From Word to World: CAN LARGE LANGUAGE MODELS BE IMPLICIT TEXT-BASED WORLD MODELS?

Yixia Li[1,2] **Hongru Wang**[3,*] **Jiahao Qiu**[4] **Zhenfei Yin**[5] **Dongdong Zhang**[2]
Cheng Qian[6] **Zeping Li**[7] **Pony Ma**[8] **Guanhua Chen**[1,*] **Heng Ji**[6]

[1]Southern University of Science and Technology [2]Microsoft Research [3]University of Edinburgh
[4]Princeton University [5]Oxford University [6]University of Illinois Urbana-Champaign
[7]Fudan University [8]Mind Lab
liyixia@me.com hongru.carrywang@gmail.com chengh3@sustech.edu.cn

ABSTRACT

Agentic learning increasingly hinges on interaction, yet real-world experience is expensive, limited, and often irreversible at inference time. World models promise to mitigate these limitations, but it remains unclear whether large language models can actually serve as reliable world models and deliver concrete benefits to downstream agents. We investigate these questions in text-based environments, a controlled testbed that reframes language modeling as next-state prediction under interaction. We propose a three-level framework to evaluate LLM-based world models: (i) fidelity and consistency, (ii) scalability and robustness, and (iii) agent utility. Across five representative environments, we show that sufficiently trained world models capture coherent environment dynamics, scale predictably with data and model capacity, and unlock tangible agent improvements—for example, action verification boosts GPT-4o by 5.5% on WebShop, and warm-started RL achieves a 15% gain on SciWorld. Crucially, these benefits hinge on behavioral coverage and environment complexity, sharply characterizing when world modeling meaningfully advances agent learning. [1]

1 INTRODUCTION

Despite rapid advances in agentic development, learning from interaction remains a central constraint. As agents become more capable, further progress increasingly demands larger, more diverse, and more challenging environments (Zeng et al., 2025; Zhang et al., 2025a; Tong et al., 2025). Unlike static pretraining corpora, such experience must be acquired through direct interaction with real environments, making data collection expensive, slow, and limited in coverage. Moreover, interaction at inference time is often irreversible: agents cannot rewind actions to correct mistakes or recover from suboptimal decisions. Together, these constraints restrict both scalable data acquisition and reliable decision making, ultimately throttling the pace of progress in agentic learning (Wei et al., 2025; Jiang et al., 2025; Guo et al., 2025).

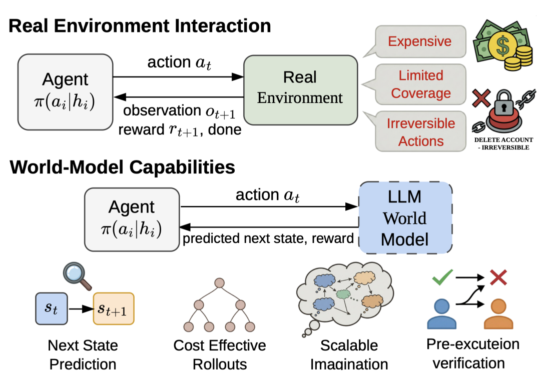

Figure 1: **LLMs as text-based world models for agent learning.** Real-world interaction is expensive, slow, and often irreversible. By predicting environment dynamics from actions, LLM-based world models enable cost-effective data synthesis, scalable learning and rewindable decision making.

A promising way to break these constraints is *world modeling* (Hafner et al., 2024; 2025; Zhao et al., 2025; Hu et al., 2025a), which seeks to learn a

*Corresponding author
[1]Our code is available at https://github.com/X1AOX1A/Word2World.

predictive abstraction of environment dynamics conditioned on agent actions. By enabling imagined interaction—rolling out consequences, comparing counterfactuals, and evaluating candidate actions before execution—world models reduce reliance on costly trial-and-error and provide a rewindable substrate for decision making. At the same time, large language models trained at a massive scale via next-token prediction demonstrate remarkable generalization and encode rich world knowledge (Grattafiori et al., 2024; Qwen et al., 2025; Hu et al., 2025b). This convergence raises a natural question: *Can large language models serve as effective world models, thereby improving agents' ability to learn from experience?*

While prior work has explored LLMs as simulators, experience generators, or planning interfaces (Chen et al., 2025; Li et al., 2025b; Wu et al., 2025; Gu et al., 2025; Wang et al., 2025; He et al., 2025), it remains unclear *how* to learn reliable world models and *when* they are reliable enough to benefit downstream agents. A useful world model must go beyond locally plausible text, maintaining coherent state over time, remaining robust to distribution shift, and delivering measurable utility. To study these questions in a controlled yet expressive setting, we focus on **text-based environments** as a unifying interface between language modeling and world modeling. This abstraction retains core challenges of agent–environment interaction, such as long-horizon dependencies, compounding errors and sim-to-real gaps, while reframing the objective from *next-token prediction* to *next-state prediction* under a fixed interaction protocol. Through this lens, we introduce a three-level characterization of world modeling for agent learning:

- **Fidelity and Consistency:** Whether a world model sustains coherent environment dynamics over both short and long horizons, forming the basis for reliable rollouts.
- **Scalability and Robustness:** How world modeling performance scales with data, model capacity, and environment complexity, and how it generalizes under distribution shift.
- **Agent Performance Improvement:** Whether high-fidelity world models translate into concrete gains for downstream agents.

Across five representative text-based environments, our study yields three key findings. (i) LLMs can act as reliable world models in structured settings: they capture implicit transition regularities that support in-context world modeling, and supervised fine-tuning substantially improves both short-term fidelity and long-horizon consistency. (ii) Reliable world modeling demands systematic scaling of data and model capacity with environment complexity, as well as broad behavioral coverage to remain robust under distribution shift. (iii) High-fidelity world models unlock tangible agent benefits, enabling verification of high-stakes actions, synthetic experience generation, and warm-started reinforcement learning that improves both efficiency and stability.

Taken together, these findings illuminate both the promise and the limits of LLM-based world models in text environments. From words to worlds, from next-token to next-state prediction, we provide an empirical grounding for treating LLMs as world models for agentic learning and chart a path toward domains beyond text.

2 Related Works

Large language models have recently been explored as world models across a variety of text-based and structured settings. Prior efforts in world modeling largely focus on predicting environment dynamics through structured or discrete state representations. Patch-based approaches prompt LLMs to estimate state deltas in ByteSized32 (Wang et al., 2024; Yang et al., 2024), while in web navigation, systems such as WMA (Chae et al., 2025) and RLVR-World (Wu et al., 2025) reason over updates to the Accessibility Tree. Other lines of work adopt closed-form prediction schemes where the model outputs predefined symbolic labels or categories, including preconditions and effects in cooking environments (Xie et al., 2024), disaster impact ratings (Li et al., 2025a), or classifier-head predictions trained on LLM embeddings (Yang et al., 2025). Although these methods illustrate the utility of structured prediction in specific domains, they typically rely on environment-specific abstractions and fixed output spaces. In contrast, we formulate world modeling as a multi-turn natural language simulation task, where the LLM generates next-state transitions in free text, enabling more general and compositional interaction patterns.

Regarding model adaptation, much prior work employs zero-shot or few-shot prompting (Wang et al., 2024; Yang et al., 2024; Li et al., 2025a; Zuo et al., 2025) or attaches lightweight classifier heads for closed-form prediction (Yang et al., 2025). Although these approaches reveal the latent capabilities of LLMs, they often yield limited accuracy and constrain downstream applicability. Moving beyond prompting-based methods, we finetune LLMs on large-scale multi-turn interaction trajectories to better internalize environment dynamics over extended horizons.

Prior evaluation efforts mainly focus on single-step prediction in limited environments (Wang et al., 2024; Xie et al., 2024; Chae et al., 2025; Li et al., 2025a), and seldom examine long-horizon consistency or compounding errors that are critical for reliable simulation. As a result, it remains unclear whether LLM-based world models can generate coherent multi-step trajectories executable in real environments. To address this gap, we conduct a systematic evaluation across five representative environments, assessing one-step fidelity, rollout stability, WM-to-Real transfer, and generalization across agents, environments, and scales.

3 LLM AS TEXT-BASED WORLD MODELS

3.1 FORMALIZATION OF WORLD MODELS

We formalize the interaction between an **agent** and a **text-based world model** as a multi-turn language-based decision process, where both perception and action are represented in natural language.

Agent A text-based agent $\mathcal{A}$ operates in a ReAct style (Yao et al., 2023b), yielding a simple, unified interface where each step involves internal reasoning and external action. Formally, the agent is defined as:

$$\mathcal{A} : \{S_0, (T_i, A_i, S_i)_{i=1}^{n-1}\} \rightarrow (T_n, A_n), \tag{1}$$

where S_i denotes the textual observation (or environment response) at step i, T_i represents the agent's internal reasoning trace, and A_i denotes the explicit action expressed in natural language.

World Model The environment or world model $\mathcal{W}$ defines the complementary mapping:

$$\mathcal{W} : \{S_0, (A_i, S_i')_{i=1}^{n-1}, A_n\} \rightarrow (S_n', R_n'), \tag{2}$$

where S_n' denotes the next state predicted by the world model, and $R_n' \in \{0, 1\}$ is a binary reward indicating task success or termination. A value of $R_n' = 1$ corresponds to a successful completion, while $R_n' = 0$ denotes either an unfinished or failure state (e.g., triggering validation at the wrong time). Through these textual transitions, the world model functions as an implicit **next-state predictor** of environment dynamics. This capability can be realized through **in-context learning**, where the model leverages few-shot examples of state transitions in its prompt, or through **supervised fine-tuning** on trajectory data to learn the underlying dynamics.

In practice, text-based environments are inherently POMDPs (Partially Observable Markov Decision Processes): the true environment state is richer than what is described to the agent. For example, in ALFWorld a room may contain objects and spatial details that are never mentioned (e.g., what is inside a closed drawer), yet these hidden factors matter for predicting how the world evolves. Thus, although the agent only receives a partial view of the initial state S_0, the world model can be initialized with a more comprehensive context, such as full environment configurations or randomized setups, allowing it to better approximate the environment's implicit dynamics.

Interactive Process Together, the agent and world model form an iterative process:

$$S_n', R_n' = \mathcal{W}\big(\mathcal{A}(S_0, (T_i, A_i, S_i')_{i=1}^{n-1})\big), \tag{3}$$

which unrolls into a multi-turn textual trajectory generated within the world model:

$$\tau_{\text{wm}} = \{S_0, T_1, A_1, S_1', \ldots, T_T, A_T, S_T'\}. \tag{4}$$

Correspondingly, the real environment produces the trajectory

$$\tau_{\text{real}} = \{S_0, T_1, A_1, S_1, \ldots, T_T, A_T, S_T\}, \tag{5}$$

which serves as the reference for evaluating the fidelity and consistency of $\mathcal{W}$.

By formulating text-based environments as multi-turn interactive processes, the world model can be prompted with few-shot exemplars or trained on real trajectories τ_{real} to predict next-state transitions. This formulation enables $\mathcal{W}$ to capture long-horizon dependencies and cumulative effects across interaction steps. While prior works (Wang et al., 2024; Xie et al., 2024; Yang et al., 2025) primarily focus on next-state prediction, we explicitly model, train and evaluate the world model's long-horizon consistency, which is critical for applications such as data synthesis, test-time simulator, and model-based reinforcement learning.

3.2 Text-based Environments

To examine the range of knowledge and dynamics required for text-based world modeling in a broad way, we adopt five representative environments spanning both structured and open-ended settings. The structured environments ALFWorld (Shridhar et al., 2021), SciWorld (Wang et al., 2022), and TextWorld (Côté et al., 2018) feature bounded state spaces. They provide deterministic or rule-governed transitions grounded in embodied, scientific, or narrative regularities. In contrast, the open-ended environments WebShop (Yao et al., 2023a) and StableToolBench (Guo et al., 2025) exhibit broad, compositional, and open-world dynamics, with diverse entities and flexible task formulations that require stronger generalization beyond fixed schemas. Table 5 in Appendix C.1 summarizes these environments and their key characteristics, with examples in Appendix D. Together, these settings provide a comprehensive and diversified testbed for evaluating language models as text-based world simulators.

4 World Model Training and Evaluation

We summarize the world model training and evaluation setup and defer full implementation details to Appendix C.1. Unless otherwise specified, all experiments follow the default settings below.

Data We collect interaction trajectories using GPT-4o as the behavior policy. To match environment complexity (see Section 6.1), we gather 40K trajectories each for ALFWorld, SciWorld, and TextWorld, and 70K for WebShop. We retain both successful and failed episodes to broaden behavioral coverage for world model training.[2] For StableToolBench, we use the public single-turn dataset with 160K samples.

