online self-evolution or adaptation. As SimuRA was originally proposed for web navigation, where HTML serves as observation, and does not incorporate a vision encoder, we likewise provide privileged textual observations and evaluate its performance under the assumption of perfect perception.

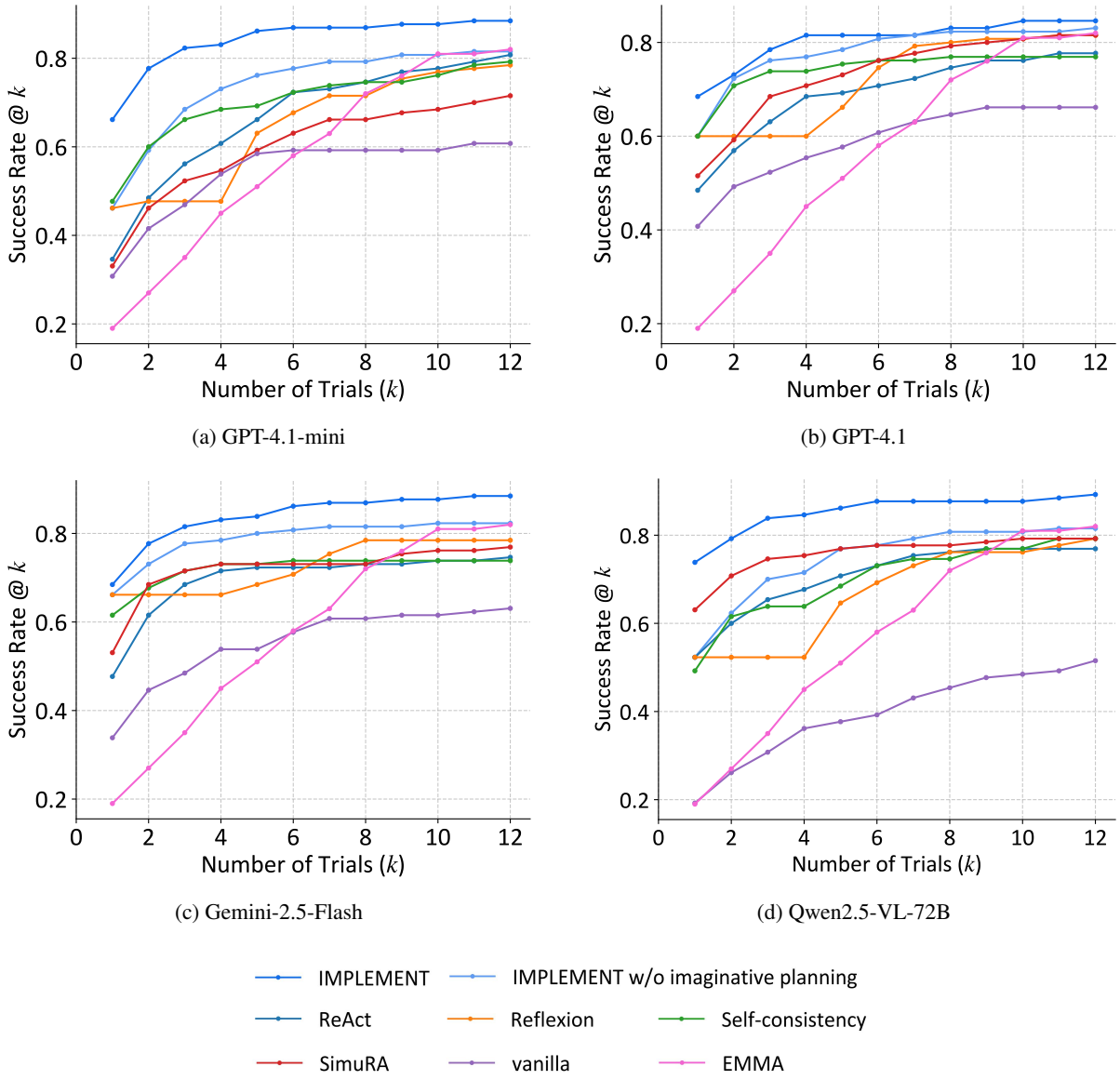
C Complete Success Rate Curves Across Policy Backbones

This section provides a detailed visualization of success rate curves over interaction trials for all policy backbones evaluated in our experiments. Each policy backbone corresponds to a subplot in Figure 3 that compares `IMPLEMENT` against vanilla VLM, test-time scaling methods, and the most competitive finetuning-based baseline (EMMA). These curves complement the results in main text by illustrating in greater detail the adaptation dynamics of various methods under increasing trial budgets and the consistent advantage of `IMPLEMENT`.

D Symbolic State Update and History Conditioning

For the purpose of illustration, here we provide an example of symbolic state representation and its update (both generated by the world model in its perception mode) in Figure 4, where the agent locates an apple on a countertop and then heats it with a microwave.

