Reinforcing VLAs in Task-Agnostic World Models

Anonymous Author(s)
Affiliation
Address
email

Abstract

Post-training Vision-Language-Action (VLA) models via reinforcement learning (RL) in learned world models has emerged as an effective strategy to adapt to new tasks without costly real-world interactions. However, while using imagined trajectories reduces the sample complexity of policy training, existing methods still heavily rely on task-specific data to fine-tune both the world and reward models, fundamentally limiting their scalability to unseen tasks. To overcome this, we argue that world and reward models should capture transferable physical priors that enable zero-shot inference. We propose **RAW-Dream** (**R**einforcing VLAs in task-**A**gnostic **W**orld **Dream**s), a new paradigm that completely disentangles world model learning from downstream task dependencies. RAW-Dream utilizes a world model pre-trained on diverse task-free behaviors for predicting future rollouts, and an off-the-shelf Vision-Language Model (VLM) for reward generation. Because both components are task-agnostic, VLAs can be readily finetuned for any new task entirely within this zero-shot imagination. Furthermore, to mitigate world model hallucinations, we introduce a dual-noise verification mechanism to filter out unreliable rollouts. Extensive experiments across simulation and real-world settings demonstrate consistent performance gains, proving that generalized physical priors can effectively substitute for costly task-dependent data, offering a highly scalable roadmap for VLA adaptation.

1 Introduction

Vision-Language-Action (VLA) policies trained via imitation learning have advanced robotic manipulation considerably [3, 6, 12, 15], yet remain brittle beyond their training distribution. Reinforcement learning (RL) can in principle address this through trial-and-error improvement, but online RL on physical robots is prohibitively costly. A rapidly growing line of work resolves this tension by leveraging action-conditioned video world models (WMs) as virtual simulators, in which VLA policies are refined entirely through imagination [8, 10, 13, 18, 21, 30, 37, 39, 43, 44, 46], yielding substantial gains over pure imitation baselines.

However, a critical limitation remains: **existing methods operate exclusively on tasks that their WMs were already trained on**. While real-world deployment routinely demands rapid adaptation to novel tasks (*e.g.*, new objects, rearranged layouts, unfamiliar instructions), prior methods rely on substantial rollout data collected specifically on the target task to train their WMs [8, 13, 21, 30, 46]. Reward mechanisms are similarly entangled, requiring task-specific classifiers or reference trajectories[13, 18, 21, 46]. The primary appeal of using a WM is to bypass costly real-world interactions and act as a general-purpose simulator; yet, current methods require knowing the target tasks in advance and collecting extensive real-world data before the pipeline can even begin. Consequently, the WM must be rebuilt whenever new tasks arise, reducing it to a costly, single-use data-augmentation tool and structurally precluding unseen-task generalization.

To break this task-specific data bottleneck, we draw on a fundamental observation: the physical dynamics of a robotic workspace are inherently task-independent. A bowl slides identically regardless

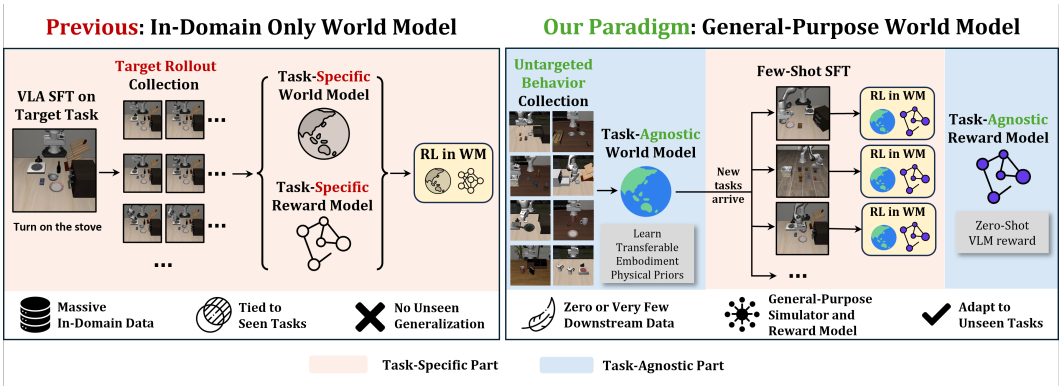


Figure 1: *Left:* Previous WM-based RL pipelines for VLA post-training tightly couple the WM and reward models to known target tasks, requiring thousands of in-domain rollouts, precluding unseen adaptation. *Right:* RAW-Dream decouples dynamics learning from task semantics. A general-purpose WM pre-trained on diverse task-free behaviors captures transferable physical priors, while a foundation VLM provides zero-shot rewards. Together, they enable data-efficient RL on novel tasks.

of whether the instruction is "put the bowl on the shelf" or "push the bowl aside". Following this insight, we propose **RAW-Dream** (**R**einforcing VLAs in task-**A**gnostic **W**orld **Dream**s), a new paradigm for VLA post-training where both the WM and the reward function are strictly *task-agnostic*. To achieve this, the WM is pre-trained on diverse, broad-coverage interaction data (*e.g.*, play data) to serve as a reusable, general-purpose physical simulator [4, 23, 25, 28, 35, 40]. Concurrently, we leverage the generalized reasoning of off-the-shelf vision-language models (VLMs) [30, 43], which can provide reliable, zero-shot reward signals by judging task success directly from the imagined video rollout and text instruction.

Under RAW-Dream, adapting to novel downstream tasks becomes highly data-efficient. The VLA policy is first anchored to the new task semantics via lightweight supervised fine-tuning (SFT) on minimal expert demonstrations. It is then optimized via RL entirely inside the frozen, task-agnostic WM using the zero-shot VLM reward. We argue this represents a significantly more scalable paradigm for WM-based RL: the simulator and reward function are prepared once and generalize broadly, leaving only the policy to adapt to new instructions. Consequently, the target-task rollout data requirement for the WM and reward drops from thousands of trajectories to **zero**. Even if optional few-shot WM fine-tuning is desired to bridge visual domain gaps, the total data budget remains vastly below that of training from scratch.

We build our action-conditioned WM upon the WAN 2.1 video generation model [34] and adopt OpenVLA-OFT [13, 16, 17, 46] as the policy. We optimize the VLA via GRPO [29], guided exclusively by binary, zero-shot success rewards from a frozen Qwen3-VL model [2]. While we build upon established architectures, integrating them within our task-agnostic paradigm effectively eliminates target-task rollout constraints, paving the way for highly efficient downstream adaptation. Finally, because querying a general-purpose WM on unseen tasks naturally increases the risk of hallucinated successes, we introduce a lightweight *dual-noise verification* mechanism that requires the VLM reward to remain consistent across rollouts generated under different initial diffusion noises. This effectively filters out hallucinations and curbs false-positive reward hacking.

We validate our paradigm in simulation and real-world settings where WMs are built strictly from broad, target-task-independent data. On the LIBERO benchmark [20], we train the WM exclusively on LIBERO-90 data and optimize the VLA across four held-out task suites (Spatial, Object, Goal, Long) entirely unseen during WM training. Our approach lifts the performance of a 1-shot SFT baseline by **+8.9%** and outperforms online RL that consumes $50\times$ more real-world data. On physical robots, we adapt policies to novel manipulation tasks using a WM pre-trained solely on diverse, uncurated play data. Starting from a 3-shot SFT baseline per task, RAW-Dream delivers a **+21.7%** absolute improvement in success rate. Our results validate that broad physical priors can effectively substitute for costly in-domain data, pointing to a clean and actionable scaling roadmap for WM-based VLA RL optimization.

2 Related work

Finetuning VLA with reinforcement learning Finetuning is a crucial step for adapting pre-trained Vision-Language-Action (VLA) models to unseen tasks. Inspired by advancements in reinforcement learning (RL) for large language models, there is growing interest in applying RL to finetune VLAs. We categorize existing approaches based on the source of their RL interaction data. One line of work (*e.g.*, VLA-RL [24], SimpleVLA-RL [17], π_{RL} [5]) uses simulated environments for data collection. However, these methods are often bottlenecked by the sim-to-real gap and the substantial engineering effort required to build high-fidelity digital twins. Alternatively, methods such as RECAP [11] and RLT [38] collect data directly from physical robots, effectively bypassing sim-to-real challenges but suffering from sample inefficiency. In contrast, our approach aligns with a recent paradigm that generates interaction data using a learned world model (which we review in detail in the next subsection). Driven by the strong predictive capabilities of modern world models to synthesize realistic future rollouts [27, 31, 33], performing RL finetuning within a world model enjoys high sample efficiency while significantly mitigating the sim-to-real gap.