Initialization Context For ALFWorld and SciWorld, we condition the world model on simulator-provided initial state configurations (see Figures 9 and 10 in Appendix C.4). In contrast, TextWorld does not expose initial states, and WebShop/StableToolBench are inherently partially observable, placing greater demands on history-based state tracking and prior knowledge to infer unobserved state variables.

Finetuning Models We use Qwen2.5-7B (base) and Llama-3.1-8B (base) as backbone models for text-based world modeling. A comparison across different model sizes is provided in Section 6.2.

Training Each trajectory is formatted as a multi-turn dialogue of alternating agent actions and environment responses (see Eq. 5). During supervised fine-tuning, the world model predicts the next environment response conditioned on the dialogue history and the current action.

Metrics We evaluate world models along two dimensions: one-step prediction accuracy and multi-step rollout consistency. **Fidelity.** We compute exact-match (EM) accuracy by conditioning on a real trajectory prefix $\{S_0, (A_i, S_i)_{i=1}^{n-1}, A_n\}$ and predicting the next state and reward (S'_n, R'_n). A prediction is correct if (S'_n, R'_n) matches the ground truth (S_n, R_n). EM is a conservative lower bound, especially in TextWorld where multiple surface forms may describe the same state. For StableToolBench, whose outputs are highly open-ended, we additionally report word-level F1. We omit semantic similarity metrics, as they are weakly correlated with state prediction performance; instead, we assess practical reliability via consistency and downstream performance. **Consistency.** We report: (1) **Real**, success rate in real environment; (2) **WM**, success rate inside the world

[2] The success/failure mixture is induced by GPT-4o's native success rate without additional filtering.

model; (3) **W2R**, success rate when replaying world-model actions in real environment; and (4) the **Consistency Ratio**, CR $=$ W2R/Real, where higher values indicate better long-horizon transfer (CR > 1 indicates simulation rollouts outperform real execution).

5 Fidelity & Consistency

5.1 Next-state Prediction Accuracy

Table 1 demonstrates that pretrained LLMs exhibit meaningful in-context world modeling ability. Models such as Gemini-2.5-flash and Claude-sonnet-4.5 achieve strong next-state prediction in structured environments like ALFWorld and SciWorld, where a handful of demonstrations provides substantial improvements (e.g., Claude rises from 56.83 to 73.08 accuracy on SciWorld with only three examples). This suggests that contemporary LLMs encode latent knowledge of environment dynamics and can rapidly adapt their transition rules with minimal supervision. However, these capabilities do not fully transfer to open-ended settings such as WebShop, where few-shot prompting plateaus around 55, indicating that implicit world knowledge alone is insufficient for generating unconstrained, context-dependent state updates.

Environment	AW	SW	TW	WS	STB	STB_{F1}
Zero-shot						
GPT-4o-mini	45.20	40.68	0.36	56.59	0.00	13.94
GPT-4o	44.45	45.78	7.86	58.20	0.00	11.88
GPT-4-turbo	42.64	34.14	0.00	52.45	0.00	12.64
GPT-4.1	43.56	35.65	0.00	58.07	0.00	12.83
GPT-5	35.09	13.06	9.20	46.12	0.00	8.02
Gemini-2.5-flash	50.00	44.81	3.51	57.64	0.00	8.74
Claude-sonnet-4.5	64.73	56.83	17.70	58.80	0.00	11.36
Few-shot (3 shot)						
GPT-4o-mini	63.79	56.26	11.43	61.93	0.00	13.44
GPT-4o	56.88	48.98	14.11	64.62	0.00	11.08
GPT-4-turbo	62.56	50.08	11.66	62.76	0.00	10.72
GPT-4.1	63.37	51.56	13.39	64.23	0.00	10.33
GPT-5	67.13	49.44	44.27	65.90	0.00	6.28
Gemini-2.5-flash	61.85	61.20	40.35	66.09	0.00	8.47
Claude-sonnet-4.5	77.04	73.08	49.12	56.65	0.00	13.11
SFT						
Qwen2.5-7B	99.87	98.60	70.60	79.05	48.90	79.15
Llama3.1-8B	99.71	98.64	70.45	77.24	49.25	78.97

Table 1: Next-state prediction EM accuracy (%) of prompt-based and finetuned models across five environments. AW, SW, TW, WS and STB denote ALFWorld, SciWorld, TextWorld, WebShop and StableToolBench, respectively. STB_{F1} denotes the word-level F1 score for StableToolBench, given its open-ended output space.

Supervised fine-tuning yields substantial improvements. Open-source models trained directly on transition trajectories achieve 99%/98% accuracy on ALFWorld and SciWorld and reach 49% F1 on StableToolBench. These results indicate that robust world modeling requires dynamics-aligned training: prompting alone cannot capture the full diversity of transition patterns, whereas supervised fine-tuning enables even relatively small models to internalize them effectively.

Findings 1: [Short-term Fidelity] LLMs exhibit implicit environment dynamics that support in-context world modeling, while high-fidelity modeling depends on supervised fine-tuning.

5.2 Rollout Consistency

A reliable world model requires not only high single-step prediction accuracy, but more critically, the ability to maintain consistency during extended interactions with agents. We examine two key dimensions: (1) whether small local errors compound into significant failures over long-horizon rollouts, and (2) whether the world model generalizes across different agent behaviors beyond its training distribution. Table 2 reports consistency metrics across four environments and multiple agents; StableToolBench is omitted due to its single-turn nature.

Consistency Across Environments World models largely preserve single-step fidelity in long-horizon rollouts, especially in structured environments. In ALFWorld, SciWorld, and TextWorld, the fine-tuned Qwen2.5 world model attains high consistency ratios of 96%, 91%, and 92%, indicating that multi-step trajectories generated within the world model remain executable when transferred to the real environment. WebShop, however, exhibits lower consistency (typically below 80%), primarily due to its open-ended nature and diverse search results that the world model struggles to simulate accurately. This error can be substantially mitigated by grounding model rollouts with real observations. When the rollout is initialized with real search results, the consistency with GPT-4o

Agent	ALFWorld				SciWorld				TextWorld				WebShop			
	Real	WM	W2R	CR	Real	WM	W2R	CR	Real	WM	W2R	CR	Real	WM	W2R	CR
Qwen2.5-7B WorldModel																
GPT-4o-mini	7.69	7.69	7.69	1.00	12.64	12.04	8.90	0.70	97.44	100.00	69.36	0.71	5.99	4.85	0.97	0.16
GPT-4o	58.00	55.90	57.44	0.99	34.97	37.63	31.44	0.90	98.84	100.00	96.53	0.98	29.36	17.43	16.51	0.56
GPT-4-turbo	74.21	62.56	64.62	0.87	36.79	50.00	36.60	0.99	100.00	99.42	98.84	0.99	17.73	14.89	11.70	0.66
GPT-4.1	67.20	68.56	69.59	1.04	43.41	45.79	46.32	1.07	100.00	100.00	100.00	1.00	21.14	12.22	12.22	0.58
GPT-5	91.00	84.62	86.67	0.95	68.21	64.10	61.03	0.89	100.00	100.00	100.00	1.00	51.00	33.03	31.19	0.61
Gemini-2.5-flash	50.50	51.79	52.31	1.04	56.00	39.49	45.64	0.82	100.00	100.00	76.30	0.76	25.00	21.10	18.35	0.73
Claude-sonnet-4.5	82.00	76.00	76.00	0.93	66.00	45.64	57.95	0.88	100.00	100.00	100.00	1.00	61.00	49.00	50.00	0.82
Average	61.51	58.16	59.19	0.96	45.43	42.10	41.13	0.91	99.47	99.92	91.58	0.92	30.17	21.79	20.13	0.67
Llama3.1-8B WorldModel																
GPT-4o-mini	7.69	9.74	9.74	1.27	12.64	10.78	8.21	0.65	97.44	92.48	57.80	0.59	5.99	2.75	0.00	0.00
GPT-4o	58.00	58.46	56.92	0.98	34.97	37.63	32.99	0.94	98.84	97.11	90.17	0.91	29.36	23.81	22.62	0.77
GPT-4-turbo	74.21	67.53	67.01	0.90	36.79	52.31	44.10	1.20	100.00	97.69	93.06	0.93	17.73	25.47	17.92	1.01
GPT-4.1	67.20	68.72	68.21	1.02	43.41	45.13	35.38	0.82	100.00	98.27	94.22	0.94	21.14	19.27	17.43	0.82
GPT-5	91.00	82.56	81.54	0.90	68.21	63.07	57.44	0.84	100.00	98.84	94.80	0.95	51.00	31.19	30.28	0.59
Gemini-2.5-flash	50.50	53.33	53.33	1.06	56.00	57.95	52.31	0.93	100.00	99.42	93.06	0.93	25.00	22.02	17.43	0.70
Claude-sonnet-4.5	82.00	84.00	84.00	1.02	66.00	58.46	53.33	0.81	100.00	93.33	90.00	0.90	61.00	60.00	55.00	0.90
Average	61.51	60.62	60.11	0.98	45.43	46.48	30.52	0.89	99.47	96.73	87.59	0.88	30.17	26.36	22.95	0.76

Table 2: Task success rate (%) of different agents across four environments. "Real", "WM", and "W2R" denote the success rate under real environment, world model, and world model-to-real execution. The last column reports the consistency ratio (CR=W2R/Real), with higher values (darker green color) indicating better rollout fidelity.

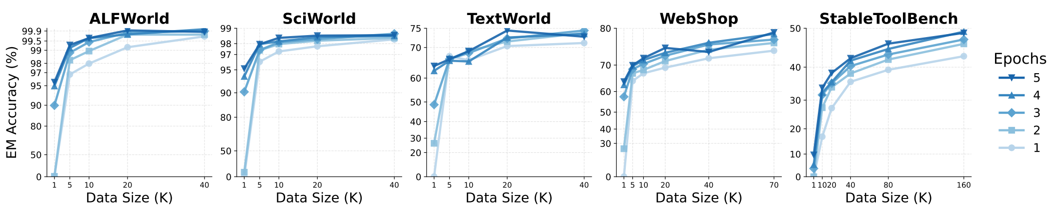

Figure 2: Next-state prediction accuracy under varying training data sizes on Qwen2.5-7B. Structured settings saturate with modest data (~20K), whereas open-ended settings continue to benefit from larger datasets.
Note. We apply a nonlinear y-axis transform $f(y) = 100 - 20\log_{10}(\max(100 - y, 0.01) + 1)$ to better reveal growth trends.

agent increases dramatically from 56% to nearly 100%, demonstrating that partial real-environment anchoring effectively reduces simulation drift.

Findings 2: [Long-term Consistency] World models maintain consistent long-horizon rollouts in well-structured domains, but tend to drift in open-ended environments due to high diversity, necessitating anchoring to real-world signals.

How does behavior shift affect consistency? Beyond environment-specific factors, world model consistency also depends on how well agent behaviors match the training distribution. Lower-capacity agents such as GPT-4o-mini yield consistency ratios frequently below 70%, whereas stronger agents like GPT-4.1, GPT-5, and Claude reliably exceed 90%. This disparity stems from weaker agents taking actions misaligned with task objectives, causing their trajectories to drift outside the training distribution. In contrast, higher-capacity agents preserve goal-directed behavior that aligns with the expert policy (GPT-4o) used for trajectory sampling, enabling higher consistency. These results highlight the importance of diversifying training trajectories rather than relying solely on a single strong agent, as further discussed in Section 6.5.

6 Scalability & Robustness

6.1 Data Scaling Laws for World Models

To investigate how world model performance scales with data, we vary training trajectories from 1K to 160K and evaluate single-step accuracy. As shown in Figure 2, structured environments (ALFWorld,

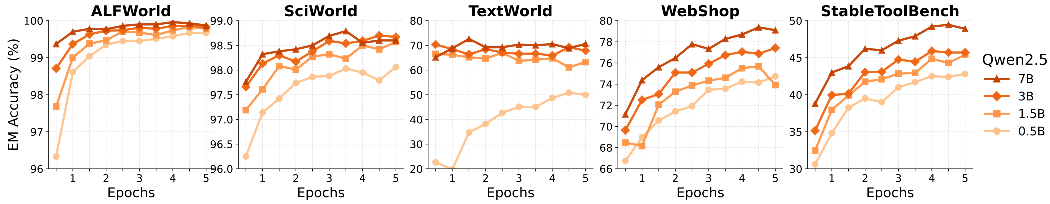

Figure 3: Next-state prediction accuracy on Qwen2.5 family. Smaller models (~1.5B) capture structured dynamics effectively, whereas more complex settings benefit markedly from increased model capacity.

SciWorld, TextWorld) improve rapidly and saturate around 20K trajectories, consistent with their low-entropy, rule-driven dynamics. In contrast, open-ended environments scale more gradually: WebShop benefits from additional data up to roughly 70K trajectories, while StableToolBench shows no saturation at 160K samples due to long-tail linguistic variation and highly compositional API behaviors. These results indicate that world modeling exhibits environment-dependent scaling: structured environments are highly data-efficient, whereas open-ended domains require larger datasets.