The conditioning context used for Meta In-Context Learning consists of two components. The first is a set of **history observations**, including receptacles visited in previous trajectories and their associated object lists. The second component captures **failed interactions**, where each failure is represented by the executed action, the agent's



(a) GPT-4.1-mini

(b) GPT-4.1

(c) Gemini-2.5-Flash

(d) Qwen2.5-VL-72B

Figure 3: Success rate curves ($k = 1$ to 12) across policy backbones.

location, its inventory, and the state of the target receptacle prior to the action. These failure cases implicitly encode knowledge about environment dynamics and affordances, such as action preconditions, object-receptacle compatibility, and state-dependent constraints, which are critical for refining future predictions and decisions. To prevent excessive context length during both training and test time, we only include failed actions that are relevant to the current decision step, defined as those sharing at least one common attribute with previous failures. We note that more sophisticated retrieval mechanisms, such as vector-based similarity matching, may further improve the accuracy and relevance of the conditioning context, which we leave for future work.

E Performance Analysis of World Model

Table 3 reports the state prediction accuracy of the learned world model on out-of-distribution tasks. In particular, we focus on two complementary aspects of state prediction that are critical for embodied planning: (i) predicting the presence of unobserved objects in the environment, and (ii) predicting whether a candidate action would fail and thus induce no state change.

For unobserved object prediction, conditioning on historical trajectories substantially improves both recall and precision compared to the unconditional variant. Notably, this improvement holds even when the target receptacle has not been directly explored before, suggesting that the world model captures a sense of *joint distribution* over

Table 3: Prediction accuracy of the learned world model. For object prediction, we report Recall / Precision.

Unobserved Object Prediction		**Failed Action Prediction**	
w/ condition	w/o condition	w/ condition	w/o condition
70.61 / 89.13	56.00 / 71.21	95.30	62.96

Table 4: Perception accuracy of the world model on object recognition.

Object Recall (%)	**Object Precision (%)**
90.40	93.76

state components, rather than treating different parts of the environment as independent fragments. As a result, information acquired from one part of the environment can inform predictions in others, enabling a form of generalization that goes beyond local observations and providing empirical support for the effectiveness of MetaICL.

We further observe a pronounced gain in failed action prediction when conditioning on past failed interactions. In this setting, an action failure corresponds to a no-op transition, where the predicted future state remains identical to the previous state. The strong improvement under conditioning indicates that incorporating historical failures enables the world model to better infer environment affordances and action feasibility, which is critical for downstream planning and action refinement.

Table 4 evaluates the perception capability of the world model, focusing on object recognition accuracy. We primarily report object-level recall and precision, as we find object identification constitutes the most challenging component of state abstraction, while other state attributes (e.g., receptacle status or agent location) are comparatively easier to infer. All metrics are computed on 134 held-out evaluation tasks that are excluded from the world model training set. We note that object recognition performance does not reach 100%, and recognition errors often arise when target objects occupy only a small region in the image, are partially occluded, or appear under unfavorable viewpoints. Improving perception accuracy remains a promising direction for further performance gains. For instance, incorporating pretraining on dedicated object detection datasets could enhance robustness to small-scale and partially visible objects. In addition, introducing dynamic viewpoint adjustment mechanisms—where the agent actively changes its observation angle or distance based on perception confidence—may further mitigate ambiguity in visual inputs and lead to more reliable state grounding.

F Efficiency Analysis

Here we perform an efficiency analysis of **`IMPLEMENT`** in terms of time and token costs, using GPT-4.1-mini as the policy backbone. The world model, instantiated with Qwen2.5-VL-7B, is deployed via vLLM with tensor-parallel inference across four NVIDIA A800 GPUs, with no parameter quantization or model compression applied. Table 5 reports the average runtime of each operation in our framework. For each environment step, the full model performs one perception update, followed by an average of 1.4 LLM-world-model interactions. This results in an average per-step computation time of:

$$2.16 + 5.33 + 1.4 \times (5.33 + 3.05 + 3.12) = 23.59 \text{ seconds.}$$

In contrast, the variant without imaginative planning performs only a single perception update and one round of action sampling per step, yielding a per-step runtime of:

$$2.16 + 5.33 = 7.49 \text{ seconds.}$$

While the full **`IMPLEMENT`** incurs higher computational costs per environment step, it achieves substantially faster task-level convergence. Specifically, as seen in Figure 2 (a), the full variant requires only 3 trials to surpass the success rate of the variant without imaginative planning over 12 trials. Accounting for this difference, the effective computational costs to reach similar task performances are approximately:

$3 \times 23.59 = 70.77$ (Full) **vs.** $12 \times 7.49 = 89.88$ (w/o IP).

Hence, in terms of per-step computation, imaginative planning does not incur higher time costs. Meanwhile, we note that beyond wall-clock computation, reducing the number of interaction trials confers additional practical advantages. For instance, in embodied environments, failed trials often incur non-negligible costs, such as safety-related penalties. Resetting the environment between trials also typically introduces extra latency. From this perspective, achieving *robust* task completion with substantially fewer trials is inherently preferable.

Table 5: Average runtime of individual operations measured on the 134 OOD tasks in ALFWorld.

Operation	Time per Call (s)
Policy Action Sampling	5.33
World Model Perception	2.16
World Model Prediction	3.05
Belief State Summarization	3.12

We further analyze token consumption incurred by API-based LLM calls. Since the world model is locally deployed, we only account for the token usage of policy action sampling and belief state summarization. On average, the variant without imaginative planning consumes 2,716.62 tokens per action sampling call. In contrast, the full **IMPLEMENT** consumes 3,393.58 tokens per sampling call, reflecting the additional context introduced by evaluated candidates and belief states. Each belief state summarization incurs an average of 2,629.76 tokens. Taking the aforementioned task-level convergence speed into consideration, imaginative planning only incurs 8.83% higher token cost.

