Decoupled Planning and Execution with LLM-Driven World Model for Efficient Reinforcement learning

Guoqing Ma [1,2]
[1] Institute of Automation, Chinese Academy of Sciences, Beijing, China 100190
[2] School of Future Technology, University of Chinese Academy of Sciences, Beijing, China 100049
maguoqing22@mails.ucas.ac.cn

Abstract

Reinforcement learning (RL) agent relies on experiential data obtained through environmental interactions to achieve task objectives, but the efficacy of such learning paradigms remains fundamentally bounded by the high cost of the interactions. Current researches advocate for the integration of large language models (LLMs) to enhance decision making. However, previous methods predominantly require domain-specific fine-tuning of foundation models, incurring extra computational costs with limited generalization capabilities. Some others rely on control primitives and often produce rigid plans that fail to adapt to environmental dynamics. To this end, we introduce LWM-DPO: **L**LM-Based **W**orld **M**odel with **D**istilled **P**olicy **O**ptimization in Task Planning, a few-shot framework that uses LLMs' autoregressive reasoning with lightweight policy distillation. Our method decouples state and action representations through the separation of planning and execution. LLM is only used to plan the trajectory of the state, enabling LLMs to generate consistent trajectories via programmatic feedback. Experiments conducted with challenging robotics tasks demonstrate superior sample efficiency (11.3× improvement over baseline RL after 10k steps) and task success rates (91.2% vs. 8.1% in dynamic environments).

1 Introduction

Deep Reinforcement Learning (DRL) has demonstrated exceptional performance in fields ranging from robotic manipulation to complex game environments [1, 2]. Apart from these achievements, this paradigm is well-known to be sample inefficient, with limited real-world applications [3–5]. Recent advances in world models establish a promising framework for addressing this challenge [6–8], achieving disentanglement of policy learning from environment interactions through internal simulator construction.

Meanwhile, the integration of large language models (LLMs) into RL algorithms has shown remarkable potential [9, 10] by leveraging their linguistic priors and semantic reasoning capabilities [11]. To better increase sampling efficiency, there come works focusing on using LLMs as learned environment dynamics [12–14]. However, domain-specific finetuning on foundation models (FMs) is necessitated in most cases [15–17], incurring substantial computational overhead while constraining generalizability.

Although zero-shot and few-shot algorithms have been explored as alternatives [18–20], predefined action primitives required by previous approaches severely constrain adaptability to dynamic environments. This limitation stems primarily from the underdeveloped physics and geometric reasoning abilities in previous-generation LLMs [13, 21]. As shown in Fig. 1(a), experiments are made in a maze environment with obstacles positioned at coordinates (1,2), (2,2), etc. An agent initially located at the starting position (1,1) is required to navigate to the target coordinate (1,8) through single-cell orthogonal movements per time step. Results show that modern LLMs with enhanced

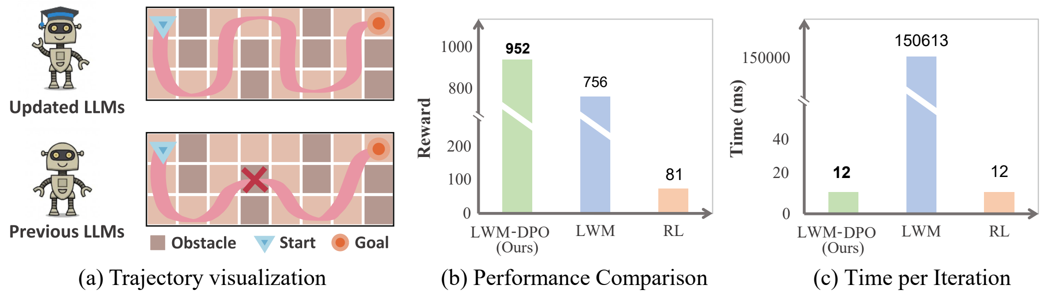


Figure 1: Comparative analysis of navigation planning and algorithm performance: (a) Trajectory visualization generated by different LLMs in maze environments, demonstrating policy evolution. (b) Success rate comparison on mobile manipulation tasks between LWM-DPO (ours), LLM-based agent without distilled policy optimization (LWM), and baseline RL approach. (c) Computational efficiency metrics showing average iteration runtime for the three algorithms in (b).

world knowledge manage to complete the task through optimized prompting strategies, indicating that they already contain information necessary to accomplish goals without any additional training. This emergent capability reveals the growing capacity of updated LLMs to generate trajectories that satisfy both task requirements and physical constraints [22, 23].

Another critical limitation in LLM-based agent systems lies in the inherent rigidity of generating single static plan from linguistic instructions, which often fails to accommodate diverse environments [24–26]. While dynamic adjustment mechanisms have been proposed to enhance grounding [27, 28], this approach remains vulnerable to output inconsistency in multi-turn interactions, significantly compromising policy reliability [29, 30]. This challenge presents an opportunity to exploit LLMs' code-writing capabilities for creating stable outputs through executable computational structures and interactions [31–33].

Moreover, as illustrated in Fig. 1(b), LLMs help to highlight meaningful regions in the state space for exploration, thereby alleviating the slow convergence issue inherent to RL with large observation and action spaces. This mechanism enables significant advancement over baseline RL paradigms. However, policies exclusively derived from LLMs may yield suboptimal solutions [34], as evidenced by comparison in Fig. 1(b)-(c), necessitating subsequent optimization stages.

Given this, here we propose LWM-DPO: **D**istilled **P**olicy **O**ptimization for **L**LM-Based **W**orld **M**odel in Task Planning. This novel framework decouples state and action representations through code-space encoding and action planning, leveraging LLMs' inherent code-generation strengths to maintain consistent environmental modeling while enabling adaptive planning. Besides, our distilled policy optimization (DPO) introduces additional refinement on LLM-guided exploration. We evaluate our method on various challenging visual RL domains, including Kitchen [35], Meta-World [36] and Robosuite [37]. Experimental results demonstrate LWM-DPO's superiority in sample efficiency and task success rates compared to existing methods. The main contribution of this paper can be summarized as follows:

- Our approach leverages LLM as predictive world model, generating physics-compliant trajectories through autoregressive reasoning. LWM-DPO bypasses the high computational costs of direct fine-tuning on FMs via lightweight policy distillation.
- By modeling the environmental state independently from action generation, our decoupled planner avoids the tight coupling seen in end-to-end approaches with task-specific action outputs. Instead, our design effectively handles variations in action dimensions across different tasks and promotes robustness in dynamic environments.
- By compiling LLM outputs into low-level code space, we construct consistent trajectory representations through programmatic feedback. This leverages LLMs' native proficiency in textual processing while compensating for their inability to directly generate executable control strategies, addressing the instability in human-LLM interaction loops.

2 Related work

Language Model as World Model: The integration of LLMs into RL frameworks has achieved significant advances in direct policy prediction [9, 10, 15]. KALM introduces a novel approach via environment-grounded imaginary rollouts, allowing agents to efficiently acquire new skills through offline learning [17]. By integrating a pretrained multimodal foundation model with an action tokenizer, GEA achieves cross-domain generalization and enables unified agent operation across various task environments [14]. However, most methods require finetuning on FMs, which imposes extra computational costs. LWM-DPO, on the other hand, uses few-shot LLMs as a world model to generate state trajectories, which are subsequently distilled into lightweight policies through SAC-based optimization. This dual-stage framework bypasses direct foundation model fine-tuning while maintaining generalization ability across diverse domains. Besides, there exist works using the zero/few-shot capabilities of LLMs for policy prediction [18, 38, 34]. While CaP [31] also benefits from code-writing LLMs, our framework incorporates policy distillation by generating trajectories instead of policies, thus ensuring enhanced stability without prompt oversaturation. ExploRLLM enhances FM capabilities through off-policy RL compensation while accelerating training via observation space reduction [34], whereas LWM-DPO implements online policy optimization through real-time trajectory generation and distillation.

