VLAW: Iterative Co-Improvement of Vision-Language-Action Policy and World Model

Anonymous Authors[1]

Abstract

The goal of this paper is to improve the performance and reliability of vision-language-action (VLA) models through iterative online interaction. Since collecting policy rollouts in the real world is expensive, we investigate whether a learned simulator—specifically, an action-conditioned video generation model—can be used to generate additional rollout data. Unfortunately, existing world models lack the physical fidelity necessary for policy improvement: they are predominantly trained on demonstration datasets that lack coverage of many different physical interactions (particularly failure cases) and struggle to accurately model small yet critical physical details in contact-rich object manipulation. We propose a simple iterative improvement algorithm that uses real-world roll-out data to improve the fidelity of the world model, which can then, in turn, be used to generate supplemental synthetic data for improving the VLA model. In our experiments on a real robot, we use this approach to improve the performance of a state-of-the-art VLA model on multiple downstream tasks. We achieve a 39.2% absolute success rate improvement over the base policy and 11.6% improvement from training with the generated synthetic rollouts. Videos can be found at this anonymous website: https://sites.google.com/view/vla-w.

1. Introduction

Vision-language-action (VLA) models have achieved great success in robot manipulation by training on large-scale demonstration data (Intelligence et al., 2025b; Kim et al., 2024; Shi et al., 2025; Guo et al., 2025b; Zhang et al., 2024). Recent studies further show that VLA models can benefit substantially from post-training on online interaction rollous (Intelligence et al., 2025a). However, in real-world robotic settings, collecting online policy rollout trajectories requires significant human labor, such as resetting the environment and monitoring robot execution, which is expensive and time-consuming (Atreya et al., 2025; Jain et al., 2025). As a result, the number of online rollouts available for VLA models is often limited, restricting the effectiveness and scalability of post-training.

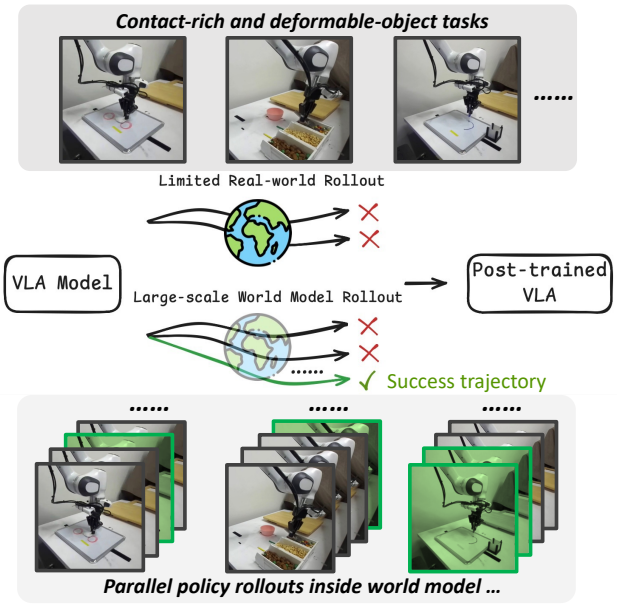

Figure 1. VLA model roll-outs in the real world are time-consuming and unscalable. In VLAW, we first learn an action-conditioned world model using limited real-world online rollouts, which in turn generates large-scale synthetic data in imagination.

Instead of relying solely on real-world policy rollouts, learning an action-conditioned world model to generate synthetic rollouts in imagination offers a promising alternative (Team et al., 2025; Li et al., 2024; Team, 2025b). However, we find that existing world models lack the physical fidelity required for effective policy improvement. As noted in prior works, these models tend to be overly optimistic about predicted trajectories, as they are trained predominantly on demonstration datasets that lack coverage of diverse physical interactions, especially failure cases (Quevedo et al.,

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

2025). Moreover, they struggle to accurately model small yet critical physical details in contact-rich manipulation and can produce blurry visual predictions (Guo et al., 2025a). Consequently, existing action-conditioned world models have largely focused on relatively simple pick-and-place motions and often fail to generate reliable synthetic data for complex tasks involving frequent collisions or deformable objects.

In this paper, we propose a simple yet scalable framework, VLAW, that iteratively improves VLA models via world-model rollouts, as shown in Figure 2. We first learn a physically-grounded world model by finetuning on online rollout data, which includes many failure cases. We find that after training on online rollout data, the world model learns to capture the complex dynamics encountered during policy execution, substantially improving its ability to model both success and failure cases. The improved world model is subsequently used to generate large-scale, high-fidelity synthetic trajectories, which are automatically annotated using a vision–language reward model (Lee et al., 2026). During policy optimization, we only use stable supervised learning objectives that can easily scale to large expressive models (e.g., flow-matching policies with intractable action probabilities (Intelligence et al., 2025b)), as opposed to dynamic programming/bootstrapping or policy gradients.

The core contribution of this paper is a simple and scalable world-model-based reinforcement learning framework for improving state-of-the-art VLA policies in the real world. In our experiments, we use the widely used real-robot platform DROID (Khazatsky et al., 2024). We start from a pretrained VLA policy, $\pi_{0.5}$ (Intelligence et al., 2025b) and an action-conditioned world model, Ctrl-World (Guo et al., 2025a). We first verify that, using policy online rollout data, we learn a physically grounded generative world model that can accurately model both success and failure trajectories, which is essential for generating useful synthetic data. In addition, to obtain a reward model for robot tasks, we fine-tune Qwen3-VL (Team, 2025a; Lee et al., 2026) on real-robot rollout data. Finally, using the synthetic data generated by the world model, we improve the pretrained $\pi_{0.5}$ across many downstream contact-rich manipulation tasks that involve deformable objects in a multi-task setup, outperforming baseline with 11.6%.

2. Related Works

2.1. Post-training Vision-Language-Action Models

Vision–language–action (VLA) models have achieved remarkable success in robotic manipulation tasks (Intelligence et al., 2025b; Pertsch et al., 2025; Liu et al., 2025a; Cui et al., 2025; Hu et al., 2024; Guo et al., 2024). A common approach is to train the VLA on large-scale data and

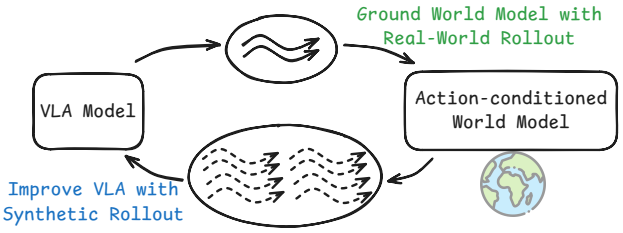

Figure 2. Policy online rollout data can help **ground** the pretrained world model in downstream tasks. Once the world model is grounded, we can generate massive data for policy learning.

then perform supervised fine-tuning on target tasks (Black et al., 2024; Zhang et al., 2025). Beyond supervised fine-tuning, improving VLA policies using online rollout data has emerged as a promising direction (Intelligence et al., 2025a; Guo et al., 2025b; Lu et al., 2025; Zang et al., 2025). Some prior works adopt on-policy reinforcement learning methods, such as PPO (Schulman et al., 2017) or GRPO (Shao et al., 2024), to improve VLA policies.

However, standard on-policy reinforcement learning typically requires a large number of rollouts and is therefore primarily validated in simulation environments (Li et al., 2025b;a; Liu et al., 2025b). Moreover, state-of-the-art VLA models are often trained with flow-matching objectives, which do not provide explicit policy likelihoods, making conventional policy-gradient methods difficult to apply. To enable policy learning in real-world settings, $\pi^*_{0.6}$ (Intelligence et al., 2025a) instead adopts an offline or batch reinforcement learning formulation with an advantage-conditioned supervised learning objective. Similarly, in our setting, we perform iterative policy improvement using batches of real-world rollout data together with world-model–generated synthetic data, and update the policy exclusively through stable supervised fine-tuning objectives.