G Implementation Details

World model training. We instantiate the world model using Qwen2.5-VL-7B (Bai et al., 2025) and train it via supervised fine-tuning. We sample approximately 80K environment transitions from 50% of training tasks provided in ALFWorld, each yielding paired perception and prediction targets. Training is performed with Low-Rank Adaptation (Hu et al., 2022), consuming 40 A800 GPU hours.

Policy LLMs. In order to examine the adaptability of **IMPLEMENT**, We evaluate it across multiple language models that differ in scale and reasoning style, including proprietary LLMs (GPT-4.1, GPT-4.1-mini, Gemini-2.5-Flash) and open-sourced models (Qwen2.5-VL-72B). Belief state summarization is consistently performed using GPT-4.1-mini to ensure cost efficiency. For low-level control, we utilize ALFWorld's built-in oracle controllers across all compared methods.

System deployment. After training, the world model is deployed on the server via FastAPI while the ALFWorld visual environments are executed locally. All the policy LLMs are accessed through APIs. We set the maximum number of LLM-world-model interaction rounds to 5 per environment step, sample 8 candidate actions at each planning iteration, and use a planning horizon of 1 following prior work (Zhou et al., 2024; Deng et al., 2025). The task horizon is capped at 30 steps, consistent with standard ALFWorld evaluation. LLM temperatures are set as 1.0, and the world model conducts 10 Monte Carlo rollouts to capture uncertainty. To control computational overhead, we further impose an upper bound of 5 on the number of LLM–world-model interaction rounds per environment step.

Evaluation metrics. We evaluate performance using success rates measured over 1, 6, and 12 trials, which quantify both zero-shot task performance and the agent's ability to progressively improve through interactions. This evaluation protocol, widely adopted in the literature, is particularly well-suited for challenging embodied tasks, where effective agents must rapidly adapt from failures and leverage experience to solve long-horizon problems (Liu et al., 2023b; Yang et al., 2024b). We note that the reported success rates are derived from a single evaluation run due to the considerable computation cost. Nevertheless, the results are averaged over 134 evaluation tasks that span six diverse categories, which we find already substantially helps mitigate randomness.

H Full Prompts

H.1 Prompt for Action Generation

```
 You are a text adventure game player, who interacts
  with a household environment to complete a task.

The following is a list of actions that can be used
in the game. You must strictly follow the syntax of
the actions in your answer.
1. go to (receptacle): move to a receptacle
2. open (receptacle): open a receptacle
3. close (receptacle): close a receptacle
4. take (object) from (receptacle): take an object
from a receptacle
5. put (object) to (receptacle): place an object in
your inventory into or onto a receptacle
6. use (object): turn on the object, which is
typically a light source like a desklamp or a
floorlamp
7. heat (object) with (receptacle): heat an object
with a receptacle.
8. cool (object) with (receptacle): cool an object
with a receptacle.
9. clean (object) with (receptacle): clean an object
 with a receptacle.

The following is your task goal.
{goal}

The following is your history interactions.
{history}

The following is the current state of the game.
Specifically, the state of the environment is
described as a dictionary with the following keys:
- "agent_location": your current location in the
environment;
- "inventory": the object you are holding (if any);
- "world_state": a dictionary describing the state
of each receptacle in the room, with the following
keys:
    - "visited": whether you have examined the
    objects in/on the receptacle;
```

```
    - "open": whether the receptacle is open; This
    is only applicable to openable receptacles;
    - "objects": a list of objects in the receptacle
    .
Current state: {current_state}

A world model is available that can predict the
state after an action is executed. Here are a few
actions, along with their predicted future states:
{evaluations}
When proposing actions, you have only two options:
Option 1. Choose an action from the above provided
actions to execute (must output "False" in "help").
    - This option is available only when any of the
    provided actions' predicted future state is
    helpful for completing the task.
    - This option is not available when there is no
    action provided above.
    - You must EXPLICITLY rank the helpfulness of
    the above provided actions according to their
    predicted future states in your <think> section,
     and then output the most promising one in <
    action>.
Option 2. Propose up to 8 different actions (must
output "True" in "help").
    - This option is feasible when none of the above
     provided actions' future state is
    helpful enough for completing the task.
    - In this case, you should propose up to 8
    DIFFERENT actions that are promising and worth
    trying.
    - The actions you propose will not be executed.
    Instead, their predicted future states will be
    returned to you to evaluate their helpfulness.

Your answer must be formatted as:
<think>
your reasoning process
</think>
<action>
action proposal(s) separated by commas
</action>
<help>
whether need world model to evaluate the action(s)
</help>.
```