Another paradigm that has been particularly effective is leveraging FMs as a skill planner [24, 39, 25]. Song et al. propose a framework for few-shot embodied instruction following, enabling dynamic adaptation to visual environments through physically-grounded plan generation and iterative refinement [27]. PCBC introduces Language-World, a natural language extension of the Meta-World robotic benchmark [36] enabling LLM-driven task planning and achieves few-shot task generalization by fine-tuning high-level plans with demonstration data [40]. PSL bridges abstract language planning in LLMs with RL to solve long-horizon robotic tasks from scratch, bypassing pre-defined skills by integrating motion planning and low-level control adaptation [12]. Yet previous methods require RL training from scratch for sub-goal decomposition, resulting in inefficiency. Instead, our method directly converts LLM-generated task semantics into executable code space with fine-grained trajectories and upsampled sub-states. This approach effectively bridges the planning-execution gap [13] while enhancing operational performance.

Table 1: Comparison between LWM-DPO and recent approaches.

Method	Free of LLM-FT	Policy-Indep.	Iterative	Optimal	Few/Zero-Shot	Online-RL	LLM Feedback
GEA [14]	✗	✗	✓	✓	✗	✓	✗
KALM [17]	✗	✗	✓	✓	✗	✗	✓
RL-SaLLM-F [41]	✓	✗	✓	✓	✗	✓	✓
PCBC [40]	✓	✓	✗	✗	✓	✓	✗
Cap [31]	✓	✗	✗	✗	✓	✗	✗
PSL [12]	✓	✓	✓	✓	✗	✓	✗
LWM-DPO (Ours)	✓	✓	✓	✓	✓	✓	✓

LLM-Guided Reward Design: Recent researches have also explored using large pretrained models instead of human supervision for reward design [**?** 42, 43]. RL-SaLLM-F replaces privileged "scripted teachers" in online preference-based RL by leveraging LLMs to generate trajectory preferences and refine feedback through ambiguity-aware discrimination and double-check mechanisms [41]. In order to improve sampling efficiency, Q-shaping introduces a RL framework that replaces reward shaping with direct LLM-guided Q-value initialization and adjustment, ensuring theoretical optimality while accelerating training across tasks [44]. EUREKA translates LLMs' high-level reasoning into robotic control via evolutionary reward code optimization, achieving human-exceeding performance across diverse embodiments [45]. However, existing reward design requires completing entire training episodes to assess reward efficacy. By contrast, LWM-DPO enables dynamic adjustment during policy rollouts, achieving higher training efficiency over prior approaches.

We highlight the distinctions between LWM-DPO and recent approaches in the way of leveraging LLMs in Table 1:

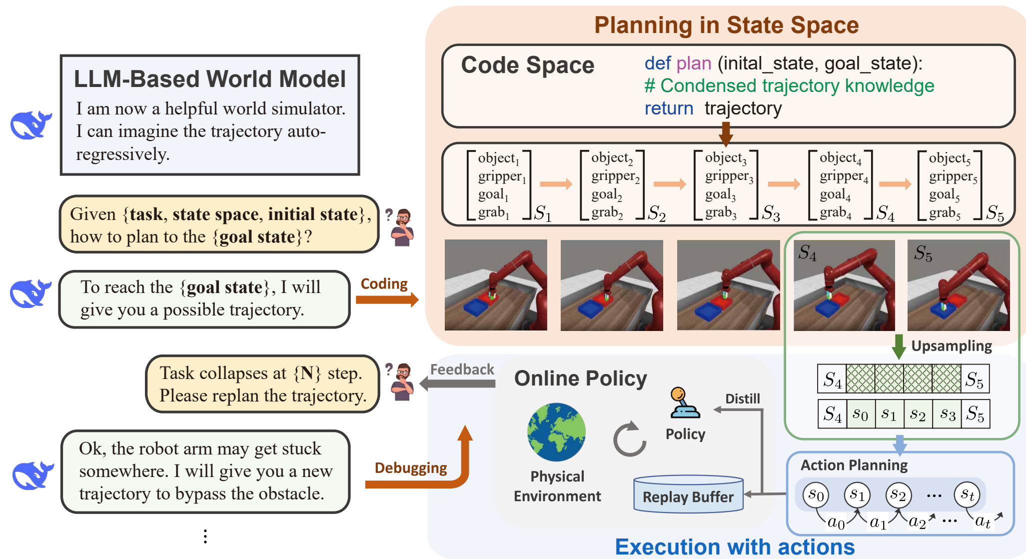


Figure 2: LWM-DPO overview. Starting with the inputs {`task, state space, initial state`}, the LLM-based world model generates candidate state trajectories toward the `goal state` in code space. These trajectories are refined into fine-grained action sequences through upsampling, action planning and further optimized to enhance physical feasibility. Upon task collapse at step **N**, the system dynamically incorporates feedback, triggers replanning, and generates alternative trajectories to bypass the obstacle.

3 Method

3.1 Framework

In this section, we present LWM-DPO for robotic task solving (outlined in Fig. 2 and Alg. 1). The method operates through two sequential phases: 1) Language-guided trajectory generation where an LLM translates textual task descriptions into state-space trajectories, followed by 2) Policy distillation that transforms these trajectories into executable control policies through reinforcement learning.

The core challenge lies in developing an effective interface between high-level language-based planning and low-level continuous control. Building upon the initial trajectory generated by the LLM-guided world model, we implement online policy optimization using a reward and termination model. This configuration enables TD (temporal difference) learning for training an actor-critic RL agent.

Algorithm 1 LWM-DPO

Require: LLM, task description g_l, state space description $space$, action planner AP, replay buffer D, planned trajectory buffer D_p

Initialize actor π and expert actor π^E

`// LWM construction`

1: LWM = LLM(g_l,$space$)

`// Generate condensed trajectory`

2: CTK = LWM.code

3: **for all** $e = 1, \cdots, E$ **do**

4: get initial state s_{init} and final state s_{final}

`// state trajectory`

5: S = CTK (s_{init},s_{final})

`// Upsample the state trajectory`

6: S_{up} = upsample(S)

`// Action planning`

7: traj = ActionPlan(AP, S_{up})

`// Exception handling`

8: **if** s_t^p fails to achive s_i after n steps **then**

9: CTK = RePlan(s_i,s_t^p)

10: go back to step 3

11: **end if**

12: save trajectory (s_t^p, a_t^p) in D_p

13: update π, π^E

14: **end for**

3.2 LLM-Guided World Model

Given task, state space, initial state and the LLM-based world model is prompted to initiate trajectory prediction (e.g. [[[x1, y1, z1],[xo1, yo1, zo1],[xg1, yg1, zg1], g1], [[x2, y2, z2], ...]) to reach the goal state. The proposed trajectory is then encoded in the code space to establish consistent state representations $\{S_i\}$, which are subsequently upsampled to get sub-states $\{s_i\}$ with linear interpolation. This hierarchical state representation enables multi-granularity reasoning while preserving geometric consistency. When policy execution fails at N step, an automated recovery protocol is triggered sending feedback to the world model after the implementation of optimized policy and require the foundation model to regenerate trajectories through replanning procedures. The world model will debug in the code space based on the feedback, iteratively refining trajectory until the agent form a collision-free path that connects a given start and goal configuration. This closed-loop refinement process demonstrates superior sample efficiency compared to traditional RL methods.