2.2. World Models for Decision Making

Action-conditioned world models predict future outcomes given current observations and actions, and are also referred to as forward dynamics models. Many works leverage such models for model-based reinforcement learning (Hafner et al., 2019; 2020; Hansen et al., 2022; Oh et al., 2015; Wu et al., 2024) and visual planning (Finn & Levine, 2017; Ebert et al., 2018; Xie et al., 2019; Dasari et al., 2019; Yang et al., 2023). Among these, the most closely related approaches to ours are DayDreamer (Wu et al., 2023), SOLAR (Zhang et al., 2019) and World4rl (Jiang et al., 2025), which also operate in real-world visual model-based reinforcement learning settings. However, due to limited model capacity and data scale, these earlier methods often learned task-specific dynamics models.

With recent advances in video diffusion models (Ren et al., 2025; Ball et al., 2025; Mei et al., 2026), it has become feasible to train multi-task action-conditioned world models that

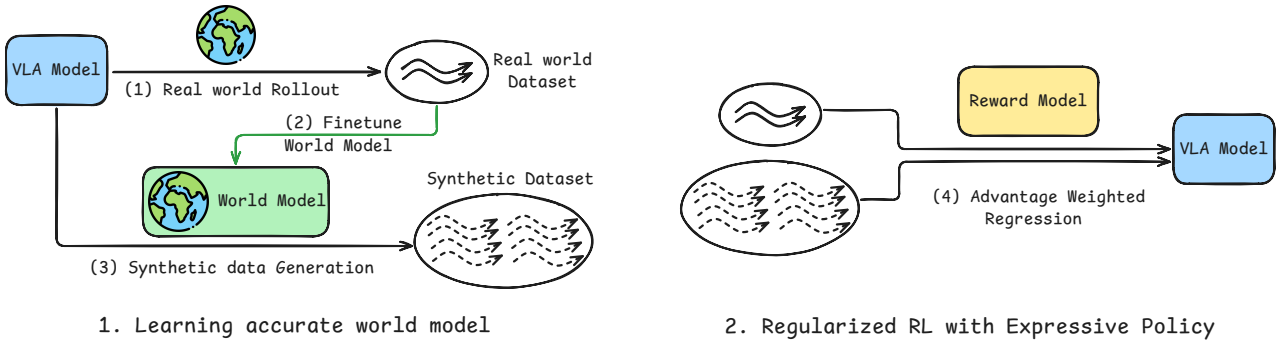

Figure 3. Detailed pipeline for VLAW: (1) We first roll out the policy in the real world to collect a small set of online trajectories. (2) We then fine-tune a pretrained action-conditioned world model on these policy rollout data, grounding the world model in the target tasks and improving its predictive fidelity. (3) Using the resulting world model, we generate large-scale synthetic trajectories through closed-loop interactions between the policy and the world model. (4) Finally, we optimize the VLA policy using both real-world and synthetic data, with reward automatically assessed by a vision–language reward model.

can generate realistic future visual observations (Quevedo et al., 2025; Chen et al., 2024; Gao et al., 2025; Zhu et al., 2024; 2025). Despite this progress, accurately modeling complex physical dynamics remains a fundamental challenge, as widely observed in prior world-model literature (Guo et al., 2025a), likely because these models are trained on offline robotics datasets usually consisting primarily of demonstrations. To address this challenge, we leverage online policy rollout data to ground a pretrained world model in new environments, thereby improving its accuracy around the policy's state–action distribution.

3. Preliminaries

Problem Setting. We study a multi-task robotic manipulation problem, where each task is specified by a language instruction I and is modeled as a Markov decision process (MDP) $\mathcal{M}_I = (\mathcal{S}, \mathcal{A}, P, R_I, \gamma)$. Here, $\mathcal{S}$ denotes the state space, $\mathcal{A}$ the action space, $P(s_{t+1} \mid s_t, a_t)$ the transition dynamics, R_I the task-dependent reward function, and γ the discount factor. At the beginning of training, we are given a pretrained vision–language–action (VLA) policy π_θ and an action-conditioned world model M_ϕ. The policy maps the current state and instruction to an action distribution, $a_t \sim \pi_\theta(\cdot \mid s_t, I)$, while the world model predicts the next state conditioned on the current state and action, $\hat{s}_{t+1} \sim M_\phi(\cdot \mid s_t, a_t)$, where $\hat{s}_{t+1}$ denotes the predicted next state.

The policy is allowed to collect online roll-outs in the real environment, resulting in trajectories $\tau^i_{\text{real}} = \{s_0, a_0, \ldots, a_{T-1}, s_T\}$. Each trajectory is labeled with a task-level reward r_i indicating success or failure. Our goal is to leverage online interaction to iteratively improve the policy so that it performs well across all tasks.

World Model Generated Trajectories. In addition to real-world interaction, we can roll out the policy inside the world model. Starting from an initial state s_0 sampled from a real trajectory, the policy and world model interact in a closed loop via $a_t \sim \pi_\theta(\cdot \mid \hat{s}_t, I)$ and $\hat{s}_{t+1} \sim M_\phi(\cdot \mid \hat{s}_t, a_t)$. By iterating this process, we auto-regressively generate a complete imagined trajectory $\tau^j_{\text{syn}} = \{s_0, a_0, \hat{s}_1, a_1, \ldots, a_{T-1}, \hat{s}_T\}$.

4. Co-Improvement of VLA and World Model

In this section, we describe the details of our method. The overall pipeline consists of the following steps:

1. **World model post-training (Sec. 4.1):** We finetune the world model M using real-world rollout data $\mathcal{D}_{\text{real}}$, jointly training it with the original DROID dataset $\mathcal{D}_{\text{DROID}}$ to maintain broad coverage. In addition, we finetune the vision-language reward model R on $\mathcal{D}_{\text{real}}$ to improve reward accuracy.
2. **VLA policy post-training (Sec. 4.2):** Using the updated world model, we generate a synthetic dataset $\mathcal{D}_{\text{syn}}$ and apply the reward model R to identify successful trajectories, yielding a filtered dataset $\mathcal{D}^+_{\text{syn}}$. This dataset is then used to finetune the VLA policy.
3. We alternate between Steps 1 and 2, iteratively improving both the world model and the policy.

The overall pipeline is summarized in Algorithm 1 and Figure 3. In Sec. 4.3, we provide a detailed analysis showing that our update procedure can be interpreted as an approximation to policy optimization under a regularized reinforcement learning framework.

4.1. World Model Learning with Real Roll-outs

Real World Policy Roll-outs. Previous work has identified two major challenges in learning effective world models: (1) *over-optimism*, as training data is dominated by successful demonstrations; and (2) *limited physical fidelity*, particu-

larly when modeling complex dynamics involving frequent contacts or deformable objects.

To address these issues, we get K trajectories by rolling out the policy in the real world, forming a dataset $\mathcal{D}_{\text{real}} = \{\tau^1_{\text{real}}, ..., \tau^K_{\text{real}}\}$, we also assign a sparse reward $r_\tau \in \{0, 1\}$ to each trajectory to indicate success or not every time we reset robot.

Training Objective. $\mathcal{D}_{\text{real}}$ captures diverse physical interactions encountered during execution, including both success and failure cases, and is used to finetune a pretrained world model. Specifically, we initialize from the pretrained Ctrl-World model (Guo et al., 2025a), a strong diffusion-based world model trained on the full DROID dataset $\mathcal{D}_{\text{DROID}}$. Finetuning on the online rollout dataset $\mathcal{D}_{\text{real}}$ follows the original diffusion objective (Blattmann et al., 2023):

$$\mathcal{L}_{\mathcal{D}_{\text{real}}} = \mathbb{E}_{x_0,\,\epsilon,\,t'} \left\| \hat{x}_0(x_{t'}, t', c) - x_0 \right\|^2, \tag{1}$$

where the prediction target $x_0 = o_{t+1:t+H}$ is sampled from $\mathcal{D}_{\text{real}}$, $x_{t'} = \sqrt{\bar{\alpha}_{t'}}\, x_0 + \sqrt{1-\bar{\alpha}_{t'}}\, \epsilon_{t'}$ denotes the noised future at diffusion step $t' \in [0, T']$ under the noise schedule $\bar{\alpha}_{t'}$, and c represents all conditioning inputs, including the action chunk $a_{t:t+H}$ and the current observation o_t.