H.2 Prompt for Belief State Summarization

```
A text adventure game player is interacting with a
household environment to complete a task. The
following is the current state of the environment.
Specifically, the state of the environment is
described as a dictionary with the following keys:
- "agent_location": the location of the agent in the
 environment;
- "inventory": the object the agent is holding (if
any);
- "world_state": a dictionary describing the state
of each receptacle in the room, with the following
keys:
    - "visited": whether the agent has examined the
    objects in/on the receptacle;
    - "open": whether the receptacle is open; This
    is only applicable to openable receptacles (
    drawer, cabinet, fridge, microwave, safe, etc.);
    - "objects": a list of objects in the receptacle
    .

Current state: {current_state}

Here is an action that the agent has proposed.
Action: {action}

The following are one or more possible future states
 of the environment after the action is executed,
predicted by a world model.
Your task is to summarize the predicted future
states into a coherent belief state that better aids
 the game player's decision.
You should focus on and clearly describe the changes
 in the future state compared to the current state (
such as the locations of agent and objects, the
states of objects and receptacles, etc.), and how
likely the changes are to happen (based on their
frequencies in the predicted results).
Importantly, your summary must be neutral and
objective, without any suggestions or speculations
beyond the predicted results, or any judgment
regarding the helpfulness of the action. Limit your
answer to within 150 words.

Predicted future states: {predicted_future_states}
```

H.3 Prompt for World Model Perception

```
[PER]
You are a world model that perceives and tracks the
state of a household environment.

State of this environment is described as a
dictionary with the following keys:
- "agent_location": the location of the agent in the
 environment;
- "inventory": the object the agent is holding (if
any);
- "world_state": a dictionary describing the state
of each receptacle in the room, with the following
keys:
    - "visited": whether the agent has examined the
    objects in/on the receptacle;
    - "open": whether the receptacle is open; This
    is only applicable to openable receptacles;
    - "objects": a list of objects in the receptacle
    .

Now you will be given the following information:
- action: an action that the agent has executed in
the environment;
- action_success: whether the action was
successfully executed;
- old_state: the state of the environment before the
 action;
- new_frame: the first-person view of the
environment after the action;
- other_visible_receptacles: for "go to" actions,
you will also be given the names of other visible
receptacles in the new frame, and you should update
the object list of these receptacles as well.

Your task is to update and return the state of the
environment in the right format based on the above
information. No verbal explanation is needed.

action: {action}
action_success: {action_success}
old_state: {old_state}
other_visible_receptacles: {
other_visible_receptacles}

The new_frame is provided right after the textual
prompt.
```

H.4 Prompt for World Model Prediction

```
[PRE]
You are a world model that predicts the state of a
household environment after an action.

State of this environment is described as a
dictionary with the following keys:
- "agent_location": the location of the agent in the
 environment;
- "inventory": the object the agent is holding (if
any);
- "world_state": a dictionary describing the state
of each receptacle in the room, with the following
keys:
    - "visited": whether the agent has examined the
    objects in/on the receptacle;
    - "open": whether the receptacle is open; this
    is only applicable to openable receptacles;
    - "objects": a list of objects in the receptacle
    .

You are given the following information:
- action: an action that the agent will execute in
the environment;
```

```
- old_state: the state of the environment before the
 action.

Your task is to predict the state of the environment
 after the action and return it in the right format.
 No verbal explanation is needed.

action: {action}
old_state: {old_state}

Here are observations from previous trajectories (
receptacles and object lists):
{condition_1}

Here are failed actions from previous trajectories (
action, agent location, inventory, state of the
target receptacle):
{condition_2}
```

I ALFWorld Task Examples

The following descriptions describe the processes involved in each of the six task types in ALFWorld, which are directly taken from the original paper (Shridhar et al., 2020). Please find visualized task examples in Figure 5.

- **Pick & Place** (e.g., "put a plate on the coffee table"): The agent must find an object of the desired type, pick it up, find the correct location to place it, and put it down there.
- **Examine in Light** (e.g., "examine a book under the lamp"): The agent must find an object of the desired type, locate a light source, and turn it on while holding the desired object.
- **Clean & Place** (e.g., "clean the knife and put it in the drawer"): The agent must find an object of the desired type, pick it up, go to a sink or a basin, wash the object by turning on the faucet, then find the correct location to place it, and put it down there.
- **Heat & Place** (e.g., "heat a mug and put it on the coffee table"): The agent must find an object of the desired type, pick it up, go to a microwave, heat the object by turning on the microwave, then find the correct location to place it, and put it down there.
- **Cool & Place** (e.g., "put a cool bottle on the countertop"): The agent must find an object of the desired type, pick it up, go to a fridge, put the object inside the fridge to cool it, then find the correct location to place it, and put it down there.
- **Pick Two & Place** (e.g., "put two pencils in the drawer"): The agent must find an object of the desired type, pick it up, find the correct location to place it, put it down there, then look for another object of the desired type, pick it up, return to the previous location, and put it down there together with the other object.

J Examples of Physical Affordances in ALFWorld

As illustrated in Figure 6, the visual environments in ALFWorld exhibit a range of physical affordance constraints that go beyond purely text-based environments, motivating the need for predictive modeling of action feasibility and state transitions.

K Potential Risks

Our paper focuses on improving decision-making of large language models in simulated embodied environments. Potential risks mainly stem from inaccurate world-model predictions or misinterpretation of physical affordances, which could lead to unsafe actions if directly transferred to real robotic systems. We emphasize that our framework is evaluated only in simulation, and extending it to real-world settings would require careful validation and safety constraints.