> **Prompt:** Generate trajectory function for {task_description} with state space: positions of {obj1_name} (initial: {obj1_position}), {goal_name} (target: {goal_position}), end-effector (current: {endeff_position}, range: {position_range}), and gripper state (current: {gripper_state}, range: (0,1]).
>
> **Output format:** A list where each element is of the form [[<end-effector position>], [<obj1 position>], [<goal position>], <gripper opening>].
>
> **Implement as:** def generate_trajectory(init_first_obj: list, goal_position: list, init_endeff: list, init_gripper: float) -> list

Action Planning. Building upon the dense sub-state representations $\{s_i\}$ generated by the upsampling module, the action planner formulates precise control sequences $\{a_i\}$ through inverse kinematics. For each consecutive state pair (s_i, s_{i+1}), we solve the optimal action a_i by computing the kinematic residual. We use Newton-Raphson method [46, 47] to calculate the action:

$$a_i = -J^{-1} \cdot \Delta T, \tag{1}$$

where $\Delta T = T_d - T$ represents the difference between the current pose of the end-effector and the desired target pose. The current pose of the end-effector is T and the desired target pose is T_d , the action variables are updated by computing the error of the pose, where T and T_d are components of s_i and s_{i+1}. J is the Jacobian matrix of the end-effector with respect to the controller variables (i.e., joint angles or end-effector velocities).

Overall, our framework employs a hierarchical refinement process for LLM-generated state trajectories. It begins with semantic-aware trajectory synthesis, followed by upsampling and kinematic constraint resolution. This layered architecture inherently unifies high-level task intentions with dynamic feasibility, thereby bypassing the dimensional complexity challenges of conventional motion planning methods.

3.3 Translate trajectory to policy

In the second stage, the agent interacts with the real environment to learn to refine the control of the robotic arm, performing tasks such as object transportation and manipulation. We propose DPO, which distills trajectory knowledge into a smaller network, allowing the neural network to be efficiently fine-tuned, achieving higher success rates and more optimal trajectories.

Policy optimization. We modeled the environment as a Markov decision process defined by $(\mathcal{SA}, \mathcal{P}, r, \gamma)$, with states $s \in \mathcal{S}$, actions $a \in \mathcal{A}$, transition probabilities $\mathcal{P}(s'|s, a)$, reward function $r(s)$, distribution over initial states μ_0, and discount factor $\gamma \in [0, 1)$. At each time step t, the agent observes the state $s_t \in S$ as described in the "Environment" section. Actions are continuous and applied to the robot arm controller. The reward varies depending on the test environment used. A trajectory is defined as $\xi = \{(s_t, a_t, r_{t+1})\}_{t=0}^{\infty}$. A policy $\pi(a|s)$, along with the system dynamics $\mathcal{P}$ and initial state distribution μ_0, gives rise to a distribution over trajectories: $\mu_\pi(\xi) = \mu_0(s_0) \prod_{t=0}^{\infty} \pi(a_t|s_t)\mathcal{P}(s_{t+1}|s_t, a_t)$. The aim is to obtain a policy π that maximizes the expected discounted cumulative reward or return

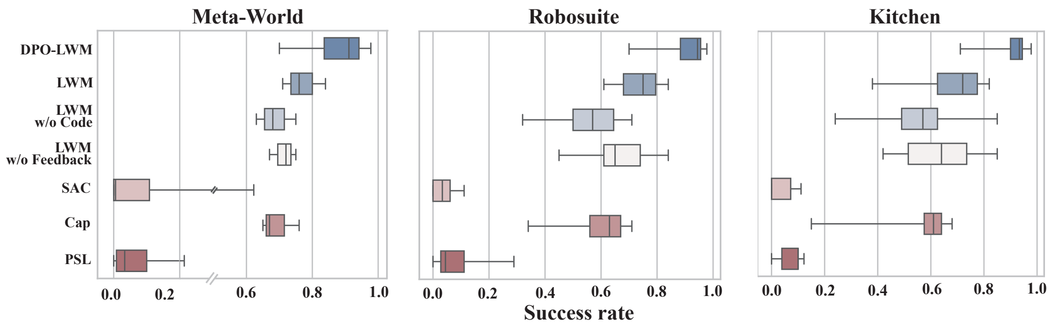


Figure 3: Comparative success rates of LWM-DPO and baseline methods after 10k steps across three robotic manipulation environments: Meta-World, Robosuite, and Kitchen.

$$J(\pi)\mathbb{E}_{\xi\sim\mu_\pi}[\sum_{t=0}^{T}\gamma^t r(s_t)]. \tag{2}$$

Behavioral Cloning. An suggested dataset $\mathcal{D} := \{(\mathbf{s}, \mathbf{a})\}$ is constructed through LLM-based trajectories by planing and translation. Afterward, a imitation learning (IL) RL policy π^E is constructed as the expert policy, which is trained to mimic the behavior of the LLM planner. In this work, the π^E is trained by minimizing the following loss function on the expert dataset $\mathcal{D}$ via behavioral cloning. The objective is to maximize the likelihood of the demonstrations

$$\underset{\pi^E}{max}\mathbb{E}_{\tau\sim\mathcal{D}}\left[\sum_{(s,a)\in\tau} ln(\pi^E(a|s))\right]. \tag{3}$$

Although BC has a strong theoretical foundation, it ignores environmental dynamics and its confinement to offline learning can lead to unstable performance [48].

Policy Distillation To better integrate the two learning processes, behavior cloning and reinforcement learning, we propose policy distillation. Specifically, let $\rho^\pi(a, s)$ be the occupancy measure of visiting state s and taking action a, under policy π. Let ρ^{π^E} the state action distribution given by expert policy π^E. The distribution matching approach proposes to learn π to minimize the discrepancy between ρ^π and ρ^{π^E} with KL-divergence:

$$\mathcal{L}_{DP} = KL(\rho^\pi||\rho^{\pi^E}) = \underset{(\mathbf{s},\mathbf{a})\sim\rho^\pi}{\mathbb{E}}\left[ln\frac{\rho^{\pi^E}(s,a)}{\rho^\pi(s,a)}\right]. \tag{4}$$

Policy objective. The policy prior π is a stochastic maximum entropy [49, 50] policy that learns to maximize the objective

$$\mathcal{J}_\pi(\theta) \doteq \underset{(\mathbf{s},\mathbf{a})_{0:H}\sim\mathcal{B}}{\mathbb{E}}\left[\sum_{t=0}^{H}\lambda^t\left[\alpha Q(\mathbf{s_t}, \pi(\mathbf{s_t})) + \beta\mathcal{H}(\pi(\cdot|s_t))\right]\right] - \gamma\mathcal{L}_{DP}, s_{t+1} = \mathcal{P}(s_t, a_t), \tag{5}$$

where $\mathcal{H}$ is the entropy of π, which can be computed in closed form. Gradients of $\mathcal{J}_\pi(\theta)$ are taken wrt. π only. As magnitude of the value estimate $Q(\mathbf{s_t}, \pi(\mathbf{s_t}))$ and entropy $\mathcal{H}$ can vary greatly between datasets and different stages of training, it is necessary to balance the two losses to prevent premature entropy collapse [51]. A common choice for tuning α, β, γ is to keep one of them constant, and adjusting the other based on an entropy target [50] or moving statistics [52].