Progressively Growing Dataset and Co-training. During successive iterations, we continuously append newly collected real-world trajectories into the dataset: $\mathcal{D}_{\text{real}} = \mathcal{D}_{\text{real}} \cup \tau^i_{\text{real}}$. To prevent overfitting to the limited online rollout data, we also co-train with the original DROID dataset $\mathcal{D}_{\text{DROID}}$ for regularization. The final training objective is:

$$\mathcal{L} = \mathcal{L}_{\mathcal{D}_{\text{real}}} + \lambda\, \mathcal{L}_{\mathcal{D}_{\text{DROID}}} \tag{2}$$

where λ controls the strength of the regularization.

Finetuning Reward Model. To keep our pipeline simple and scalable, we leverage a general-purpose vision-language model, Qwen3-VL-4B-Instruct (Team, 2025a; Lee et al., 2026), to assess whether a trajectory succeeds or not. However, we find that the zero-shot VLM is not accurate enough, so in the first iteration, we fine-tune the VLM with the success labels r_τ in $\mathcal{D}_{\text{real}}$.

In implementation, the reward model takes as input a trajectory video τ^i_{real} together with a query asking whether the task instruction I^i is successfully completed. We classify a trajectory as successful if the probability assigned to the `'yes'` token exceeds a threshold α. By adjusting α, we can make the reward model more or less conservative.

$$R(\tau^i) \;=\; \mathbf{1}\big[P(\texttt{'yes'} \mid \tau^i, I^i) > \alpha\big], \tag{3}$$

4.2. Iterative Improvement for VLA Policy

Scalable Training Pipeline. Once we have a good learned world model and reward model, then we can use it to cheaply generate a large amount of synthetic data. In principle, many different algorithms could be used to leverage this data, including a variety of sophisticated reinforcement learning methods. Because we want to easily scale to large, flow-matching based VLA policies, we choose to use the one of the simplest possible methods for incorporating this synthetic data.

Algorithm 1 VLAW

Require: Pretrained VLA policy π_θ; pretrained world model M_ϕ; reward model R; real-world rollout budget K; synthetic rollout budget N; iterations K_{iter}; reward threshold α

Output: Post-trained policy π_θ and world model M_ϕ

Initialize real-world dataset $\mathcal{D}_{\text{real}} \leftarrow \emptyset$
for $i = 1$ to K_{iter} **do**
  (1) Real-world rollouts
  Roll out π_θ in the real world to collect $\tau^1_{\text{real}}, \ldots, \tau^K_{\text{real}}$
  Append collected trajectories to $\mathcal{D}_{\text{real}}$, success trajectories in $\mathcal{D}^+_{\text{real}}$
  (2) World model and reward model post-training
  Update M_ϕ using $\mathcal{D}_{\text{real}}$ and $\mathcal{D}_{\text{DROID}}$ according to Eq. (1) and Eq. (2)
  (3) Synthetic rollout generation with reward label
  Roll out π_θ in M_ϕ to generate $\mathcal{D}_{\text{syn}} = \tau^1_{\text{syn}}, \ldots, \tau^N_{\text{syn}}$
  Apply reward model R with threshold α (Eq. (3)) to obtain $\mathcal{D}^+_{\text{syn}}$
  (4) Policy post-training
  Update π_θ on $\mathcal{D}^+_{\text{real}} \cup \mathcal{D}^+_{\text{syn}}$ using the flow-matching objective in Eq. (4)
end for
return $\pi_\theta,\ M_\phi$

Specifically, we generate N trajectories by rolling out the policy in imagination: $\mathcal{D}_{\text{syn}} = \{\tau^1_{syn}, ..., \tau^N_{syn}\}$. We then apply the finetuned reward model to identify successful trajectories and construct a filtered dataset containing only success cases: $\mathcal{D}^+_{\text{syn}} = \{\tau^{i_1}_{syn}, ..., \tau^{i_n}_{syn}\}$, where $i_1, ..., i_n$ is the index of success trajectory.

Policy Learning Objective. We update the $\pi_{0.5}$ policy using a weighted flow-matching objective over both real-world rollouts and world-model–generated data. After filtering for successful trajectories, we assign a binary weight $w(o, a) = 1$ to transitions from successful trajectories and $w(o, a) = 0$ to transitions from failed trajectories:

$$\begin{aligned} \mathcal{L} &= \mathbb{E}_{(o,a)\sim\mathcal{D}_{\text{syn}}\cup\mathcal{D}_{\text{real}}}\, w(o,a)\, \mathcal{L}_{\text{FM}}(\theta; o, a) \\ &= \mathbb{E}_{(o,a)\sim\mathcal{D}^+_{\text{syn}}\cup\mathcal{D}^+_{\text{real}}}\, \mathcal{L}_{\text{FM}}(\theta; o, a), \end{aligned} \tag{4}$$

where $\mathcal{L}_{\text{FM}}(\theta; o, a)$ denotes the flow-matching loss for an observation–action pair (o, a).

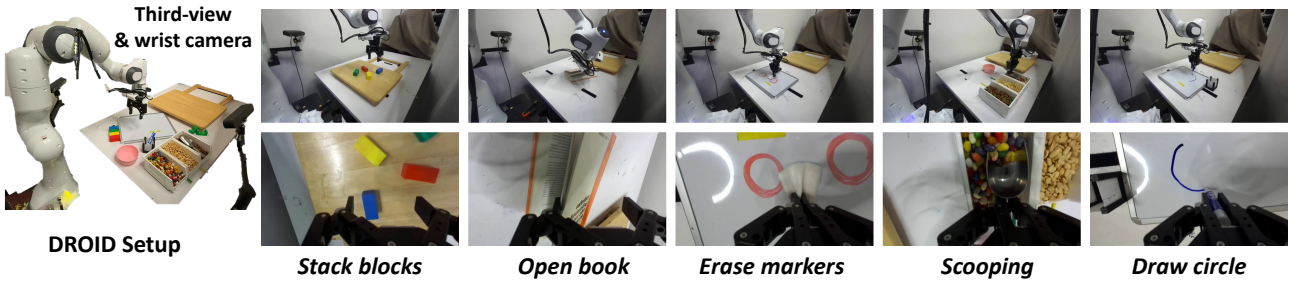

Figure 4. Our experiments are conducted on the DROID platform and cover five task categories, as illustrated in the figure. These tasks involve complex physical interactions, including frequent contact and deformable objects, which are challenging to model in traditional simulations.

4.3. Relation to Regularized Reinforcement Learning

In this subsection, we show that the policy update in Eq. 4 can be view as policy optimization under a regularized reinforcement learning (RL) framework (Peng et al., 2019) with certain approximations.