Learning world models for VLA finetuning Closely related to our work is a recent line of research that finetunes VLAs with the help of learned world models. These works typically use a world model to generate synthetic rollouts and apply RL on these imagined trajectories to optimize the VLA policy without physical execution. Some frameworks [13, 30, 37, 44, 46] utilize world models as static simulators for post-training policies, while others [8, 21, 39] explore closed-loop systems where the policy and world model are iteratively co-trained. To provide reward signals for these imagined rollouts, existing literature generally relies on specifically engineered [18] or task-finetuned [8, 13, 46]) reward functions, though a few methods [10, 30, 43] explored zero-shot VLM rewards. However, a fundamental limitation shared across these approaches is their reliance on task-dependent world models. Unlike these approaches, our work takes a step towards a more generalizable framework by pairing a task-agnostic world model, capable of simulating unseen tasks out-of-the-box, with off-the-shelf VLM rewards, entirely eliminating the need for task-specific finetuning on both fronts. We believe this paradigm is inherently more scalable and can be readily upgraded as more powerful foundational world models and VLMs become available.

3 Method

Unlike prior pipelines that directly optimize for known target tasks, RAW-Dream operates under the strict constraint of target-task-agnostic VLA-RL optimization in WMs. This forces us to specially design the WM training / inference process and the RL optimization loop in the WM to handle unseen generalization. RAW-Dream focuses on two core components: **(i)** An action-conditioned video world model that learns transferable physical dynamics from the embodiment's diverse behaviors, enabling reliable rollouts even under severe target-domain data constraints. **(ii)** An imagination-based RL pipeline driven by a zero-shot VLM reward in GRPO, equipped with a dual-noise verification mechanism to filter the severe WM hallucinations caused by target-data scarcity.

3.1 World Modeling under Target-Domain Data Scarcity

Establishing physical priors. Under strict data constraints for target domains, the WM must learn generalizable physical dynamics from task-agnostic data. We train our WM on a broad mixture of diverse task-free behaviors (*e.g.*, general uncurated play data or noisy exploratory rollouts collected by executing VLA policies), covering varied success and failure patterns across a wide range of physical interactions and transitions, forcing the model to learn versatile physical dynamics rather than overfitting to the semantics of any single task or expert demonstration. More details can be found in our experiment sections.

Architecture and action-conditioning. We build our WM on the pre-trained Wan 2.1-T2V-1.3B Diffusion Transformer (DiT) video generation model backbone, operating within the VAE latent space. We finetune it into an action-conditioned generator via adaptive layer normalization (AdaLN) [26]. The action embeddings are projected into action-specific scale, shift, and gate parameters by a from-scratch MLP, which are then fused with the original diffusion timestep AdaLN parameters, and modulate the attention results in each DiT block. This mechanism provides control signals to condition the denoising process on the actions [1, 9, 36, 45]. To prevent generation from future-action

leakage, we strictly enforce causal masking for temporal attention operation in the DiT blocks. The network v_θ is supervised using a unified rectified flow matching objective [22] to predict the continuous velocity field $\mathbf{v}^* = \boldsymbol{\epsilon} - \mathbf{z}^0$:

$$\mathcal{L} = \mathbb{E}_{\tau,\, \mathbf{z},\, \boldsymbol{\epsilon},\, \mathbf{a}} \left[\left\| v_\theta(\mathbf{z}^\tau, \tau, \mathbf{a}, \mathbf{z}_{ctx}) - \mathbf{v}^* \right\|^2 \right], \tag{1}$$

where τ is the diffusion timestep, $\mathbf{z}^\tau = \tau\boldsymbol{\epsilon} + (1-\tau)\mathbf{z}^0$ is the noisy latent state, $\mathbf{z}^0$ is the ground-truth latent state, $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ is the noise, $\mathbf{a}$ denotes the conditioned actions, and $\mathbf{z}_{ctx}$ represents the context latent states.

Long-horizon autoregressive rollout. To make long-horizon prediction, the WM generates video autoregressively, conditioned on recently generated states as context. To accommodate the policy's action chunk size (8 in OpenVLA-OFT), the WM predicts 2 states per step in inference. We employ a large context window of 6 for reliable generation. During training, context frames are corrupted with diffusion noise (timesteps sampled from $[0, 300]$), compelling the model to make predictions from imperfect contexts. Also, we apply conditioning masks to construct context and generated states of variable lengths. In inference, we set a fixed noise timestep $\tau = 50$ for the context frames.

First-frame anchoring and progressive noise. When rolling out in entirely unseen target domains, the initial observation $\mathbf{o}_0$ is the only ground-truth visual reference. We condition every autoregressive step on this first frame to ensure scene consistency, using 7 context and predicting 2 latent states per step. However, this also induces *first-frame ghosting* in zero-shot WM transfer: lacking domain familiarity, the model over-anchors to the reliable $\mathbf{o}_0$, stubbornly pasting initial object layouts into later autoregressive chunks, creating "ghosts" of the first frame that corrupt both dynamics and VLM reward judgments in RL. To resolve this, we introduce *progressive anchor noise* at inference. We gradually increase diffusion timesteps on the anchor as the autoregressive rollout progresses. This smoothly degrades the anchor's influence over time, forcing the model to rely on recent context while preserving $\mathbf{o}_0$'s role in maintaining scene consistency. We provide further WM implementation details in Appendix A.1 and qualitative results of first-frame ghosting in Appendix C.1.

3.2 Policy Optimization in Task-Agnostic World Model

Starting from an initial visual observation and a task instruction, the OpenVLA-OFT policy samples actions to interact with the learned WM autoregressively, generating a group of imagined trajectories $\{\tau_i\}_{i=1}^G$. Since our paradigm strictly precludes collecting target-task rollouts to train success classifiers, we evaluate these imagined videos using an off-the-shelf VLM (Qwen3-VL), yielding a binary outcome R_i for each trajectory τ_i. This group is then directly used to update the policy via GRPO.

While this provides a stable optimization pipeline, it inevitably incurs hallucinations on unseen tasks. The policy may discover visually plausible but dynamically unfaithful rollouts that fool the VLM to give false positive rewards, a risk amplified by our task-agnostic WM's lack of downstream training data. To address this, we draw inspiration from model-based offline RL that penalizes the rewards of unreliable transitions based on prediction uncertainty [14, 42], and present a tractable analog tailored for modern DiT-based WMs and RL based on binary outcome rewards. Because our WM employs long-horizon autoregressive generation via diffusion denoising processes, the randomness and uncertainty naturally compounds over steps, especially when generalizing to unseen tasks and scenes. Therefore, rollouts regenerated under different initial diffusion noise are more likely to yield divergent outcomes under out-of-distribution dynamics compared to reliable transitions.

Building on this, we introduce **Dual-Noise Verification** (DNV) to detect and penalize hallucinations. For any imagined trajectory τ_i within the GRPO group that was marked successful by the VLM, we replay its action sequence $\{a_{i,t}\}_{t=1}^{T_i}$ in the WM using independently resampled initial diffusion noise at each autoregressive step. If the VLM evaluates this newly generated video as a failure, the initial success is flagged as unfaithful and the rollout is discarded from the group. Crucially, we detect hallucinations directly through the disagreement of the VLM reward rather than video generation discrepancies. Low-level video metrics are often unaligned with task semantics; by contrast, utilizing the VLM's judgment provides a semantic validation that directly grounds the final reward penalty.

In our GRPO objective based on DNV, we compute advantages using group-relative returns over the reliable subset $\mathcal{R}$ of size $G_{\mathcal{R}}$. More details and analysis on DNV can be found in our Appendix.

$$\mathcal{J}(\theta) = \mathbb{E}_{s_0 \sim \mathcal{D}, \{\tau_i\} \sim \pi_{\theta_{\text{old}}}} \left[\frac{1}{G_{\mathcal{R}}} \sum_{i \in \mathcal{R}} \frac{1}{T_i} \sum_{t=1}^{T_i} \min \left(r_{i,t}(\theta) \hat{A}_i,\ \text{clip}(r_{i,t}(\theta), 1 - \epsilon_{\text{low}}, 1 + \epsilon_{\text{high}}) \hat{A}_i \right) \right], \tag{2}$$

with

$$r_{i,t}(\theta) = \frac{\pi_\theta(a_{i,t} \mid s_{i,t})}{\pi_{\theta_{\text{old}}}(a_{i,t} \mid s_{i,t})}, \qquad \hat{A}_i = \frac{R_i - \text{mean}(\{R_j\}_{j \in \mathcal{R}})}{\text{std}(\{R_j\}_{j \in \mathcal{R}})}. \tag{3}$$

Here $r_{i,t}(\theta)$ is the probability ratio between new and old policies at step t of trajectory τ_i, $R_i = R(\tau_i)$, and $\hat{A}_i$ is the normalized group-relative advantage over the reliable subset.