6.2 Model Size Effects

We next analyze how model capacity shapes world model performance (Figure 3). Mirroring data-scaling trends, model size interacts strongly with environment complexity. In structured environments, performance saturates quickly: 1.5B models already capture core transition dynamics, with further scaling yielding only marginal improvements. In open-ended environments, however, capacity matters substantially. Smaller models struggle to represent rich linguistic variability and compositional tool usage, whereas larger models offer steady accuracy gains. Together with the data-scaling results, these findings indicate that success in open-ended world modeling requires both extensive trajectories and sufficient model capacity to internalize long-tailed, high-entropy dynamics.

Findings 3: [LLMs are Scalable World Models] World modeling performance scales systematically with data volume, model size and environment complexity.

6.3 Beyond Memorized Environments

A central question in world model design is how well they generalize across unseen settings. Using ALFWorld as a representative case, we analyze two out-of-distribution test splits following the original environment settings (Shridhar et al., 2021): *OOD-Seen*, which keeps the room type but alters the layout, and *OOD-Unseen*, which introduces entirely new room types or unseen layout configurations. As shown in Figure 4, the world models maintain success rates closely aligned with the real environment across both OOD settings even when the spatial configuration shifts or novel room types appear. These results indicate that the LLM world model captures transferable transition dynamics rather than memorizing specific layouts, demonstrating strong robustness to structural variations in the environment's state space.

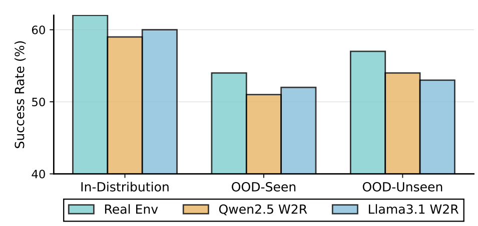

Figure 4: Task success rate (%) in ALFWorld under different OOD settings. Success rate averaged over different agents, with full results provided in Table 10 of Appendix E. World models maintain strong performance even when layouts or room types change.

6.4 Cross-Env Transfer via Joint Training

Training world models in isolation often limits their ability to generalize beyond a single environment, motivating us to investigate whether jointly training on multiple environments can yield transferable gains. We therefore evaluate three mixed-training configurations: Mix3 (ALFWorld, SciWorld, TextWorld), Mix4 (with WebShop), and Mix5 (with StableToolBench), allocating 1K trajectories per environment to match the data budget of individually trained models. As shown in Figure 5, mixed training consistently accelerates learning and improves final accuracy, with particularly strong gains in TextWorld and WebShop, suggesting that the model effectively internalizes and reuses shared physical, procedural, and narrative dynamics across tasks. The exception is StableToolBench, whose schema-centric, single-turn structure is underrepresented in the mixture, causing the separately trained model to outperform. Overall, these results show that mixed data provides stable positive gains and, importantly, enables practical deployments where a single world model can robustly serve multiple environments.

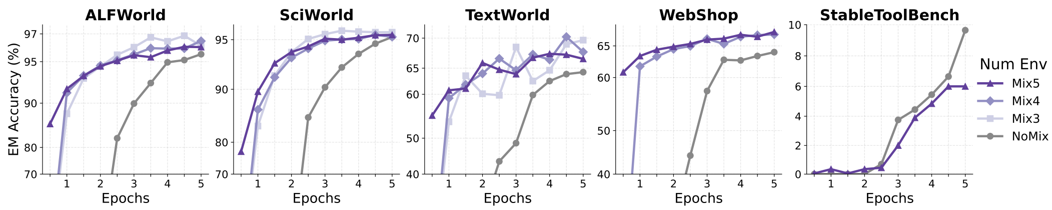

Figure 5: Next-state prediction accuracy under mixed and separate training on Qwen2.5-7B, with 1K samples per environment. We begin by mixing structured environments (ALFWorld, SciWorld, TextWorld) and incorporate open-ended environments (WebShop, StableToolBench), yielding the Mix3, Mix4, and Mix5 settings.

6.5 Behavioral Coverage for Robust World Modeling

As behavior shifts reduce consistency, we ask whether broader behavioral coverage improves generalization. We train a world model on mixed-agent trajectories and compare it to a GPT-4o-only baseline. Table 3 shows marked OOD gains for weaker agents: GPT-4o-mini's consistency ratio rises from 0.49 to 0.81, and GPT-4-turbo also improves. This indicates that expert-only trajectories are insufficient under distribution shift; incorporating diverse agent behaviors is crucial for improving generalization and rollout stability.

Agent	Real	Single Agent Traj			Mix Agent Traj		
		WM	W2R	CR	WM	W2R	CR
Qwen3-235B	24.00	24.00	18.00	0.75	26.00	18.00	0.75
GPT-4o	34.97	32.31	26.67	0.76	32.31	26.67	0.76
GPT-5	68.21	55.38	59.49	0.87	69.74	60.00	0.88
Claude-sonnet-4.5	66.00	40.51	57.44	0.87	52.31	49.74	0.75
ID Agent Average	48.30	38.05	40.40	0.81	45.09	38.60	0.79
GPT-4o-mini	12.64	5.64	6.15	0.49	13.39	10.26	0.81
GPT-4-turbo	36.79	32.31	38.97	1.06	51.28	42.56	1.16
GPT-4.1	43.41	28.72	36.41	0.84	52.31	36.41	0.84
Gemini-2.5-flash	56.00	36.92	51.79	0.92	56.92	45.64	0.82
OOD Agent Average	37.21	25.90	33.33	0.83	43.48	33.72	0.91

Table 3: Task success rate (%) in SciWorld under different training data compositions. "Single Agent Traj" uses only 4K GPT-4o trajectories for training, whereas "Mix Agent Traj" combines trajectories from ID agents, with 1K trajectories from each.

Findings 4: [World Models Learn Generalizable Dynamics] World models generalize beyond memorized configurations and transfer across environments and agent policies, where coverage of environments and diversity of agent behaviors play critical roles.

7 Agent Utility

7.1 Can World Models Prevent Irreversible Mistakes?

In real-world decision-making, some actions are irreversible and costly, creating a safety bottleneck: a single mistaken commitment can end an episode or cause unrecoverable loss. This motivates using world models as a *rewindable imagined world* to evaluate high-stakes actions before execution. WebShop exemplifies this setting: once the agent checks out, the episode ends and errors cannot be undone. We therefore use the world model as a lightweight pre-execution verifier. Before committing to checkout, the agent simulates the outcome; it executes the action only when the prediction indicates success, otherwise it continues interacting with the environment. We vary the verification budget (0, 2, 4, 10). As shown in Table 4,

Agent	0	2	4	10
GPT-4o-mini	5.99	7.50 (+1.51)	7.55 (+1.56)	7.59 (+1.60)
GPT-4o	29.36	32.41 (+3.05)	33.94 (+4.58)	34.86 (+5.50)
GPT-4-turbo	17.73	33.33 (+15.60)	27.05 (+9.32)	29.37 (+11.64)
GPT-4.1	21.14	23.59 (+2.45)	23.59 (+2.45)	23.08 (+1.94)
GPT-5	51.00	53.27 (+2.27)	53.77 (+2.77)	53.27 (+2.77)
Gemini-2.5-flash	25.00	31.00 (+6.00)	29.50 (+4.50)	28.00 (+3.00)
Claude-sonnet-4.5	61.00	62.00 (+1.00)	65.00 (+4.00)	64.00 (+3.00)

Table 4: Task success rate (%) of different agents in WebShop with varying numbers of max pre-execution verification attempts using the world model. The numbers in parentheses indicate the improvement over the baseline without verification.

adding pre-execution verification consistently improves success rates over no verification across agents, with the largest gains for medium-capacity models.

7.2 Synthetic Data Competes with Real

When real interaction is expensive, slow, or constrained, agents face an experience bottleneck. A world model can alleviate this bottleneck by synthesizing trajectories that substitute for part of real experience. To examine this, we collect 1,000 successful trajectories from either the real environment or the world model[3], and construct four SFT datasets for Qwen2.5-7B-Instruct: Real 1K, Syn 1K, 0.5K+0.5K, and a 1K+1K mixture, all trained under identical procedures. Figure 6 shows that world model–generated trajectories are competitive with real data. In SciWorld, Syn 1K matches Real 1K, while the 1K+1K mixture outperforms either source alone. In WebShop, synthetic data remains effective, and mixed regimes yield the most stable gains. Overall, these results suggest that synthetic experience can reduce reliance on real-environment interaction, providing an alternative pathway for scaling agent learning when real experience is limited.

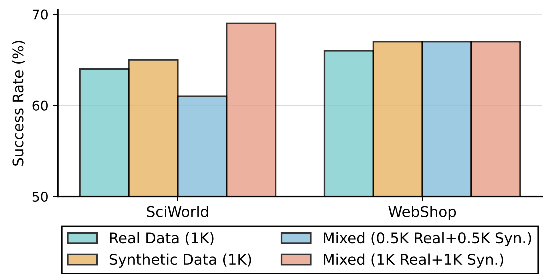

Figure 6: Task success rate (%) of Qwen2.5-7B-Instruct SFT trained agents with different data synthesis strategies in SciWorld and WebShop.

7.3 Early Experience for Policy Learning

Recent work (Zhang et al., 2025b) suggests that exposing a model to environment dynamics before explicit policy learning can provide a useful inductive bias: anticipating consequences may reduce unguided exploration and stabilize early RL. To study this in our text-based decision environments, we compare (1) standard Agent-SFT → RL baseline; and (2) world-model warmup pipeline (WM-SFT → Agent-SFT → RL), where the agent is first exposed to environment dynamics with same objective as world model training[4].

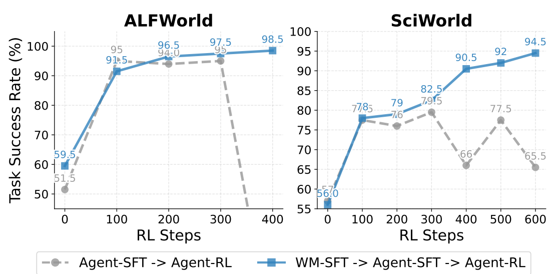

Figure 7: Task success rate (%) of Qwen2.5-7B-Instruct RL trained agents with and without early experience.

Figure 7 indicates that early experience delivers consistent gains on both ALFWorld and SciWorld. By exposing the agent to environment dynamics before policy learning, early experience stabilizes RL training, reducing failures driven by incorrect commonsense priors, and ultimately yields higher final success rates than the baseline. Overall, early experience provides a promising direction for improving learning effectiveness.

8 Conclusion

Using text-based environments as a controlled testbed, we cast world modeling as multi-turn next-state prediction and evaluate it along fidelity, scalability, and agent utility. Our results show that with dynamics-aligned supervision and sufficient scale and coverage, LLMs can serve as implicit text-based world models that maintain coherent states over long horizons and improve downstream agents via safer decision-making, scalable experience generation, and better learning efficiency. We further demonstrate that robustness depends critically on behavioral coverage, distributional alignment, and environmental complexity, providing insights for future work on world modeling and agent learning.

[3] Implementation details in Appendix C.2

[4] Implementation details in Appendix C.3

Reproducibility Statement

In Section 4 and Appendix C.1, we provide detailed implementation details for the world model training and evaluation. We will open-source the code and data for reproducibility.

Ethical Considerations

Our study is conducted in controlled, text-only benchmark environments and does not involve human subjects or the collection of personal data. As with other agentic and world-modeling capabilities, misuse (e.g., enabling harmful or deceptive behavior) and bias propagation are possible; we encourage responsible deployment with appropriate safeguards and oversight.

References

Hyungjoo Chae, Namyoung Kim, Kai Tzu iunn Ong, Minju Gwak, Gwanwoo Song, Jihoon Kim, Sunghwan Kim, Dongha Lee, and Jinyoung Yeo. Web agents with world models: Learning and leveraging environment dynamics in web navigation, 2025. URL https://arxiv.org/abs/2410.13232.

Zhaorun Chen, Zhuokai Zhao, Kai Zhang, Bo Liu, Qi Qi, Yifan Wu, Tarun Kalluri, Sara Cao, Yuanhao Xiong, Haibo Tong, Huaxiu Yao, Hengduo Li, Jiacheng Zhu, Xian Li, Dawn Song, Bo Li, Jason Weston, and Dat Huynh. Scaling agent learning via experience synthesis, 2025. URL https://arxiv.org/abs/2511.03773.