Under the regularized RL setting, we constrains the learned policy to remain close to a reference policy π_{ref} while optimizing reward. This yields the following regularized objective:

$$J(\theta) = \mathbb{E}_{\tau\sim\rho_{\pi_\theta}}[R(\tau)] - \beta\,\mathbb{E}_{o\sim\rho_{\pi_\theta}}[D(\pi_\theta(\cdot\mid o)\,\|\,\pi_{\text{ref}}(\cdot\mid o))] \tag{5}$$

where $D(\cdot\|\cdot)$ denotes a KL divergence measure and $\beta > 0$ controls the strength of the regularization. The optimal improved policy admits a closed-form solution given by:

$$\pi^\star(a\mid o) \propto w(o,a)\pi_{\text{ref}}(a\mid o), w(o,a) = \exp\left(\frac{A^{\pi_{\text{ref}}}(o,a)}{\beta}\right)$$

where π_{ref} denotes a reference policy, and $A^{\pi_{\text{ref}}}(o,a)$ is the corresponding advantage function, and β is a temperature parameter controlling the strength of the regularization. We can define a surrogate divergence which measures how well π_θ matches samples drawn from $\pi^\star$ under the flow-matching loss:

$$D_{\text{FM}}(\pi^\star(\cdot\mid o), \pi_\theta(\cdot\mid o)) \;\triangleq\; \mathbb{E}_{a\sim\pi^\star(\cdot|o)}\big[\mathcal{L}_{\text{FM}}(\theta; o, a)\big], \tag{6}$$

Using this divergence, we can project policy to the optimal solution with :

$$\theta^\star = \arg\min_\theta\; \mathbb{E}_{(o,a)\sim\mathcal{D}}\Big[w(o,a)\,\mathcal{L}_{\text{FM}}(\theta; o, a)\Big], \tag{7}$$

which is the weighted regression objective used in our policy update equation 4. More detailed derivations are provided in Appendix A.

5. Experiments

In this section, we conduct extensive experiments on complex real-world tasks involving frequent collisions and deformable objects. Our experiments are designed to answer the following questions:

1. Can we learn a high-fidelity action-conditioned world model for contact-rich and deformable-object tasks that accurately models both successful and failed trajectories?
2. Can the synthetic data generated by the world model improve VLA policy performance?
3. Can the policy and world model be continuously improved through an iterative training process in a multi-task setting?

5.1. Experimental Settings

Setups and Tasks. We conduct experiments on the DROID platform (Khazatsky et al., 2024). In the DROID setup, a Franka Panda arm is equipped with a Robotiq gripper. Observations are captured using two third-person cameras and one wrist-mounted camera, as illustrated in Figure 4. We evaluate our method on five categories of contact-rich tasks, described below. More task details can be found in Appendix B.

- **Stacking:** Four colored blocks are randomly placed on the table at the beginning of each episode. The robot receives the instruction: "stack block A on block B," where $A, B \in \{\text{red}, \text{green}, \text{blue}, \text{yellow}\}$.
- **Open Book:** A book is randomly placed on the table at the start of each episode. We evaluate performance across four different books. The robot is instructed to "open the book cover."
- **Erase Marks:** One to three marker drawings are randomly drawn on a whiteboard. The robot receives the instruction: "erase all marks using a tissue."
- **Scooping:** The robot uses a scoop to transfer snacks into a bowl. Both the scoop and the bowl are randomly placed within the workspace. The instruction is: "transfer some A to the bowl," where $A \in \{\text{peanuts}, \text{candies}, \text{almonds}\}$.
- **Drawing:** The robot is instructed to draw a complete circle on a whiteboard using a marker.

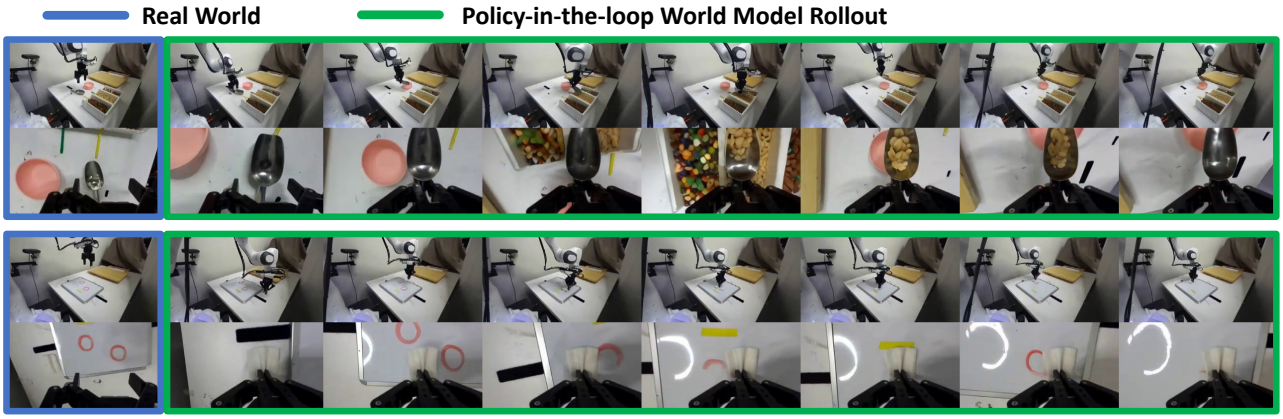

Figure 5. Examples of long-horizon policy-in-the-loop rollouts within the world model starting from the initial observation. The policy $\pi_{0.5}$ is rolled out for 20 iterations (20 seconds). The post-trained world model accurately captures contact-rich physical dynamics. Top: scooping peanuts into a new bowl. Bottom: erasing marker drawings with a tissue.

Method	(1) Video Quality Metrics					(2) Event Confusion Matrix			
	PSNR ↑	SSIM ↑	LPIPS ↓	FID ↓	FVD ↓	TP ↑	FN ↓	**TN** ↑	**FP** ↓
Pretrained Ctrl-world	16.32	0.634	0.347	41.03	225.13	-	-	-	-
Pretrained Ctrl-world + Expert Rollout	19.87	0.748	0.189	12.76	99.98	28	2	9	11
Pretrained Ctrl-world + Expert Rollout + Online Rollout	**21.77**	**0.784**	**0.136**	**9.58**	**64.12**	26	4	19	1

Table 1. We replay recorded action sequences in the world model. (1) We evaluate video quality metrics on 256 replayed clips, each 5 seconds long. All metrics are computed using the wrist-view camera, as this viewpoint best captures object interactions during manipulation. (2) The interaction phase is the primary source of errors. Therefore, we report an event-level confusion matrix on 50 clips involving physical interactions. For each clip, we label the interaction outcome (success or failure) and compare the model predictions against real-world outcomes.

Base Models and Hyperparameters. We use $\pi_{0.5}$ (Intelligence et al., 2025b) as the base vision–language–action (VLA) model and Ctrl-World (Guo et al., 2025a) as the base world model. For each task category, we collect 25 expert demonstrations and finetune $\pi_{0.5}$ on this data to warm-start the policy, which serves as our base policy. The reward model is initialized from Qwen3-VL-4B-Instruct (Team, 2025a).

In each iteration, we roll out 50 trajectories per task category in the real world. We finetune the world model for 50K training steps using these rollout trajectories. We then generate 500 synthetic trajectories per task using the updated world model to form the synthetic dataset. The reward model is additionally finetuned using rollout data from the first iteration to improve reward accuracy. The policy is updated with 2k steps with batch size 256. We perform a total of two iterations of this procedure.

5.2. Can we learn an accurate action-conditioned world model for contact-rich tasks?

Action replay inside the world model. We evaluate the fidelity of the learned world model and study the contribution of online rollout data by replaying real-world action sequences inside the world model. Specifically, we randomly select a starting frame from a real-world trajectory and auto-regressively feed a 5-second sequence of recorded action chunks to the world model, starting from the same frame. We compare our post-trained world model against two baselines: the original pretrained world model and a model finetuned only on expert demonstration data.

We use two categories of metrics to quantitatively evaluate video prediction quality:

- **(1) Video distance metrics:** These include pixel-level metrics (PSNR (Hore & Ziou, 2010) and SSIM (Wang et al., 2004)) as well as learned perceptual and distributional metrics (LPIPS (Zhang et al., 2018), FID (Heusel et al., 2017), and FVD (Unterthiner et al., 2018)).
- **(2) Interaction event confusion matrix:** Correctly predicting the outcome of object interactions is the most challenging aspect of action-conditioned world modeling. We filter replayed clips that involve object interactions and classify each interaction as success or failure. We then evaluate whether the predicted outcome aligns with the real-world result.