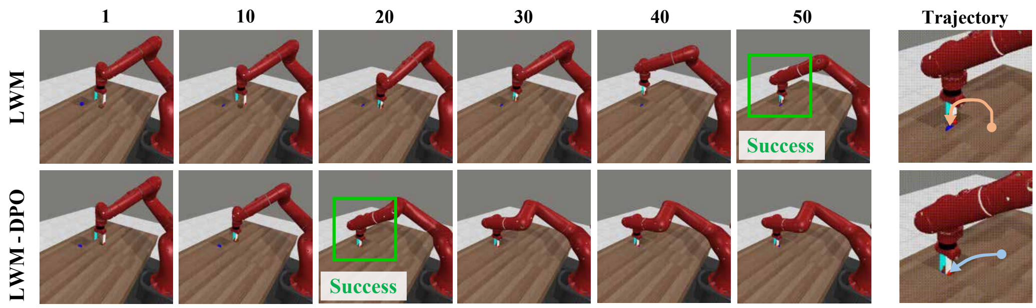


Figure 5: LWM achieves high success rate by state planning and execution in a zero-shot setting. DPO further optimizes trajectories, improving execution speed and smoothness, and achieving higher rewards.

4 Experiments

4.1 Environments

We evaluate our method on three robotic simulation frameworks: Meta-World, Robosuite and Kitchen. We perform three repeated experiments using different random seeds.

Meta-World [36]: Being a widely used RL benchmark, this platform provides procedurally-generated tasks with varying dynamics and reward structures. We select several tasks for evaluation: `MW-Assembly` (attaching nut to peg), `MW-Basketball` (shooting ball into hoop), `MW-Bin-Picking` (picking and placing a cube), `MW-Box-Close` (closing container lid), `MW-Hammer` (hammering a nail), `MW-Button-Press` (pressing vertical switches), `MW-Button-Down` (pressing horizontal switches), `MW-Down-Wall` (navigating under barrier), `MW-Hand-Insert` (inserting object into slot), `MW-Push` (pushing object to target).

Robosuite [37]: This physics-accurate simulator enables rigorous testing of robotic manipulators in instrumented environments. Our selection covers: `RS-Lift`: cube lifting, `RS-Door`: door opening and `RS-Bread/Cereal/Can`: pick-place object(s) into appropriate bin(s).

Kitchen [35]: Designed for household automation research, this environment tests sequential task execution in connected appliance systems. The tasks used in our work include `K-Slide` (opening slide cabinet), `K-Kettle` (pushing kettle forward), `K-Burner` (turning burner), `K-Light` (flicking light switch), and `K-Microwave` (opening microwave).

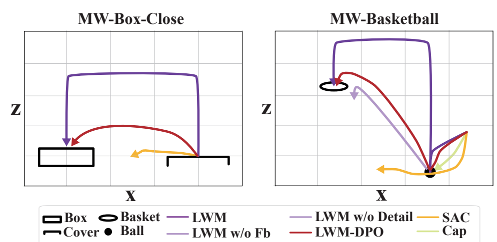
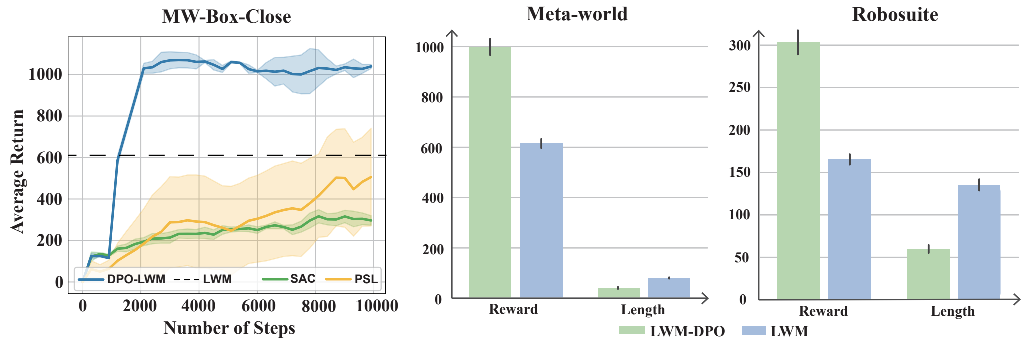


Figure 4: Two examples of 2D projections (onto the x-z plane) of trajectories generated by the different methods. Fb represents feedback. LWM complete this task successfully. RL-finetuned trajectories achieve the task goals more quickly and smoothly.

4.2 Results and Analyses

Fig. 3 shows LWM-DPO's success rate superiority over baselines across 20 Robosuite/Kitchen/Meta-World tasks. Fig. 4 comparatively demonstrates its trajectory advantages: our method achieves goal-directed navigation without deadlock, whereas baseline approaches exhibit failure modes including premature termination. The observed trajectories further validate its geometric efficiency in path planning.

Fig. 5 demonstrates the efficiency gains from DPO in `MW-Hand-Insert`. The DPO-enhanced planner (blue trajectory) achieves successful insertion in 20 frames through a direct, near-minimal path, while the baseline without DPO (orange trajectory) requires 50 frames with visible detours. The quantitative benefits of DPO are further validated in Fig. 6. Across both Meta-World and Robosuite



(a) Sample efficiency (b) Reward lifting

Figure 6: Performance advantages of DPO. **(a)** Dual metric comparison (reward ↑/length ↓) between LWM-DPO and traditional LWM method without DPO in Meta-World and Robosuite. **(b)** Training efficiency on `MW-Box-Close`. Shaded regions denote performance variance across seeds.

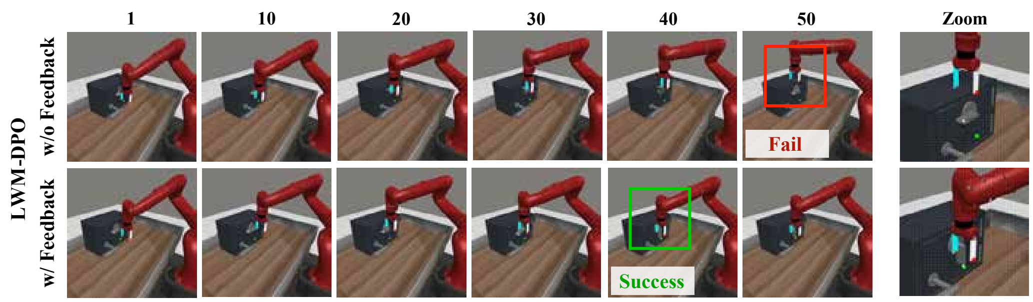


Figure 7: In cases where the planned trajectory encounters obstacles, the world model incorporating feedback mechanisms performs real-time adjustments and replanning to maintain optimal navigation.

benchmarks, LWM-DPO shows significant performance improvements over the non-DPO baseline, achieving higher task rewards while executing more efficient trajectories (Fig. 6(a)). Moreover, as depicted in Figure 6(b), DPO notably accelerates policy convergence, demonstrating significantly enhanced sample efficiency.