L Pseudo code

We summarize our proposed method, **IMPLEMENT**, in Algorithm 1.

M Belief State Summarization Example

To illustrate the full reasoning pipeline of **IMPLEMENT**, we provide a concrete example of how symbolic state, world model prediction, belief state summarization, and policy reasoning interact during a single planning step.

Setting. The agent is at the middle of the room with an empty inventory. The task goal is: *"find a mug and put it on the desk."*. The candidate actions include: `go to desk 1`, `go to desk 2`, `go to drawer 1`, `open drawer 1`, `go to drawer 2`, `open drawer 2`.

World model predictions. We show 3 of 10 Monte Carlo samples for the candidate action `go to desk 1`:

```
Sample 1: {"agent_location": "desk 1",
 "inventory": null, "world_state":
 {"desk 1": {"visited": true, "objects":
 ["alarmclock 1", "creditcard 1",
 "pencil 1"]}, "drawer 1": {"visited":
 false, "open": false, "objects": []},
```

Algorithm 1 `IMPLEMENT`: Model-based Online Policy Iteration

Require: Environment $\mathcal{M}$, goal g; policy LLM π_θ; world model M_ϕ with modes `[PER]` and `[PRE]`; planning horizon H and task horizon T; MC rollouts M; max interaction rounds R per step; history trajectories τ.

Initialize interaction history $h_0 = \varnothing$; initialize s_{-1} and a_{-1} as dummy placeholders for notational convenience.
for $t = 0, 1, \ldots, T$ **do**
  Observe o_t from the environment.
  Perception: $s_t \leftarrow M_\phi(s_{t-1}, a_{t-1}, o_t; \texttt{[PER]})$ ▷ update symbolic state
  Update interaction history $h_{t+1} \leftarrow h_t \cup \{(a_{t-1}, s_t)\}$
  Initialize evaluation memory $\mathcal{E}_t \leftarrow \varnothing$.
  for $r = 1, 2, \ldots, R$ **do**
    Action sampling: $(\mathcal{A}_t, \texttt{help}) \sim \pi_\theta(\cdot \mid g, h_t, \mathcal{E}_t)$
    if `help` = `False` or $r = R$ **then**
      Select $a_t \leftarrow \textsc{Commit}(\mathcal{A}_t)$ ▷ e.g., pick top-ranked action
      break
    end if
    for each candidate action $\tilde{a} \in \mathcal{A}_t$ **do**
      for $m = 1, 2, \ldots, M$ **do**
        Prediction: $\hat{s}_t^{(m)} \leftarrow M_\phi(s_{t-1}, \tilde{a}, \tau; \texttt{[PRE]})$ ▷ sampled with temperature
      end for
      Belief state summarization: $b(\tilde{a}) \leftarrow \textsc{Summarize}(\{\hat{s}_t^{(m)}\}_{m=1}^{M})$
      $\mathcal{E}_t \leftarrow \mathcal{E}_t \cup \{(\tilde{a}, b(\tilde{a}))\}$
    end for
  end for
  Execute: apply a_t in the environment and receive next observation.
end for

```
"drawer 2": {"visited": false, "open":
false, "objects": []}, "desk 2":
{"visited": false, "objects": []},
... other receptacles omitted ...}}

Sample 2: {"agent_location": "desk 1",
"inventory": null, "world_state":
{"desk 1": {"visited": true, "objects":
["pencil 2", "cd 1", "book 1"]},
"drawer 1": {"visited": false, "open":
false, "objects": []}, "drawer 2":
{"visited": false, "open": false,
"objects": []}, "desk 2": {"visited":
false, "objects": []},
... other receptacles omitted ...}}

Sample 3: {"agent_location": "desk 1",
"inventory": null, "world_state":
{"desk 1": {"visited": true, "objects":
["laptop 1"]}, "drawer 1": {"visited":
false, "open": false, "objects": []},
"drawer 2": {"visited": false, "open":
false, "objects": []}, "desk 2":
{"visited": false, "objects": []},
... other receptacles omitted ...}}
```

Belief state summarization. The Monte Carlo samples are compressed into a concise natural-language summary:

> *After the agent goes to desk 1, the agent's location updates from "middle of the room" to "desk 1," and desk 1's "visited" status changes to True. The objects visible on desk 1 differ across predicted states but consistently include at least one of the following: pencil (4 of 10), book (3 of 10), alarmclock (2 of 10), creditcard (2 of 10), cd (2 of 10), and laptop (1 of 10).* ***No mug is predicted on desk 1.*** *All other receptacles remain unvisited and unchanged. Inventory stays empty.*

Policy reasoning conditioned on belief state. The LLM planner reasons over the summarized belief states for all candidate actions:

> *Task is to put some mug on desk. Currently, the inventory is empty and the agent is at the middle of the room. Among the provided actions, going to desk 2 is predicted to reveal a mug (mug 1), which directly helps identify a mug. Going to desk 1 also reveals objects but no mugs are predicted, and drawers rarely contain mugs. Therefore, "go to desk 2" is the most promising action for finding a mug.*

As shown, the belief state distills distributional information across sampled futures (*e.g.*, which objects are likely or unlikely to appear, and with what frequency) into a concise natural-language summary, bridging the gap between raw structured predictions and the LLM planner's reasoning.