Quantitative results are reported in Table 1. Finetuning with

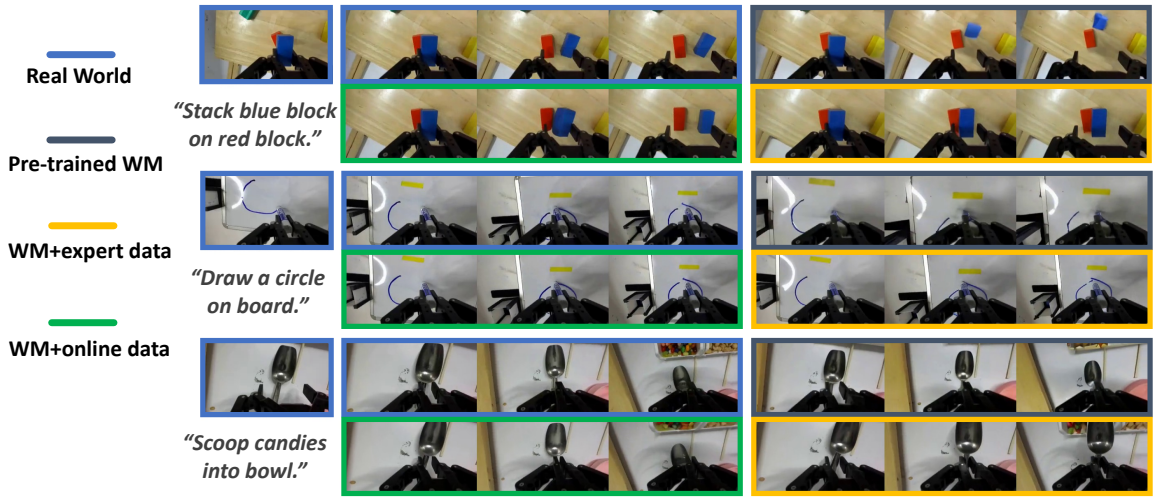

Figure 6. Conditioned on the same initial frame and identical action sequences (five chunks), we roll out trajectories inside different world models. The pretrained Ctrl-World model is insufficiently accurate for these contact-rich tasks. World models fine-tuned only on expert trajectories tend to be overly optimistic. In contrast, the world model fine-tuned on policy online rollout data accurately captures the underlying physical dynamics and is well aligned with real-world outcomes. Only the wrist-view camera is shown due to space limitations. Zoom in for better comparisons.

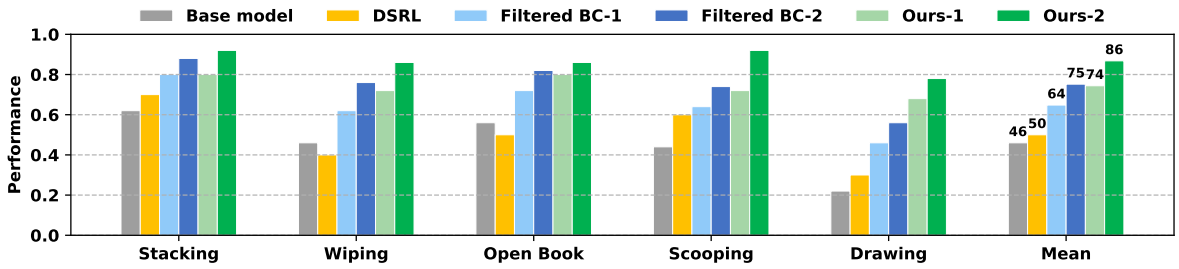

Figure 7. Success Rate Improvement Comparison with Baselines. We perform two rounds of iterative training. "Ours-1" denotes the VLAW method after the first round of online rollouts. Overall, VLAW consistently outperforms both the filtered BC and DSRL baselines in the multi-task setting.

online rollout data is crucial for world model performance: all video quality metrics improve substantially compared to both baselines. Moreover, by training on mixed success and failure trajectories, the world model largely eliminates the over-optimistic bias observed when training only on expert demonstrations. In particular, false-positive interaction predictions are significantly reduced. We provide qualitative visualizations of interaction replay in Figure 6.

Policy-in-the-loop rollout. We further evaluate the world model by rolling out the policy directly inside the learned model. Although evaluated tasks involve complex, contact-rich interactions, and we find that the post-trained world model maintains high visual fidelity and physical plausibility even for long-horizon rollouts of up to 20 seconds. Example rollouts are shown in Figure 5. This long-horizon stability enables effective search for successful trajectories within the world model, which we subsequently leverage for policy improvement.

5.3. Can world model generated data improve VLA policy performance?

Baselines. Our goal is to leverage real-world online interaction data to improve the VLA policy while minimizing physical rollouts. Under this setting, we compare our method against two baselines that do not utilize a world model:

- **(1) Filtered BC**, which filters successful trajectories from real-world rollouts and performs supervised fine-tuning on these trajectories. We control the real world rollout number the same as our method for fair comparison (50 rollouts for each category of tasks).
- **(2) DSRL (Wagenmaker et al., 2025)**, which improves the $\pi_{0.5}$ policy by optimizing its noise space through online exploration, we control the online rollout number the same as other methods.

Large-scale rollout visualizations. We visualize parallel rollouts generated by the world model in Figure 8. Starting from an initial frame recorded in the real world (GT), we search for successful trajectories entirely within the world

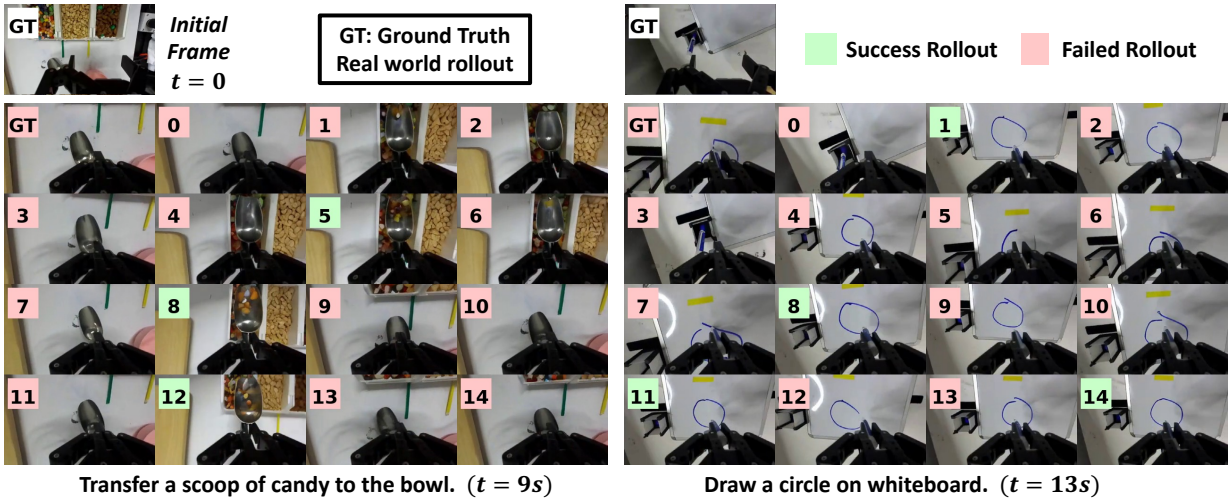

Figure 8. GT denotes the real-world rollout, while $0 \sim 14$ denotes diverse trajectories imagined by the world model, all rollouts from the same GT initial frame with $\pi_{0.5}$. In the real-world rollout, the robot fails to grasp the scoop (left, GT) and fails to draw a complete circle (right, GT). With the help of a world model, we can **search successful trajectories for failure cases**, which can be useful for policy learning.

model. These successful imagined trajectories provide additional supervision for policy learning, enabling the policy to progressively overcome failure cases and improve task performance.