Ablation study on RL method. To evaluate the effectiveness of DPO, we experimented with directly combining an expert policy with SAC. The results indicate that while the DPO does not play a decisive role in performance improvement, it still outperforms pure BC. This may be attributed to distillation better transferring expert knowledge when optimizing multiple objectives (RL and BC) [53].

Effectiveness of feedback loop. The critical role of feedback loop in dynamic manipulation tasks is demonstrated through `MW-Door-Lock` (shown in Fig. 7). The feedback-enabled LWM-DPO detects handle resistance through joint torque signatures and adjusts the gripper's trajectory to prevent jamming, enabling successful door unlocking. In contrast, methods without feedback rigidly follow the pre-planned trajectory, forcing the robotic arm into an inescapable mechanical lock with the handle.

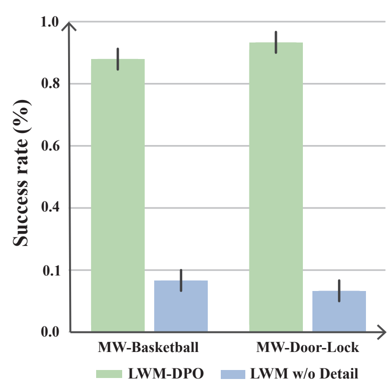


Figure 8: Success rates on both `MW-Basketball` and `MW-Door-Lock` demonstrate the significance of detailed prompts.

Effectiveness of detailed information. In addition, as shown in Fig. 9, the performance of LLM-guided world model can be improved when provided with detailed environmental prompts. For instance, in `MW-Basketball`, incorporating precise state information such as the hoop diameter (approximately 0.1) enables the robotic arm to avoid potential jamming during execution, thereby enhancing operational reliability. A quantitative comparison presented in Fig. 8 further demonstrates this advantage,

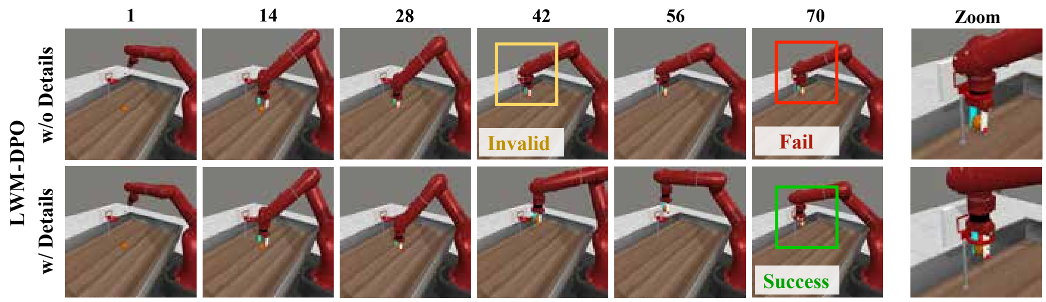


Figure 9: With additional prompts, the LLM-guided world model acquires richer state information, enabling it to generate more rational and optimized trajectories.

where the detailed state description prompt achieves significantly higher success rates than its non-detailed counterpart across both `MW-Door-Lock` and `MW-Basketball`.

Effectiveness of code space integration. The empirical advantages of code space integration are quantitatively validated in Table 2. When equipped with code space knowledge, all tested LLMs demonstrate improvement in task success rates while reducing planning trajectory length (measured in steps). These results confirm that code space serves as an effective abstraction layer for compressing environmental state trajectory representations while preserving critical task semantics.

Table 2: Impact of Code Space Knowledge on Planning Performance. Gray-shaded rows highlight configurations utilizing code space.

LLM	Task	Code Space	Length ↓	Success rate ↑
Deepseek-R1 [54]	Reach	w/o	52	20/20
		w/	42	20/20
	Hand-Insert	w/o	78	12/20
		w/	76	15/20
GPT-4o [55]	Reach	w/o	47	19/20
		w/	45	20/20
	Hand-Insert	w/o	87	12/20
		w/	85	16/20
Deepseek-V3 [56]	Reach	w/o	64	20/20
		w/	54	20/20
	Hand-Insert	w/o	92	11/20
		w/	87	13/20
GPT-4o-mini [57]	Reach	w/o	51	18/20
		w/	43	20/20
	Hand-Insert	w/o	89	10/20
		w/	72	11/20

5 Conclusions and limitations

We proposed LWM-DPO, a framework that integrates LLMs with distilled policy optimization to enhance decision-making efficiency in reinforcement learning. By decoupling state and action representations through code-space encoding, our method enables LLMs to generate adaptive trajectories while avoiding costly fine-tuning. Experiments on robotics tasks demonstrate improved sample efficiency and policy reliability compared to conventional RL and zero-shot LLM-based approaches. The limitation lies in restricting inputs to structured state representations, which precludes direct application to visual RL scenarios requiring raw pixel processing. Future work will extend the framework by incorporating visual encoders to bridge textual reasoning with pixel-based observations.

References

[1] N. Brown, A. Bakhtin, A. Lerer, and Q. Gong, "Combining deep reinforcement learning and search for imperfect-information games," *Annual Conference on Neural Information Processing Systems*, 2020.

[2] J. Schrittwieser, I. Antonoglou, T. Hubert, K. Simonyan, L. Sifre, S. Schmitt, A. Guez, E. Lockhart, D. Hassabis, T. Graepel, T. P. Lillicrap, and D. Silver, "Mastering atari, go, chess and shogi by planning with a learned model," *Nature*, vol. 588, pp. 604 – 609, 2019.

[3] M. Hessel, J. Modayil, H. V. Hasselt, T. Schaul, G. Ostrovski, W. Dabney, D. Horgan, B. Piot, M. G. Azar, and D. Silver, "Rainbow: Combining improvements in deep reinforcement learning," in *Association for the Advancement of Artificial Intelligence*, 2017.

[4] W. Zhang, G. Wang, J. Sun, Y. Yuan, and G. Huang, "Storm: Efficient stochastic transformer based world models for reinforcement learning," *Annual Conference on Neural Information Processing Systems*, 2023.

[5] Y. Zhang, G. Ma, G. Hao, L. Guo, Y. Chen, and S. Yu, "Efficient reinforcement learning through adaptively pretrained visual encoder," in *Association for the Advancement of Artificial Intelligence*, 2025, pp. 22 668–22 676.

[6] D. Hafner, J. Pasukonis, J. Ba, and T. Lillicrap, "Mastering diverse domains through world models," *arXiv preprint arXiv:2301.04104*, 2023.

[7] E. Alonso, A. Jelley, V. Micheli, A. Kanervisto, A. J. Storkey, T. Pearce, and F. Fleuret, "Diffusion for world modeling: Visual details matter in atari," *Annual Conference on Neural Information Processing Systems*, 2024.

[8] G. Ma, Z. Wang, X. Yuan, and F. Zhou, "Improving model-based deep reinforcement learning with learning degree networks and its application in robot control," *J. Robotics*, pp. 7 169 594:1–7 169 594:14, 2022.

[9] J.-C. Pang, X. Yang, S.-H. Yang, X.-H. Chen, and Y. Yu, "Natural language instruction-following with task-related language development and translation," in *Neural Information Processing Systems*, 2023.

[10] A. Szot, M. Schwarzer, H. Agrawal, B. Mazoure, W. Talbott, K. Metcalf, N. Mackraz, D. Hjelm, and A. Toshev, "Large language models as generalizable policies for embodied tasks," *International Conference on Learning Representations*, 2023.

