WORLD-VLA-LOOP: CLOSED-LOOP LEARNING OF VIDEO WORLD MODEL AND VLA POLICY

Anonymous authors
Paper under double-blind review

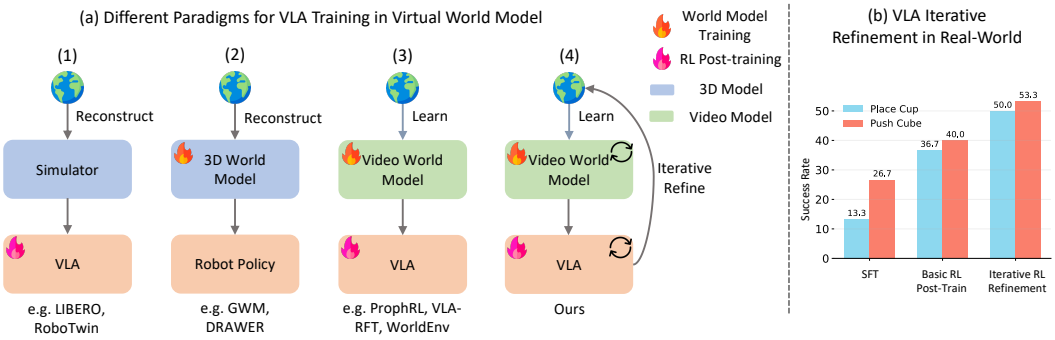


Figure 1: (a) Paradigms for world-model-based VLA reinforcement learning. Comparison of existing methodologies: current approaches typically rely on reconstructing the environment within 3D world or training video world models that simulate the environment. To address the imprecise action-following inherent in existing video-based simulators, we propose World-VLA-Loop, structured around two fundamental designs and a higher-level co-evolving paradigm. (b) We show that the real-world policy success rate is improved by 36.7% and 26.6% on two tasks after two iterations of joint optimization with VLA model and world model.

ABSTRACT

Reinforcement learning (RL) can refine Vision-Language-Action (VLA) policies beyond behavior cloning, but real-world RL remains expensive due to extensive rollouts, resets, supervision, and safety risks. Action-conditioned video world models offer an option to train in virtual environments, yet they exhibit imprecise action following, particularly on subtle near-success failures. Besides, they lack native reward signals for RL. Computing rewards based on inaccurate visual predictions remain unreliable. We introduce World-VLA-Loop, structured around two foundational designs and a higher-level co-evolving paradigm. We first curate SANS, dedicatedly mixing successful and near-success trajectories to improve action–outcome alignment. Then, we train a state-aware video world model that jointly predicts future frames and binary rewards from diffusion latents. It couples reward estimation to the generator rather than a separate module, and in turn, benefits visual prediction. Since VLA behavior shifts during RL, a fixed simulator can misalign with the updated policy, World-VLA-Loop therefore closes the loop by using the refined world model for iterative VLA post-training while feeding rollouts from each improved policy back to augment and fine-tune the world model. Across simulation and real-robot experiments, World-VLA-Loop substantially improves VLA performance while reducing reliance on costly physical interaction.

1 INTRODUCTION

Vision-Language-Action (VLA) models have emerged as a dominant paradigm for robotic manipulation by leveraging large language model priors to map natural language directly to low-level control (Black et al., 2024; Kim et al., 2024; 2025; Liu et al., 2025b). Beyond behavior cloning, which requires large-scale human demonstration collection, recent works (Lu et al., 2025a; Li et al., 2025a;

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Intelligence et al., 2025) integrate reinforcement learning (RL) to refine policies through interaction. Despite this promise, existing RL post-training remains largely confined to simulated environments (Liu et al., 2023; Tao et al., 2024), because real-world RL incurs substantial costs: extensive physical rollouts, safety risks during exploration, and operational overhead for resets and supervision.

A promising way to mitigate these challenges is to treat learned world models as virtual environments. As shown in Fig. 1, we group such models into three paradigms: (1) *Handcrafted digital twins* (Liu et al., 2023; Chen et al., 2025), which rely on manual asset creation and physics engine modeling but often lack the photorealism and physical fidelity needed for real-world adaptation; (2) *3D-based reconstruction* (Lu et al., 2025b; Xia et al., 2025), which use geometric 3D representations but struggle to generalize across diverse environments, and rarely support rich stochastic exploration; (3) *Action-conditioned video world models* (Zhang et al., 2025; Li et al., 2025b; Xiao et al., 2025), which leverage pretrained priors from large-scale video and offer high photorealism and stronger scene generalization.

Video world models are attractive, yet they face two critical limitations. First, our empirical analysis shows that they often follow actions imprecisely, so predicted trajectories diverge from actual execution outcomes. This is particularly prominent in *near-success* failure cases, where the robot fails to achieve a specific goal due to minor action errors. As shown in Fig. 2, models such as Cosmos-Predict 2 (Ali et al., 2025) frequently hallucinate successful outcomes even under erroneous actions, reflecting weak grounding in fine-grained physical dynamics. Second, they do not natively provide reward signals for RL. Although recent work (Xiao et al., 2025) constructs rewards from visual predictions, such rewards remain unreliable when the underlying video predictions are inaccurate, especially in the (near-)success cases.

We address these limitations with two complementary *foundational* designs. First, to improve action–outcome alignment, we curate the **SANS** dataset, combining **S**uccessful **A**nd **N**ear-**S**uccess trajectories to sharpen the boundary between success and subtle failure. Second, we introduce a **state-aware video world model** that jointly predicts future frames and binary rewards. This joint supervision sharpens the success–failure boundary in latent space, while the predicted rewards directly provide RL signals at inference. Together, these designs improve world-model quality and enable effective RL post-training of VLA policies.

However, RL post-training continually changes VLA behaviors and failure modes, potentially misaligning a static world model trained on earlier policy data and causing trajectory and reward drift. We therefore introduce World-VLA-Loop, a higher-level *co-evolving* paradigm built on these foundations. It closes the loop between the simulator and policy: the world model supports iterative VLA post-training in the virtual environment, while rollouts from each updated policy augment the training data used to refine the world model. Combined with the foundational dataset and world-model design, this closed-loop co-evolution transforms one-shot simulation into an iterative system that remains aligned with the evolving policy. Our contributions are as follows:

- We characterize *near-success* failure modes in video world models and curate SANS to improve action–outcome alignment.
- We develop a state-aware world model that achieves strong action-following precision and native reward prediction by joint reward-and-video supervision on diffusion latents.
- We introduce World-VLA-Loop, a closed-loop framework that establishes a co-evolving cycle between world model and VLA policy learning through iterative refinement.
- We conduct evaluation across both simulation and real-world settings, showing that World-VLA-Loop significantly improves VLA performance across simulation and physical deployment while reducing reliance on costly real-world interaction.