N Discussion on Privileged Learning and Scalability of Symbolic State Representations

Privileged learning. The world model in `IMPLEMENT` is trained on symbolic state transitions derived from ALFWorld's built-in symbolic engine, which provides ground-truth object-centric state annotations. This constitutes a form of *privileged learning*—a paradigm widely adopted in robotics for efficiently bootstrapping perception and dynamics models (Pinto et al., 2017; Yamada et al., 2024; Wang et al., 2024b). While such privileged supervision is typically unavailable in open real-world settings, we emphasize that it is used here for *experimental convenience*—to assess the planning mechanism without conflating it with the separate challenge of learning state abstraction from scratch—rather than being a fundamental requirement of the framework. The core contribution of this work is the test-time imaginative planning mechanism (IP + MetaICL), which is *orthogonal* to the data collection and supervision strategy.

In practice, equivalent state annotations could be obtained through several alternative pipelines that do not rely on a built-in symbolic engine: (i) *automatic labeling* via pretrained open-vocabulary object detectors or VLM-based captioning systems, which can extract object identities, locations, and attributes from raw images; (ii) *semi-automatic*

bootstrapping, where a small set of human annotations is used to train an initial state extractor that is then scaled via self-training on unlabeled interaction data; and (iii) *sim-to-real transfer*, leveraging simulator-provided symbolic data as pretraining priors whose learned abstractions are subsequently adapted to real-world environments with limited fine-tuning. These alternatives make the framework applicable beyond environments with built-in symbolic engines.

Scalability of symbolic representations. The current implementation employs a fixed, object-centric schema sufficient for the structured household environments in ALFWorld, where objects belong to a closed set of categories and carry a bounded number of attributes. We acknowledge that real-world environments with unconstrained action spaces, open-vocabulary entities, and high object density would require more flexible representations. The linear text serialization of symbolic states could grow substantially in dense scenes, potentially consuming a large portion of the planner's context window or diluting its attention.

Several directions show promise for addressing these scalability concerns. *Neuro-symbolic representations*—combining neural perception with compositional symbolic structure—can accommodate open-vocabulary entity discovery while retaining the compositional reasoning advantages of symbolic interfaces. In fact, our current approach already takes a step in this direction by using a VLM to generate symbolic states from visual inputs; extending this so that the schema itself can be dynamically constructed rather than predefined is a natural next step. In addition, *state-pruning or forgetting mechanisms*—such as filtering objects by task relevance, recency, or spatial proximity—could further improve scalability in cluttered environments. We also note that more complex real-world tasks may involve other agents with volition and internal states; world models will need to develop suitable representations of these actors, connecting to the growing body of work on machine Theory of Mind (Rabinowitz et al., 2018; Zhang et al., 2025c).

O Supplementary Experimental Results

O.1 Results with Qwen3.5-Plus and SimuRA Symbolic-State Variant

To further validate the generalizability of **IMPLEMENT** and isolate the contribution of our explicit world-modeling mechanism, we report additional experiments using Qwen3.5-Plus (qwen3.5-plus-2026-02-15)—a latest-generation LLM distinct from the four backbones evaluated in the main paper—as the policy model. In addition, we evaluate a variant of SimuRA (Deng et al., 2025) that consumes the same symbolic state input produced by our world model's perception module (denoted "SimuRA (WM Percep.)"), while keeping SimuRA's planning procedure unchanged. This aligns the perception interface and enables a direct comparison focused on differences in transition modeling and planning. All results below are SR@1 (success rate on the first trial) averaged over the full set of 134 evaluation tasks.

Table 6: SR@1 on 134 OOD tasks with Qwen3.5-Plus as the policy backbone.

Agent	SR@1
IMPLEMENT w/ IP (ours)	**67.9%** (91/134)
IMPLEMENT w/o IP	58.9% (79/134)
SimuRA (WM Percep.)	44.8% (60/134)

The results confirm two key findings: (i) Imagined Policy Iteration provides consistent gains (+9.0%) that are not substitutable by increased backbone capacity; and (ii) under aligned perception, **IMPLEMENT** outperforms SimuRA by 23.1 percentage points, attributable solely to differences in world modeling and planning (*i.e.*, our explicit learned dynamics model with uncertainty-aware Monte Carlo rollouts vs. SimuRA's LLM-prior-based simulation). The fact that these advantages hold with a new backbone further demonstrates that **IMPLEMENT** generalizes across LLM families and versions.

O.2 Multi-Seed Variance Estimation

To assess the stability of our results, we report the mean and standard deviation of SR@1 across 12 repeated runs for all frozen-LLM agents in Table 7.

Across all four policy backbones, **IMPLEMENT** consistently achieves the highest average SR@1 while exhibiting comparable or smaller standard deviations, indicating stable and reliable performance across repeated runs.

Table 7: SR@1 mean (std) across 12 repeated runs for all frozen-LLM agents. **Bold** indicates the best average performance within each policy backbone group.