[11] K. Cobbe, V. Kosaraju, M. Bavarian, M. Chen, H. Jun, L. Kaiser, M. Plappert, J. Tworek, J. Hilton, R. Nakano, C. Hesse, and J. Schulman, "Training verifiers to solve math word problems," *arXiv preprint arXiv:2110.14168*, 2021.

[12] M. Dalal, T. Chiruvolu, D. S. Chaplot, and R. Salakhutdinov, "Plan-seq-learn: Language model guided rl for solving long horizon robotics tasks," *International Conference on Learning Representations*, 2024.

[13] W. Huang, P. Abbeel, D. Pathak, and I. Mordatch, "Language models as zero-shot planners: Extracting actionable knowledge for embodied agents," *International Conference on Machine Learning*, vol. 162, pp. 9118–9147, 2022.

[14] A. Szot, B. Mazoure, O. Attia, A. Timofeev, H. Agrawal, D. Hjelm, Z. Gan, Z. Kira, and A. Toshev, "From multimodal llms to generalist embodied agents: Methods and lessons," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 10 644–10 655, 2025.

[15] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, K. Choromanski, T. Ding, D. Driess, K. A. Dubey, C. Finn, P. R. Florence, C. Fu, M. G. Arenas, K. Gopalakrishnan, K. Han, K. Hausman, A. Herzog, J. Hsu, B. Ichter, A. Irpan, N. J. Joshi, R. C. Julian, D. Kalashnikov, Y. Kuang, I. Leal, S. Levine, H. Michalewski, I. Mordatch, K. Pertsch, K. Rao, K. Reymann, M. S. Ryoo, G. Salazar, P. R. Sanketi, P. Sermanet, J. Singh, A. Singh, R. Soricut, H. Tran, V. Vanhoucke, Q. H. Vuong, A. Wahid, S. Welker, P. Wohlhart, T. Xiao, T. Yu, and B. Zitkovich, "Rt-2: Vision-language-action models transfer web knowledge to robotic control," *Conference on Robot Learning*, vol. 229, pp. 2165–2183, 2023.

[16] D. Driess, F. Xia, M. S. M. Sajjadi, C. Lynch, A. Chowdhery, B. Ichter, A. Wahid, J. Tompson, Q. H. Vuong, T. Yu, W. Huang, Y. Chebotar, P. Sermanet, D. Duckworth, S. Levine, V. Vanhoucke, K. Hausman, M. Toussaint, K. Greff, A. Zeng, I. Mordatch, and P. R. Florence, "Palm-e: An embodied multimodal language model," in *International Conference on Machine Learning*, 2023.

[17] J.-C. Pang, S.-H. Yang, K. Li, J. Zhang, X.-H. Chen, N. Tang, and Y. Yu, "Knowledgeable agents by offline reinforcement learning from large language model rollouts," *Conference on Neural Information Processing Systems*, 2024.

[18] Y. Jiang, A. Gupta, Z. Zhang, G. Wang, Y. Dou, Y. Chen, L. Fei-Fei, A. Anandkumar, Y. Zhu, and L. J. Fan, “Vima: General robot manipulation with multimodal prompts,” *arXiv preprint arXiv:2210.03094*, 2022.

[19] G. Wang, Y. Xie, Y. Jiang, A. Mandlekar, C. Xiao, Y. Zhu, L. J. Fan, and A. Anandkumar, “Voyager: An open-ended embodied agent with large language models,” *Trans. Mach. Learn. Res.*, vol. 2024, 2023.

[20] R. Mon-Williams, G. Li, R. Long, W. Du, and C. Lucas, “Embodied large language models enable robots to complete complex tasks in unpredictable environments,” *Nature Machine Intelligence*, 2025.

[21] J. Duan, R. Zhang, J. Diffenderfer, B. Kailkhura, L. Sun, E. Stengel-Eskin, M. Bansal, T. Chen, and K. Xu, “Gtbench: Uncovering the strategic reasoning limitations of llms via game-theoretic evaluations,” *arXiv preprint arXiv:2402.12348*, 2024.

[22] S. Jassim, M. S. Holubar, A. Richter, C. Wolff, X. Ohmer, and E. M. B. Bruni, “Grasp: A novel benchmark for evaluating language grounding and situated physics understanding in multimodal language models,” in *International Joint Conference on Artificial Intelligence*, 2023.

[23] G. Polverini and B. Gregorcic, “How understanding large language models can inform the use of chatgpt in physics education,” *European Journal of Physics*, vol. 45, 2023.

[24] M. Ahn, A. Brohan, N. Brown, Y. Chebotar, O. Cortes, B. David, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, D. Ho, J. Hsu, J. Ibarz, B. Ichter, A. Irpan, E. Jang, R. M. J. Ruano, K. Jeffrey, S. Jesmonth, N. J. Joshi, R. C. Julian, D. Kalashnikov, Y. Kuang, K.-H. Lee, S. Levine, Y. Lu, L. Luu, C. Parada, P. Pastor, J. Quiambao, K. Rao, J. Rettinghouse, D. M. Reyes, P. Sermanet, N. Sievers, C. Tan, A. Toshev, V. Vanhoucke, F. Xia, T. Xiao, P. Xu, S. Xu, and M. Yan, “Do as i can, not as i say: Grounding language in robotic affordances,” in *Conference on Robot Learning*, 2022.

[25] I. Singh, V. Blukis, A. Mousavian, A. Goyal, D. Xu, J. Tremblay, D. Fox, J. Thomason, and A. Garg, “Progprompt: Generating situated robot task plans using large language models,” *IEEE International Conference on Robotics and Automation*, pp. 11 523–11 530, 2022.

[26] Y. Lu, W. Feng, W. Zhu, W. Xu, X. E. Wang, M. P. Eckstein, and W. Y. Wang, “Neuro-symbolic procedural planning with commonsense prompting,” in *International Conference on Learning Representations*, 2022.

[27] C. H. Song, J. Wu, C. Washington, B. M. Sadler, W.-L. Chao, and Y. Su, “Llm-planner: Few-shot grounded planning for embodied agents with large language models,” *IEEE/CVF International Conference on Computer Vision*, pp. 2986–2997, 2022.

[28] T. Carta, C. Romac, T. Wolf, S. Lamprier, O. Sigaud, and P.-Y. Oudeyer, “Grounding large language models in interactive environments with online reinforcement learning,” *International Conference on Machine Learning*, vol. 202, pp. 3676–3713, 2023.

[29] M. Eric, R. Goel, S. Paul, A. Kumar, A. Sethi, A. K. Goyal, P. Ku, S. Agarwal, S. Gao, and D. Z. Hakkani-Tür, “Multiwoz 2.1: Multi-domain dialogue state corrections and state tracking baselines,” *International Conference on Language Resources and Evaluation*, pp. 422–428, 2019.

[30] M. Rohmatillah and J.-T. Chien, “Robust multi-domain multi-turn dialogue policy via student-teacher offline reinforcement learning,” *APSIPA Transactions on Signal and Information Processing*, 2024.

[31] J. Liang, W. Huang, F. Xia, P. Xu, K. Hausman, B. Ichter, P. R. Florence, and A. Zeng, “Code as policies: Language model programs for embodied control,” *IEEE International Conference on Robotics and Automation*, pp. 9493–9500, 2022.