2 Related Work

VLA for Robot Manipulation By leveraging the strong priors of foundation VLMs, VLA has emerged as a mainstream paradigm for robot manipulation. Pretrained VLMs offer strong generalization and are typically fine-tuned on robotic demonstration data to adapt to action spaces (Driess et al., 2023; Karamcheti et al., 2024; Kim et al., 2024; 2025; Black et al., 2024). However, the dominant training paradigm remains imitation learning, which suffers from data scarcity and compounding errors at test time, ultimately limiting generalization. To mitigate these limitations, reinforcement learning has become an attractive option, enabling policies to learn from online rollouts beyond static

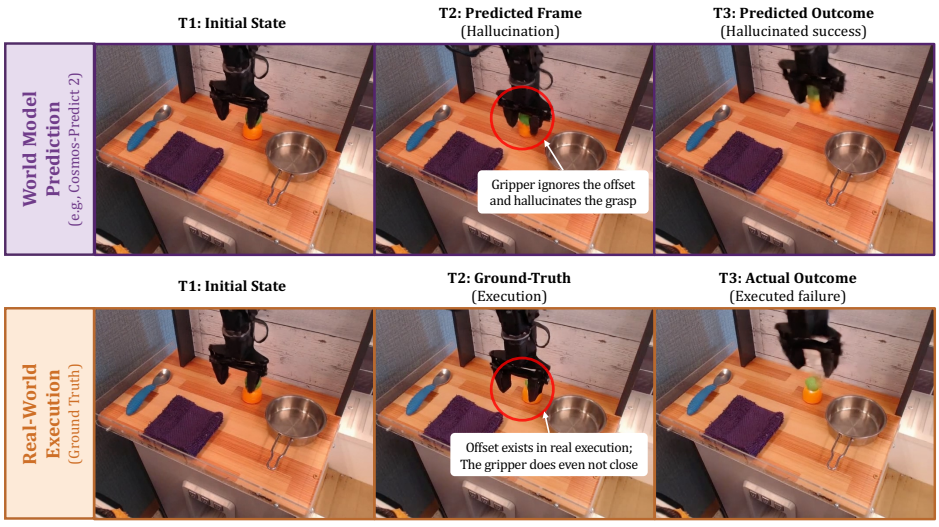


Figure 2: Current world models struggle to accurately simulate failure cases stemming from minor action errors, largely because they inadequately model fine-grained interaction dynamics and precise action conditioning. Transparent overlays denote ground-truth gripper trajectories, illustrating cases where the robot fails to grasp the object.

human demonstrations. Representative works include VLA-RL (Lu et al., 2025a), RL-VLA (Liu et al., 2025a), SimpleVLA-RL (Li et al., 2025a), RLinf-VLA (Zang et al., 2025), among others. A fundamental bottleneck for adapting RL post-training to the real world is prohibitive cost: extensive rollouts, operational overhead for resets and supervision, and safety risks during exploration. To address this, our framework eliminates the costs of physical interaction by simulating the real-world environment within a high-fidelity video world model, enabling efficient policy RL.

Video World Models for Robotics With recent advances in video generation (Wan et al., 2025; Blattmann et al., 2023), controllable video-based world models have become increasingly feasible. Unlike standard generative models that rely solely on text prompts, world models incorporate action sequences as conditioning signals. Many works leverage the spatiotemporal priors of large pretrained video models to enable action-conditioned simulation in domains such as gaming (Bruce et al., 2024; Yu et al., 2025), autonomous driving (Ni et al., 2025; Hu et al., 2024), and robotics (Li et al., 2025b; Xiao et al., 2025; Quevedo et al., 2025; Ali et al., 2025; Zhang et al., 2025; Zhu et al., 2025; Guo et al., 2026; Jiang et al., 2026). Despite this progress, empirical evidence suggests that current video world models still struggle with precise action following, yielding misalignment between predicted frames and the outcomes implied by the actions. To integrate world models into VLA training, particularly RL post-training, practitioners need not only a visual environment but also a reward signal. Recent work explores several designs. VLA-RFT (Li et al., 2025b) employs a visual-similarity-based heuristic reward. World-Env (Xiao et al., 2025) uses a VLM to estimate success from the world model's visual predictions. WMPO (Zhu et al., 2025) implements the reward model with a video encoder and a linear head. Most such approaches rely on an additional trained module and on visual predictions that may themselves be unreliable. We instead predict rewards directly from the world model's latents. Joint training not only enables end-to-end reward prediction but also benefits the visual prediction branch. Concurrent works, including WoVR (Jiang et al., 2026) and VLAW (Guo et al., 2026), also identify the imagination-outcome misalignment issue of world models and propose the similar idea of co-evolving of world model and VLA. However, they still rely on external reward model yet does not investigate the intrinsic reward and its relation with the world state.

3 METHOD

As shown in Figure 3, our framework comprises four phases. We first curate a dataset of success and near-success trajectories across diverse outcomes. We then train an action-conditioned video world model with reward supervision on top of Cosmos-Predict (Ali et al., 2025). The trained model can be efficiently adapted to unseen scenarios using a few demonstrations, and serves as a virtual

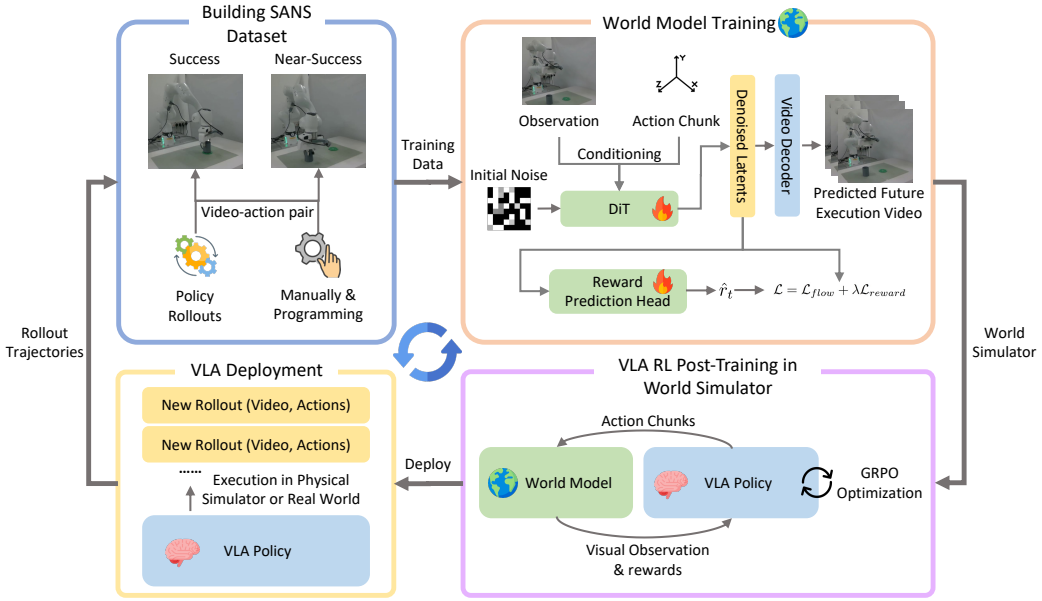


Figure 3: Full pipeline of our proposed framework. The process comprises four phases: (1) Curating the SANS dataset via manual teleoperation and policy rollouts; (2) Pretraining the action-conditioned world model on SANS with joint reward and video supervision; (3) Executing VLA policy rollouts within the world model to perform GRPO optimization; and (4) Deploying the refined policy to collect new failure and success data for further SANS augmentation. This cycle enables the joint optimization of the world model and the VLA policy, iteratively enhancing both performance.

environment for policy evaluation and RL post-training. Real-world policy rollouts are further used to augment the dataset. We detail each stage below.