Reward model analysis. We use a learned reward model to filter successful trajectories from world model–generated rollouts. As described in the method section, a trajectory is considered successful only if the probability assigned to the `'yes'` token exceeds a predefined threshold. This thresholding strategy substantially reduces false-positive trajectories. Additional details and analyses of the reward model are provided in Appendix C.

Results. The success rate improvements are shown in Figure 7. DSRL achieves limited gains in our multi-task setting. We hypothesize that this is because reinforcement learning becomes significantly harder to optimize across diverse tasks, and because DSRL constrains optimization to the noise space of the $\pi_{0.5}$ policy rather than updating the model parameters directly, which limits the expressive capacity of the policy. Filtered BC improves performance over two iterations by leveraging successful real-world trajectories. In contrast, by generating large-scale synthetic rollouts and selectively filtering successful trajectories, VLAW achieves substantially larger performance gains across all tasks.

Ablations. We conduct ablation studies on (1) the number of world model rollouts and (2) whether real-world rollout data is included during policy finetuning. We evaluate these ablations on the most challenging drawing task, with results shown in Figure 9. Reducing the amount of synthetic rollout data leads to noticeable performance degradation, and removing real-world success trajectories during finetuning further harms performance, highlighting the importance of both components.

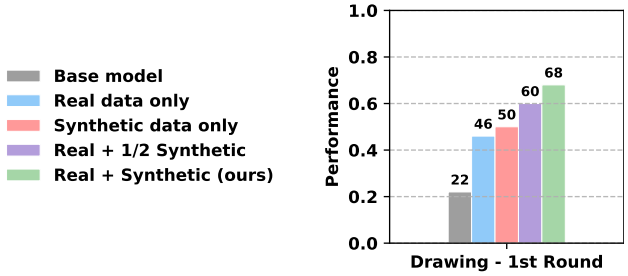

Figure 9. We conduct ablation studies on (1) the amount of synthetic data used for policy fine-tuning (reducing from 500 to 250 trajectories) and (2) whether real-world rollout data (50 trajectories) is included during fine-tuning. We observe that either decreasing the number of synthetic trajectories or removing the real-world dataset leads to a performance degradation.

6. Conclusions and discussions

In this paper, we propose VLAW, an iterative improvement pipeline that jointly enhances both the vision–language–action (VLA) policy and the action-conditioned world model. We demonstrate that VLAW consistently improves performance across multiple contact-rich manipulation tasks. Although the learned world model achieves high fidelity on the downstream tasks from which online data are collected, our current evaluation is limited to five task categories. Scaling online rollout data to a broader and more diverse set of tasks is a promising direction for future work. We believe that, as base video models continue to advance and large-scale robot interaction data become increasingly available, world-model-based training will provide a powerful new paradigm for learning generalist robotic policies.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Atreya, P., Pertsch, K., Lee, T., Kim, M. J., Jain, A., Kuramshin, A., Eppner, C., Neary, C., Hu, E., Ramos, F., et al. Roboarena: Distributed real-world evaluation of generalist robot policies. *arXiv preprint arXiv:2506.18123*, 2025.

Ball, P. J., Bauer, J., Belletti, F., Brownfield, B., Ephrat, A., Fruchter, S., Gupta, A., Holsheimer, K., Holynski, A., Hron, J., Kaplanis, C., Limont, M., McGill, M., Oliveira, Y., Parker-Holder, J., Perbet, F., Scully, G., Shar, J., Spencer, S., Tov, O., Villegas, R., Wang, E., Yung, J., Baetu, C., Berbel, J., Bridson, D., Bruce, J., Buttimore, G., Chakera, S., Chandra, B., Collins, P., Cullum, A., Damoc, B., Dasagi, V., Gazeau, M., Gbadamosi, C., Han, W., Hirst, E., Kachra, A., Kerley, L., Kjems, K., Knoepfel, E., Koriakin, V., Lo, J., Lu, C., Mehring, Z., Moufarek, A., Nandwani, H., Oliveira, V., Pardo, F., Park, J., Pierson, A., Poole, B., Ran, H., Salimans, T., Sanchez, M., Saprykin, I., Shen, A., Sidhwani, S., Smith, D., Stanton, J., Tomlinson, H., Vijaykumar, D., Wang, L., Wingfield, P., Wong, N., Xu, K., Yew, C., Young, N., Zubov, V., Eck, D., Erhan, D., Kavukcuoglu, K., Hassabis, D., Gharamani, Z., Hadsell, R., van den Oord, A., Mosseri, I., Bolton, A., Singh, S., and Rocktäschel, T. Genie 3: A new frontier for world models. 2025.

Black, K., Brown, N., Driess, D., Esmail, A., Equi, M., Finn, C., Fusai, N., Groom, L., Hausman, K., Ichter, B., et al. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

Blattmann, A., Dockhorn, T., Kulal, S., Mendelevitch, D., Kilian, M., Lorenz, D., Levi, Y., English, Z., Voleti, V., Letts, A., et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023.

Chen, B., Martí Monsó, D., Du, Y., Simchowitz, M., Tedrake, R., and Sitzmann, V. Diffusion forcing: Next-token prediction meets full-sequence diffusion. *Advances in Neural Information Processing Systems*, 37:24081–24125, 2024.

Cui, C., Ding, P., Song, W., Bai, S., Tong, X., Ge, Z., Suo, R., Zhou, W., Liu, Y., Jia, B., et al. Openhelix: A short survey, empirical analysis, and open-source dual-system vla model for robotic manipulation. *arXiv preprint arXiv:2505.03912*, 2025.

Dasari, S., Ebert, F., Tian, S., Nair, S., Bucher, B., Schmeckpeper, K., Singh, S., Levine, S., and Finn, C. Robonet: Large-scale multi-robot learning. *arXiv preprint arXiv:1910.11215*, 2019.

Ebert, F., Finn, C., Dasari, S., Xie, A., Lee, A., and Levine, S. Visual foresight: Model-based deep reinforcement learning for vision-based robotic control. *arXiv preprint arXiv:1812.00568*, 2018.

Finn, C. and Levine, S. Deep visual foresight for planning robot motion. In *2017 IEEE international conference on robotics and automation (ICRA)*, pp. 2786–2793. IEEE, 2017.

Gao, S., Zhou, S., Du, Y., Zhang, J., and Gan, C. Adaworld: Learning adaptable world models with latent actions. *arXiv preprint arXiv:2503.18938*, 2025.

Guo, Y., Hu, Y., Zhang, J., Wang, Y.-J., Chen, X., Lu, C., and Chen, J. Prediction with action: Visual policy learning via joint denoising process. *Advances in Neural Information Processing Systems*, 37:112386–112410, 2024.

Guo, Y., Shi, L. X., Chen, J., and Finn, C. Ctrl-world: A controllable generative world model for robot manipulation. *arXiv preprint arXiv:2510.10125*, 2025a.

Guo, Y., Zhang, J., Chen, X., Ji, X., Wang, Y.-J., Hu, Y., and Chen, J. Improving vision-language-action model with online reinforcement learning. *arXiv preprint arXiv:2501.16664*, 2025b.

Hafner, D., Lillicrap, T., Ba, J., and Norouzi, M. Dream to control: Learning behaviors by latent imagination. *arXiv preprint arXiv:1912.01603*, 2019.

Hafner, D., Lillicrap, T., Norouzi, M., and Ba, J. Mastering atari with discrete world models. *arXiv preprint arXiv:2010.02193*, 2020.

Hansen, N., Wang, X., and Su, H. Temporal difference learning for model predictive control. *arXiv preprint arXiv:2203.04955*, 2022.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

Hore, A. and Ziou, D. Image quality metrics: Psnr vs. ssim. In *2010 20th international conference on pattern recognition*, pp. 2366–2369. IEEE, 2010.

Hu, Y., Guo, Y., Wang, P., Chen, X., Wang, Y.-J., Zhang, J., Sreenath, K., Lu, C., and Chen, J. Video prediction policy: A generalist robot policy with predictive visual representations. *arXiv preprint arXiv:2412.14803*, 2024.