Agent	Pick	Clean	Heat	Cool	Look	Pick2	Avg.
GPT-4.1-mini							
Vanilla	0.37 (0.06)	0.30 (0.04)	0.42 (0.06)	0.19 (0.08)	0.26 (0.08)	0.10 (0.06)	0.29 (0.02)
ReAct	0.68 (0.05)	0.54 (0.07)	0.16 (0.05)	0.22 (0.07)	0.22 (0.10)	0.45 (0.08)	0.40 (0.02)
Self-consistency	0.53 (0.05)	0.66 (0.05)	0.52 (0.08)	0.38 (0.07)	0.56 (0.08)	0.29 (0.11)	0.51 (0.04)
SimuRA	0.60 (0.07)	0.31 (0.08)	0.14 (0.07)	0.08 (0.08)	0.31 (0.06)	0.19 (0.10)	0.29 (0.02)
IMPLEMENT w/o IP	0.56 (0.07)	0.62 (0.09)	0.49 (0.05)	0.37 (0.09)	0.52 (0.15)	0.23 (0.11)	0.49 (0.03)
IMPLEMENT (ours)	**0.66** (0.04)	**0.78** (0.04)	**0.72** (0.07)	**0.73** (0.04)	**0.65** (0.07)	**0.40** (0.09)	**0.68** (0.02)
GPT-4.1							
Vanilla	0.36 (0.06)	0.34 (0.06)	0.50 (0.07)	0.40 (0.08)	0.38 (0.08)	0.17 (0.08)	0.37 (0.04)
ReAct	0.65 (0.04)	0.84 (0.03)	0.14 (0.05)	0.26 (0.07)	0.33 (0.09)	0.57 (0.07)	0.48 (0.03)
Self-consistency	0.52 (0.04)	0.76 (0.05)	0.31 (0.10)	0.79 (0.04)	0.69 (0.03)	0.33 (0.13)	0.58 (0.03)
SimuRA	0.72 (0.04)	0.56 (0.09)	0.20 (0.09)	0.33 (0.09)	0.47 (0.11)	0.46 (0.12)	0.47 (0.03)
IMPLEMENT w/o IP	0.60 (0.07)	0.75 (0.04)	0.40 (0.10)	0.80 (0.06)	0.71 (0.05)	0.27 (0.13)	0.61 (0.03)
IMPLEMENT (ours)	**0.68** (0.06)	**0.85** (0.07)	**0.78** (0.06)	**0.50** (0.07)	**0.81** (0.04)	**0.42** (0.05)	**0.69** (0.03)
Gemini-2.5-Flash							
Vanilla	0.50 (0.06)	0.27 (0.06)	0.44 (0.07)	0.23 (0.06)	0.50 (0.09)	0.04 (0.06)	0.35 (0.03)
ReAct	0.71 (0.05)	0.75 (0.06)	0.17 (0.07)	0.22 (0.07)	0.54 (0.09)	0.59 (0.08)	0.51 (0.03)
Self-consistency	0.63 (0.06)	0.75 (0.05)	0.73 (0.04)	0.47 (0.04)	0.63 (0.05)	0.38 (0.09)	0.62 (0.02)
SimuRA	0.63 (0.18)	0.61 (0.23)	0.24 (0.10)	0.25 (0.19)	0.51 (0.21)	0.48 (0.25)	0.46 (0.18)
IMPLEMENT w/o IP	0.62 (0.05)	0.77 (0.04)	0.72 (0.04)	0.46 (0.08)	0.78 (0.07)	0.39 (0.11)	0.64 (0.02)
IMPLEMENT (ours)	**0.72** (0.04)	**0.85** (0.04)	**0.79** (0.09)	**0.61** (0.06)	**0.81** (0.04)	**0.43** (0.10)	**0.72** (0.03)
Qwen2.5-VL-72B							
Vanilla	0.15 (0.06)	0.13 (0.05)	0.36 (0.09)	0.31 (0.09)	0.17 (0.05)	0.01 (0.03)	0.20 (0.03)
ReAct	0.71 (0.07)	0.77 (0.03)	0.16 (0.07)	0.44 (0.09)	0.31 (0.05)	0.57 (0.05)	0.51 (0.02)
Self-consistency	0.50 (0.03)	0.71 (0.06)	0.48 (0.09)	0.68 (0.04)	0.56 (0.09)	0.09 (0.13)	0.53 (0.03)
SimuRA	0.64 (0.08)	0.65 (0.10)	0.58 (0.11)	0.63 (0.06)	0.54 (0.12)	0.24 (0.07)	0.57 (0.06)
IMPLEMENT w/o IP	0.55 (0.05)	0.71 (0.05)	0.52 (0.09)	0.73 (0.06)	0.57 (0.07)	0.25 (0.10)	0.58 (0.03)
IMPLEMENT (ours)	**0.71** (0.08)	**0.81** (0.06)	**0.80** (0.07)	**0.63** (0.09)	**0.80** (0.03)	**0.42** (0.12)	**0.72** (0.03)

Figure 4: Example of state representation and update, when the agent locates and heats an apple.