3.1 Success and Near-Success Dataset

Existing open robotic datasets with action-trajectory annotations, regardless of their diversity, predominantly focus on successful executions as they are primarily curated for imitation learning. This convention restricts the diversity required to train robust world models, which consequently struggle to simulate physically plausible outcomes across diverse failure modes. While recent works like RoboFAC (Lu et al., 2025c) and AHA (Duan et al., 2024) have begun exploring failure data, they are designed for QA-style reasoning and lack action annotations. Moreover, these datasets are largely confined to simulation, leaving its practical utility underexplored.

In this paper, we introduce the Success and Near-Success Dataset (SANS), which leverages trajectories that nearly accomplish the target goal or sub-goal but fail due to minor inaccuracies in end-effector positioning. Such data is critical for training robust world models for two reasons: (1) these trajectories are difficult to distinguish from successful executions, they force the world model to focus on fine-grained nuances in spatial dynamics; and (2) since robot policies frequently exhibit these "near-success" behaviors, including them ensures the virtual environment more accurately reflects the actual failure modes encountered during policy rollouts. We curate the SANS dataset across ManiSkill (Tao et al., 2024), LIBERO (Liu et al., 2023), and our own real-world robotic settings.

For ManiSkill (Tao et al., 2024), we collect success trajectories using simple control policies with ground-truth object poses, and generate failures via pose perturbations. We also capture natural failure modes from policy rollouts, a strategy we follow in LIBERO using OpenVLA-OFT (Kim et al., 2025), and in real-world settings via both teleoperation and OpenVLA-OFT rollouts. Each trajectory includes video, actions, and step-wise binary reward signals indicating success.

Using this pipeline, we build a large ManiSkill dataset for world model pretraining, comprising 35k video-action pairs across 23 tasks with diverse success and failure outcomes. For LIBERO and real-world scenarios, we collect smaller SANS datasets (roughly 50 success and 50 failure trajectories per task), and show that the pretrained model can be efficiently transferred to these unseen tasks.

3.2 STATE-AWARE VIDEO WORLD SIMULATOR

We build our video world simulator on top of Cosmos-Predict 2 (Ali et al., 2025). Cosmos-Predict 2 is pretrained on a large corpus of embodiment-related video data and can be further adapted to incorporate robot action conditions. Given the first h observed frames $x_0, \ldots, x_{h-1}$ and next T timesteps of robot actions $a_1, \ldots, a_T \in \mathbb{R}^6 \cup \{0, 1\}$ (represented as 6-DoF end-effector poses together with the gripper open/close state) the model synthesizes future execution frames for the next T steps, denoted as $x_h, \ldots, x_{h+T-1}$.

Cosmos-Predict utilizes a Diffusion Transformer (DiT) backbone to predict future video chunks autoregressively. To accommodate the action modality, an action embedder MLP maps each input pose into a latent tensor. These action embeddings are integrated into the DiT modules by adding them directly to the diffusion timestamp embeddings. This design enables the model to effectively ground its visual predictions in the provided control sequence while leveraging its strong pretrained generative priors.

To enable reward prediction without relying on external reward models, we augment the original generative model with a reward prediction head, as illustrated in Figure 3. After the DiT model produces diffusion latents, we add a projection layer that maps these latents to a sequence of reward values, supervised by ground-truth rewards collected in the SANS dataset. Specifically, for each sampled timestep t, let z_t denote the denoised latent at the final step. The reward head ϕ predicts the scalar reward $\hat{r}_t = \phi(z_t)$, where ϕ is implemented as a lightweight MLP. During training, we jointly optimize the reward prediction head alongside the original flow matching loss (Lipman et al., 2022): $\mathcal{L} = \mathcal{L}_{flow} + \lambda \|\hat{r}_t - r_t\|^2$, where the weighting factor λ is modulated according to the sampled noise level following the EDM framework (Karras et al., 2022). This ensures the reward head remains robust to the high-variance latents encountered during the early stages of the denoising process.

This design brings two key advantages: **(1)** it provides a reliable reward generation mechanism within the world model. Since rewards are predicted based on the generated video latent states, they are intrinsically aligned with the actual visual outcomes and tend to be more accurate than low-level heuristic proxies or external VLM-based reward models. **(2)**, joint training of the reward head and the video model encourages the generator to better distinguish successful versus failed execution outcomes under different action conditions. This leads to more accurate future video predictions—a crucial capability often lacking in existing video world models.

In practice, we initialize from the Cosmos-Predict checkpoint, first pretrain it on the collected ManiSkill SANS dataset, and then fine-tuning on a small set of success and near-success demonstrations for new scenarios. The ManiSkill-pretrained model provides strong action following ability, while fine-tuning imparts knowledge of new environments, object textures, and task-specific interactions.

3.3 WORLD SIMULATOR FOR POLICY RL POST-TRAINING

Once trained to precisely generate execution outcomes conditioned on action inputs, our video world model serves as a reliable virtual environment for downstream policy evaluation and RL post-training.

We adopt OpenVLA-OFT (Kim et al., 2025) as the base policy and build a GRPO-based post-training pipeline following SimpleVLA-RL (Li et al., 2025a). For each target task with a sparse SANS dataset (typically fewer than 50 success and 50 near-success trajectories), we first fine-tune both the world model and policy to the specific scenario. During RL, the learned world model replaces physics-based simulators, autoregressively generating multi-step visual observations and reward predictions conditioned on policy actions. These predictions are fed back to the policy in a closed loop to enable full rollout generation. The world model also provides step-wise rewards, which are thresholded into binary success signals and used to compute advantages over groups of rollouts for policy updates.

As depicted in Figure 3, a key advantage of our method is the establishment of a comprehensive, iterative joint-optimization framework for both the world model and the VLA policy. Following each RL update, we collect success and near-success rollouts generated by the newly trained VLA policy when deployed in its target environment. These rollouts are then used to augment the SANS dataset for subsequent iterations of world model fine-tuning. This iterative refinement ensures that the world model progressively improves its ability to synthesize precise execution outcomes, which in turn

Table 1: Video generation performance. ↑ indicates higher is better; ↓ indicates lower is better.

Scenario	SSIM ↑	PSNR ↑	LPIPS ↓	MSE ↓
LIBERO	0.90	26.57	0.031	0.0024
Real-World	0.91	29.61	0.059	0.0019
Average	0.91	28.09	0.045	0.0022

enhances the effectiveness of the next RL phase for the VLA policy. In the experimental section, we validate that this joint-optimization process does yield superior VLA RL performance.

4 EXPERIMENTS

The following section evaluates our framework across multiple dimensions, validating the effectiveness and broader applicability of the proposed approach. Our results demonstrate that: **(1)** The world model achieves an average visual alignment of 88.5% and a reward alignment of 87.25%, establishing it as a high-fidelity simulator for VLA policy rollouts. **(2)** Through RL post-training within our simulator, the OpenVLA-OFT policy achieves an average success rate improvement of 12.7% on the LIBERO benchmarks, as well as 23.4% and 13.3% in real-world scenarios. **(3)** The proposed iterative refinement process further increases real-world task accuracy by an average of 13.3% relative to the first RL checkpoint, underscoring the efficacy of the joint optimization cycle. For completeness, we compare our method with additional baselines in Appendix C.

