WORLD-GYMNAST: TRAINING ROBOTS WITH REINFORCEMENT LEARNING IN A WORLD MODEL

Ansh Kumar Sharma[1] **Yixiang Sun**[1] **Ninghao Lu**[2] **Yunzhe Zhang**[1]
Jiarao Liu[1,3] **Sherry Yang**[1]
[1]New York University [2]NYU Shanghai [3]UC Berkeley
{as20482,sherryyang}@nyu.edu

ABSTRACT

Robot learning from interacting with the physical world is fundamentally bottlenecked by the cost of physical interaction. The two alternatives, supervised finetuning (SFT) from expert demonstrations and reinforcement learning (RL) in a software-based simulator, are limited by the amount of expert data available and the sim-to-real gap for manipulation. With the recent emergence of world models learned from real-world video-action data, we ask the question of whether training a policy in a world model can be more effective than supervised learning or software simulation in achieving better real-robot performance. We propose World-Gymnast, which performs RL finetuning of a vision-language-action (VLA) policy by rolling out the policy in an action-conditioned video world model and rewarding the rollouts with a vision-language model (VLM). On the Bridge robot setup, World-Gymnast outperforms SFT by as much as 18x and outperforms software simulator by as much as 2x. More importantly, World-Gymnast demonstrates intriguing capabilities of RL with a world model, including training on diverse language instructions and novel scenes from the world model, test-time training in a novel scene, and online iterative world model and policy improvement. Our results suggest learning a world model and training robot policies in the cloud could be the key to bridging the gap between robots that work in demonstrations and robots that can work in anyone's household.[1]

1 INTRODUCTION

Robots that learn by trial and error in the real world face an inherent constraint: physical interaction is expensive (Kormushev et al., 2013). Every policy update that depends on executing actions on hardware consumes operator time, risks wear-and-tear, and compounds safety concerns (Brunke et al., 2022), especially for manipulation, where failures are frequent early in learning. This cost creates a fundamental bottleneck for scaling robot learning from interaction. As a result, many real-robot systems rely on alternatives that reduce or replace on-robot exploration (Matas et al., 2018; Schaal, 1999).

One alternative is supervised finetuning (SFT) from expert demonstrations, where a robot is trained to imitate trajectories collected by teleoperation or scripted controllers (Ross et al., 2011). However, demonstration data tend to cover only a narrow slice of the long tail situations (Hu et al., 2024), and rarely expose the robot to the kinds of compounding errors and recovery behaviors needed for robust deployment (Lu et al., 2022). The second alternative is reinforcement learning (RL) in a software-based simulator (Zhao et al., 2020). However, software simulators are costly to create for every new scenarios. Furthermore, they often suffer from the sim-to-real gap where visual features differ from real-world images (Salvato et al., 2021).

Recent work have shown that world models learned from real-robot data can approximate real-robot execution outcomes (Yang et al., 2023; Quevedo et al., 2025; Guo et al., 2025; Tseng et al., 2025; Li et al., 2025b). These models aim to predict how the visual world evolves under the robot's actions, effectively serving as an action-conditioned video simulator learned from real-world data. Compared

[1]See videos and code at world-gymnast.github.io

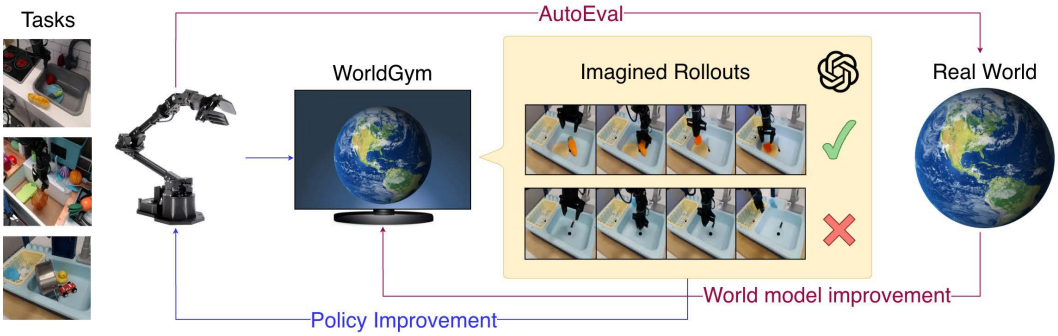

Figure 1: **Overview of World-Gymnast.** The policy is trained on tasks specified by an initial frame and language instruction. During training, the policy outputs actions which are then passed to the world model (WorldGym Quevedo et al. (2025)) which generates imagined rollouts. These rollouts are then passed to a VLM which returns a binary task completion reward. This reward is used to update the policy. Once trained, we evaluate the policy on real robots using the AutoEval (Zhou et al., 2025) setup. The resulting real world rollouts (frame-action sequences) from AutoEval can be further used to improve the world model on the particular environment.