Marc-Alexandre Côté, Ákos Kádár, Xingdi Yuan, Ben Kybartas, Tavian Barnes, Emery Fine, James Moore, Ruo Yu Tao, Matthew Hausknecht, Layla El Asri, Mahmoud Adada, Wendy Tay, and Adam Trischler. Textworld: A learning environment for text-based games. *CoRR*, abs/1806.11532, 2018.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, and Chloe Bi et al. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Yu Gu, Kai Zhang, Yuting Ning, Boyuan Zheng, Boyu Gou, Tianci Xue, Cheng Chang, Sanjari Srivastava, Yanan Xie, Peng Qi, Huan Sun, and Yu Su. Is your llm secretly a world model of the internet? model-based planning for web agents, 2025. URL https://arxiv.org/abs/2411.06559.

Zhicheng Guo, Sijie Cheng, Hao Wang, Shihao Liang, Yujia Qin, Peng Li, Zhiyuan Liu, Maosong Sun, and Yang Liu. Stabletoolbench: Towards stable large-scale benchmarking on tool learning of large language models, 2025. URL https://arxiv.org/abs/2403.07714.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models, 2024. URL https://arxiv.org/abs/2301.04104.

Danijar Hafner, Wilson Yan, and Timothy Lillicrap. Training agents inside of scalable world models, 2025. URL https://arxiv.org/abs/2509.24527.

Haoran He, Yang Zhang, Liang Lin, Zhongwen Xu, and Ling Pan. Pre-trained video generative models as world simulators. In *ICLR 2025 Workshop on World Models: Understanding, Modelling and Scaling*, 2025. URL https://openreview.net/forum?id=oTYF8WUadL.

Mengkang Hu, Tianxing Chen, Yude Zou, Yuheng Lei, Qiguang Chen, Ming Li, Qiwei Liang, Yao Mu, Hongyuan Zhang, Wenqi Shao, and Ping Luo. Text2world: Benchmarking large language models for symbolic world model generation. In *ICLR 2025 Workshop on World Models: Understanding, Modelling and Scaling*, 2025a. URL https://openreview.net/forum?id=dIQNOxuBay.

Zhengyu Hu, Jianxun Lian, Zheyuan Xiao, Seraphina Zhang, Tianfu Wang, Nicholas Jing Yuan, Xing Xie, and Hui Xiong. Unveiling the learning mind of language models: A cognitive framework and empirical study. *arXiv preprint arXiv:2506.13464*, 2025b.

Dongfu Jiang, Yi Lu, Zhuofeng Li, Zhiheng Lyu, Ping Nie, Haozhe Wang, Alex Su, Hui Chen, Kai Zou, Chao Du, Tianyu Pang, and Wenhu Chen. Verltool: Towards holistic agentic reinforcement learning with tool use, 2025. URL `https://arxiv.org/abs/2509.01055`.

Lingyao Li, Dawei Li, Zhenhui Ou, Xiaoran Xu, Jingxiao Liu, Zihui Ma, Runlong Yu, and Min Deng. Llms as world models: Data-driven and human-centered pre-event simulation for disaster impact assessment, 2025a. URL `https://arxiv.org/abs/2506.06355`.

Yuetai Li, Huseyin A Inan, Xiang Yue, Wei-Ning Chen, Lukas Wutschitz, Janardhan Kulkarni, Radha Poovendran, Robert Sim, and Saravan Rajmohan. Simulating environments with reasoning models for agent training, 2025b. URL `https://arxiv.org/abs/2511.01824`.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025. URL `https://arxiv.org/abs/2412.15115`.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. Alfworld: Aligning text and embodied environments for interactive learning, 2021. URL `https://arxiv.org/abs/2010.03768`.

Jingqi Tong, Jixin Tang, Hangcheng Li, Yurong Mou, Ming Zhang, Jun Zhao, Yanbo Wen, Fan Song, Jiahao Zhan, Yuyang Lu, Chaoran Tao, Zhiyuan Guo, Jizhou Yu, Tianhao Cheng, Zhiheng Xi, Changhao Jiang, Zhangyue Yin, Yining Zheng, Weifeng Ge, Guanhua Chen, Tao Gui, Xipeng Qiu, Qi Zhang, and Xuanjing Huang. Game-rl: Synthesizing multimodal verifiable game data to boost vlms' general reasoning, 2025. URL `https://arxiv.org/abs/2505.13886`.

Ruoyao Wang, Peter Jansen, Marc-Alexandre Côté, and Prithviraj Ammanabrolu. Scienceworld: Is your agent smarter than a 5th grader?, 2022. URL `https://arxiv.org/abs/2203.07540`.

Ruoyao Wang, Graham Todd, Ziang Xiao, Xingdi Yuan, Marc-Alexandre Côté, Peter Clark, and Peter Jansen. Can language models serve as text-based world simulators? In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 1–17, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-short.1. URL `https://aclanthology.org/2024.acl-short.1/`.

Siyin Wang, Zhaoye Fei, Qinyuan Cheng, Shiduo Zhang, Panpan Cai, Jinlan Fu, and Xipeng Qiu. World modeling makes a better planner: Dual preference optimization for embodied task planning. In *ICLR 2025 Workshop on World Models: Understanding, Modelling and Scaling*, 2025. URL `https://openreview.net/forum?id=HCXshVxxdg`.

Zhepei Wei, Wenlin Yao, Yao Liu, Weizhi Zhang, Qin Lu, Liang Qiu, Changlong Yu, Puyang Xu, Chao Zhang, Bing Yin, Hyokun Yun, and Lihong Li. Webagent-r1: Training web agents via end-to-end multi-turn reinforcement learning, 2025. URL `https://arxiv.org/abs/2505.16421`.

Jialong Wu, Shaofeng Yin, Ningya Feng, and Mingsheng Long. Rlvr-world: Training world models with reinforcement learning, 2025. URL `https://arxiv.org/abs/2505.13934`.

Zhiheng Xi, Yiwen Ding, Wenxiang Chen, Boyang Hong, Honglin Guo, Junzhe Wang, Dingwen Yang, Chenyang Liao, Xin Guo, Wei He, Songyang Gao, Lu Chen, Rui Zheng, Yicheng Zou, Tao Gui, Qi Zhang, Xipeng Qiu, Xuanjing Huang, Zuxuan Wu, and Yu-Gang Jiang. Agentgym: Evolving large language model-based agents across diverse environments, 2024.

Zhiheng Xi, Jixuan Huang, Chenyang Liao, Baodai Huang, Honglin Guo, Jiaqi Liu, Rui Zheng, Junjie Ye, Jiazheng Zhang, Wenxiang Chen, Wei He, Yiwen Ding, Guanyu Li, Zehui Chen, Zhengyin Du, Xuesong Yao, Yufei Xu, Jiecao Chen, Tao Gui, Zuxuan Wu, Qi Zhang, Xuanjing Huang, and Yu-Gang Jiang. Agentgym-rl: Training llm agents for long-horizon decision making through multi-turn reinforcement learning, 2025. URL https://arxiv.org/abs/2509.08755.

Kaige Xie, Ian Yang, John Gunerli, and Mark Riedl. Making large language models into world models with precondition and effect knowledge, 2024.

Chang Yang, Xinrun Wang, Junzhe Jiang, Qinggang Zhang, and Xiao Huang. Evaluating world models with llm for decision making, 2024. URL https://arxiv.org/abs/2411.08794.

Chang Yang, Xinrun Wang, Qinggang Zhang, Qi Jiang, and Xiao Huang. Efficient integration of external knowledge to LLM-based world models via retrieval-augmented generation and reinforcement learning. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 9484–9501, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-335-7. doi: 10.18653/v1/2025.findings-emnlp.504. URL https://aclanthology.org/2025.findings-emnlp.504/.

Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. Webshop: Towards scalable real-world web interaction with grounded language agents, 2023a. URL https://arxiv.org/abs/2207.01206.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models, 2023b. URL https://arxiv.org/abs/2210.03629.

Zhiyuan Zeng, Hamish Ivison, Yiping Wang, Lifan Yuan, Shuyue Stella Li, Zhuorui Ye, Siting Li, Jacqueline He, Runlong Zhou, Tong Chen, Chenyang Zhao, Yulia Tsvetkov, Simon Shaolei Du, Natasha Jaques, Hao Peng, Pang Wei Koh, and Hannaneh Hajishirzi. Rlve: Scaling up reinforcement learning for language models with adaptive verifiable environments, 2025. URL https://arxiv.org/abs/2511.07317.

Jiayi Zhang, Yiran Peng, Fanqi Kong, Cheng Yang, Yifan Wu, Zhaoyang Yu, Jinyu Xiang, Jianhao Ruan, Jinlin Wang, Maojia Song, HongZhang Liu, Xiangru Tang, Bang Liu, Chenglin Wu, and Yuyu Luo. Autoenv: Automated environments for measuring cross-environment agent learning, 2025a. URL https://arxiv.org/abs/2511.19304.

Kai Zhang, Xiangchao Chen, Bo Liu, Tianci Xue, Zeyi Liao, Zhihan Liu, Xiyao Wang, Yuting Ning, Zhaorun Chen, Xiaohan Fu, Jian Xie, Yuxuan Sun, Boyu Gou, Qi Qi, Zihang Meng, Jianwei Yang, Ning Zhang, Xian Li, Ashish Shah, Dat Huynh, Hengduo Li, Zi Yang, Sara Cao, Lawrence Jang, Shuyan Zhou, Jiacheng Zhu, Huan Sun, Jason Weston, Yu Su, and Yifan Wu. Agent learning via early experience, 2025b. URL https://arxiv.org/abs/2510.08558.

Yi Zhao, Aidan Scannell, Yuxin Hou, Tianyu Cui, Le Chen, Dieter Büchler, Arno Solin, Juho Kannala, and Joni Pajarinen. Generalist world model pre-training for efficient reinforcement learning. In *ICLR 2025 Workshop on World Models: Understanding, Modelling and Scaling*, 2025. URL https://openreview.net/forum?id=WtJnrr4BGO.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, Zheyan Luo, Zhangchi Feng, and Yongqiang Ma. Llamafactory: Unified efficient fine-tuning of 100+ language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, Bangkok, Thailand, 2024. Association for Computational Linguistics. URL http://arxiv.org/abs/2403.13372.

Dongqi Zuo, Zheng CHEN, Chuan Zhou, Yandong Guo, Xiao He, and Mingming Gong. RADI: LLMs as world models for robotic action decomposition and imagination. In *ICLR 2025 Workshop on World Models: Understanding, Modelling and Scaling*, 2025. URL https://openreview.net/forum?id=cPo2iS6lwP.

APPENDIX

A LIMITATIONS

While we show that LLMs can serve as effective text-based world models under dynamics-aligned supervision, robustness remains sensitive to data diversity and behavioral coverage, suggesting that stronger generalization will require broader and more diverse interaction data that better spans the relevant state–action space. To control confounding factors and enable systematic evaluation, we restrict our study to purely text-based environments. Future work can extend this framework to multimodal and embodied settings, where perception and action introduce additional challenges for world modeling.

B LLM USAGE

In the preparation of this paper, we only used large language models (LLMs) as an assistive tool for grammar correction and text polishing.

C IMPLEMENTATION DETAILS

C.1 WORLD MODEL TRAINING AND EVALUATION

Environments We evaluate five text-based environments, including ALFWorld (Shridhar et al., 2021), SciWorld (Wang et al., 2022), TextWorld (Côté et al., 2018), WebShop (Yao et al., 2023a), and StableToolBench (Guo et al., 2025). Table 5 summarizes these environments along four dimensions: the nature of the environment, the abilities required of an agent, the form of the underlying world state, and the modeling capabilities demanded of a world model.

Environment	Description	Required Agent Ability	World Model State	Required World Model Ability
ALFWorld (Shridhar et al., 2021)	Embodied environment where agents accomplish household tasks by issuing text-based commands.	Spatial and physical commonsense, reasoning about containers and locations, and multi-step executions.	Room layout with hundreds of container–object combinations, agent inventory, and task progression.	Track physical configurations, maintain object relations, and predict stable multi-step state transitions.
SciWorld (Wang et al., 2022)	Text-based interactivate laboratory environment involving simplified physics & chemistry experiments.	Scientific concepts, causal reasoning, experiment planning, hypothesis testing with outcome evaluation.	Ten interconnected labs with $\sim$200 materials, intermediate substance states, and experiment progress.	Scientific dynamics modeling, physical reasoning, chemical simulation, experiment progress estimation.
TextWorld (Côté et al., 2018)	Text-based open-world environment supporting exploration, interaction, and diverse quest-like tasks.	Environment understanding, open-ended task planning, temporal tracking, and structured exploration.	Multiple connected rooms with $\sim$10 objects, exploration and discovery status, and task advancement.	Long-horizon state prediction, symbolic transition feedback, and exploration progress estimation.
WebShop (Yao et al., 2023a)	Simulated shopping website where agents search, browse, and shop through multi-step interactions.	Goal decomposition, product evaluation, and robust reasoning over diverse semi-structured attributes.	Metadata for over 1M product attributes, search-query items surface, item details, and cart states.	Simulation of search engines, multi-step web navigation, product attributes, and constraint satisfaction.
StableToolBench (Guo et al., 2025)	API-based tool-use environment requiring schema adherence and structured output generation.	Doc understanding, symbolic reasoning, and executing schema-compliant action sequences.	Over 10K API tools, input/output schemas, intermediate tool-call states, and execution context.	Symbolic world state simulation, doc understanding, schema constraint satisfaction, structured generation.