4.1 EXPERIMENT SETTINGS

Experiment Scenarios. We conduct experiments on the LIBERO benchmark for simulated tasks, complemented by a self-constructed laboratory setup to evaluate real-world applicability. In both settings, the configuration consists of a Franka research arm (Haddadin et al., 2022) and a single third-person-view RealSense D435 camera (Keselman et al., 2017) mounted on a fixed position to provide observation input. For the real-world experiments, we design two major tasks *Pick and Place Cup* and *Pushing Cube*. We use a unified chunk size of 24 for all tasks in simulation and real-world. For more implementation details and real-world settings, please refer to Appendix A.

Evaluation Methods. We evaluate our pipeline from two perspectives. First, we assess the generative performance of the World-VLA-Loop world model, focusing on visual fidelity and action-following precision. Building upon this, we quantify the model's utility as a simulator by measuring the success rate improvements of a VLA policy during RL post-training.

4.2 WORLD MODEL EVALUATION

To ensure that the world model provides reliable generation suitable for accurate policy rollouts, we first evaluate its generative quality. Specifically, given a test dataset of image-actions pairs, the world model takes an initial image and a sequence of absolute robot end-effector poses as input. It then autoregressively predicts the video frames corresponding to the execution of those actions. We evaluate the resulting generations based on two primary dimensions: video quality and action-following accuracy.

Video Quality Since the generated frames serve as the major visual observations for the policy, in order to maintain the stable behavior of VLA, ensuring high visual fidelity is crucial. As indicated in Table 1, World-VLA-Loop world model achieves high-quality generation results across both simulation and real-world scenarios, yielding high-fidelity results with negligible structural and perceptual distortion. The consistency between the LIBERO and real-world metrics demonstrates that our world model generalizes effectively across different environments, providing a reliable visual foundation for downstream policy rollouts.

Generation Accuracy A critical requirement for a world simulator is the ability to faithfully reflect the causal consequences of an action trajectory. While pixel-level metrics like MSE provide a baseline, a more robust evaluation measures the alignment between generated and ground-truth outcomes. To this end, we categorize the final state of each trajectory as either a success or failure and calculate the

Table 2: Outcome Alignment Performance across specific LIBERO and Real-World tasks. For each task we evaluate 50 samples within set, and report the percentage of samples where the predicted success/failure matches the ground truth.

Metric	LIBERO-Object		LIBERO-Goal		LIBERO-Spatial		Real-World	
	Task 1	Task 2	Task 1	Task 2	Task 1	Task 2	Place Cup	Push Cube
Visual Alignment	92%	90%	94%	78%	86%	94%	90%	84%
Reward Alignment	88%	90%	90%	76%	88%	94%	94%	78%

Table 3: Success rate of OpenVLA-OFT before and after RL training. Success rates are computed across 500 rollouts for the LIBERO suites and 30 physical rollouts for our real-world experiments. The Oracle results are obtained by RL training inside LIBERO simulator with ground truth videos and rewards, and the numbers are averaged across the entire tasksuite.

Model	LIBERO-Object		LIBERO-Goal		LIBERO-Spatial		Real-World	
	Task 1	Task 2	Task 1	Task 2	Task 1	Task 2	Place Cup	Push Cube
SFT Base	73.9%	73.9%	91.9%	86.1%	83.9%	87.9%	13.3%	26.7%
RL Post-Training (Ours)	97.9%	91.9%	100%	96.2%	93.9%	94.0%	36.7%	40.0%
Δ vs SFT	+24.0%	+18.0%	+8.1%	+10.1%	+10.0%	+6.1%	+23.4%	+13.3%
RL Post-Training (Oracle)	98.7%	98.7%	98.8%	98.8%	98.2%	98.2%	-	-

alignment rate—the percentage of samples where the predicted outcome matches the ground truth. Since our world model simultaneously generates video frames and scalar rewards, we report two distinct alignment metrics:

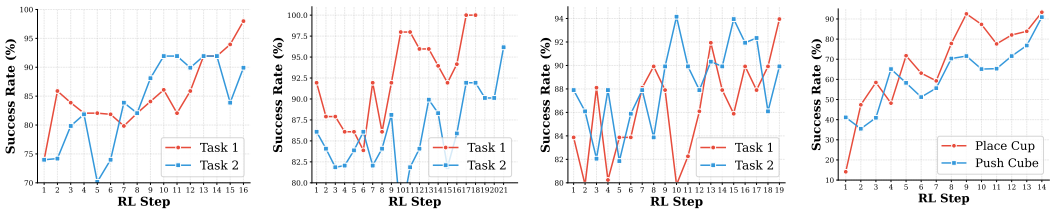
- *Visual Alignment*: Measures the faithfulness of the pixel-level generation. Success or failure is determined by evaluating the final generated video frames to judge the task outcome.
- *Reward Alignment*: Measures the world model reward prediction. A predicted trajectory is classified as a success if the generated reward exceeds a threshold of 0.9.

As shown in Table 2, our method effectively distinguishes between successful and failed trajectories across both simulation and real-world scenarios, achieving an average alignment accuracy exceeding 80% for both metrics. Furthermore, the outcomes classified via reward alignment are highly consistent with those from visual alignment. This strong correlation underscores the reliability of the internal reward prediction head and demonstrates its learned success criteria are well-aligned with human judgment.

4.3 World Simulator for VLA Post-training

Basic RL Training Leveraging its reliable generative performance, World-VLA-Loop world model serves as a learned simulator for VLA policy post-training. For each task, we first conduct task-specific finetuning for both the OpenVLA policy and the world model. Subsequently, the policy is deployed within the world model environment for RL post-training. We adopt the RL pipeline from SimpleVLA-RL (Li et al., 2025a), adapting the rollout process to operate entirely within our neural simulator. In this configuration, only the initial frame is sourced from the original dataset; all subsequent observations are autoregressively generated by the world model conditioned by the policy's predicted actions. Finally, the RL reward is derived from the world model's reward head, utilizing a thresholding mechanism to determine the binary success signals required for group relative optimization (Shao et al., 2024).

We conduct experiments across three task suites from the LIBERO benchmark and tasks in real-world settings. Since LIBERO-100 primarily involves long-horizon tasks requiring the generation of over 300 video frames, where current autoregressive video models often suffer from severe quality drift, we leave its exploration for future work. For each task, 80–100 success and near-success trajectories are utilized to fine-tune the world model; among these, a subset of approximately 50 success trajectories is used to fine-tune the OpenVLA policy. As illustrated in Table 3 and Figure 4, success rates across both the LIBERO suites and the real-world tasks exhibit significant improvements throughout the RL post-training process. To clarify, the real-world success rate curves in Figure 4 are evaluated



(a) LIBERO-Object (b) LIBERO-Goal (c) LIBERO-Spatial (d) Real-World

Figure 4: Success rate improvements along World-VLA-Loop RL training steps.