to software simulators, video-based world models hold the promise of closing the visual gap and generalizing to a novel initial frame. However, it is unclear whether video world model offers more realistic physics than traditional simulators due to hallucinations (Yang et al., 2024). While evaluating physical realism is difficult, we instead tackle the end-to-end problem: does training robot policies inside a learned world model result in better real-robot performance than SFT or RL in a traditional simulator?

In this paper, we explore this question through the lens of large vision-language-action (VLA) policies that map images and language instructions to robot actions (Brohan et al., 2022; Kim et al., 2024; Black et al.). Specifically, we propose World-Gymnast, a training framework that performs RL fine-tuning a VLA policy using a world model (Figure 1). Concretely, World-Gymnast uses the action-conditioned video generation model similar to Quevedo et al. (2025) as its world model, enabling the policy to generate imagined rollouts conditioned on action sequences sampled from the VLA and uses a vision-language model (VLM) to compute rewards from predicted video frames. The resulting rewards are used to perform policy gradient updates to the VLA policy. More importantly, World-Gymnast opens up many intriguing possibilities of RL training with a world model, including (i) RL training from an arbitrary image frame, (ii) test-time training on a novel initial frame, and (iii) online iterative world model and policy improvement.

We evaluate World-Gymnast on the Bridge robot platform through AutoEval (Zhou et al., 2025), an automated real-robot evaluation platform open to public. Across a suite of manipulation tasks from AutoEval, we show World-Gymnast substantially outperforms SFT using the original Bridge data (Walke et al., 2023) and RL in SIMPLER (Li et al., 2024), a software simulator for Bridge created through real-to-sim techniques. Furthermore, since the world model only requires a single initial frame to perform rollouts, we demonstrate intriguing usage of the world model including training on novel language instructions and initial frames injected with distractor objects, test-time training from a real-robot frame, and iterative world model and policy improvement, all of which positively contribute to improved real-robot performance.

2 Preliminaries

In this section, we define notations and review model-based RL. We then discuss how foundation world models and vision language models can serve as general dynamics and reward models under the model-based RL formulation.

Markov Decision Process. We consider a multi-task, finite-horizon, partially observable Markov Decision Process (POMDP) (Puterman, 2014; Kaelbling et al., 1995), specified by $\mathcal{M} = (S, A, O, G, R, T, \mathcal{E}, H)$, which consists of state, action, observation, and task spaces, reward, tran-

sition, and emission functions, and horizon length. A policy π interacts with the environment for a task starting from an initial state $g, s_0 \sim G, o_0 \sim \mathcal{E}(s_0)$, producing a distribution $\pi(\cdot|o_t, g)$ over A from which an action a_t is sampled and applied to the environment at each step $t \in [0, H]$. The environment produces a scalar reward $r_t = R(s_t, g)$, and transitions to a new state $s_{t+1} \sim T(s_t, a_t)$ and emits a new observation $o_{t+1} \sim \mathcal{E}(s_{t+1})$.

The value of a policy π can be defined as the total expected future reward:

$$\begin{aligned}\rho(\pi) =& \mathbb{E}[R(s_H, g)|s_0, g \sim G, o_t \sim \mathcal{E}(s_t), a_t \sim \pi(o_t, g), \\ & s_{t+1} \sim T(s_t, a_t) \quad \forall t \in [0, H]].\end{aligned} \tag{1}$$