Intelligence, P., Amin, A., Aniceto, R., Balakrishna, A., Black, K., Conley, K., Connors, G., Darpinian, J., Dhabalia, K., DiCarlo, J., et al. $\pi^*_{0.6}$: a vla that learns from experience. *arXiv preprint arXiv:2511.14759*, 2025a.

Intelligence, P., Black, K., Brown, N., Darpinian, J., Dhabalia, K., Driess, D., Esmail, A., Equi, M., Finn, C., Fusai, N., et al. $\pi_{0.5}$: a vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025b.

Jain, A., Zhang, M., Arora, K., Chen, W., Torne, M., Irshad, M. Z., Zakharov, S., Wang, Y., Levine, S., Finn, C., et al. Polaris: Scalable real-to-sim evaluations for generalist robot policies. *arXiv preprint arXiv:2512.16881*, 2025.

Jiang, Z., Liu, K., Qin, Y., Tian, S., Zheng, Y., Zhou, M., Yu, C., Li, H., and Zhao, D. World4rl: Diffusion world models for policy refinement with reinforcement learning for robotic manipulation. *arXiv preprint arXiv:2509.19080*, 2025.

Khazatsky, A., Pertsch, K., Nair, S., Balakrishna, A., Dasari, S., Karamcheti, S., Nasiriany, S., Srirama, M. K., Chen, L. Y., Ellis, K., et al. Droid: A large-scale in-the-wild robot manipulation dataset. *arXiv preprint arXiv:2403.12945*, 2024.

Kim, M. J., Pertsch, K., Karamcheti, S., Xiao, T., Balakrishna, A., Nair, S., Rafailov, R., Foster, E., Lam, G., Sanketi, P., et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

Lee, T., Wagenmaker, A., Pertsch, K., Liang, P., Levine, S., and Finn, C. Roboreward: General-purpose vision-language reward models for robotics. *arXiv preprint arXiv:2601.00675*, 2026.

Li, H., Ding, P., Suo, R., Wang, Y., Ge, Z., Zang, D., Yu, K., Sun, M., Zhang, H., Wang, D., et al. Vla-rft: Vision-language-action reinforcement fine-tuning with verified rewards in world simulators. *arXiv preprint arXiv:2510.00406*, 2025a.

Li, H., Zuo, Y., Yu, J., Zhang, Y., Yang, Z., Zhang, K., Zhu, X., Zhang, Y., Chen, T., Cui, G., et al. Simplevla-rl: Scaling vla training via reinforcement learning. *arXiv preprint arXiv:2509.09674*, 2025b.

Li, X., Hsu, K., Gu, J., Pertsch, K., Mees, O., Walke, H. R., Fu, C., Lunawat, I., Sieh, I., Kirmani, S., et al. Evaluating real-world robot manipulation policies in simulation. *arXiv preprint arXiv:2405.05941*, 2024.

Liu, J., Chen, H., An, P., Liu, Z., Zhang, R., Gu, C., Li, X., Guo, Z., Chen, S., Liu, M., et al. Hybridvla: Collaborative diffusion and autoregression in a unified vision-language-action model. *arXiv preprint arXiv:2503.10631*, 2025a.

Liu, J., Gao, F., Wei, B., Chen, X., Liao, Q., Wu, Y., Yu, C., and Wang, Y. What can rl bring to vla generalization? an empirical study. *arXiv preprint arXiv:2505.19789*, 2025b.

Lu, G., Guo, W., Zhang, C., Zhou, Y., Jiang, H., Gao, Z., Tang, Y., and Wang, Z. Vla-rl: Towards masterful and general robotic manipulation with scalable reinforcement learning. *arXiv preprint arXiv:2505.18719*, 2025.

Mei, Z., Yin, T., Shorinwa, O., Badithela, A., Zheng, Z., Bruno, J., Bland, M., Zha, L., Hancock, A., Fisac, J. F., et al. Video generation models in robotics-applications, research challenges, future directions. *arXiv preprint arXiv:2601.07823*, 2026.

Oh, J., Guo, X., Lee, H., Lewis, R. L., and Singh, S. Action-conditional video prediction using deep networks in atari games. *Advances in neural information processing systems*, 28, 2015.

Peng, X. B., Kumar, A., Zhang, G., and Levine, S. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning. *arXiv preprint arXiv:1910.00177*, 2019.

Pertsch, K., Stachowicz, K., Ichter, B., Driess, D., Nair, S., Vuong, Q., Mees, O., Finn, C., and Levine, S. Fast: Efficient action tokenization for vision-language-action models. *arXiv preprint arXiv:2501.09747*, 2025.

Quevedo, J., Liang, P., and Yang, S. Evaluating robot policies in a world model. *arXiv preprint arXiv:2506.00613*, 2025.

Ren, X., Lu, Y., Cao, T., Gao, R., Huang, S., Sabour, A., Shen, T., Pfaff, T., Wu, J. Z., Chen, R., et al. Cosmos-drive-dreams: Scalable synthetic driving data generation with world foundation models. *arXiv preprint arXiv:2506.09042*, 2025.

Schulman, J., Levine, S., Abbeel, P., Jordan, M., and Moritz, P. Trust region policy optimization. In *International conference on machine learning*, pp. 1889–1897. PMLR, 2015.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Shao, Z., Wang, P., Zhu, Q., Xu, R., Song, J., Bi, X., Zhang, H., Zhang, M., Li, Y., Wu, Y., et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Shi, L. X., Ichter, B., Equi, M., Ke, L., Pertsch, K., Vuong, Q., Tanner, J., Walling, A., Wang, H., Fusai, N., et al. Hi robot: Open-ended instruction following with hierarchical vision-language-action models. *arXiv preprint arXiv:2502.19417*, 2025.

Team, G. R., Devin, C., Du, Y., Dwibedi, D., Gao, R., Jindal, A., Kipf, T., Kirmani, S., Liu, F., Majumdar, A., et al. Evaluating gemini robotics policies in a veo world simulator. *arXiv preprint arXiv:2512.10675*, 2025.

Team, Q. Qwen3-vl: Sharper vision, deeper thought, broader action. *Qwen Blog. Accessed*, pp. 10–04, 2025a.

Team, X. W. M. 1x world model: Evaluating bits, not atoms. 2025b. URL https://www.1x.tech/1x-world-model.pdf.

Unterthiner, T., Van Steenkiste, S., Kurach, K., Marinier, R., Michalski, M., and Gelly, S. Towards accurate generative models of video: A new metric & challenges. *arXiv preprint arXiv:1812.01717*, 2018.

Wagenmaker, A., Nakamoto, M., Zhang, Y., Park, S., Yagoub, W., Nagabandi, A., Gupta, A., and Levine, S. Steering your diffusion policy with latent space reinforcement learning. *arXiv preprint arXiv:2506.15799*, 2025.

Wang, Z., Bovik, A. C., Sheikh, H. R., and Simoncelli, E. P. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

Wu, J., Yin, S., Feng, N., He, X., Li, D., Hao, J., and Long, M. ivideogpt: Interactive videogpts are scalable world models. *Advances in Neural Information Processing Systems*, 37:68082–68119, 2024.

Wu, P., Escontrela, A., Hafner, D., Abbeel, P., and Goldberg, K. Daydreamer: World models for physical robot learning. In *Conference on robot learning*, pp. 2226–2240. PMLR, 2023.

Xie, A., Ebert, F., Levine, S., and Finn, C. Improvisation through physical understanding: Using novel objects as tools with visual foresight. *arXiv preprint arXiv:1904.05538*, 2019.

Yang, M., Du, Y., Ghasemipour, K., Tompson, J., Schuurmans, D., and Abbeel, P. Learning interactive real-world simulators. *arXiv preprint arXiv:2310.06114*, 1(2): 6, 2023.