4 Experiments

We conduct extensive experiments across both the LIBERO simulation benchmark [20] and our real-world robotic setups to systematically validate the effectiveness of RAW-Dream. Our experiments are designed to answer the following four questions:

- **Q1:** Can an action-conditioned video world model trained on broad, downstream-task-agnostic trajectory data effectively predict and simulate unseen target tasks? (Section 4.2)
- **Q2:** Can RL within the task-agnostic world model using task-agnostic VLM reward effectively improve VLA performance under minimal task-specific data constraints? (Section 4.3)
- **Q3:** How do the zero-shot VLM reward and the dual-noise verification mechanism contribute to the success of imagination-based RL? (Section 4.4)
- **Q4:** Does this paradigm successfully transfer to physical robots to facilitate real-world manipulation tasks? (Section 4.5)

4.1 Experimental settings for simulation

We instantiate our paradigm on the LIBERO benchmark using a strict evaluation protocol to mirror our proposed downstream-agnostic paradigm: constructing a foundational WM from LIBERO-90, and subsequently evaluating its simulation fidelity and RL performance on four entirely unseen downstream task suites under minimal target-task data constraints.

A general-purpose WM from LIBERO-90. The **LIBERO-90** benchmark (90 tasks, containing $\sim$4,500 expert demonstrations) offers a rich diversity of scenes and manipulation tasks. We leverage this benchmark as a foundational source to construct a general-purpose WM for the LIBERO embodiment. Specifically, we first train an OpenVLA-OFT policy via multi-task SFT on the curated LIBERO-90 expert demonstrations. To build a comprehensive physical prior, we collect broad exploration data by executing this SFT policy with injected Gaussian noise on its action outputs across all 90 tasks. This yields a large, diverse corpus of rollouts spanning successes and failures across a broad range of tasks with diverses, yet targeting no specific downstream task, which can be seen as the *proxy play data* for LIBERO. The WM is then trained entirely on these rollouts, forcing the model to internalize versatile, broadly applicable physical dynamics.

Unseen downstream evaluation. We then employ four highly distinct held-out downstream task suites for evaluation: **Spatial** (novel item arrangements), **Object** (novel objects), **Goal** (novel semantic goals), and **Long** (long-horizon sequences). Each suite comprises 10 distinct tasks. Crucially, the WM has never observed any data from these four suites during pretraining on LIBERO-90. When confronted with a novel target suite, we follow a two-phase VLA policy learning pipeline. First, via semantic anchoring (multi-task **1-shot SFT**), the VLA policy previously trained on LIBERO-90 is further adapted using exactly *one* expert demonstration per task (10 demonstrations per suite), anchoring novel task semantics with minimal demonstration data. Second, via multi-task **RL in WM imagination**, the anchored policy is improved through trial-and-error entirely within the WM via GRPO, using the zero-shot binary success reward provided by the frozen Qwen3-VL [2].

World model adaptation conditions. Since the RL phase is conducted entirely within the WM, the simulator's fidelity directly governs the policy's improvement ceiling. To systematically probe this

Table 1: **Action-conditioned video prediction quality on unseen task suites using different WMs.** "Tgt. data" denotes the number of target-task-suite trajectories used for WM training. Average trajectory length of validation datasets in parentheses lies below each suite name.

Suite	WM Variant	Tgt. data	PSNR↑	SSIM↑	LPIPS↓	FVD↓
Spatial (Avg Length: 164)	Zero-Shot WM	0	19.17	80.24	12.51	92.29
	Co-Train WM	10	21.34	84.58	8.26	60.76
	ID-FT WM	500	**24.99**	**89.58**	**4.20**	**23.52**
	WoVR WM	2500	22.80	86.95	6.61	45.39
Object (Avg Length: 208)	Zero-Shot WM	0	19.36	76.59	13.85	233.29
	Co-Train WM	10	19.94	82.23	9.77	114.06
	ID-FT WM	500	**25.91**	**90.28**	**3.53**	**26.82**
	WoVR WM	2500	22.73	86.59	6.15	92.11
Goal (Avg Length: 212)	Zero-Shot WM	0	19.50	83.11	11.19	89.83
	Co-Train WM	10	21.94	87.21	7.17	50.33
	ID-FT WM	500	**25.33**	**91.15**	**3.79**	**21.65**
	WoVR WM	2500	22.92	88.67	6.40	41.78
Long (Avg Length: 472)	Zero-Shot WM	0	19.14	82.87	12.96	56.43
	Co-Train WM	10	19.60	84.00	11.15	44.48
	ID-FT WM	500	**20.32**	**84.85**	**8.96**	**38.84**
	WoVR WM	2500	18.03	82.04	12.97	80.52

dependency, we evaluate the WM fidelity and the above WM-based RL pipeline under four conditions that span a wide spectrum of target-domain data exposure:

- **Zero-Shot WM:** The foundational WM trained on LIBERO-90 is fully frozen, observing exactly zero target-suite rollouts.
- **Co-Train WM:** Since the zero-shot WM may lack visual familiarity with novel target scenes, we anchor it to the target domain at minimal cost by mixing the 10 expert demonstrations from 1-shot SFT into the LIBERO-90 WM training data and jointly fine-tuning, avoiding overfitting on scarce expert-only data while incurring zero additional collection budget.
- **ID-FT WM (In-Domain FineTuning):** The 1-shot SFT VLA policy is evaluated on the target suite, yielding 500 rollouts (50 per task) that are used to fine-tune the foundational LIBERO-90 WM. This serves as an upper bound for our architecture.
- **WoVR WM [13]:** A recent state-of-the-art method that trains a 5B, WAN2.2-based WM from scratch on 2,500 per-target-task-suite rollouts without any broad pre-training, providing a direct comparison against data-intensive, task-specific WM construction.

Evaluation. For WM fidelity, we evaluate the action-conditioned video generation quality of each WM on a fixed validation dataset for each task suite (each containing 500 trajectories unseen in training), using standard metrics: PSNR, SSIM, LPIPS and FVD. LPIPS and SSIM scores are scaled ×100 for compact display. For VLA policy evaluation in SFT and RL, each method is evaluated by executing the learned policy in the real LIBERO simulator for 50 episodes per task, reporting per-suite average success rate (%). Further evaluation details can be found in our Appendix A.

4.2 World model simulation quality

Table 1 compares four WM variants across all four unseen target suites. Under the strict zero-shot setting, apart from LIBERO-Object which introduces entirely novel objects that severely shock the input distribution (FVD=233), the WM maintains a remarkably capable baseline across other suites. This demonstrates that physical priors learned from broad, downstream-agnostic data can effectively generalize to unseen domains. Incorporating the exact same 10 demonstrations used for SFT semantic anchoring into the WM (Co-Train), which demands zero additional data collection budget, yields a stark enhancement in fidelity, rapidly approximating the performance of WoVR (particularly on Spatial and Goal), which relies on massive in-domain data.

Furthermore, on LIBERO-Long, the closest suite to LIBERO-90, the Zero-Shot WM already eclipses WoVR despite WoVR having access to massive in-domain data. This highlights that when underlying

Table 2: **Success rate (%) on four unseen LIBERO suites.** The VLA is first trained on LIBERO-90, then adapted to each suite using 1-shot SFT. Bold rows correspond to our method with different WM conditions. Bold rows correspond to our method with different WM conditions. "Tgt. data" refers to target-suite real rollout budget used in the whole policy learning process, including data for WM and VLA fine-tuning. Deltas in parentheses relative to 1-shot SFT.

Category	Method	Tgt. data	Spatial	Object	Goal	Long	Avg.
No RL	Zero-Shot from 90	0	3.4	0.0	4.2	7.0	3.7
	1-shot SFT	10	54.6	46.4	52.2	20.2	43.4
RL in simulator (GT reward)	Online RL (Short)	522	58.4 (+3.8)	60.2 (+13.8)	55.2 (+3.0)	17.6 (-2.6)	47.9 (+4.5)
	Online RL (Long)	2570	68.8 (+14.2)	78.8 (+32.4)	65.2 (+13.0)	22.4 (+2.2)	58.8 (+15.4)
RL in WM (VLM reward)	**Zero-Shot WM**	**10**	**65.8 (+11.2)**	**47.2 (+0.8)**	**60.2 (+8.0)**	**35.8 (+15.6)**	**52.3 (+8.9)**
	Co-Train WM	**10**	**73.2 (+18.6)**	**60.2 (+13.8)**	**58.4 (+6.2)**	**36.6 (+16.4)**	**57.1 (+13.7)**
	ID-FT WM	**510**	**82.0 (+27.4)**	**79.8 (+33.4)**	**63.4 (+11.2)**	**38.6 (+18.4)**	**66.0 (+22.6)**
	WoVR WM	2510	81.6 (+27.0)	71.6 (+25.2)	66.4 (+14.2)	24.0 (+3.8)	60.9 (+17.5)

scenes overlap with the pre-training distribution, zero-shot transfer is highly effective, and validates the superiority of our WM architecture and inference machanism in long-horizon simulation.