[32] A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H. W. Chung, C. Sutton, S. Gehrmann, P. Schuh, K. Shi, S. Tsvyashchenko, J. Maynez, A. Rao, P. Barnes, Y. Tay, N. M. Shazeer, V. Prabhakaran, E. Reif, N. Du, B. Hutchinson, R. Pope, J. Bradbury, J. Austin, M. Isard, G. Gur-Ari, P. Yin, T. Duke, A. Levskaya, S. Ghemawat, S. Dev, H. Michalewski, X. García, V. Misra, K. Robinson, L. Fedus, D. Zhou, D. Ippolito, D. Luan, H. Lim, B. Zoph, A. Spiridonov, R. Sepassi, D. Dohan, S. Agrawal, M. Omernick, A. M. Dai, T. S. Pillai, M. Pellat, A. Lewkowycz, E. Moreira, R. Child, O. Polozov, K. Lee, Z. Zhou, X. Wang, B. Saeta, M. Díaz, O. Firat, M. Catasta, J. Wei, K. S. Meier-Hellstern, D. Eck, J. Dean, S. Petrov, and N. Fiedel, “Palm: Scaling language modeling with pathways,” *J. Mach. Learn. Res.*, vol. 24, pp. 240:1–240:113, 2023.

[33] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. L. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, J. Schulman, J. Hilton, F. Kelton, L. E. Miller, M. Simens, A. Askell, P. Welinder, P. F. Christiano, J. Leike, and R. J. Lowe, “Training language models to follow instructions with human feedback,” *Conference on Neural Information Processing Systems*, 2022.

[34] R. Ma, J. Luijkx, Z. Ajanović, and J. Kober, “Explorllm: Guiding exploration in reinforcement learning with large language models,” *arXiv preprint arXiv:2403.09583*, 2024.

[35] A. Gupta, V. Kumar, C. Lynch, S. Levine, and K. Hausman, “Relay policy learning: Solving long-horizon tasks via imitation and reinforcement learning,” *ArXiv*, vol. abs/1910.11956, 2019.

[36] T. Yu, D. Quillen, Z. He, R. C. Julian, K. Hausman, C. Finn, and S. Levine, “Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning,” *Conference on Robot Learning*, vol. 100, pp. 1094–1100, 2019.

[37] Y. Zhu, J. Wong, A. Mandlekar, and R. Mart'in-Mart'in, “robosuite: A modular simulation framework and benchmark for robot learning,” *arXiv preprint arXiv:2009.12293*, 2020.

[38] A. Zeng, M. Attarian, brian ichter, K. M. Choromanski, A. Wong, S. Welker, F. Tombari, A. Purohit, M. S. Ryoo, V. Sindhwani, J. Lee, V. Vanhoucke, and P. Florence, “Socratic models: Composing zero-shot multimodal reasoning with language,” in *The Eleventh International Conference on Learning Representations*, 2023.

[39] W. Huang, F. Xia, T. Xiao, H. Chan, J. Liang, P. Florence, A. Zeng, J. Tompson, I. Mordatch, Y. Chebotar, P. Sermanet, T. Jackson, N. Brown, L. Luu, S. Levine, K. Hausman, and b. ichter, “Inner monologue: Embodied reasoning through planning with language models,” in *Proceedings of The 6th Conference on Robot Learning*, vol. 205, 2023, pp. 1769–1782.

[40] K. R. Zentner, R. C. Julian, B. Ichter, and G. S. Sukhatme, “Conditionally combining robot skills using large language models,” *IEEE International Conference on Robotics and Automation*, pp. 14 046–14 053, 2023.

[41] S. Tu, J. Sun, Q. Zhang, X. Lan, and D. Zhao, “Online preference-based reinforcement learning with self-augmented feedback from large language model,” in *Proceedings of the 24th International Conference on Autonomous Agents and Multiagent Systems*, 2025, p. 2069–2077.

[42] Y. Wang, Z. Xian, F. Chen, T.-H. Wang, Y. Wang, Z. Erickson, D. Held, and C. Gan, “Robogen: Towards unleashing infinite data for automated robot learning via generative simulation,” *International Conference on Machine Learning*, 2024.

[43] Y. Zeng, Y. Mu, and L. Shao, “Learning reward for robot skills using large language models via self-alignment,” *International Conference on Machine Learning*, 2024.

[44] X. Wu, “From reward shaping to q-shaping: Achieving unbiased learning with llm-guided knowledge,” *arXiv preprint arXiv:2410.01458*, 2024.

[45] Y. J. Ma, W. Liang, G. Wang, D.-A. Huang, O. Bastani, D. Jayaraman, Y. Zhu, L. Fan, and A. Anandkumar, “Eureka: Human-level reward design via coding large language models,” *International Conference on Learning Representations*, 2024.

[46] S. Tsuboi, H. Kino, and K. Tahara, “End-point stiffness and joint viscosity control of musculoskeletal robotic arm using muscle redundancy,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2022, pp. 4997–5002.

[47] I. Singh, O. Lakhal, Y. Amara, V. Coelen, P. M. Pathak, and R. Merzouki, “Performances evaluation of inverse kinematic models of a compact bionic handling assistant,” in *IEEE International Conference on Robotics and Biomimetics*, 2017, pp. 264–269.

[48] S. Ross, G. Gordon, and D. Bagnell, “A reduction of imitation learning and structured prediction to no-regret online learning,” in *International Conference on Artificial Intelligence and Statistics*, 2011, pp. 627–635.

[49] B. D. Ziebart, A. L. Maas, J. A. Bagnell, A. K. Dey *et al.*, “Maximum entropy inverse reinforcement learning.” in *Association for the Advancement of Artificial Intelligence*, vol. 8, 2008, pp. 1433–1438.

[50] T. Haarnoja, A. Zhou, K. Hartikainen, G. Tucker, S. Ha, J. Tan, V. Kumar, H. Zhu, A. Gupta, P. Abbeel *et al.*, “Soft actor-critic algorithms and applications,” *arXiv preprint arXiv:1812.05905*, 2018.

[51] D. Yarats, R. Fergus, A. Lazaric, and L. Pinto, “Mastering visual continuous control: Improved data-augmented reinforcement learning,” in *International Conference on Learning Representations*, 2022.

[52] D. Hafner, J. Pasukonis, J. Ba, and T. Lillicrap, “Mastering diverse domains through world models,” *arXiv preprint arXiv:2301.04104*, 2023.

[53] H. Hoang, T. Mai, and P. Varakantham, “Imitate the good and avoid the bad: An incremental approach to safe reinforcement learning,” in *Association for the Advancement of Artificial Intelligence*, vol. 38, no. 11, 2024, pp. 12 439–12 447.