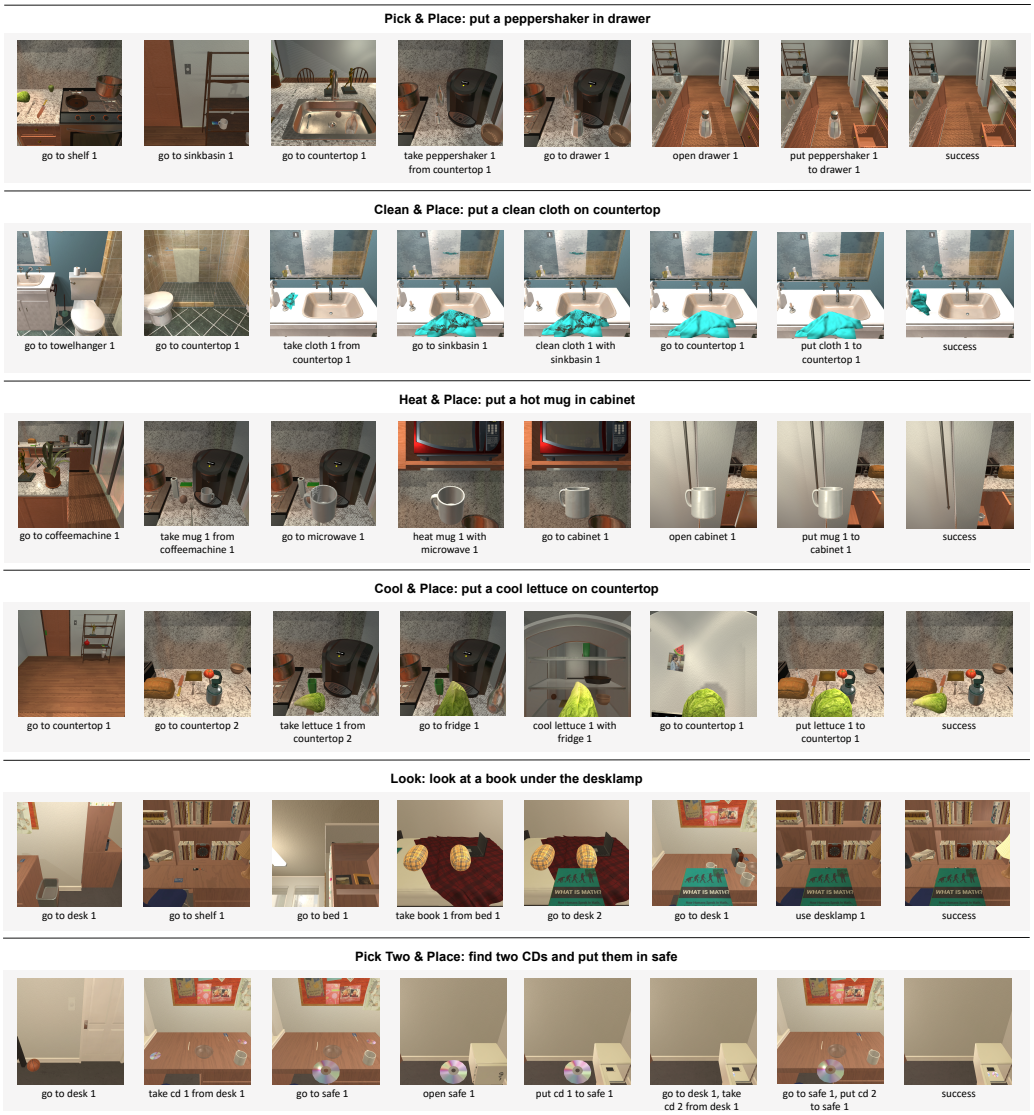


Figure 5: Visualized task examples of visual embodied environments in ALFWorld (Shridhar et al., 2020).

Spatial Reachability Constraint
Inadmissible action: take cd 1 from desk 1
Reason: the agent's current location differs from the target receptacle (desk 1). It must first navigate to the corresponding location before any interaction.

Inventory Capacity Constraint
Inadmissible action: take soapbar 1 from toilet 1
Reason: The inventory is already occupied. Pick-up actions will fail unless the agent places the held object (toiletpaper) onto a valid receptacle beforehand.

Receptacle Occupancy Constraint
Inadmissible action: clean cloth 1 with sinkbasin 1
Reason: certain interactions require an empty or specific receptacle state. For example, if the sinkbasin already contains a soap bar, the agent must remove it before cleaning a cloth.

Object-Receptacle Compatibility Constraint
Inadmissible action: put statue 1 to fridge 1
Reason: Not all objects are compatible with all receptacles. Attempting to place an incompatible object-receptacle pair will consistently fail, regardless of spatial proximity.

Receptacle Capacity Constraint
Inadmissible action: put toiletpaper 1 to cabinet 1
Reason: Some receptacles have limited capacity. When a receptacle (e.g., a cabinet) is full, placing additional objects will fail unless space is freed.

Occlusion and Access Ordering Constraint
Inadmissible action: examine drawer 3
Reason: Certain receptacles exhibit hierarchical or occlusion relationships. For instance, a lower drawer (drawer 3) may be inaccessible for interaction until an upper drawer (drawer 2) is closed.

Figure 6: Several non-exhaustive examples of affordance constraints in ALFWorld, highlighting the types of environment dynamics and interaction constraints that must be reasoned about for successful task execution in visually grounded embodied environments.