The most critical takeaway arises when comparing our ID-FT upper-bound against WoVR. By injecting just 500 target rollouts into our pre-trained foundation, the WM outperforms WoVR (trained from scratch on 2,500 target rollouts) across all suites and metrics, confirming that broad, downstream-independent physical priors outweigh brute-force data stacking within a single target domain.

4.3 VLA Policy improvement via RL in WMs

Results in Table 2 translate the WM fidelity analysis into policy performance. We compare our WM-based RL methods against two categories of baselines: **No-RL baselines**: the LIBERO-90 zero-shot policy and the 1-shot SFT policy. **Online RL in the real simulator**: using ground-truth success rewards, we run GRPO with two real-rollout budgets—Online RL (Short) uses $\sim$500 target-suite episodes (matching the ID-FT WM data budget), and Online RL (Long) uses $\sim$2,500 episodes (matching WoVR's data budget). For WM-based RL, we evaluate under the same four WM conditions as in Table 1, all using the zero-shot Qwen3-VL reward for fair comparison. Online RL and WM-based RL are both initialized from the 1-shot SFT policy. More details are in Appendix A.

Across all suites, every WM-based RL variant yields a clear improvement over the 1-shot SFT baseline, confirming that task-agnostic world modeling combined with a VLM reward is sufficient to drive meaningful RL gains on unseen tasks. Using a zero-shot WM and a VLM reward, our method already outperforms Online RL (Short), which requires $\sim$500 real-rollout with ground-truth rewards (52.3% v.s 47.9%). Co-Train WM RL approaches the performance of Online RL (Long) that consumes $\sim$2,500 real episodes, while incurring zero additional data collection beyond the 10 demonstrations already used for SFT (57.1% v.s 58.8%). With only 500 in-domain trajectories for WM fine-tuning, RL in ID-FT WM surpasses Online RL (Long) using $5\times$ real rollouts and gt rewards (66.0% v.s 58.8%). These results validate that imagination-only RL can be a competitive substitute for costly online interaction even under strict target-suite data constraints. Remarkably, ID-FT WM RL also outperforms RL in WoVR's WM, which is learned from scratch with massive target-suite data. This highlights extreme data efficiency, proving that the solid foundation established by the pre-trained WM needs only minimal in-domain data to significantly benefit RL.

Furthermore, the improvement ordering of our WM variants (Zero-shot $\rightarrow$ Co-Train $\rightarrow$ ID-FT) consistently predicts the ordering of downstream RL gains, proving that WM fidelity is a ceiling for imagination-based policy optimization. The Object suite reflects this most sharply: the Zero-Shot WM has the worst simulation quality (FVD=233) and correspondingly shows almost no RL gain (+0.8%), but as Co-Train and ID-FT improve prediction fidelity, the policy success rate increases dramatically (+13.8% and +33.4%). On Long suite, without any target rollouts for the WM or reward, RL in Zero-Shot WM even significantly outperforms long-horizon Online RL (+15.6% v.s +2.2%), showcasing the power of the pretrained WM's transferable physical dynamics on hard, long-horizon

Table 3: **Component ablation.** We ablate the reward model and DNV under two WM conditions. "1-shot RM" is a VideoMAE-based classifier fine-tuned on the 10 1-shot SFT demonstrations per suite. Robometer serves as an oracle reference, trained on LIBERO data that includes the target suites. DNV status is indicated per row. Bold rows correspond to our full method in Table 2.

WM	Configuration	Spatial	Object	Goal	Long	Avg.
Zero-Shot	**Qwen3-VL, w/ DNV**	65.8 (+11.2)	47.2 (+0.8)	60.2 (+8.0)	**35.8 (+15.6)**	**52.3 (+8.9)**
	Qwen3-VL, w/o DNV	64.0 (+9.4)	46.0 (-0.4)	56.8 (+4.6)	30.8 (+10.6)	49.4 (+6.0)
	1-shot RM, w/o DNV	50.6 (-4.0)	23.2 (-23.2)	37.2 (-15.0)	15.6 (-4.6)	31.7 (-11.7)
	Robometer, w/o DNV	**68.4 (+13.8)**	**50.0 (+3.6)**	**61.8 (+9.6)**	22.0 (+1.8)	50.6 (+7.2)
Co-Train	**Qwen3-VL, w/ DNV**	**73.2 (+18.6)**	60.2 (+13.8)	**58.4 (+6.2)**	**36.6 (+16.4)**	**57.1 (+13.7)**
	Qwen3-VL, w/o DNV	72.4 (+17.8)	56.4 (+10.0)	58.2 (+6.0)	30.8 (+10.6)	54.5 (+11.1)
	1-shot RM, w/o DNV	49.4 (-5.2)	45.8 (-0.6)	44.0 (-8.2)	20.8 (+0.6)	40.0 (-3.4)
	Robometer, w/o DNV	73.0 (+18.4)	**64.8 (+18.4)**	53.2 (+1.0)	21.8 (+1.6)	53.2 (+9.8)

tasks. Notably, on Goal suite, the zero-shot VLM reward quality degrades (as shown in Appendix B) and directly limits the policy improvement, but still improves upon the 1-shot SFT baseline.

4.4 Ablation studies

We ablate the reward model usage and the DNV mechanism under Zero-Shot and Co-Train WM. A central question of our task-agnostic paradigm is that, when only minimal target-task data (one expert demo per task) is available, whether a zero-shot VLM reward can replace the finetuned reward model as a success classifier. Since DNV itself relies on the reward signal for its second-pass judgment, we first compare reward models *without* DNV to isolate the reward signal's intrinsic quality. We evaluate three reward variants: **(i)** *Qwen3-VL* (zero-shot): our default, requiring no target-task data. **(ii)** *1-shot RM*: Following [46], we finetune a binary classifier from VideoMAE [32] for each suite, using the 10 expert demonstrations in 1-shot SFT. We construct a small supervised dataset by sampling terminal segments of each demonstration as positive examples and early or cross-task segments as negatives, learning a multi-task reward model per suite from minimal data. **(iii)** *Robometer* [19] (oracle): a Qwen3-VL-4B-Instruct-based reward model finetuned on RBM-1M robotic dataset including LIBERO data, serving as an oracle upper bound with access to rich in-domain knowledge.

The results are in Table 3. The zero-shot Qwen3-VL reward significantly outperforms the 1-shot finetuned RM. With only 10 positive examples, the learned classifier overfits to expert demonstration visuals and catastrophically misjudges the diverse, often imperfect outputs of the WM, actively degrading the policy below the SFT baseline on every suite. By contrast, Qwen3-VL's broad visual commonsense generalizes robustly, performing comparably to the oracle Robometer. When in-domain data is scarce, foundation VLM commonsense is far more reliable than a few-shot learned classifier.

We then ablate the effect of DNV with the reward fixed to Qwen3-VL. DNV yields consistent gains, particularly on Libero-Long. Since stochasticity and prediction uncertainty compound over autoregressive generation, long-horizon rollouts are most prone to hallucinations, which are effectively filtered by DNV, stabilizing RL. Notably, combining Qwen3-VL with DNV approximates or outperforms the oracle Robometer reward. However, DNV offers limited help when the upstream signal is itself unreliable. On Object under zero-shot WM, the simulation quality is too poor for any rollout to be faithful; on Goal the VLM reward quality degrades (Appendix B), resulting in modest performance gain under Co-Train WM. More results of DNV can be found in Appendix C.3.

4.5 Real-World Experiments

We deploy our pipeline and RAW-Dream on an AgileX Piper robotic arm with 7-DoF continuous control (6-DoF end-effector poses + 1-DoF continuous gripper within $[0, 0.1]$). Our evaluation strictly adheres to the zero-target-data paradigm: the WM is never exposed to any downstream task data, and the VLA policy is improved from extremely-few-shot SFT base models entirely through imagination.