Table 5: Summary of the five text-based environments used in our paper, highlighting the knowledge demands placed on both agents and world models. Task examples are provided in Figures 14–18 in Appendix D.

Data Sources and Sizes For ALFWorld, SciWorld and WebShop, we follow the data splits provided in AgentGym[5] (Xi et al., 2024). For TextWorld, we follow the official TextWorld repository[6] to generate game files and randomly split them into 2.5K training games and 200 test games. For StableToolBench, we filtered the StableToolBench MirrorAPI dataset[7] and removed samples with errors or incomplete information, and used 160K API pairs for training and 2K pairs for testing. The data sizes for different environments are summarized in Table 6.

[5] `https://github.com/WooooDyy/AgentGym`

[6] `https://github.com/microsoft/TextWorld`

[7] `https://huggingface.co/datasets/stabletoolbench/MirrorAPI-Training`

Trajectories Collection We utilize the AgentGym (Xi et al., 2024) framework to collect long-horizon interaction trajectories using GPT-4o as the agent across four interactive environments: ALFWorld, SciWorld, TextWorld, and WebShop. We maintain consistent system prompts (Appendix F), interaction protocols, and environment configurations as in AgentGym. The sampling temperature is set to 1.0 with Top-p of 1.0, and a maximum of 50 interaction turns per trajectory. System prompts used for trajectory collection are provided in Figure 19 to 23. Ultimately, we collect 40K trajectories each for ALFWorld, SciWorld, and TextWorld, and 70K trajectories for WebShop on their respective training sets, as summarized in Table 6.

Environment	Train Games	Test Games	Trajectories
ALFWorld	2420	200	40K
SciWorld	2120	200	40K
TextWorld	2500	200	40K
WebShop	3930	200	70K
StableToolBench	160K	2000	None

Table 6: Training data sizes for different environments. StableToolBench only contains single-turn training data without interactive trajectories.

World Model Training Hyper-parameters We utilize LLaMa-Facotry[8] (Zheng et al., 2024) for SFT training of LLM-based world models. The training parameters are summarized in Table 7. Parameters unspecified in the table follow the default settings of LLaMA-Factory. Training data size varies across different environments, as detailed in Table 6 if not otherwise specified. Experiments are conducted on 4xH100-80GB GPUs.

Parameter	Value
Global Train Batch Size	128
Learning Rate	1.0e-5
Number of Training Epochs	5
LR Scheduler Type	Constant with Warmup
Warmup Steps	10
BF16	True
Max Gradient Norm	100

Table 7: SFT hyper-parameters for training LLM-based world models.

World Model Backbones We utilize Qwen2.5-7B (Qwen et al., 2025) and Llama3.1-8B (Grattafiori et al., 2024) as the primary backbone for LLM-based world models. To study the impact of model scale, we train Qwen2.5 models of four sizes: 0.5B, 1.5B, 3B, and 7B parameters. The specific model checkpoints used are as follows:

Model	Checkpoint URL
Qwen2.5-7B	`https://huggingface.co/Qwen/Qwen2.5-7B`
Qwen2.5-3B	`https://huggingface.co/Qwen/Qwen2.5-3B`
Qwen2.5-1.5B	`https://huggingface.co/Qwen/Qwen2.5-1.5B`
Qwen2.5-0.5B	`https://huggingface.co/Qwen/Qwen2.5-0.5B`
Llama3.1-8B	`https://huggingface.co/meta-llama/Llama-3.1-8B`

Table 8: Model checkpoints used for world model training.

API Models We list the API models and their versions used in the paper in Table 9.

C.2 Synthetic Data Competes with Real

To compare the quality of world-model–synthesized trajectories with those from the real environment, we construct matched SFT datasets using 1,000 successful trajectories collected from (i) the real

[8] `https://github.com/hiyouga/LLaMA-Factory`

Model	Version
GPT-4o-mini	gpt-4o-mini-2024-07-18
GPT-4o	gpt-4o-2024-11-20
GPT-4-turbo	gpt-4-turbo-2024-04-09
GPT-4.1	gpt-4.1-2025-04-14
GPT-5	gpt-5-2025-08-07
Gemini-2.5-flash	gemini-2.5-flash
Claude-sonnet-4.5	claude-sonnet-4-5-20250929
Qwen3-235B	qwen3-235b-a22b-instruct-2507

Table 9: API models and versions used for evaluations.

environment and (ii) the world model. To control for the behavior policy used during data collection, both datasets are generated by the same agent: a Qwen2.5-7B-Instruct policy trained via direct RL (i.e., without any SFT). This design avoids reusing the world-model training policy (GPT-4o) as the collector, thereby reducing the risk that the world model "self-replays" trajectories through the behavior policy. For world-model rollouts, trajectory success is determined by the model's own predicted outcome.

C.3 Early Experience for Policy Learning

Early Experience (WM-SFT) To provide early dynamics exposure before policy learning, we warm-start the agent with a *world-model style* supervised objective (Eq. 2): predicting the next environment response and termination signal conditioned on the dialogue history and the current action. We use the same data sources described in Appendix C.1 and sample 1,000 trajectories to construct the WM-SFT dataset. Training follows the same SFT hyper-parameters as world model training (Table 7). For the baseline without early experience, this stage is skipped.

Agent Warmup (Agent-SFT) After WM-SFT, we perform a standard policy warmup stage by supervised fine-tuning on real-environment trajectories collected in Appendix C.1. Specifically, we sample 1,000 trajectories and fine-tune the agent to generate its next turn (reasoning trace and action; Eq. 1) from the interaction history. We use the same SFT hyper-parameters as in Table 7.

Reinforcement Learning (Agent-RL) We utilize the AgentGymRL framework[9] (Xi et al., 2025) to run GRPO training for agent policy training, with the suggested hyper-parameters as suggested in the paper. The command is shown in Figure 8.

C.4 World Model Initialization Context

In *ALFWorld* and *SciWorld*, each game instance involves random initialization of the environment. For example, in *ALFWorld*, the positions and contents of objects within rooms vary, while in *SciWorld*, the connectivity of houses changes with each initialization. Consequently, even for humans, accurately predicting the next state of the environment based solely on task descriptions is challenging. Similar to RAWM (Yang et al., 2025), we include the initial state information of the environment for the world model's predictions. This design aligns with practical applications where the world model is used with knowledge of the initial environment state. In data synthesis scenarios, such random states can be sampled through similar random generation methods, enhancing the diversity and generalization capabilities of the world model. Examples of initial state information are provided in Figure 9 and 10. While *TextWorld* lacks full initial state information due to environment limits. *WebShop* and *StableToolBench* are inherently open environments where comprehensive initial state information cannot be provided, so they also do not include initial state information.

[9] `https://github.com/WooooDyy/AgentGym-RL`

```
python3 -m verl.agent_trainer.main_ppo \
  algorithm.adv_estimator=grpo \
  algorithm.rounds_ctrl.type=fixed \
  algorithm.rounds_ctrl.rounds=20 \
  data.train_file=${DATA_FILE} \
  data.train_batch_size=16 \
  data.max_prompt_length=1024 \
  data.max_response_length=4096 \
  actor_rollout_ref.agentgym.task_name=${TASK_NAME} \
  actor_rollout_ref.agentgym.env_addr=${ENV_ADDR} \
  actor_rollout_ref.agentgym.timeout=600 \
  actor_rollout_ref.model.path=${MODEL_PATH} \
  actor_rollout_ref.actor.use_kl_loss=True \
  actor_rollout_ref.actor.kl_loss_coef=0.001 \
  actor_rollout_ref.actor.kl_loss_type=low_var_kl \
  actor_rollout_ref.rollout.gpu_memory_utilization=0.6 \
  actor_rollout_ref.rollout.n=8 \
  actor_rollout_ref.rollout.max_model_len=32768 \
  actor_rollout_ref.rollout.max_tokens=200 \
  actor_rollout_ref.rollout.tensor_model_parallel_size=1 \
  actor_rollout_ref.actor.ppo_epochs=1 \
  actor_rollout_ref.actor.optim.lr=1e-6 \
  actor_rollout_ref.actor.ppo_mini_batch_size=8 \
  actor_rollout_ref.actor.ppo_micro_batch_size_per_gpu=1 \
  algorithm.kl_ctrl.kl_coef=0.001 \
  trainer.project_name="agentgym" \
  trainer.experiment_name="${EXPERIMENT_NAME}" \
  trainer.save_freq=10 \
  trainer.total_epochs=10 \
  trainer.n_gpus_per_node=4
```

Figure 8: Reinforcement Learning Command for Agent Training.

```
# Environment Information (Only visible to Assistant)

=== Objects on Receptacles ===
- cabinet 1 is closed, if opened, in it, you see nothing.
- cabinet 2 is closed, if opened, on the cabinet 2, you see a dishsponge 1
- cabinet 3 is closed, if opened, on the cabinet 3, you see a mug 1
- cabinet 4 is closed, if opened, in it, you see nothing.
- cabinet 5 is closed, if opened, on the cabinet 5, you see a soapbottle 3
- cabinet 6 is closed, if opened, on the cabinet 6, you see a plate 1
- On the cabinet 9, you see a mug 3
- On the countertop 1, you see a bread 1, a dishsponge 3, a egg 2, a fork 2, a mug 2, a pen 2, a peppershaker 1, a potato 1, a potato 2, a winebottle 2, a winebottle 3
- On the diningtable 1, you see a apple 1, a bowl 1, a bowl 2, a butterknife 1, a butterknife 2, a butterknife 3, a fork 1, a glassbottle 1, a glassbottle 2, a knife 1, a knife 2, a lettuce 1, a pen 1, a pot 1, a saltshaker 1, a saltshaker 2, a soapbottle 1, a soapbottle 2, a spatula 1, a spatula 2, a winebottle 1
- drawer 2 is closed, if opened, on the drawer 2, you see a dishsponge 2, a pencil 1
- fridge 1 is closed, if opened, on the fridge 1, you see a bowl 3, a cup 1, a lettuce 2, a tomato 1
- On the garbagecan 1, you see a apple 2, a egg 1
- microwave 1 is closed, if opened, on the microwave 1, you see a apple 3
- On the sinkbasin 1, you see a glassbottle 3, a spoon 1
- On the stoveburner 1, you see a pan 1
- On the stoveburner 3, you see a pan 2
- On the stoveburner 4, you see a pan 2

# User Environment Information (Displayed to User)

You are in the middle of a room. Looking quickly around you, you see a cabinet 10, a cabinet 9, a cabinet 8, a cabinet 7, a cabinet 6, a cabinet 5, a cabinet 4, a cabinet 3, a cabinet 2, a cabinet 1, a coffeemachine 1, a countertop 1, a diningtable 1, a drawer 2, a drawer 1, a fridge 1, a garbagecan 1, a microwave 1, a sinkbasin 1, a stoveburner 4, a stoveburner 3, a stoveburner 2, a stoveburner 1, and a toaster 1.
Your task is to: cool some mug and put it in coffeemachine.
AVAILABLE ACTIONS: go to cabinet 1,go to cabinet 10,go to cabinet 2,go to cabinet 3,go to cabinet 4,go to cabinet 5,go to cabinet 6,go to cabinet 7,go to cabinet 8,go to cabinet 9,go to coffeemachine 1,go to countertop 1,go to diningtable 1,go to drawer 1,go to drawer 2,go to fridge 1,go to garbagecan 1,go to microwave 1,go to sinkbasin 1,go to stoveburner 1,go to stoveburner 2,go to stoveburner 3,go to stoveburner 4,go to toaster 1,help,inventory,look
```

Figure 9: Initialization Context Example of ALFWorld

Environment Information (Only visible to Assistant)

=== Goal Progress ===
Completed keys:

Sequential Subgoals:

---

0 false GoalFind focus on thermometer
1 false GoalFind focus on substance
2 false GoalFindAnswerBox focus on correct answer box

Unordered and Optional Subgoals:

---

0 false GoalInRoomWithObject be in same location as thermometer
1 false GoalSpecificObjectInDirectContainer have thermometer in inventory
2 false GoalMoveToNewLocation move to a new location
3 false GoalMoveToLocation move to the location asked by the task (substance location)
4 false GoalMoveToLocation move to the location asked by the task (answer box location)
5 false GoalMoveToLocation move to a location with a heating device (kitchen)
6 false GoalMoveToLocation move to a location with a heating device (outside)
7 false GoalMoveToLocation move to a location with a heating device (foundry)
8 false GoalSpecificObjectInDirectContainer have task object in inventory
9 false GoalPastActionUseObjectOnObject use thermometer on substance
10 false GoalPastActionUseObjectOnObject use thermometer on substance (after it has been heated)
11 false GoalObjectsInSingleContainer have substance alone in a single container
12 false GoalActivateDeviceWithName activate heater (stove)
13 false GoalActivateDeviceWithName activate heater (blast furnace)
14 false GoalActivateDeviceWithName activate heater (oven)
15 false GoalActivateDeviceWithName activate heater (hot plate)
16 false GoalSpecificObjectInDirectContainer have lighter in inventory
17 false GoalSpecificObjectInDirectContainer move wood into fire pit
18 false GoalTemperatureOnFire ignite wood
19 false GoalObjectInContainer have substance on heater (stove)
20 false GoalObjectInContainer have substance on heater (blast furnace)
21 false GoalObjectInContainer have substance on heater (oven)
22 false GoalObjectInContainer have substance on heater (hot plate)
23 false GoalObjectInContainer have substance on heater (fire pit)
24 false GoalTemperatureIncrease heat substance by at least 20C

=== Possible Actions ===
- activate OBJ
- close OBJ
- connect OBJ to OBJ
- deactivate OBJ
- disconnect OBJ
- dunk OBJ in OBJ
- eat OBJ
- flush OBJ
- focus on OBJ
- go OBJ
- inventory
- look around
- look at OBJ
- look in OBJ
- mix OBJ
- move OBJ to OBJ
- open OBJ
- pick up OBJ
- pour OBJ in OBJ
- put down OBJ
- read OBJ
- reset task
- task
- teleport OBJ
- use OBJ on OBJ
- wait
- wait1

Figure 10: Initialization Context Example of SciWorld

```
=== Per-Room Observations ===
== Room: terminal 2 ==
This room is called the bathroom. In it, you see:
the agent
a substance called air
a bathtub, which is turned off. In the bathtub is: nothing.
a glass cup (containing nothing)
a picture
a sink, which is turned off. In the sink is: nothing.
a toilet. In the toilet is: A drain, which is open, a substance called water.
You also see:
A door to the kitchen (that is closed)
Possible Objects: agent, air, bathroom, bathtub, cup, door, kitchen, picture, sink, substance in toilet, toilet

== Room: kitchen ==
This room is called the bathroom. In it, you see:
the agent
a substance called air
a bathtub, which is turned off. In the bathtub is: nothing.
a glass cup (containing nothing)
a picture
a sink, which is turned off. In the sink is: nothing.
a toilet. In the toilet is: A drain, which is closed, a substance called water.
You also see:
A door to the kitchen (that is closed)
Possible Objects: agent, air, bathroom, bathtub, cup, door, kitchen, picture, sink, substance in toilet, toilet

== Room: sewer ==
This room is called the bathroom. In it, you see:
the agent
a substance called air
a bathtub, which is turned off. In the bathtub is: nothing.
a glass cup (containing nothing)
a picture
a sink, which is turned off. In the sink is: nothing.
a toilet. In the toilet is: A drain, which is closed, a substance called water.
You also see:
A door to the kitchen (that is closed)
Possible Objects: agent, air, bathroom, bathtub, cup, door, kitchen, picture, sink, substance in toilet, toilet

== Room: bedroom ==
This room is called the bedroom. In it, you see:
the agent
a substance called air
a bed. On the bed is: a mattress. On the mattress is: a white pillow..
a book shelf (containing nothing)
a closet. The closet door is closed.
a drawing
a table. On the table is: nothing.
You also see:
A door to the hallway (that is closed)
Possible Objects: agent, air, bed, bedroom, book shelf, closet, cloth sittable, door, drawing, hallway, object, table

== Room: art studio ==
This room is called the art studio. In it, you see:
the agent
a substance called air
a large cupboard. The large cupboard door is closed.
a table. On the table is: a jug (containing nothing).
a wood cup (containing yellow paint)
a wood cup (containing blue paint)
a wood cup (containing red paint)
You also see:
A door to the hallway (that is closed)
Possible Objects: agent, air, art studio, blue paint, cup containing blue paint, cup containing red paint, cup containing yellow paint,
cupboard, door, hallway, jug, paint in cup containing red paint, paint in cup containing yellow paint, table

== Room: foundry ==
This room is called the foundry. In it, you see:
the agent
a substance called air
a blast furnace, which is turned off. The blast furnace door is closed.
a sink, which is turned off. In the sink is: nothing.
a table. On the table is: nothing.
```

Figure 11: Initialization Context Example of SciWorld (Continued)

You also see:
A door to the outside (that is closed)
Possible Objects: agent, air, blast furnace, door, foundry, outside, sink, steel table

== Room: workshop ==
This room is called the workshop. In it, you see:
the agent
a substance called air
a table. On the table is: a battery, a black wire, a blue light bulb, which is off, a green light bulb, which is off, a red wire, a switch, which is off, a violet light bulb, which is off, a yellow wire.
a ultra low temperature freezer. The ultra low temperature freezer door is closed.
You also see:
A door to the hallway (that is closed)
Possible Objects: agent, air, anode in battery, anode in blue light bulb, anode in component, anode in green light bulb, anode in violet light bulb, battery, battery cathode, black wire, black wire terminal 1, black wire terminal 2, blue light bulb, blue light bulb cathode, cathode in component, cathode in green light bulb, cathode in violet light bulb, door, freezer, green light bulb, hallway, red wire, red wire terminal 1, red wire terminal 2, switch, table, terminal 1 in yellow wire, terminal 2 in yellow wire, violet light bulb, workshop, yellow wire

== Room: hallway ==
This room is called the hallway. In it, you see:
the agent
a substance called air
a drawing
You also see:
A door to the art studio (that is closed)
A door to the bedroom (that is closed)
A door to the greenhouse (that is closed)
A door to the kitchen (that is closed)
A door to the living room (that is closed)
A door to the workshop (that is closed)
Possible Objects: agent, air, art studio, art studio door, bedroom, bedroom door, door to greenhouse, door to kitchen, door to living room, door to workshop, drawing, greenhouse, hallway, kitchen, living room, workshop

== Room: bathroom ==
This room is called the bathroom. In it, you see:
the agent
a substance called air
a bathtub, which is turned off. In the bathtub is: nothing.
a glass cup (containing nothing)
a picture
a sink, which is turned off. In the sink is: nothing.
a toilet. In the toilet is: A drain, which is closed, a substance called water.
You also see:
A door to the kitchen (that is closed)
Possible Objects: agent, air, bathroom, bathtub, cup, door, kitchen, picture, sink, substance in toilet, toilet

== Room: terminal 1 ==
This room is called the bathroom. In it, you see:
the agent
a substance called air
a bathtub, which is turned off. In the bathtub is: nothing.
a glass cup (containing nothing)
a picture
a sink, which is turned off. In the sink is: nothing.
a toilet. In the toilet is: A drain, which is closed, a substance called water.
You also see:
A door to the kitchen (that is closed)
Possible Objects: agent, air, bathroom, bathtub, cup, door, kitchen, picture, sink, substance in toilet, toilet

== Room: living room ==
This room is called the bathroom. In it, you see:
the agent
a substance called air
a bathtub, which is turned off. In the bathtub is: nothing.
a glass cup (containing nothing)
a picture
a sink, which is turned off. In the sink is: nothing.
a toilet. In the toilet is: A drain, which is closed, a substance called water.
You also see:
A door to the kitchen (that is closed)
Possible Objects: agent, air, bathroom, bathtub, cup, door, kitchen, picture, sink, substance in toilet, toilet

Figure 12: Initialization Context Example of SciWorld (Continued)

== Room: outside ==
This outside location is called the outside. Here you see:
the agent
a substance called air
an axe
a fire pit (containing nothing)
a fountain (containing a substance called water)
the ground
a shovel
a substance called wood
You also see:
A door to the foundry (that is closed)
A door to the greenhouse (that is closed)
A door to the kitchen (that is closed)
Possible Objects: agent, air, axe, door to foundry, door to greenhouse, door to kitchen, fire pit, foundry, fountain, greenhouse, ground, kitchen, outside, shovel, substance in fountain, wood

== Room: greenhouse ==
This room is called the greenhouse. In it, you see:
the agent
a substance called air
a bee hive. The bee hive door is closed.
a jug (containing nothing)
a sink, which is turned off. In the sink is: nothing.
You also see:
A door to the hallway (that is closed)
A door to the outside (that is closed)
Possible Objects: agent, air, bee hive, door to hallway, door to outside, greenhouse, hallway, jug, outside, sink

User Environment Information (Displayed to User)

Your task is to measure the melting point of tin, which is located around the kitchen. First, focus on the thermometer. Next, focus on the tin. If the melting point of tin is above 200.0 degrees celsius, focus on the blue box. If the melting point of tin is below 200.0 degrees celsius, focus on the orange box. The boxes are located around the kitchen.
This room is called the bathroom. In it, you see:
the agent
a substance called air
a bathtub, which is turned off. In the bathtub is: nothing.
a glass cup (containing nothing)
a picture
a sink, which is turned off. In the sink is: nothing.
a toilet. In the toilet is: A drain, which is open, a substance called water.
You also see:
A door to the kitchen (that is closed)

Figure 13: Initialization Context Example of SciWorld (Continued)

D Task Examples and Case Studies

We provide task examples and case studies on world model across five environments in Figures 14–18.

Instruction: API doc:

{
"api_name": "SWIFT codes by country",
"api_description": "Returns list of SWIFT codes for a country (use two letter ISO country code)",
"required_parameters": [
{ "name": "country_code", "type": "string", "description": ", "default": "AU" }
],
"optional_parameters": [ ],
"tool_description": "Lookup country banks and their SWIFT codes",
"tool_name": "Bank SWIFT codes",
"tool_category": "Financial"
}

Request:

{
"category": "Financial",
"tool_name": "Bank SWIFT codes",
"api_name": "SWIFT codes by country",
"tool_input": "{"country_code": "JP"}",
"strip": "filter"
}

Real Trajectory

{
"error": "",
"response": "{ 'data': [
{ 'country': 'JP', 'code': 'FCTOJPJ1XXX', 'bank': 'ABN AMRO CLEARING TOKYO CO., LTD.' },
{ 'country': 'JP', 'code': 'ACSRJPJ1XXX', 'bank': 'ACE SECURITIES CO., LTD.' },
{ 'country': 'JP', 'code': 'ADCTJPJJXXX', 'bank': 'ADEKA CORPORATION' },
{ 'country': 'JP', 'code': 'AEONJPJ1XXX', 'bank': 'AEON BANK, LTD.' },
{ 'country': 'JP', 'code': 'AEONJPJTXXX', 'bank': 'AEON BANK, LTD.' }...
] }"
}

World Model Trajectory

{
"error": "",
"response": "{ 'data': [
{ 'country': 'JP', 'code': '01234567', 'bank': 'MUKAMAL BANK' },
{ 'country': 'JP', 'code': '09876543', 'bank': 'NIPPON BANK' },
{ 'country': 'JP', 'code': '12345678', 'bank': 'YAMATO BANK' },
{ 'country': 'JP', 'code': '87654321', 'bank': 'TOHOKU BANK' }
] }"
}

Figure 14: Task Example and Case Study of StableToolBench

E Detailed Results

We provide detailed results in Table 10 for the OOD generalization of world models.

F System Prompts for Agent Trajectory Collection

We provide system prompts for agent trajectory collection in Figure 19 to 23.

Instruction: You are in the middle of a room. Looking quickly around you, you see a armchair 1, a bed 1, a diningtable 1, a drawer 2, a drawer 1, a garbagecan 1, a sidetable 2, and a sidetable 1.
Your task is to: put some pillow on armchair.
AVAILABLE ACTIONS: go to armchair 1,go to bed 1,go to diningtable 1,go to drawer 1,go to drawer 2,go to garbagecan 1,go to sidetable 1,go to sidetable 2,help,inventory,look

Real Trajectory

Thought 1: My task is to put a pillow on the armchair. Before proceeding, I should first find out where the pillows are located. A likely place could be the bed or the sidetables. Starting with checking the bed might be a good idea as pillows are commonly found there.

Action 1: go to bed 1

Observation 1: You arrive at bed 1. On the bed 1, you see a book 2, a laptop 1, a pillow 2, and a pillow 1.

Thought 2: I have found two pillows on the bed. My task requires putting a pillow on the armchair. I need to pick up one of the pillows and then go to the armchair.

Action 2: inventory

Observation 2: You are not carrying anything.

Thought 3: I need to pick up a pillow from the bed to carry it to the armchair. Once I have the pillow, I'll proceed to go to the armchair to complete the task.

Action 3: pick up pillow 1

Observation 3: Nothing happens.