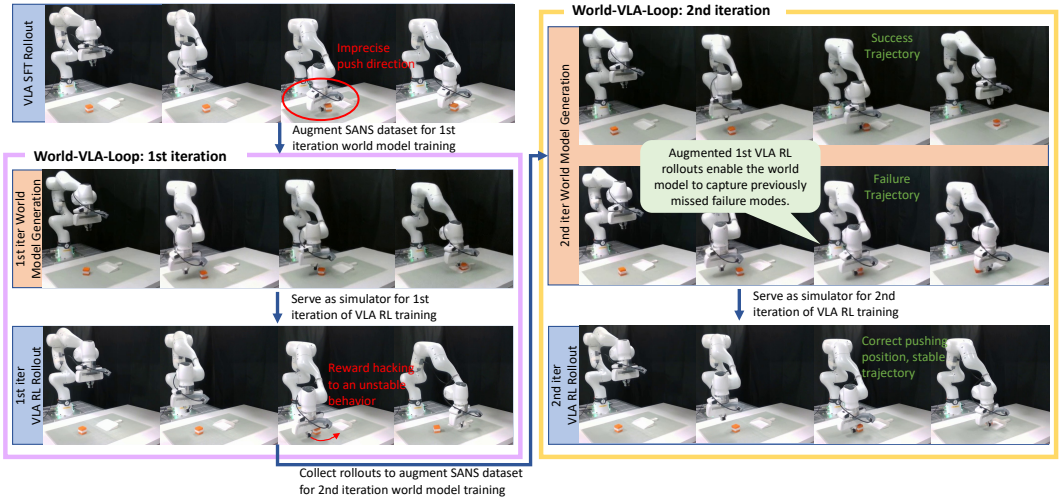


Figure 5: Examples of world model generated rollouts and actual execution videos by both SFT policy and RL post-trained policy. The policy rollouts augment the SANS dataset for world model training, which in turn serves as better simulators for RL iterations, leading to improved policies.

within the world simulator, whereas the final results reported in Table 3 are obtained through physical, real-world experiments.

Iterative Refinement through Policy Rollouts As discussed in Section 3.3, once a functional policy is established, its rollout trajectories can be leveraged to augment the SANS dataset, which is then utilized to further refine the world model, enhancing its precision and action-following capabilities. Ultimately, this iterative cycle improves the overall efficiency of VLA reinforcement learning within the neural simulator.

We conduct iterative experiments within our real-world setting. In the initial phase (Step 0), the SUPA dataset comprises manually collected success and near-success trajectories, alongside rollouts from the SFT OpenVLA-OFT baseline. For the next iteration (Step 1), the dataset is expanded to include newly generated trajectories from the RL-optimized policy produced in the previous step. Both world models are initialized from the ManiSkill-pretrained checkpoint, while the VLA policies for each RL stage begin from the base SFT version. As illustrated in Figure 1 (b), this iterative refinement successfully boosts performance: the base SFT policy achieved only the success rate of 13.3% and 26.7%, after two RL iterations within our loop, it eventually reached 50.0% and 53.3% on two real-world tasks. These results demonstrate the overall effectiveness of our proposed closed-loop framework in enabling continuous policy and world model improvement.

4.4 Ablation Study

Reward Prediction Head We conduct ablation experiments to validate the effectiveness of the reward prediction head. First, we examine the world model's performance after removing this head.

As indicated in Table 4, visual alignment drops by approximately 30% without reward supervision, suggesting that joint training of the reward head and the diffusion backbone significantly improves the model's ability to discriminate between different action outcomes. Second, we compare our reward head against Qwen3-VL (Bai et al., 2025) acting as a success-failure judge—a common approach in prior literature. Specifically, we provide Qwen3-VL with the video frames generated by our world model and prompt it to evaluate the task's final success status. Notably, these are the same frames used to produce our internal reward predictions. The results in Table 4 show that the VLM suffers from hallucination in specific domains and gets weaker accuracy, making it a less reliable reward function for RL compared to our method. While task-specific fine-tuning might improve VLM accuracy, it would significantly increase pipeline complexity. In contrast, our reward prediction head, which shares the backbone with the video generator, is not only more accurate but also enhances video quality through the benefits of joint training. For more results please refer to Appendix B.

Near-Success Data during World Model Training We also fine-tune the world model while excluding near-success trajectories. As indicated in Table 4, the visual alignment of the generated videos decreases significantly under this configuration. This confirms that near-success trajectories are indispensable for world model training, as they enable the model to internalize potential failure modes possible during VLA execution and more faithfully reflect resulting action outcomes.

Table 4: Ablation results. We investigate how different design choices influence the world model prediction accuracy.

Metric	LIBERO-Object		Real-World
	Task 1	Task 2	
Visual Alignment			
w/o near-success data	60%	66%	54%
w/o reward prediction head	68%	70%	80%
ours	**92%**	**90%**	90%
Reward Alignment			
Qwen3-VL-8B-Instruct	84%	58%	84%
ours	**88%**	**90%**	**94%**

4.5 Qualitative Results

Figure 5 presents qualitative examples of synthesized world model rollouts alongside actual VLA policy executions. Through our iterative refinement loop, the world model progressively learns to cover a broader action space and better represent the stochasticity of VLA exploration. In the policy rollouts, we observe that the SFT-baseline tends to generate imprecise pushing poses. While the first RL iteration mitigates this issue, the policy initially engages in reward hacking, converging on suboptimal behavior such as moving in front of the cube, since the first iteration world model fails to accurately model the spatial nuances of such trajectory. After augmenting the SANS dataset, the world model develops a finer awareness of these edge cases. Consequently, the VLA policy trained within this refined environment achieves significantly more precise and robust grasping poses. For additional qualitative examples, please refer to Appendix D.

5 Conclusion

We presented World-VLA-Loop, a novel framework for world-model-based VLA reinforcement learning through the joint optimization of our world model and VLA policy. By leveraging the SANS dataset curated with near-success trajectories, we trained a high-fidelity world model with integrated reward supervision, enabling effective policy refinement within a virtual environment. Furthermore, our co-evolving paradigm uses real-world rollouts to augment training data, progressively enhancing both world model grounding and policy performance. Experiments demonstrate significant success rate gains across simulation and real-world benchmarks, offering a scalable direction for real-world RL by reducing reliance on costly physical interactions.

Limitations and Future Work: Current autoregressive video models suffer from limited context memory and quality drift, which diminishes performance in long-horizon tasks exceeding 300 frames. Future research will explore integrating video backbones with enhanced long-term stability. Additionally, transitioning from sparse final-state rewards to step-wise intermediate sub-goals could further improve reward head accuracy and RL convergence.

AI USE STATEMENT

In this work, we used generative AI tools for implementation for parts of the modules in the coding projects, as well as polishing the paper language. We have not used generative AI tools for original paper writting. We have reviewed all AI-assisted work. LLM-generated code was verified and tested for correctness by 2 authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We provide the anonymous downloadable source code in the supplementary experiments. The source code will be further sorted up and open-sourced after the paper being accepted.

REFERENCES

Arslan Ali, Junjie Bai, Maciej Bala, Yogesh Balaji, Aaron Blakeman, Tiffany Cai, Jiaxin Cao, Tianshi Cao, Elizabeth Cha, Yu-Wei Chao, et al. World simulation with video foundation models for physical ai. *arXiv preprint arXiv:2511.00062*, 2025.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023.

Jake Bruce, Michael D Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, et al. Genie: Generative interactive environments. In *Forty-first International Conference on Machine Learning*, 2024.