Task-agnostic WM from play data. We first pre-train our WM on Open X-Embodiment datasets [7], which is then fine-tuned on approximately 4 hours of uncurated, teleoperated play data, collected via a master-slave control system. The tabletop workspace contains a rich variety of everyday

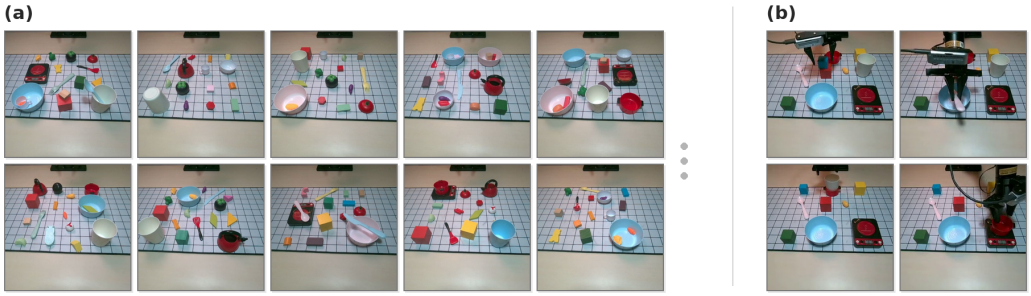


Figure 2: (a) Sample scenes from our collected play data spanning diverse object arrangements and tabletop layouts. (b) The four downstream evaluation tasks for VLA fine-tuning and RL.

objects arranged across different scene layouts with significantly randomized object positions and compositions, as is illustrated in Figure 2 (a). During collection, the human operator freely performs whatever manipulation comes to mind —grasping, placing, inserting, pushing, tumbling—without following any task definition or success criterion. The data provides a generalizable physical prior, capturing a broad distribution of arm-object dynamics, contact patterns, and recovery behaviors, while remaining agnostic to the downstream evaluation tasks. We provide qualitative WM rollout results on unseen downstream scene layouts during WM training in Appendix C.2.

Evaluation protocols and Results. We evaluate on four contact-rich manipulation tasks in entirely unseen scene layouts: *(1)* **Stack Block** (stack the blue block onto the red block), *(2)* **Place Pot** (place the pot on the stove), *(3)* **Put Spoon** (pick up the spoon and put it into the bowl), and *(4)* **Place Cup** (place the cup into the pot), illustrated in Figure 2 (b). While similar manipulation semantics may appear in the play data, the specific scene layouts are entirely novel. The VLA policy is first anchored to each task via 3-shot SFT on teleoperated demonstrations, then improved via RL entirely within the play-data WM using zero-shot Qwen3-VL rewards, where the zero-shot WM is not finetined any more. Each method is evaluated over 30 real-world trials per task.

Results. As shown in Table 4, starting from a 3-shot SFT baseline at only 50.0% average success rate, RAW-Dream yields a striking **+21.7%** absolute gain (50.0% $\rightarrow$ 71.7%), with improvements on every task. The largest gains appear on Put Spoon (+30.0%) and Place Cup (+30.0%), where compounding execution errors most severely limit the few-shot SFT policy. Notably, all gains are achieved using only 3 teleoperated demonstrations per task for SFT and a fully frozen, play-data-only WM with zero-shot VLM rewards. No rollouts from the downstream scene are collected for either the WM or the reward. These results confirm that broad physical priors captured by a downstream-agnostic WM are sufficient to drive reliable real-world policy improvement entirely through imagination. We provide video demonstrations of our real-world experiments on our anonymous website.

Table 4: Real-world success rates (%, 30 trials per task) on the AgileX Piper robot arm.

Method	Stack block	Place pot	Put spoon	Place cup	Average
3-shot SFT	40.0 (12/30)	73.3 (22/30)	36.7 (11/30)	50.0 (15/30)	50.0
RL from 3-shot SFT	**50.0** (15/30)	**90.0** (27/30)	**66.7** (20/30)	**80.0** (24/30)	**71.7**

5 Conclusion

We propose RAW-Dream, a paradigm enabling WM-based RL post-training of VLAs on unseen tasks. By combining a task-agnostic WM with a zero-shot VLM reward, RAW-Dram eliminates the need for massive in-domain data and target-task exposure. A dual-noise verification mechanism further stabilizes RL by filtering WM hallucinations. Experiments demonstrate that our method consistently outperforms SFT baselines and data-heavy paradigms like online RL or RL in WMs learned from scratch, unlocking a highly data-efficient and scalable pathway for WM-based VLA RL post-training. Promising future directions include improving zero-shot WM fidelity by scaling model capacity and data diversity, enhancing zero-shot VLM reward quality through lightweight calibration, and extending our paradigm to flow-based VLA architectures.

References

[1] Ali, A. et al. World simulation with video foundation models for physical ai. *arXiv preprint arXiv:2511.00062*, 2025.

[2] Bai, S. et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

[3] Black, K. et al. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

[4] Chandra, A.L. et al. Diwa: Diffusion policy adaptation with world models. *arXiv preprint arXiv:2508.03645*, 2025.

[5] Chen, K. et al. π_{RL}: Online rl fine-tuning for flow-based vision-language-action models. *arXiv preprint arXiv: 2510.25889*, 2025.

[6] Chen, X. et al. Villa-x: enhancing latent action modeling in vision-language-action models. *arXiv preprint arXiv:2507.23682*, 2025.

[7] Collaboration, O.X.E. et al. Open X-Embodiment: Robotic learning datasets and RT-X models. `https://arxiv.org/abs/2310.08864`, 2023.

[8] Guo, Y. et al. Vlaw: Iterative co-improvement of vision-language-action policy and world model. *arXiv preprint arXiv:2602.12063*, 2026.

[9] He, H. et al. Pre-trained video generative models as world simulators. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 4645–4653, 2026.

[10] Hung, C.Y. et al. Nora-1.5: A vision-language-action model trained using world model-and action-based preference rewards. *arXiv preprint arXiv:2511.14659*, 2025.

[11] Intelligence, P. et al. $\pi^*_{0.6}$: a vla that learns from experience. *arXiv preprint arXiv: 2511.14759*, 2025.

[12] Intelligence, P. et al. $\pi_{0.5}$: a vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025.

[13] Jiang, Z. et al. Wovr: World models as reliable simulators for post-training vla policies with rl. *arXiv preprint arXiv:2602.13977*, 2026.

[14] Kidambi, R. et al. Morel: Model-based offline reinforcement learning. *Advances in neural information processing systems*, 33:21810–21823, 2020.

[15] Kim, M.J. et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

[16] Kim, M.J., Finn, C. and Liang, P. Fine-tuning vision-language-action models: Optimizing speed and success. *arXiv preprint arXiv:2502.19645*, 2025.

[17] Li, H. et al. Simplevla-rl: Scaling vla training via reinforcement learning. *arXiv preprint arXiv:2509.09674*, 2025.

[18] Li, H. et al. Vla-rft: Vision-language-action reinforcement fine-tuning with verified rewards in world simulators. *arXiv preprint arXiv:2510.00406*, 2025.

[19] Liang, A. et al. Robometer: Scaling general-purpose robotic reward models via trajectory comparisons. *arXiv preprint arXiv:2603.02115*, 2026.

[20] Liu, B. et al. Libero: Benchmarking knowledge transfer for lifelong robot learning. *Advances in Neural Information Processing Systems*, 36:44776–44791, 2023.

[21] Liu, X. et al. World-vla-loop: Closed-loop learning of video world model and vla policy. *arXiv preprint arXiv:2602.06508*, 2026.

[22] Liu, X., Gong, C. and Liu, Q. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.

[23] Lu, C. et al. Challenges and opportunities in offline reinforcement learning from visual observations. *arXiv preprint arXiv:2206.04779*, 2022.

[24] Lu, G. et al. Vla-rl: Towards masterful and general robotic manipulation with scalable reinforcement learning. *arXiv preprint arXiv:2505.18719*, 2025.

[25] Mazzaglia, P. et al. Genrl: Multimodal-foundation world models for generalization in embodied agents. *Advances in neural information processing systems*, 37:27529–27555, 2024.

[26] Peebles, W. and Xie, S. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4195–4205, 2023.

[27] Quevedo, J. et al. Worldgym: World model as an environment for policy evaluation. *arXiv preprint arXiv: 2506.00613*, 2025.

[28] Sekar, R. et al. Planning to explore via self-supervised world models. In *International conference on machine learning*, pages 8583–8592. PMLR, 2020.

[29] Shao, Z. et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

[30] Sharma, A.K. et al. World-gymnast: Training robots with reinforcement learning in a world model. *arXiv preprint arXiv:2602.02454*, 2026.

[31] Team, G.R. et al. Evaluating gemini robotics policies in a veo world simulator. *arXiv preprint arXiv: 2512.10675*, 2025.

[32] Tong, Z. et al. Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. *Advances in neural information processing systems*, 35:10078–10093, 2022.

[33] Tseng, W.C. et al. Scalable policy evaluation with video world models. *arXiv preprint arXiv: 2511.11520*, 2025.

[34] Wan, T. et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

[35] Wang, Y. et al. Founder: Grounding foundation models in world models for open-ended embodied decision making. *arXiv preprint arXiv:2507.12496*, 2025.

[36] Wang, Y. et al. Co-evolving latent action world models. *arXiv preprint arXiv:2510.26433*, 2025.