Thought 4: The action to pick up the pillow didn't work, suggesting perhaps it's not a valid action at this moment, or I need to utilize another approach to be able to pick up items. I should consult the help to see if there's any guidance on how to interact with objects in the environment.

Action 4: help

Observation 4: Available commands: look: look around your current location. inventory: check your current inventory. go to (receptacle): move to a receptacle. open (receptacle): open a receptacle...

Thought 5: Based on the available commands, I should use the "take (object) from (receptacle)" command to pick up a pillow from the bed. I will proceed with the correct format to take a pillow.

Action 5: take pillow 1 from bed 1

Observation 5: You pick up the pillow 1 from the bed 1.

Thought 6: I successfully picked up a pillow. Now, I need to go to the armchair to place the pillow on it, completing the task.

Action 6: go to armchair 1

Observation 6: You arrive at armchair 1. On the armchair 1, you see nothing.

Thought 7: I'm at the armchair where I need to place the pillow. I will use the m̈ove (object) to (receptacle)c̈ommand to put the pillow on the armchair.

Action 7: move pillow 1 to armchair 1

Observation 7: You move the pillow 1 to the armchair 1.

Success: True

World Model Trajectory

Thought 1: In order to complete the task of putting a pillow on the armchair, I need to first find a pillow. Pillows are most likely to be found on a bed, in a drawer, or on a sidetable. I should explore these areas to locate a pillow and then proceed to the armchair to place it there.

Action 1: go to bed 1

Observation 1: You arrive at bed 1. On the bed 1, you see a book 2, a laptop 1, a pillow 2, and a pillow 1.

Thought 2: I have found pillows on the bed, which is helpful in achieving the task. I need to pick up one of the pillows so I can then place it on the armchair.

Action 2: inventory

Observation 2: You are not carrying anything.

Thought 3: It's confirmed that I am not carrying anything. To complete the task, I should pick up one of the pillows from the bed and then make my way to the armchair.

Action 3: Take pillow 1

Observation 3: Nothing happens.

Thought 4: It appears that "Take pillow 1" is not a valid action, or perhaps it is not the correct format for the given scenario. I will revise my approach and check if there is a specific action for picking up items or if I need to use a different method.

Action 4: help

Observation 4: Available commands: look: look around your current location. inventory: check your current inventory. go to (receptacle): move to a receptacle. open (receptacle): open a receptacle...

Thought 5: Now that I have the list of available commands, I see that I need to use the "take (object) from (receptacle)" command to pick up a pillow from the bed. I will proceed to take one of the pillows.

Action 5: take pillow 1 from bed 1

Observation 5: You pick up the pillow 1 from the bed 1.

Thought 6: I now have pillow 1 in my possession. The next step is to go to the armchair so that I can place the pillow on it to achieve the task goal.

Action 6: go to armchair 1

Observation 6: You arrive at armchair 1. On the armchair 1, you see nothing.

Thought 7: I am now at the armchair with pillow 1 in my inventory. The final step is to place the pillow onto the armchair to complete the task.

Action 7: move pillow 1 to armchair 1

Observation 7: You move the pillow 1 to the armchair 1

Success: True

Figure 15: Task Example and Case Study of ALFWorld

Instruction: Your task is to measure the melting point of tin, which is located around the kitchen. First, focus on the thermometer. Next, focus on the tin. If the melting point of tin is above 50.0 degrees celsius, focus on the yellow box. If the melting point of tin is below 50.0 degrees celsius, focus on the purple box. The boxes are located around the kitchen.
This room is called the living room. In it, you see:
the agent
a substance called air
a book shelf (containing A book (Sherlock Holmes) titled Sherlock Holmes by Arthur Conan Doyle)
a chair. On the chair is: nothing.
a couch. On the couch is: a white pillow.
a finger painting
a table. On the table is: nothing.
You also see:
A door to the hallway (that is closed)

Real Trajectory

Thought 1: To begin the task, I need to find the thermometer. The thermometer will likely be in the same area as the tin or in a place where temperature measurements are common, such as a kitchen or storage.

Action 1: go to hallway

Observation 1: The door is not open.

Thought 2: Since the door is closed, I should open it first to proceed to the hallway and eventually to the kitchen.

Action 2: open door to hallway

Observation 2: The door is now open.

Thought 3: Now that the door to the hallway is open, I can proceed through it in order to reach the kitchen and continue with the task.

Action 3: go to hallway

Observation 3: You move to the hallway.

Thought 4: Now that I'm in the hallway, I need to find the kitchen where the thermometer and tin are likely located.

Action 4: go to kitchen

Observation 4: The door is not open.

Thought 5: I need to open the door to the kitchen in order to enter and find the thermometer and tin.

Action 5: open door to kitchen

Observation 5: The door is now open.

Thought 6: Now that the door to the kitchen is open, I can proceed into the kitchen to locate the thermometer and the tin.

Action 6: go to kitchen

Observation 6: You move to the kitchen.

Thought 7: Now that I'm in the kitchen, I need to find and focus on the thermometer as the first step of the task.

Action 7: focus on thermometer

Observation 7: You focus on the thermometer.

Thought 8: I have focused on the thermometer. The next step is to focus on the tin to identify its melting point.

Action 8: focus on tin

Observation 8: You focus on the tin.

Thought 9: Now that I have focused on the tin, I need to determine its melting point. The melting point of tin is a known value, which is approximately 231.9 degrees Celsius.

Action 9: focus on yellow box

Observation 9: You focus on the yellow box.

Success: True

World Model Trajectory

Thought 1: I am currently in the living room and need to measure the melting point of tin, which is located in the kitchen. I need to find my way to the kitchen first.

Action 1: go to hallway

Observation 1: The door is not open.

Thought 2: I must open the door to the hallway so I can access the kitchen where the tin is located.

Action 2: open door to hallway

Observation 2: The door is now open.

Thought 3: Now that the door to the hallway is open, I can proceed through it to find the kitchen.

Action 3: go to hallway

Observation 3: You move to the hallway.

Thought 4: I need to continue towards the kitchen to locate the tin and thermometer.

Action 4: go to kitchen

Observation 4: The door is not open.

Thought 5: I need to open the door to the kitchen to proceed with the task.

Action 5: open door to kitchen

Observation 5: The door is now open.

Thought 6: Now that the kitchen door is open, I can enter the kitchen to look for the thermometer and tin.

Action 6: go to kitchen

Observation 6: You move to the kitchen.

Thought 7: I have reached the kitchen. My first task is to focus on the thermometer to eventually measure the melting point of tin.

Action 7: focus on thermometer

Observation 7: You focus on the thermometer.

Thought 8: I've focused on the thermometer. The next step is to focus on the tin to find out its melting point.

Action 8: focus on tin

Observation 8: You focus on the tin.

Thought 9: Now that I have focused on the tin, I need to determine its melting point. Based on its value, I will focus on either the yellow box or the purple box.

Action 9: focus on yellow box

Observation 9: You focus on the yellow box.

Success: True

Figure 16: Task Example and Case Study of SciWorld

Instruction: Hey, thanks for coming over to the TextWorld today, there is something I need you to do for me. First of all, you could, like, look and see that the antique trunk inside the bedroom is opened. Then, recover the old key from the antique trunk. Then, make absolutely sure that the wooden door inside the bedroom is unlocked. After unlocking the wooden door, open the wooden door in the bedroom. Then, try to head east. After that, try to travel south. Once you get through with that, take the milk from the couch within the living room. Having taken the milk, attempt to travel north. That done, rest the milk on the stove inside the kitchen. And if you do that, you're the winner!
-= Bedroom =-
You make a grand eccentric entrance into a bedroom.You see a closed normal looking chest drawer right there by you. You can see an antique trunk. You see a king-size bed. But the thing is empty.There is a closed wooden door leading east.
-= Bedroom =-0/1
AVAILABLE ACTIONS: examine antique trunk, examine chest drawer, examine king-size bed, examine wooden door, inventory, look, open antique trunk, open chest drawer

Real Trajectory

Thought 1: The description mentioned an antique trunk inside the bedroom that needs to be opened first. Since one of the admissible actions is to open the antique trunk, this should be the first step to proceed with the tasks.

Action 1: open antique trunk

Observation 1: You open the antique trunk, revealing an old key.
-= Bedroom =-0/2

Thought 2: Now that the antique trunk is open and the old key is revealed, the next step is to recover the old key from the antique trunk.

Action 2: take old key

Observation 2: You take the old key from the antique trunk.
-= Bedroom =-0/3

Thought 3: With the old key in hand, the next step is to ensure that the wooden door inside the bedroom is unlocked. Examining the wooden door might be necessary to unlock it.

Action 3: examine wooden door

Observation 3: it is what it is, a wooden door You can't see inside it because the lid's in your way.
-= Bedroom =-0/4

Thought 4: The door needs to be unlocked using the old key we obtained. The next logical step would be to try to unlock the door.

Action 4: unlock wooden door with old key

Observation 4: You unlock wooden door.
-= Bedroom =-0/5

Thought 5: The wooden door has been unlocked. The next step is to open the door so that you can proceed to head east.

Action 5: open wooden door

Observation 5: You open wooden door.
-= Bedroom =-0/6

Thought 6: With the wooden door now open, the next step is to proceed through the door to the east as mentioned in the instructions.

Action 6: go east

Observation 6: -= Kitchen =-
Well how about that, you are in a place we're calling a kitchen. The room is well lit.
You make out a closed refrigerator nearby. You can see a counter. The counter is normal. But the thing hasn't got anything on it. You rest your hand against a wall, but you miss the wall and fall onto a a stove. Looks like ...

[More steps omitted for brevity]

Success: True

World Model Trajectory

Thought 1: The first step is to open the antique trunk to check if it contains the old key.

Action 1: open antique trunk

Observation 1: You open the antique trunk, revealing an old key.
-= Bedroom =-0/2

Thought 2: Now that the antique trunk is open and the old key is visible, the next step is to take the old key from the trunk.

Action 2: take old key

Observation 2: You take the old key from the antique trunk.
-= Bedroom =-0/3

Thought 3: With the old key in hand, the next task is to unlock the wooden door in the bedroom.

Action 3: examine wooden door

Observation 3: The wooden door looks well-built. You can't see inside it because the lid's in your way.
-= Bedroom =-0/4

Thought 4: I attempted to examine the wooden door, but I need to unlock it now using the old key to proceed with the tasks.

Action 4: unlock wooden door with old key

Observation 4: You unlock wooden door.
-= Bedroom =-0/5

Thought 5: The wooden door is now unlocked, and the next step is to open it so that I can move to the next room.

Action 5: open wooden door

Observation 5: You open wooden door.
-= Bedroom =-0/6

Thought 6: The wooden doorf is open, so the next step is to go through it by heading east.

Action 6: go east

Observation 6: -= Kitchen =-
You are in a kitchen. A standard one.
You can see a refrigerator. You can make out a counter. The counter is normal. But the thing is empty. You can make out a stove. The stove is usual. But the thing is empty, unfortunately. You can make out a kitchen island. The kitchen island is normal. On the kitchen island you can make out a note.
There is a closed screen door leading east. There is an open wooden door leading west. You need an unblocked exit? You should try going north. You need an unguarded exit? You should try going south.
Your score has just gone up by one point.
-= Kitchen =-1/7

[More steps omitted for brevity]

Success: True

Figure 17: Task Example and Case Study of TextWorld.

Instruction: Find me men's sleep & lounge with long sleeve, elastic waistband for daily wear with color: multi 7, and size: x-large, and price lower than 80.00 dollars [SEP] Search
Your admissible actions of the current situation are: ['search[<your query>]','click[search]'].

Real Trajectory

Thought 1: I need to search for men's sleep & lounge wear that has long sleeves, an elastic waistband, is designated for daily wear, colored multi 7, size x-large, and priced under $80.

Action 1: search[mens sleep lounge long sleeve elastic waistband multi 7 x-large under 80]

Observation 1: Back to Search [SEP] Page 1 (Total results: 50) [SEP] Next > [SEP] B09ND8P2QR [SEP] InterestPrint Deer and Stars Men's Loungewear Pajama Sets - Long Sleeve Tee and Jogger Pant [SEP] $43.59 [SEP]... Your admissible actions of the current situation are: ['click[back to search]','click[next >]','click[b09nd8p2qr]','click[b09nd9dp7j]','click[b09qw2...

Thought 2: I found a men's loungewear set, B09ND8P2QR, that has long sleeves and is within the price range of under $80. It seems to be a potential match and worth investigating further.

Action 2: click[b09nd8p2qr]

Observation 2: Back to Search [SEP] < Prev [SEP] size [SEP] small [SEP] medium [SEP] large [SEP] x-large [SEP] xx-large [SEP] color [SEP] multi 1 [SEP] multi 10 [SEP] multi 2 [SEP]... Your admissible actions of the current situation are: ['click[back to search]','click[< prev]','click[description]','click[features]','click[reviews]'...