[54] DeepSeek-AI, D. Guo, D. Yang, H. Zhang, J.-M. Song, R. Zhang, R. Xu, Q. Zhu, S. Ma, P. Wang, X. Bi, X. Zhang, X. Yu, Y. Wu, Z. F. Wu, Z. Gou, Z. Shao, Z. Li, Z. Gao, A. Liu, B. Xue, B.-L. Wang, B. Wu, B. Feng, C. Lu, C. Zhao, C. Deng, C. Zhang, C. Ruan, D. Dai, D. Chen, D.-L. Ji, E. Li, F. Lin, F. Dai, F. Luo, G. Hao, G. Chen, G. Li, H. Zhang, H. Bao, H. Xu, H. Wang, H. Ding, H. Xin, H. Gao, H. Qu, H. Li, J. Guo, J. Li, J. Wang, J. Chen, J. Yuan, J. Qiu, J. Li, J. Cai, J. Ni, J. Liang, J. Chen, K. Dong, K. Hu, K. Gao, K. Guan, K. Huang, K. Yu, L. Wang, L. Zhang, L. Zhao, L. Wang, L. Zhang, L. Xu, L. Xia, M. Zhang, M. Zhang, M. Tang, M. Li, M. Wang, M. Li, N. Tian, P. Huang, P. Zhang, Q. Wang, Q. Chen, Q. Du, R. Ge, R. Zhang, R. Pan, R. Wang, R. J. Chen, R. L. Jin, R. Chen, S. Lu, S. Zhou, S. Chen, S. Ye, S. Wang, S. Yu, S. Zhou, S. Pan, S. S. Li, S. Zhou, S.-K. Wu, T. Yun, T. Pei, T. Sun, T. Wang, W. Zeng, W. Zhao, W. Liu, W. Liang, W. Gao, W.-X. Yu, W. Zhang, W. L. Xiao, W. An, X. Liu, X. Wang, X. Chen, X. Nie, X. Cheng, X. Liu, X. Xie, X. Liu, X. Yang, X. Li, X. Su, X. Lin, X. Q. Li, X. Jin, X.-C. Shen, X. Chen, X. Sun, X. Wang, X. Song, X. Zhou, X. Wang, X. Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Y. Zhang, Y. Xu, Y. Li, Y. Zhao, Y. Sun, Y. Wang, Y. Yu, Y. Zhang, Y. Shi, Y. Xiong, Y. He, Y. Piao, Y. Wang, Y. Tan, Y. Ma, Y. Liu, Y. Guo, Y. Ou, Y. Wang, Y. Gong, Y.-J. Zou, Y. He, Y. Xiong, Y.-W. Luo, Y. mei You, Y. Liu, Y. Zhou, Y. X. Zhu, Y. Huang, Y. Li, Y. Zheng, Y. Zhu, Y. Ma, Y. Tang, Y. Zha, Y. Yan, Z. Ren, Z. Ren, Z. Sha, Z. Fu, Z. Xu, Z. Xie, Z. guo Zhang, Z. Hao, Z. Ma, Z. Yan, Z. Wu, Z. Gu, Z. Zhu, Z. Liu, Z.-A. Li, Z. Xie, Z. Song, Z. Pan, Z. Huang, Z. Xu, Z. Zhang, and Z. Zhang, "Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning," *arXiv preprint arXiv:2501.12948*, 2025.

[55] OpenAI. (2024) Hello gpt-4o. [Online]. Available: https://openai.com/index/hello-gpt-4o

[56] DeepSeek-AI, A. Liu, B. Feng, B. Xue, B.-L. Wang, B. Wu, C. Lu, C. Zhao, C. Deng, C. Zhang, C. Ruan, D. Dai, D. Guo, D. Yang, D. Chen, D.-L. Ji, E. Li, F. Lin, F. Dai, F. Luo, G. Hao, G. Chen, G. Li, H. Zhang, H. Bao, H. Xu, H. Wang, H. Zhang, H. Ding, H. Xin, H. Gao, H. Li, H. Qu, J. L. Cai, J. Liang, J. Guo, J. Ni, J. Li, J. Wang, J. Chen, J. Chen, J. Yuan, J. Qiu, J. Li, J.-M. Song, K. Dong, K. Hu, K. Gao, K. Guan, K. Huang, K. Yu, L. Wang, L. Zhang, L. Xu, L. Xia, L. Zhao, L. Wang, L. Zhang, M. Li, M. Wang, M. Zhang, M. Zhang, M. Tang, M. Li, N. Tian, P. Huang, P. Wang, P. Zhang, Q. Wang, Q. Zhu, Q. Chen, Q. Du, R. J. Chen, R. L. Jin, R. Ge, R. Zhang, R. Pan, R. Wang, R. Xu, R. Zhang, R. Chen, S. S. Li, S. Lu, S. Zhou, S. Chen, S.-P. Wu, S. Ye, S. Ma, S. Wang, S. Zhou, S. Yu, S. Zhou, S. Pan, T. Wang, T. Yun, T. Pei, T. Sun, W. L. Xiao, W. Zeng, W. Zhao, W. An, W. Liu, W. Liang, W. Gao, W.-X. Yu, W. Zhang, X. Q. Li, X. Jin, X. Wang, X. Bi, X. Liu, X. Wang, X.-C. Shen, X. Chen, X. Zhang, X. Chen, X. Nie, X. Sun, X. Wang, X. Cheng, X. Liu, X. Xie, X. Liu, X. Yu, X. Song, X. Shan, X. Zhou, X. Yang, X. Li, X. Su, X. Lin, Y. K. Li, Y. Q. Wang, Y. X. Wei, Y. X. Zhu, Y. Zhang, Y. Xu, Y. Huang, Y. Li, Y. Zhao, Y. Sun, Y. Li, Y. Wang, Y. Yu, Y. Zheng, Y. Zhang, Y. Shi, Y. Xiong, Y. He, Y. Tang, Y. Piao, Y. Wang, Y. Tan, Y.-B. Ma, Y. Liu, Y. Guo, Y. Wu, Y. Ou, Y. Zhu, Y. Wang, Y. Gong, Y. Zou, Y. He, Y. Zha, Y. Xiong, Y. Ma, Y. Yan, Y.-W. Luo, Y. mei You, Y. Liu, Y. Zhou, Z. F. Wu, Z. Ren, Z. Ren, Z. Sha, Z. Fu, Z. Xu, Z. Huang, Z. Zhang, Z. Xie, Z. guo Zhang, Z. Hao, Z. Gou, Z. Ma, Z. Yan, Z. Shao, Z. Xu, Z. Wu, Z. Zhang, Z. Li, Z. Gu, Z. Zhu, Z. Liu, Z.-A. Li, Z. Xie, Z. Song, Z. Gao, and Z. Pan, "Deepseek-v3 technical report," *arXiv preprint arXiv:2412.19437*, 2024.

[57] OpenAI. (2024) Gpt-4o mini: Advancing cost-efficient intelligence. [Online]. Available: https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: We introduce the contribution in the abstract and the last paragraph of Sec. 1.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: We have discussed the limitations of this work in detail in Sec. 5.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [NA]

Justification: This paper does not include theoretical proofs.

Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: We report important details about the experiments in Sec. 3. We also provide the necessary hyper-parameters for reproducibility: the choice of the pre-trained LLM (Deepseek-R1-671B [54]).

   Guidelines:

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

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

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

Answer: [Yes]

Justification: We have released the code and data based on the submission guidelines.

Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: We selected those hyperparameters for their good performances.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: The results reported are averaged over three random runs. We use error bars or shaded area to present the statistical significance, e.g., Fig. 3.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: We illustrate the details about experiments.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: The research conforms with the NeurIPS Code of Ethics.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: We have discussed the broader impacts in conclusion.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

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

11. **Safeguards**

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

    Answer: [NA]

    Justification: This paper poses no such risks, as it does not release new pre-trained models or datasets.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: The assets/environments used in this paper are properly credited: Deepseek-R1 [54], SAC [50], Meta-World [36], Robosuite [37], Kitchen [35].

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [NA]

    Justification: This paper does not release new assets.

    Guidelines:

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

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

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

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

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

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

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

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

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

    Answer: [Yes]

    Justification: We utilize a LLM as a language-based world model for state planning. The specific methodology is detailed in the main text.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.