[37] Xiao, J. et al. World-env: Leveraging world model as a virtual environment for vla post-training. *arXiv preprint arXiv:2509.24948*, 2025.

[38] Xu, C. et al. Rl token: Bootstrapping online rl with vision-language-action models. *arXiv preprint arXiv:2604.23073*, 2026.

[39] Yang, J. et al. Rise: Self-improving robot policy with compositional world model. *arXiv preprint arXiv:2602.11075*, 2026.

[40] Yin, T. et al. Playworld: Learning robot world models from autonomous play. *arXiv preprint arXiv:2603.09030*, 2026.

[41] Yu, C. et al. Rlinf: Flexible and efficient large-scale reinforcement learning via macro-to-micro flow transformation. *arXiv preprint arXiv:2509.15965*, 2025.

[42] Yu, T. et al. Mopo: Model-based offline policy optimization. *Advances in neural information processing systems*, 33:14129–14142, 2020.

[43] Zhang, J. et al. Reinforcing action policies by prophesying. *arXiv preprint arXiv:2511.20633*, 2025.

[44] Zhang, Z. et al. Towards practical world model-based reinforcement learning for vision-language-action models. *arXiv preprint arXiv:2603.20607*, 2026.

[45] Zhu, F. et al. Irasim: A fine-grained world model for robot manipulation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9834–9844, 2025.

[46] Zhu, F. et al. Wmpo: World model-based policy optimization for vision-language-action models. *arXiv preprint arXiv:2511.09515*, 2025.

A Implementation Details

A.1 Action-Conditioned World Model

Architecture. We build on the WAN 2.1 T2V-1.3B DiT backbone with a paired VAE (latent dim $C{=}16$, stride $(4, 8, 8)$), yielding a $32{\times}32$ spatial latent grid from $256{\times}256$ pixel inputs and $4\times$ temporal compression (33 pixel frames $\rightarrow \sim$8 latent frames per training window). For action conditioning, each latent frame maps to a chunk of 4 raw action frames (matching the VAE temporal stride). Each chunk is projected by a from-scratch MLP into action tokens. These enter every DiT block via AdaLN described in Section 3.1. Temporal self-attention is block-wise causal: spatial tokens within the same timestep attend bidirectionally, but across timesteps attention is strictly causal.

Training. We train on 33-frame windows. Each window is partitioned into context frames and generation frames via a binary mask. The context–generation split is randomized per sample. Context frames are corrupted with per-position noise sampled from $\mathcal{U}[0, 300]$ out of 1000 total diffusion timesteps, training the model to predict from imperfect contexts. We use AdamW with learning rate $7.5e{-}5$, linear warmup followed by cosine decay, batch size 12, and train for $\sim$100K steps. The rectified flow objective supervises velocity prediction with MSE loss computed only over valid generation positions. Classifier-free guidance (CFG) dropout is disabled both in training and inference, which we found helpful for performance. Actions inputs are normalized on the WM training data via q01-q99 statistics.

Autoregressive inference. Each AR step generates $G{=}2$ latent states (8 pixel frames, matching the OpenVLA-OFT action chunk size) and conditions. CFG is disabled on a window of 7 context frames: 1 first-frame anchor + 6 recently generated latent states. The AR iterations are performed entirely in latent space without intermediate VAE decode–encode round-trips. Each denoising step uses the UniPC multi-step ODE solver with 5 function evaluations. CFG is disabled. Non-anchor context frames are injected at a fixed noise level $t_c{=}50$. For the first-frame anchor, we apply *progressive noise* starting from AR step $s_0{=}3$: the anchor noise level increases linearly as $t_1^{(s)} = \min(10 \cdot \max(s-3, 0),\ 300)$. The increasing noise gradually relaxes the anchor's influence. This adjustment is applied *only at inference* and does not alter the training.

A.2 Dual-Noise Verification (DNV) in GRPO

To integrate DNV into GRPO, we handle unreliable rollouts as follows. For each GRPO group of $G{=}8$ trajectories, any rollout whose VLM reward disagrees between the original and replayed generation has its reward replaced by the mean reward of the remaining trustworthy rollouts. This forces the flagged trajectory's advantage to $\hat{A}_i = 0$, rendering it invisible to the policy gradient without changing the fixed group tensor size in RL optimization. While this may shrink the standard deviation when computing advantages in the group of only reliable trajectories, the effect is merely a constant scaling factor on the gradients and does not harm optimization. If more than two rollouts within a single group are flagged, the entire group is discarded to preserve optimization stability.

A.3 VLA Policy and SFT Training

We adopt OpenVLA-OFT [16] as the VLA backbone, which outputs an action chunk of 8 steps per inference call. We do not use proprioceptive input or wrist camera images. Unlike the original OpenVLA-OFT which uses deterministic L1 regression heads, we retain the probabilistic (categorical) action tokenization, as it naturally provides the stochastic policy required for GRPO sampling during RL. For SFT on LIBERO-90, we use the full curated expert demonstrations and train for 150K steps. For 1-shot SFT on each target suite, we fine-tune the LIBERO-90 SFT checkpoint on the 10 expert demonstrations (1 per task) for 50K steps.

A.4 RL Training

We use GRPO with group size $G{=}8$. No KL reference penalty is applied ($\lambda_{\text{KL}}{=}0$). Groups with all failure or all success rollouts are discarded during training. The VLA temperature when sampling actions during rollout is 1.6. The learning rate is $5e{-}6$. All experiments are conducted using eight NVIDIA H200 GPUs. Training RL in our WM specifically takes about 2 days.

WM-based RL. Training steps per suite are: 20 for Object (early stopping since Object tends to produce all-success groups when we extend the training steps, which will be filtered, and GRPO will wait for a long time for a group that exhibits difference in rewards), and 50 for other three suites.

Online RL. Both Online RL (Short) and Online RL (Long) use GRPO with the same OpenVLA-OFT architecture, initialized from the 1-shot SFT checkpoint. Rollouts are collected in the real LIBERO simulator with ground-truth binary success rewards. Online RL (Short) uses $\sim$500 episodes (512 episode, 1 online RL step) per suite comparable to the budget of ID-FT WM fine-tuning, and this setting is the same as WMPO; Online RL (Long) uses $\sim$2,500 episodes (2560 episodes, 10 online RL steps) comparable to the budget of WoVR WM training.

WoVR. We compare against WoVR [13], which trains a WAN 2.2-TI2V-5B-based WM trained from scratch on 2,500 target-suite rollouts per suite without any broad pre-training, to validate the effect of the learned physical prior. We directly use their inference implementation in RL-Inf [41] and their released WM checkpoints on the four LIBERO suites. To fairly evaluate the RL performance on different WM variants, we also use the VLM-based reward for RL training in WoVR WM. The VLM usage, inference parameters and the suite training steps are all the same with ours.

A.5 VLM Reward Deployment

We deploy Qwen3-VL via vLLM as the zero-shot reward model. Model selection per suite: Qwen3-VL-8B-Instruct for Spatial and Object; Qwen3-VL-32B-Instruct for Goal, Long and real-world tasks. Inference configuration: temperature 0.7, n=5 samples per request (majority vote), vote threshold $\geq 4/5$ for a positive reward. Input videos are sampled at 4 FPS from the WM-generated rollout.

The prompt follows a simple task-agnostic template:

```
 You are a highly rigorous robotics operations evaluation
expert.  Your task is to observe the provided video of a robot
and objectively determine if it successfully completed the
"${task_description}" task.  Please strictly output your
evaluation in the following JSON format, without generating
any extra text or markdown formatting:  { "Reasoning":
"Briefly analyze whether the robot successfully completed
the task.", "Final Answer":  [Output ONLY one word:  Success
or Failure.]  }
```

where the ${task_description} is the original description of the Libero tasks in simulation, and for real-world experiments we also directly use the task description mentioned in Section 4.5. We found that this minimal prompt outperforms more complex chain-of-thought variants that impose robot-specific evaluation criteria, as the latter tend to be overly strict and reduce recall.

A.6 1-shot Fine-tuned Reward Model

Following [46], we finetune a VideoMAE [32] binary classifier per suite using the 10 expert demonstrations from 1-shot SFT, resulting in four multi-task binary reward models. Positive examples are sampled from terminal segments of successful demonstrations; negative examples are constructed from early segments and cross-task segments. The length of each segment is 8. We add a learnable embedding for each task id in one suite to enable multi-task prediction ability. Furthermore, since the 1-shot finetuning quickly overfits to the scarce all-expert data, the originally-produced threshold (≥ 0.9, validated on these 10 training episodes since we have no access to more in-domain validation data) is proved to be so strict that nearly none of the rollouts are predicted to be success. So we change to use a threshold of 0.5 in RL. During inference, the reward model scans the trajectory using a sliding window before finding a segment whose success probability exceeds the threshold (which is regarded as the terminal state of this success rollout, and the rest frames are discarded) . If none of segment is predited as success, the trajecory is predicted as a failure.