Tianxing Chen, Zanxin Chen, Baijun Chen, Zijian Cai, Yibin Liu, Zixuan Li, Qiwei Liang, Xianliang Lin, Yiheng Ge, Zhenyu Gu, et al. Robotwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation. *arXiv preprint arXiv:2506.18088*, 2025.

Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, Wenlong Huang, et al. Palm-e: An embodied multimodal language model. 2023.

Jiafei Duan, Wilbert Pumacay, Nishanth Kumar, Yi Ru Wang, Shulin Tian, Wentao Yuan, Ranjay Krishna, Dieter Fox, Ajay Mandlekar, and Yijie Guo. Aha: A vision-language-model for detecting and reasoning over failures in robotic manipulation. *arXiv preprint arXiv:2410.00371*, 2024.

Yanjiang Guo, Tony Lee, Lucy Xiaoyang Shi, Jianyu Chen, Percy Liang, and Chelsea Finn. Vlaw: Iterative co-improvement of vision-language-action policy and world model. *arXiv preprint arXiv:2602.12063*, 2026.

Sami Haddadin, Sven Parusel, Lars Johannsmeier, Saskia Golz, Simon Gabl, Florian Walch, Mohamadreza Sabaghian, Christoph Jähne, Lukas Hausperger, and Simon Haddadin. The franka emika

robot: A reference platform for robotics research and education. *IEEE Robotics & Automation Magazine*, 29(2):46–64, 2022. doi: 10.1109/MRA.2021.3138382.

Xiaotao Hu, Wei Yin, Mingkai Jia, Junyuan Deng, Xiaoyang Guo, Qian Zhang, Xiaoxiao Long, and Ping Tan. Drivingworld: Constructing world model for autonomous driving via video gpt. *arXiv preprint arXiv:2412.19505*, 2024.

Physical Intelligence, Ali Amin, Raichelle Aniceto, Ashwin Balakrishna, Kevin Black, Ken Conley, Grace Connors, James Darpinian, Karan Dhabalia, Jared DiCarlo, et al. $\pi_{0.6}$: a vla that learns from experience. *arXiv preprint arXiv:2511.14759*, 2025.

Zhennan Jiang, Shangqing Zhou, Yutong Jiang, Zefang Huang, Mingjie Wei, Yuhui Chen, Tianxing Zhou, Zhen Guo, Hao Lin, Quanlu Zhang, et al. Wovr: World models as reliable simulators for post-training vla policies with rl. *arXiv preprint arXiv:2602.13977*, 2026.

Siddharth Karamcheti, Suraj Nair, Ashwin Balakrishna, Percy Liang, Thomas Kollar, and Dorsa Sadigh. Prismatic vlms: Investigating the design space of visually-conditioned language models. In *Forty-first International Conference on Machine Learning*, 2024.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *Advances in neural information processing systems*, 35:26565–26577, 2022.

Leonid Keselman, John Iselin Woodfill, Anders Grunnet-Jepsen, and Achintya Bhowmik. Intel realsense stereoscopic depth cameras. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pp. 1–10, 2017.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-tuning vision-language-action models: Optimizing speed and success. *arXiv preprint arXiv:2502.19645*, 2025.

Haozhan Li, Yuxin Zuo, Jiale Yu, Yuhao Zhang, Zhaohui Yang, Kaiyan Zhang, Xuekai Zhu, Yuchen Zhang, Tianxing Chen, Ganqu Cui, et al. Simplevla-rl: Scaling vla training via reinforcement learning. *arXiv preprint arXiv:2509.09674*, 2025a.

Hengtao Li, Pengxiang Ding, Runze Suo, Yihao Wang, Zirui Ge, Dongyuan Zang, Kexian Yu, Mingyang Sun, Hongyin Zhang, Donglin Wang, et al. Vla-rft: Vision-language-action reinforcement fine-tuning with verified rewards in world simulators. *arXiv preprint arXiv:2510.00406*, 2025b.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

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

Jijia Liu, Feng Gao, Bingwen Wei, Xinlei Chen, Qingmin Liao, Yi Wu, Chao Yu, and Yu Wang. What can rl bring to vla generalization? an empirical study. *arXiv preprint arXiv:2505.19789*, 2025a.

Xiaokang Liu, Kevin Yuchen Ma, Chen Gao, and Mike Zheng Shou. Diffusion models in robotics: A survey. 2025b.

Guanxing Lu, Wenkai Guo, Chubin Zhang, Yuheng Zhou, Haonan Jiang, Zifeng Gao, Yansong Tang, and Ziwei Wang. Vla-rl: Towards masterful and general robotic manipulation with scalable reinforcement learning. *arXiv preprint arXiv:2505.18719*, 2025a.

Guanxing Lu, Baoxiong Jia, Puhao Li, Yixin Chen, Ziwei Wang, Yansong Tang, and Siyuan Huang. Gwm: Towards scalable gaussian world models for robotic manipulation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9263–9274, 2025b.

Weifeng Lu, Minghao Ye, Zewei Ye, Ruihan Tao, Shuo Yang, and Bo Zhao. Robofac: A comprehensive framework for robotic failure analysis and correction. *arXiv preprint arXiv:2505.12224*, 2025c.

Jingcheng Ni, Yuxin Guo, Yichen Liu, Rui Chen, Lewei Lu, and Zehuan Wu. Maskgwm: A generalizable driving world model with video mask reconstruction. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 22381–22391, 2025.

Julian Quevedo, Percy Liang, and Sherry Yang. Evaluating robot policies in a world model. *arXiv preprint arXiv:2506.00613*, 2025.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Stone Tao, Fanbo Xiang, Arth Shukla, Yuzhe Qin, Xander Hinrichsen, Xiaodi Yuan, Chen Bao, Xinsong Lin, Yulin Liu, Tse-kai Chan, et al. Maniskill3: Gpu parallelized robotics simulation and rendering for generalizable embodied ai. *arXiv preprint arXiv:2410.00425*, 2024.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

Hongchi Xia, Entong Su, Marius Memmel, Arhan Jain, Raymond Yu, Numfor Mbiziwo-Tiapo, Ali Farhadi, Abhishek Gupta, Shenlong Wang, and Wei-Chiu Ma. Drawer: Digital reconstruction and articulation with environment realism. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 21771–21782, 2025.

Junjin Xiao, Yandan Yang, Xinyuan Chang, Ronghan Chen, Feng Xiong, Mu Xu, Wei-Shi Zheng, and Qing Zhang. World-env: Leveraging world model as a virtual environment for vla post-training. *arXiv preprint arXiv:2509.24948*, 2025.

Jiwen Yu, Yiran Qin, Xintao Wang, Pengfei Wan, Di Zhang, and Xihui Liu. Gamefactory: Creating new games with generative interactive videos. *arXiv preprint arXiv:2501.08325*, 2025.

Hongzhi Zang, Mingjie Wei, Si Xu, Yongji Wu, Zhen Guo, Yuanqing Wang, Hao Lin, Liangzhi Shi, Yuqing Xie, Zhexuan Xu, et al. Rlinf-vla: A unified and efficient framework for vla+ rl training. *arXiv preprint arXiv:2510.06710*, 2025.