Zang, H., Wei, M., Xu, S., Wu, Y., Guo, Z., Wang, Y., Lin, H., Shi, L., Xie, Y., Xu, Z., et al. Rlinf-vla: A unified and efficient framework for vla+ rl training. *arXiv preprint arXiv:2510.06710*, 2025.

Zhang, J., Guo, Y., Chen, X., Wang, Y.-J., Hu, Y., Shi, C., and Chen, J. Hirt: Enhancing robotic control with hierarchical robot transformers. *arXiv preprint arXiv:2410.05273*, 2024.

Zhang, J., Guo, Y., Hu, Y., Chen, X., Zhu, X., and Chen, J. Up-vla: A unified understanding and prediction model for embodied agent. *arXiv preprint arXiv:2501.18867*, 2025.

Zhang, M., Vikram, S., Smith, L., Abbeel, P., Johnson, M., and Levine, S. Solar: Deep structured representations for model-based reinforcement learning. In *International conference on machine learning*, pp. 7444–7453. PMLR, 2019.

Zhang, R., Isola, P., Efros, A. A., Shechtman, E., and Wang, O. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 586–595, 2018.

Zhu, F., Wu, H., Guo, S., Liu, Y., Cheang, C., and Kong, T. Irasim: Learning interactive real-robot action simulators. *arXiv preprint arXiv:2406.14540*, 2024.

Zhu, F., Yan, Z., Hong, Z., Shou, Q., Ma, X., and Guo, S. Wmpo: World model-based policy optimization for vision-language-action models. *arXiv preprint arXiv:2511.09515*, 2025.

A. Relation to Regularized Reinforcement Learning.

In this part, we relate the policy update in Eq. 4 to policy optimization under a regularized reinforcement learning (RL) framework with certain approximations. Our VLA policy is trained with a flow-matching objective and does not provide a tractable action log-likelihood, so standard KL-based derivations do not apply directly. Under the regularized RL setting, the optimal improved policy admits a closed-form solution given by:

$$\pi^\star(a \mid o) \;\propto\; \pi_{\text{ref}}(a \mid o)\, \exp\!\left(\frac{A^{\pi_{\text{ref}}}(o,a)}{\beta}\right), \tag{8}$$

where π_{ref} denotes a reference policy, $A^{\pi_{\text{ref}}}(o,a)$ is the corresponding advantage function, and β is a temperature parameter controlling the strength of the regularization.

Since the target distribution $\pi^\star$ is generally not representable within a finite parametric policy class, policy improvement is typically performed via a *projection step*, which fits a parametric policy π_θ to $\pi^\star$ by minimizing a divergence D:

$$\theta^\star = \arg\min_\theta \; \mathbb{E}_{o\sim\mathcal{D}}\Big[D(\pi^\star(\cdot \mid o), \pi_\theta(\cdot \mid o))\Big]. \tag{9}$$

AWR for flow-matching policies. In standard Advantage-Weighted Regression (AWR) (Peng et al., 2019), the divergence D is chosen to be the KL divergence, which results in a weighted log-likelihood objective. However, because our VLA policy is trained using a flow-matching objective $\mathcal{L}_{\text{FM}}(\theta; o, a)$ and does not provide explicit action likelihoods, this formulation is not directly applicable.

Instead, we define a projection operator that is compatible with flow matching by introducing the following surrogate divergence:

$$D_{\text{FM}}(\pi^\star(\cdot \mid o), \pi_\theta(\cdot \mid o)) \;\triangleq\; \mathbb{E}_{a\sim\pi^\star(\cdot|o)}\big[\mathcal{L}_{\text{FM}}(\theta; o, a)\big], \tag{10}$$

which measures how well π_θ matches samples drawn from $\pi^\star$ under the flow-matching loss.

Using this divergence, the projection step becomes:

$$\begin{aligned} \theta^\star &= \arg\min_\theta \; \mathbb{E}_{o\sim\mathcal{D}}\, \mathbb{E}_{a\sim\pi^\star(\cdot|o)}\big[\mathcal{L}_{\text{FM}}(\theta; o, a)\big] \\ &\approx \arg\min_\theta \; \mathbb{E}_{(o,a)\sim\mathcal{D}}\Big[w(o,a)\, \mathcal{L}_{\text{FM}}(\theta; o, a)\Big], \end{aligned} \tag{11}$$

where the approximation follows a standard offline RL practice that replaces sampling from $\pi^\star$ with weighted samples from a fixed dataset (Schulman et al., 2015). The weights are proportional to the exponential advantage: $w(o,a) \;\propto\; \exp\!\Big(\frac{A^{\pi_{\text{ref}}}(o,a)}{\beta}\Big)$.

Then, by setting the discount factor $\gamma \to 1$ and assigning a large negative reward to failure trajectories, Eq. 11 reduces to Eq. 4, which is the objective used in our policy update.

B. Task Details

Success Criteria. We define task success using simple, outcome-based criteria that can be reliably judged from the final state (or a short post-action observation window):

- **Stacking:** Success if block A is stably placed on top of block B (with A supported by B, not the table) and the stack remains upright for a short holding period.
- **Open Book:** Success if the front cover is opened beyond a predefined angle (e.g., clearly separated from the pages and lying open) and remains open at the end of the episode.
- **Erase Marks:** Success if all visible marker strokes are removed from the whiteboard area (i.e., no clearly detectable marks remain) at the end of the episode.
- **Scooping:** Success if at least a minimum amount of the target object A is transferred into the bowl (with non-trivial contents remaining in the bowl at the end), while the majority of the transferred items are inside the bowl rather than spilled outside.

- **Drawing:** Success if the robot produces a single closed curve that forms a visually complete circle (i.e., endpoints meet with small gap tolerance) on the whiteboard within the designated drawing region.

Detailed success rate improvement All task is evaluated 50 times since we collect 50 online rollouts in each iteration. DSRL baseline is evaluated with 10 times since it's too time-consuming to evaluate too many rollouts during online update.

Method	Stacking	Wiping	Open Book	Scooping	Drawing	Mean
Base model	0.62	0.46	0.56	0.44	0.22	0.460
DSRL	0.70	0.40	0.50	0.60	0.30	0.500
Filtered BC-1	0.80	0.62	0.72	0.64	0.46	0.648
Filtered BC-2	0.88	0.76	0.82	0.74	0.56	0.752
Ours-1	0.80	0.72	0.80	0.72	0.68	0.744
Ours-2	0.92	0.86	0.86	0.92	0.78	0.868

Table 2. Detailed Success rates across 5 manipulation tasks.

C. Reward Model Details

We use the Qwen3-VL-4B-Instruct model (Team, 2025a) as the vision–language reward model. Each trajectory is temporally downsampled into a 16-frame video before being fed to the model. We fintune the Qwen3-VL-4B-Instruct model for 200 steps with batch size 128.

We observe that directly prompting the reward model to output a binary `yes`/`no` decision can be overly optimistic, leading to a non-negligible number of false positives. To mitigate this issue, we instead examine the model-assigned probability of the ``` ``yes'' ``` token and only label a trajectory as successful when this probability exceeds a threshold of 0.8, with this threshold, model is more conservative on generate success label.

We compare this threshold-based criterion with the naive approach of directly querying the model for a binary answer. Empirically, using a higher confidence threshold substantially reduces the number of false-positive trajectories, resulting in more reliable supervision for downstream policy learning.

Table 3. Confusion matrices comparing the original reward model decision and our threshold-based criterion. We manually label a subset of 40 trajectories and compare the predictions of each method against human-annotated ground-truth labels. The false-positive number significantly dropped.

Original Method (Direct Yes/No Output)			
		Predicted	
		Success	Failure
GT	Success	15	7
	Failure	8	10

Ours (Probability Threshold $p(\texttt{yes}) > 0.8$)			
		Predicted	
		Success	Failure
GT	Success	10	12
	Failure	2	16