A.7 Robometer

Robometer [19] is a VLM-based general-purpose robotic reward model trained on the RBM-1M dataset via a dual objective of frame-level progress prediction and trajectory-level preference com-

parison. It takes instruction and video as input and outputs dense progress and binary success labels in a zero-shot manner across tasks and embodiments. In our experiments, we use it as an oracle reward upper bound. We deploy it as an HTTP service, using its predicted success probability for each subchunk to give binary rewards, with a threshold of 0.85. Similar to 1-shot fine-tuned reward model, if a subchunk is already judged as success before the end of the sequence, we discard the rest frames and regard this step as the terminal state for RL.

B Offline Evaluation of Reward Models

To validate the reliability of our task-agnostic reward, we conduct an offline evaluation of the reward models using a real-environment rollouts dataset on LIBERO (about 10K real trajectories for each of the four suites) with ground-truth binary success labels. We compare the Qwen3-VL alongside oracle Robometer-4B as a reference baseline. We do not compare with 1-shot in-domain finetuned reward model, since the results are too low and scanning along the whole trajectory is too slow for large-scale evaluation. The VLM inference parameters are all the same as in our RL training.

The evaluation reports the micro-averaged Precision (P), Recall (R), and micro-F1-score across all tasks in the four LIBERO suites. A robust reward model needs high precision to avoid false-positive gradients during RL, while maintaining enough recall to provide a meaningful learning signal.

As shown in Table 5, Qwen3-VL delivers strong and consistent F1-scores and Precision on Spatial, Object and Long, without requiring a single target-domain trajectory. While Robometer-4B offers near-oracle performance on the Object suite (likely due to the inclusion of similar prior data), its performance weirdly drops on Spatial and Long. Both models perform poor on Goal, which is the reason for the lowest gain in RL performance across the four suites, as shown in Table 2 and Table 3.

Table 5: Offline Reward Model Evaluation (Averaged precision, recall, and F1 across suites (%)).

	Qwen3-VL			Robometer		
Suite	P	R	F1	P	R	F1
Spatial	**85.8**	**80.0**	**82.8**	73.5	20.1	31.5
Object	90.6	91.8	91.2	**94.3**	**94.8**	**94.6**
Goal	**61.1**	**24.5**	**35.0**	27.5	8.9	13.4
Long	**62.0**	**85.8**	**72.0**	53.8	73.9	62.2

C Additional Results and Analysis

C.1 Qualitative Analysis on First-Frame Ghosting

We provide qualitative examples and analysis on the first-frame ghosting problem in Figure 3, showcasing the efficacy of the progressive injected first-frame noise timestep during inference.

C.2 Qualitative WM Rollout results on Real-World Experiments

We provide qualitative WM rollout results in Figure 5, evaluated on entirely unseen scene layouts absent from the WM's play-data training set and without any fine-tuning. Despite being pre-trained only on uncurated, task-agnostic play data — with *zero* exposure to these downstream tasks or scene layouts — the WM tracks gripper trajectory, contact events, and object dynamics over horizons of 121–201 frames, producing rollouts faithful enough to drive RL post-training entirely in imagination.

C.3 Additional Results and Analysis on Dual-Noise Verification

Effect of DNV on Early Training. While Table 3 in the main text reports the final RL performance, Table 6 further reveals that DNV accelerates the early phase of training. At just 10–20 RL steps, DNV already delivers sizable gains over the no-DNV baseline on most suites, indicating that filtering hallucinated false positives from the outset provides cleaner gradient signal and enables faster initial policy improvement.

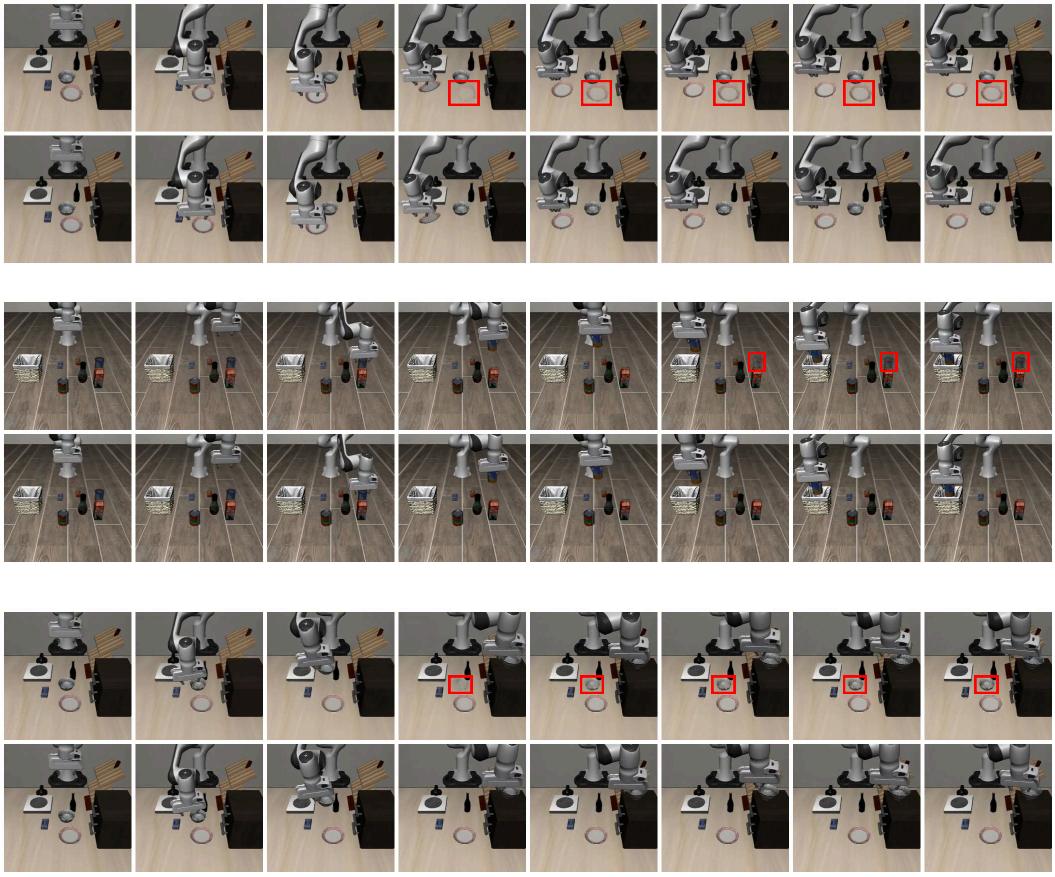


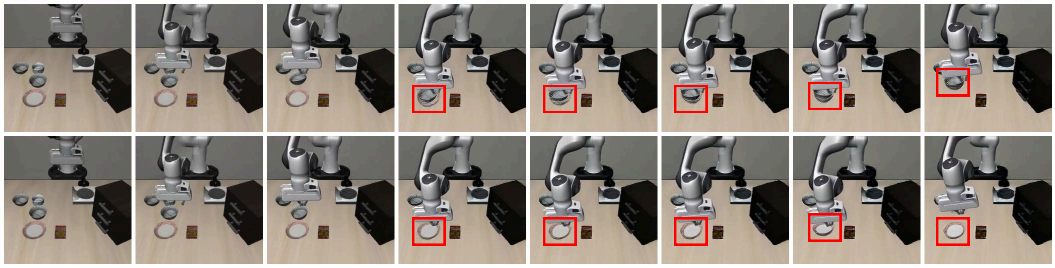
Figure 3: **Qualitative examples of first-frame ghosting and its mitigation via progressive first-frame timestep noise.** For each task, we show two world-model rollouts produced from the same initial observation and the same action sequence, differing only in whether progressive first-frame timestep noise is applied at inference. **Top row** of each subfigure: rollout *without* progressive first-frame timestep noise. The model over-anchors to the first frame $\mathbf{z}^{(0)}$ and copies stale early-frame content into later autoregressive steps, producing visible **ghost artifacts** (red boxes)—e.g., an object that has already been moved or grasped reappears at its original location. **Bottom row**: rollout *with* progressive first-frame timestep noise applied. The anchor's influence is gradually attenuated as the autoregressive horizon extends, so the model relies on recent context to unroll its learned physical dynamics, eliminating ghosting while still preserving global scene consistency.

Qualitative DNV Examples. Figure 4 visualizes how DNV detects hallucinated successes: replaying the same actions under different diffusion noise in the WM may produces divergent outcomes, and the VLM reward disagreement directly exposes rollouts whose apparent success is an artifact of the WM generation stochasticity rather than faithful physical dynamics.