Jiahui Zhang, Ze Huang, Chun Gu, Zipei Ma, and Li Zhang. Reinforcing action policies by prophesying. *arXiv preprint arXiv:2511.20633*, 2025.

Fangqi Zhu, Zhengyang Yan, Zicong Hong, Quanxin Shou, Xiao Ma, and Song Guo. Wmpo: World model-based policy optimization for vision-language-action models. *arXiv preprint arXiv:2511.09515*, 2025.

A Implementation Details

A.1 World Model Training

We transfer the world model from Cosmos Predict 2 action-conditioned version checkpoint. As stated in the main paper, we first train the world model on a collected Maniskill SANS dataset. This stage enables the model to grasp basic relationship between Franka robot arm and conditioning action, as well as some fundamental physics dynamics in manipulation scenarios. After this pretraining phase, we can adapt our model towards any downstream task with few data (less than 100 success and near-success trajectories). For the downstream task fine-tuning, we adopt a smaller learning rate and keep using full parameter tuning. For Maniskill pretraining we use the learning rate of $1e-3$, and for downstream tasks we use $1e-5$. We use FusedAdam as optimizer.

A.2 VLA Deployment in World Model

We adapt the SimpleVLA-RL codebase by replacing the physical simulator with our learned world model, while maintaining a consistent interface across both environments. The group size for GPRO to compute the advantage is set to 8. To facilitate the batch optimization required for GRPO, we implement the interaction between the VLA policy and the world model via a request-response architecture. The world model operates as a backend server that monitors incoming requests; once the VLA policy generates an action chunk, it is transmitted to the server, which allocates the task to an available model worker to compute the subsequent observation and reward signals. We use the exact same set of hyperparameters as SimpleVLA-RL (Li et al., 2025a) except for the chunk size using 24.

A.3 Real-World Experimental Settings

We conducted two real-world experiments, covering common robot manipulation actions 'pick and place' and 'push'. As depicted in Figure 6, in our real-world experiments, the robot was tasked with the prompt *"Pick up the cup and place it on the green plate."* and *"Push the cube into the dustpan."* respectively. The positions of the cup, cube and plate were randomized. These tasks are challenging for VLA policies as it requires rigorous grasping and pushing precision, otherwise in the first task, the rounded cup easily slips away, and in the second task the gripper tends to push crookedly.

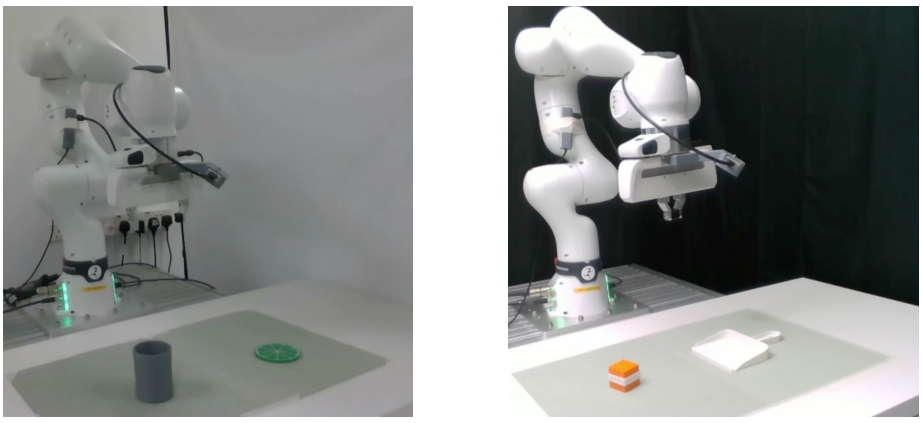


Figure 6: Real world experiments setup.

A.4 Computation Cost

Inference Efficiency A batched world model generation of 24 video frames takes approximately 6 seconds on a single NVIDIA H100 node. In our experiments, a full RL training session for a task typically converges within 20–40 optimization steps, totaling roughly 25 to 35 hours. This represents a significant improvement over traditional physical rollouts, which require labor-intensive manual resets and constant human supervision.

Training Efficiency World model fine-tuning on a specific task with 100 trajectories typically converges within 4 hours on 8 H100 GPUs, making the approach highly affordable and scalable for new task adaptation.

B Additional Ablation

B.1 Quality Analysis on Long-Horizon Tasks

To clarify on the long-horizon tasks, we measure the video quality of world model in different frame length:

Table 5: Performance Metrics across Frames

Frame	PSNR	SSIM
200	23.5890	0.7430
250	22.2148	0.6731
300	20.2351	0.6282

Quality analysis confirms that severe degradation typically occurs only after the 300-frame mark, which is sufficient for the majority of robotic manipulation tasks. LIBERO-Object, Spatial and Goal tasksuites typically require around 220 frames (10 passes), LIBERO-Long tasks extend to 300-400 frames (15 passes) and thus the video becomes too blurry for RL accuracy in the task success section of these tasks. In the future, we will leverage newer auto-regressive video generation techniques and we believe this limitation will be finally addressed.

B.2 VLM Reward for RL

In Section 4, we compared the world model generation accuracy of using VLM and reward prediction head. Here we also conducted RL experiments on LIBERO-Object using a VLM-based reward (Qwen3.5-VL-Instruct 8B). We replaced our integrated reward head with the VLM, feeding generated video frames to the VLM with prompts to output binary success states. The results are shown below:

Table 6: Performance comparison on object tasks.

Model	Object-1	Object-2
Ours	97.9%	91.9%
Ours w/ VLM	93.9%	88.0%

This experiment demonstrates two key findings: (1) High-quality video generation: The fact that VLM rewards achieve reasonable performance validates that our world model generates sufficiently realistic and accurate videos that can be effectively interpreted by external VLMs. (2) Efficiency advantage: Our integrated reward head achieves superior performance while being significantly more computationally efficient than VLM-based alternatives. This confirms that joint reward supervision successfully enables the world model to distinguish between success and failure rollouts more accurately than external reward functions.

C Additional Comparison with Baselines

C.1 Comparison with other RL Post-Training Methods

To isolate the role of the world model, we conducted an additional experiment on the real-world Push Cube task. To match our experimental setting of performing RL with an offline dataset, we post-trained the VLA policy using DPO on the same SANS dataset, which contains both successful and failed trajectories. As shown in Table 7, direct offline post-training tends to overfit to the limited failure modes covered by the collected data. Consequently, the policy becomes stuck in specific failure behaviors, resulting in only a minor improvement.

Table 7: Comparison of different RL post-training methods on the real-world Push Cube task.

Method	Success Rate
SFT Base	26.7%
DPO	33.3%
Ours	53.3%

In contrast, our method uses SANS to train a world model, which then serves as an online simulator for RL. This enables the policy to explore beyond the fixed offline dataset while avoiding large-scale physical rollouts. Therefore, the improvement does not arise solely from the SANS data itself, but from using SANS to construct a generalizable world environment for online RL exploration.

C.2 Comparison with WMPO

Compared with WMPO, our method differs in two main aspects: (i) we use an intrinsic reward prediction head that is jointly trained with the world model, and (ii) we leverage closed-loop feedback from VLA rollouts to iteratively refine the world model. To provide a matched comparison of the reward design, we train the same VideoMAE-style reward model used in WMPO on our collected SANS data, following the training paradigm of WMPO.