Thought 3: I need to select the correct size and color for the loungewear to match the requirements. The color should be multi 7, and the size should be x-large.

Action 3: click[x-large]

Observation 3: Back to Search [SEP] < Prev [SEP] size [SEP] small [SEP] medium [SEP] large [SEP] x-large [SEP] xx-large [SEP] color [SEP] multi 1 [SEP] multi 10 [SEP] multi 2 [SEP] multi 3... Your admissible actions of the current situation are: ['click[back to search]','click[< prev]','click[description]','click[features]','click[reviews]'...

Thought 4: I need to select the color "multi 7" to fully match the criteria provided.

Action 4: click[multi 7]

Observation 4: Back to Search [SEP] < Prev [SEP] size [SEP] small [SEP] medium [SEP] large [SEP] x-large [SEP] xx-large [SEP] color [SEP] multi 1 [SEP] multi 10 [SEP] multi 2... Your admissible actions of the current situation are: ['click[back to search]','click[< prev]','click[description]','click[features]','click[reviews]'...

[More steps omitted for brevity]

Thought 7: I have selected the correct size (x-large) and color (multi 7) for the loungewear. Now, it is time to proceed with purchasing the item.

Action 7: click[buy now]

Observation 7: Thank you for shopping with us! [SEP] Your code: [SEP] None [SEP] (Paste it in your MTurk interface.) [SEP] Purchased [SEP] asin [SEP] B09ND8P2QR [SEP] options [SEP] "color": "multi 7", "size": "x-large" [SEP] attrs [SEP] None [SEP] category [SEP] None [SEP] query [SEP] None [SEP] product category [SEP] None [SEP] Target [SEP] asin [SEP] options [SEP] attrs [SEP] price upper [SEP] instuction text...

Success: True

World Model Trajectory

Thought 1: I need to find men's sleep & lounge wear with specific attributes. It's best to search using the keywords related to the product, including category, features, and price.

Action 1: search[mens sleep lounge long sleeve elastic waistband multi 7 x-large under 80]

Observation 1: Back to Search [SEP] Page 1 (Total results: 50) [SEP] Next > [SEP] B09ND8P2QR [SEP] InterestPrint Deer and Stars Men's Loungewear Pajama Sets - Long Sleeve Tee and Jogger Pant [SEP] $43.59 [SEP] B09ND9DP7J [SEP] InterestPrint Gold Horse Pattern Men's 2-Piece Sleepwear Set, Long Sleeve Shirt with...

Thought 2: Based on the search result list, I need to identify an appropriate item that matches the criteria: men's sleep & lounge wear with long sleeves, an elastic waistband, multi 7 color, size x-large, and priced under $80.

Action 2: click[B09ND8P2QR]

Observation 2: Back to Search [SEP] < Prev [SEP] size [SEP] small [SEP] medium [SEP] large [SEP] x-large [SEP] xx-large [SEP] color [SEP] multi 1 [SEP] multi 10 [SEP] multi 2... Your admissible actions of the current situation are: ['click[back to search]', 'click[< prev]', 'click[description]', 'click[features]', 'click[reviews]'...

Thought 3: First, I need to ensure the selected item is available in the correct size (x-large) and color (multi 7) to match the criteria.

Action 3: click[x-large]

Observation 3: Back to Search [SEP] < Prev [SEP] size [SEP] small [SEP] medium [SEP] large [SEP] x-large [SEP] xx-large [SEP] color [SEP] multi 1 [SEP] multi 10 [SEP] multi 2 [SEP]... Your admissible actions of the current situation are: ['click[back to search]', 'click[< prev]', 'click[description]', 'click[features]', 'click[reviews]'...

Thought 4: Now, I need to confirm the color option 'multi 7' to ensure it matches the criteria.

Action 4: click[multi 7]

Observation 4: Back to Search [SEP] < Prev [SEP] size [SEP] small [SEP] medium [SEP] large [SEP] x-large [SEP] xx-large [SEP] color [SEP] multi 1 [SEP] multi 10 [SEP] multi 2... Your admissible actions of the current situation are: ['click[back to search]', 'click[< prev]', 'click[description]', 'click[features]', 'click[reviews]'...

[More steps omitted for brevity]

Thought 7: Since both the desired size and color have been confirmed, the next step is to proceed with purchasing the item.

Action 7: click[buy now]

Observation 7: Thank you for shopping with us! [SEP] Your code: [SEP] None [SEP] (Paste it in your MTurk interface.) [SEP] Purchased [SEP] asin [SEP] B09ND8P2QR [SEP] options [SEP] "color": "multi 7", "size": "x-large" [SEP] attrs [SEP] None [SEP] category [SEP] None [SEP] query [SEP] None [SEP] product category [SEP] None [SEP] Target [SEP] asin [SEP] options [SEP] attrs [SEP] price upper [SEP] instuction text...

Success: True

Figure 18: Task Example and Case Study of WebShop

Interact with a household to solve a task.
Imagine you are an intelligent agent in a household environment and your target is to perform actions to complete the task goal. At the beginning of your interactions, you will be given the detailed description of the current environment and your goal to accomplish. For each of your turn, you will be given a list of actions which you can choose one to perform in this turn. Now it's your turn to take an action. You should first think about the current condition and plan for your future actions, and then output your action in this turn. Your output must strictly follow this format: "Thought:
your thoughts.
Action:
your next action"
After your each turn, the environment will give you immediate feedback based on which you plan your next few steps. if the envrionment output "Nothing happened", that means the previous action is invalid and you should try more options.
Reminder: the action must be chosen from the given available actions. Any actions except provided available actions will be regarded as illegal.

Figure 19: Agent System Prompt for ALFWorld Trajectory Collection

You are an agent for science world. Every round I will give you an observation, you have to respond an action based on the observation to finish the given task.
Here are the actions you may take: [
{"action": "open/close OBJ", "description": "open/close a container"},
{"action": "de/activate OBJ", "description": "activate/deactivate a device"},
{"action": "connect OBJ to OBJ", "description": "connect electrical components"},
{"action": "disconnect OBJ", "description": "disconnect electrical components"},
{"action": "use OBJ [on OBJ]", "description": "use a device/item"},
{"action": "look around", "description": "describe the current room"},
{"action": "look at OBJ", "description": "describe an object in detail"},
{"action": "look in OBJ", "description": "describe a container's contents"},
{"action": "read OBJ", "description": "read a note or book"},
{"action": "move OBJ to OBJ", "description": "move an object to a container"},
{"action": "pick up OBJ", "description": "move an object to the inventory"},
{"action": "put down OBJ", "description": "drop an inventory item"},
{"action": "pour OBJ into OBJ", "description": "pour a liquid into a container"},
{"action": "dunk OBJ into OBJ", "description": "dunk a container into a liquid"},
{"action": "mix OBJ", "description": "chemically mix a container"},
{"action": "go to LOC", "description": "move to a new location"},
{"action": "eat OBJ", "description": "eat a food"},
{"action": "flush OBJ", "description": "flush a toilet"},
{"action": "focus on OBJ", "description": "signal intent on a task object"},
{"action": "wait", "description": "take no action for 10 iterations"},
{"action": "wait1", "description": "take no action for 1 iteration"},
{"action": "examine OBJ", "description": "provides a description of the objects present on or in a receptacle."},
{"action": "task", "description": "describe current task"},
{"action": "inventory", "description": "list your inventory"}
]
Your response should use the following format: "Thought:"
your thoughts.
"Action:"
your next action

Figure 20: Agent System Prompt for SciWorld Trajectory Collection

You are playing a text-based interactive fiction game (TextWorld).
You will receive observations describing the current state. When available, a list of admissible actions may be provided.
Always output strictly in the following format:
"Thought:
<your reasoning>
Action:
<the single action to take>"
Guidelines:
- Prefer actions from admissible commands when provided.
- If no list is provided, issue a valid single command (e.g., "look", "inventory", "open door", "go north", "take key").
- Avoid invalid or multiple actions in one step.

Figure 21: Agent System Prompt for TextWorld Trajectory Collection

You are web shopping.
I will give you instructions about what to do.
You have to follow the instructions.
Every round I will give you an observation and a list of available actions, you have to respond an action based on the state and instruction.
You can use search action if search is available.
You can click one of the buttons in clickables.
An action should be of the following structure:
search[keywords]
click[value]
If the action is not valid, perform nothing.
Keywords in search are up to you, but the value in click must be a value in the list of available actions.
Remember that your keywords in search should be carefully designed.
Your response should use the following format: "Thought:
I think ...

Action:
click[something]"

Figure 22: Agent System Prompt for WebShop Trajectory Collection

Imagine you are an API Server operating within a specialized tool, which contains a collection of distinct APIs. Your role is to deeply understand the function of each API based on their descriptions in the API documentation. As you receive specific inputs for individual API calls within this tool, analyze these inputs to determine their intended purpose. Your task is to craft a JSON formatted response that aligns with the expected output of the API. The JSON scheme is:
{
"error": "",
"response": ""
}

The error field should remain empty, indicating no errors in processing. The response field should contain the content you formulate based on the API's functionality and the input provided. Ensure that your responses are meaningful, directly addressing the API's intended functionality.
The key is to maintain the JSON format's integrity while ensuring that your response is an accurate reflection of the API's intended output within the tool. Please note that your answer should not contain anything other than a json format object, which should be parsable directly to json.
Note that:
- your response should contain rich information given the api input parameters.
- your response must be effective and have practical content.

API calls may fail for various reasons, such as invalid input parameters, authentication issues, or server errors. Your goal is to generate a response that accurately reflects the API's intended functionality, even if the input parameters are incorrect. Your response should be informative and relevant to the API's purpose, providing a clear and concise explanation of the expected output based on the input provided.
Here is an example:
API doc:
{
"api_name": "List Languages",
"api_description": "Get a list of currently supported languages. We are constantly adding more every few weeks.",
"required_parameters": [],
"optional_parameters": [],
"tool_description": "Introducing our cutting-edge text to speech service, designed to provide you with the most realistic human-sounding voices at an affordable price. Our service is fast and reliable, delivering high-quality audio output in a matter of seconds. Additionally, we offer a wide range of languages and a variety of voice choices, so you can find the perfect fit for your project. Whether you need a voiceover for a video, an audiobook, or any other project, our text to speech service has you covered. Ex...",
"tool_name": "TTSKraken",
"tool_category": "Artificial_Intelligence_Machine_Learning"
}
Request:
data = {
"category": "Artificial_Intelligence_Machine_Learning",
"tool_name": "TTSKraken",
"api_name": "List Languages",
"tool_input": "{}",
"strip": "filter",
}
Response:
{
"error": "",
"response": "{"status":0,"msg":"Success","languages":["en","fr-fr","pt-br"]}"
}

Figure 23: Agent System Prompt for StableToolBench Trajectory Collection

Agent	OOD - Seen				OOD - Unseen			
	Real	WM	W2R	CR	Real	WM	W2R	CR
Qwen2.5-7B WorldModel								
GPT-4o-mini	6.75	7.10	7.10	1.05	4.03	4.67	5.33	1.32
GPT-4o	52.10	43.79	45.56	0.87	52.00	44.67	44.67	0.86
GPT-4-turbo	65.00	51.23	52.47	0.81	74.50	62.42	62.42	0.84
GPT-4.1	53.37	56.80	57.40	1.08	64.19	65.33	64.67	1.01
GPT-5	71.60	69.23	71.01	0.99	74.00	76.00	76.67	1.04
Gemini-2.5-flash	39.05	40.83	41.42	1.06	51.35	48.67	49.33	0.96
Claude-sonnet-4.5	87.00	72.00	79.00	0.91	76.04	76.00	79.00	1.04
Average	53.55	48.71	50.57	0.94	56.59	53.97	54.58	0.96
Llama3.1-8B WorldModel								
GPT-4o-mini	6.75	8.88	8.88	1.32	4.03	2.67	2.67	0.66
GPT-4o	52.10	48.52	47.93	0.92	52.00	49.33	49.33	0.95
GPT-4-turbo	65.00	56.52	55.90	0.86	74.50	62.16	62.16	0.83
GPT-4.1	53.37	56.21	55.62	1.04	64.19	60.67	60.00	0.93
GPT-5	71.60	69.82	69.23	0.97	74.00	74.00	73.33	0.99
Gemini-2.5-flash	39.05	42.60	42.60	1.09	51.35	46.00	45.33	0.88
Claude-sonnet-4.5	87.00	78.00	84.00	0.97	76.04	81.00	78.00	1.03
Average	53.55	51.51	52.02	0.97	56.59	53.69	52.97	0.94

Table 10: Task success rate (%) in ALFWorld under different OOD settings. "OOD-Seen" indicates the same room with different layout as training. "OOD-Unseen" indicates tasks containing room types or environment layouts never seen during training.