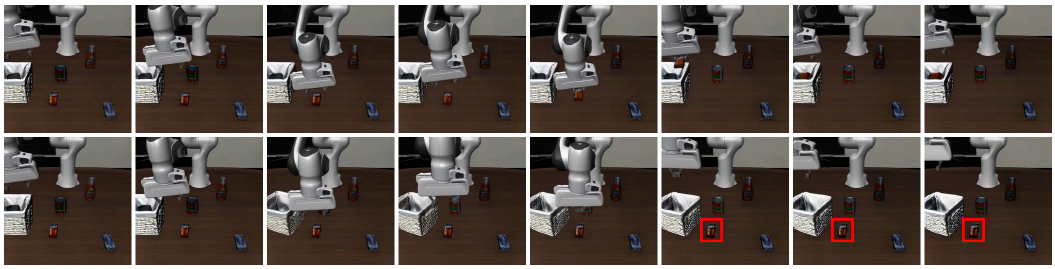
DNV Computational Overhead. DNV requires a second-pass WM generation only for rollouts initially judged as successes by the VLM, which typically constitute a minority of the GRPO group. In practice, this adds approximately $1.3\times$ wall-clock overhead to the WM inference stage per RL step, a modest cost given that DNV yields consistent gains, especially on long-horizon tasks where hallucinations compound most severely (LIBERO-Long).

Table 6: **Additional Results on dual-noise verification.** Reward is fixed to VLM zero-shot (Qwen3-VL). We report success rate (%) at two early RL training checkpoints (step 10 and step 20) to show DNV's positive effect on training at early stage.

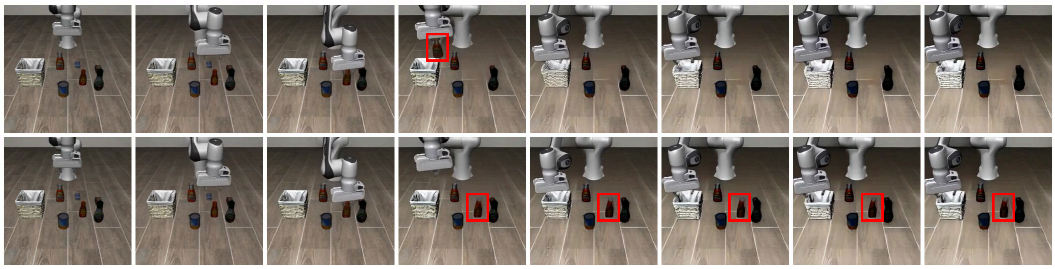
		Spatial		Object		Goal		Long	
WM	DNV	@10	@20	@10	@20	@10	@20	@10	@20
Zero-Shot	w/o	55.6(+1.0)	57.8 (+3.2)	53.2 (+6.8)	46.0 (-0.4)	54.4 (+2.2)	54.2 (+2.0)	21.2 (+1.0)	26.8 (+6.6)
	w/	55.8(+1.2)	61.4 (+6.8)	49.0 (+2.6)	47.2 (+0.8)	55.2 (+3.0)	58.6 (+6.4)	21.0 (+0.8)	26.8 (+6.6)
Co-Train	w/o	59.8(+5.2)	67.2 (+12.6)	56.2 (+9.8)	56.4 (+10.0)	56.2 (+4.0)	57.0 (+4.8)	22.2 (+2.0)	26.4 (+6.2)
	w/	62.0 (+7.4)	66.6 (+12.0)	58.0 (+11.6)	60.2 (+13.8)	58.4 (+6.2)	59.6 (+7.4)	25.6 (+5.4)	27.0 (+6.8)



(a) Task: *"pick up the black bowl between the plate and the ramekin and place it on the plate"* from LIBERO-Spatial.



(b) Task: *"put both the cream cheese box and the butter in the basket"* from LIBERO-Long.



(c) Task: *"pick up the bbq sauce and place it in the basket"* from LIBERO-Object.

Figure 4: **Qualitative examples of Dual-Noise Verification (DNV).** For each task, we show two world-model rollouts produced under the *same action sequence* but with *independently re-sampled initial diffusion noise* at every autoregressive step. **Top row** of each subfigure: the original imagined rollout, on which the VLM reward returns a **success** verdict. **Bottom row**: the second-pass rollout using the same action sequence, under fresh noise, on which the VLM returns a **failure** verdict. The disagreement reveals that the apparent first-pass success was a hallucination unsupported by the WM's true predictive distribution rather than a faithful physical outcome. DNV detects such hallucination and excludes these rollouts from the GRPO update, curbing false-positive-driven reward hacking on unseen tasks.

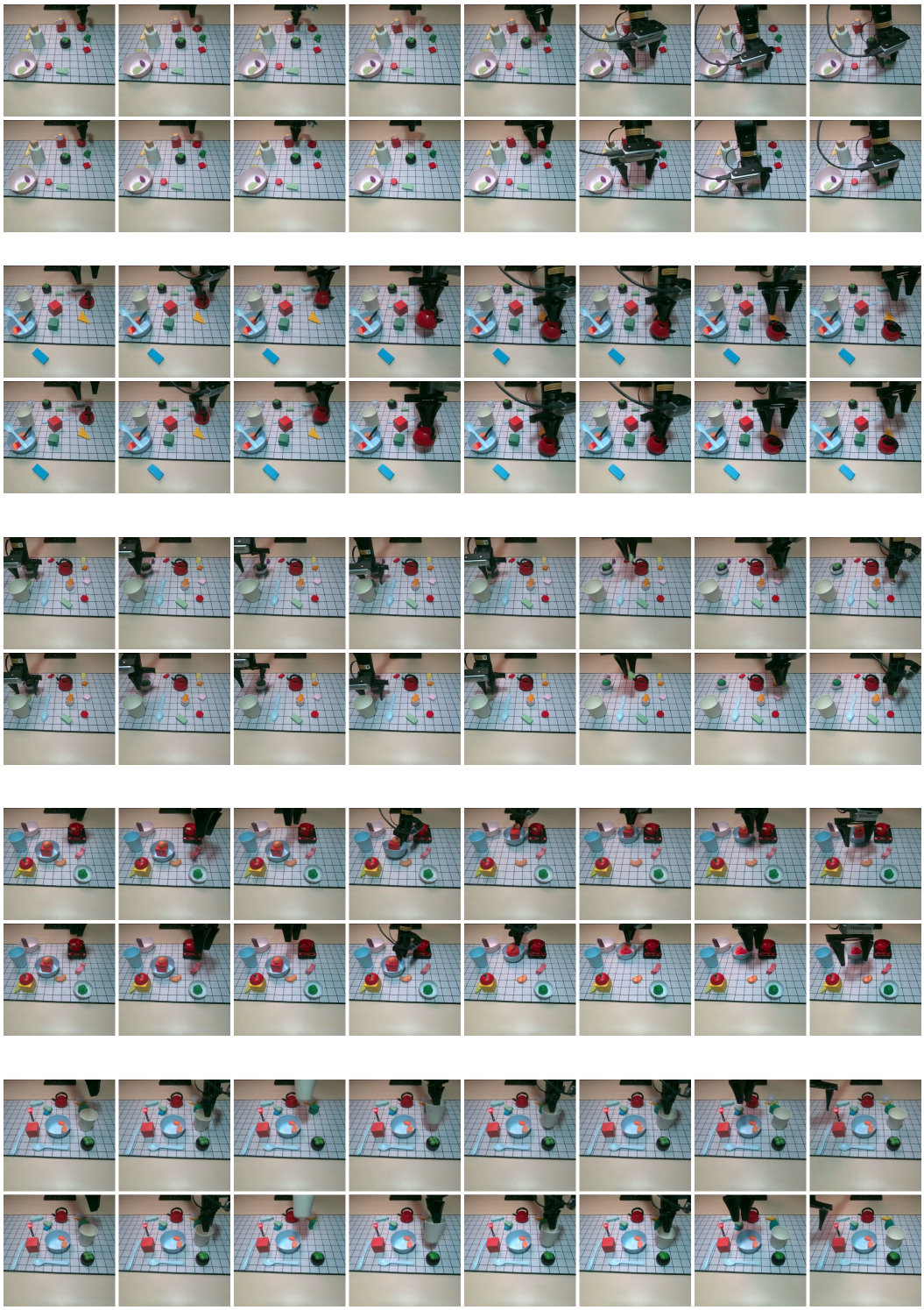


Figure 5: **Qualitative real-world rollouts of our task-agnostic world model. Top row** of each subfigure: the ground-truth real-world video executed on the AgileX Piper arm. **Bottom row**: the corresponding autoregressive prediction from our WM, conditioned on the same initial observation $\mathbf{o}_0$ and the same teleoperated action sequence. These results are evaluated on entirely unseen scene layouts absent from the WM's play-data training set.