We evaluate the reward models on both real-world collected trajectories and synthetic trajectories generated by our world simulator for the real-world Push Cube task. The reward alignment results are summarized in Table 8.

Table 8: Reward alignment on collected and synthetic trajectories for the real-world Push Cube task.

Reward Model	Collected Trajectories	Synthetic Trajectories
WMPO reward model	94%	58%
Ours	90%	78%

The WMPO reward model achieves high accuracy on the collected-trajectory domain, but its performance drops substantially on synthetic videos, suggesting limited generalization to world-model-generated rollouts. In contrast, our reward head provides more consistent reward predictions for generated outcomes, which is important when using the world model as an RL simulator. As shown in Appendix B.2, a VLM-based reward can achieve reasonable RL performance on our generated videos. This suggests that the main limitation of the separate WMPO-style reward module lies in its generalization to the generated-video domain, rather than in the quality of the generated videos themselves.

We further replace our intrinsic reward head with the WMPO reward model during world-model-based RL. As shown in Table 9, the final real-world success rate decreases substantially when using the WMPO-style reward model, indicating the impact of the lower-quality reward signal.

Table 9: Real-world success rates when using different reward models for world-model-based RL on the Push Cube task.

Method	Success Rate
SFT Base	26.7%
WMPO-style reward	13.3%
Ours	53.3%

These results suggest that, under matched settings, the intrinsic reward head is important for stable and effective RL in the learned world simulator. Moreover, our reward head is substantially more parameter-efficient than the separate reward model used in WMPO.

D VISUAL EXAMPLES

D.1 ADDITIONAL WORLD MODEL GENERATION RESULTS

We provide additional qualitative examples of our world model's generation capabilities in this section. Figure 7 illustrates failure rollouts synthesized by the world model when conditioned on erroneous action trajectories. The generated video frames demonstrate that the world model accurately adheres to the conditioning actions and precisely distinguishes between nuanced outcomes resulting from incorrect execution. This capability ensures that the model covers a diverse range of failure modes, providing the necessary grounding for effective policy refinement. Figure 8 provides some examples of generated successful trajectories. Most of the videos show that world model can clearly distinguish between failure and success action inputs.

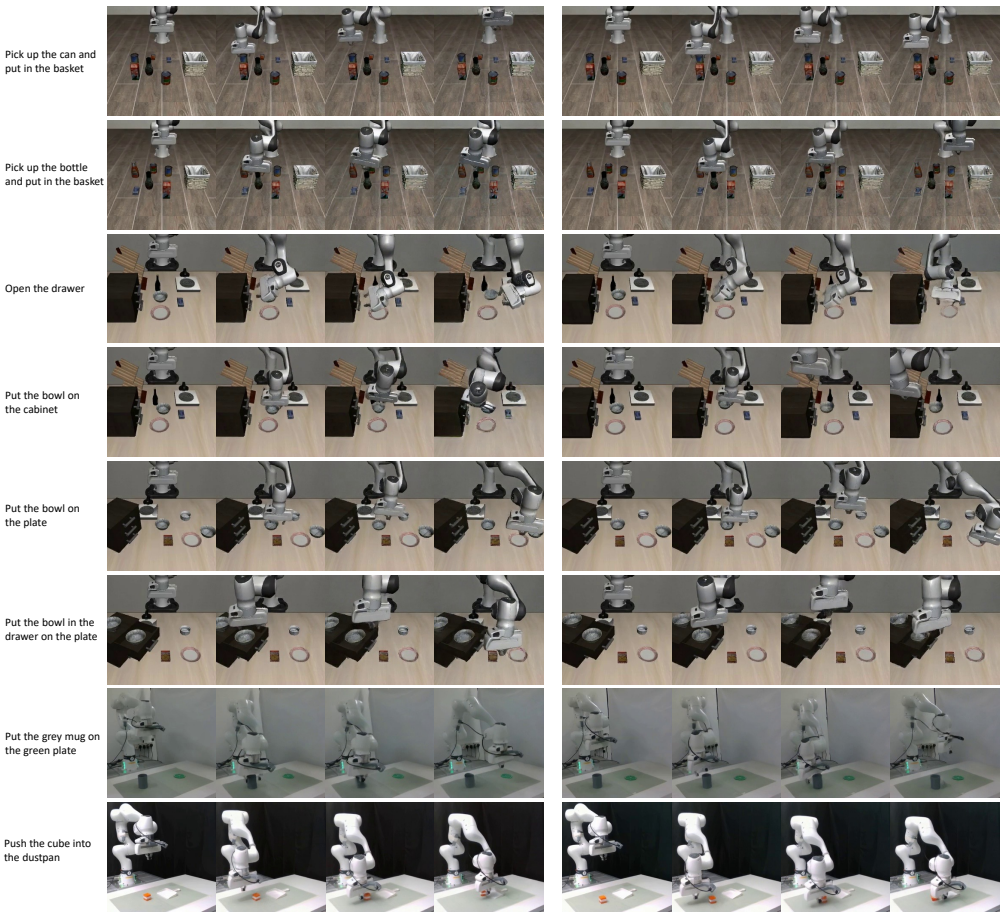


Figure 7: Examples of world model generated rollouts of failure trajectories.

D.2 WORLD MODEL IN UNSEEN CASES

Notably, our world model demonstrates the ability to synthesize trajectories unseen during the fine-tuning of the target SANS dataset, leveraging the broad priors learned during its pretraining on ManiSkill. Figure 9 illustrates cases where the world model is conditioned on entirely novel action sequences and robot arm trajectories. Despite the stochastic nature of these unseen inputs, the model faithfully adheres to the action conditioning, generating physically plausible motion aligning with ground truth. This robustness indicates that the world model has effectively internalized the underlying mapping between control actions and robotic kinematics, rather than merely memorizing training sequences.



Figure 8: Examples of world model generated rollouts of successful trajectories.

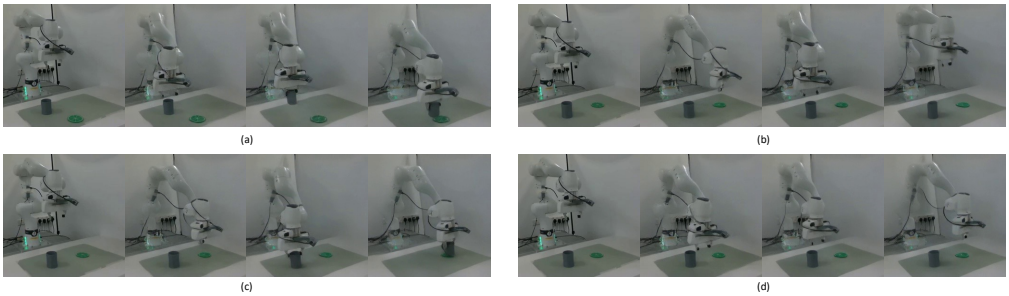


Figure 9: World model generation results on unseen action sequences. (a) The plate is initialized directly in front of the mug. (b) The gripper moves to the right, returns to the mug's overhead position, and retracts to a neutral pose. (c) The gripper first moves to the right, followed by a successful pick-and-place of the mug onto the plate. (d) A sequence of forward, backward, and lateral oscillations.