Model-Based RL with Foundation Models. RL (Sutton et al., 1998) aims to maximize $\rho(\pi)$ through trial-and-error interactions between the policy and the environment. Model-based RL (Doya et al., 2002) considers the setting where T and R are unknown and need to be estimated from samples from the environment, which can be an offline dataset logged from previous interactions $D = \{\tau_i = g, s_0, o_0, a_0, ..., s_H, o_H, r_H\}$. Motivated by characteristics of a real-world system such as image based observations and high control frequencies, the learned model $\hat{T}(\cdot|\mathbf{o}, \mathbf{a})$ can often take a sequence of previous image observations and a sequence of next actions. After $\hat{T}$ and $\hat{R}$ are estimated from data, a policy can perform rollout in the learned model

$$\begin{aligned}\hat{\rho}(\pi) =& \mathbb{E}[\hat{R}([o_0, ..., o_H], g)|s_0, g \sim G, \mathbf{a} \sim \pi(\mathbf{o}, g), \\ & \mathbf{o}' \sim \hat{T}(\mathbf{o}, \mathbf{a}), \mathbf{o} = \mathbf{o}'].\end{aligned} \tag{2}$$

Recent work has shown that $\hat{T}$ can be parametrized using an action-conditioned video generation model (world model) while $\hat{R}$ can be parametrized using a vision-language model (VLM).

Policy gradient methods (Williams, 1992) estimates the gradient of Equation 2 with respect to the policy π, and maximizes $\rho(\pi)$ directly via gradient ascent. The most commonly used gradient estimator has the form

$$\nabla_\theta \rho(\pi_\theta) = E_{\tau \sim \pi, T}\left[\sum_{t=0}^{H} \gamma^t \nabla_\theta \log \pi_\theta(o_t, g) \hat{A}(o_t, a_t)\right], \tag{3}$$

where $\hat{A}$ is some advantage function that can be separately estimated via Monte-Carlo returns from π, T, R (Williams, 1992). With model-based policy gradient, these advantages can be estimated from Monte-Carlo samples from $\pi, \hat{T}, \hat{R}$.

3 RL WITH A WORLD MODEL

In this section, we describe the RL algorithm World-Gymnast uses in Section 3.1. We then describe emerging training scenarios such as training on out-of-distribution (OOD) language and image in Section 3.2 and test-time training in Section 3.3. Lastly, we explain how World-Gymnast can be combined with classical algorithm such as Dyna Sutton (1991) to do online iterative world model and policy improvements.

3.1 MODEL-BASED GRPO WITH WORLD MODEL ROLLOUTS

To optimize the policy π_θ from equation 3, World-Gymnast uses the learned world model $\hat{T}$ from Quevedo et al. (2025). We adopt Group Relative Policy Optimization (GRPO) (Shao et al., 2024), a policy gradient algorithm that estimates $\hat{A}$ using group-based score normalization.

For a given task instruction g and an initial observation o_0, we generate a group of K independent trajectories $\{\tau_1, \ldots, \tau_K\}$ by rolling out the policy π_θ in the world model $\hat{T}$. Specifically, for the k-th trajectory, the policy samples an action $a_{t,k} \sim \pi_\theta(\cdot|o_{t,k}, g)$, and the world model predicts the next observation $o_{t+1,k} \sim \hat{T}(o_{t,k}, a_{t,k})$. This process repeats until the horizon H is reached, yielding a trajectory $\tau_k = (o_{0,k}, a_{0,k}, \ldots, o_{H,k})$. Once the rollouts are complete, we employ a VLM $\hat{R}$ to assign a binary task completion reward to each trajectory $r_k = \hat{R}(\tau_k, g)$. To compute the advantages, we treat the group of K outputs as a baseline. We compute the mean and standard deviation of the

rewards within the group:

$$\mu = \frac{1}{K}\sum_{k=1}^{K} r_k, \quad \sigma = \sqrt{\frac{1}{K-1}\sum_{k=1}^{K}(r_k-\mu)^2}. \tag{4}$$

The advantage for the k-th trajectory is then calculated via normalization:

$$\hat{A}_k = \frac{r_k - \mu}{\sigma + \epsilon}, \tag{5}$$

where ϵ is a small constant for numerical stability. We assign the trajectory-level advantage to every time step t within that trajectory. That is, $\hat{A}_{t,k} = \hat{A}_k$ for all $t \in [0, H-1]$. Finally, we optimize the policy π_θ using a PPO-style objective clipped based on the computed advantages. The loss function is defined as:

$$\mathcal{J}(\theta) = \mathbb{E}_{g,o_0\sim\mathcal{D}}\Bigg[\frac{1}{K}\sum_{k=1}^{K}\frac{1}{H}\sum_{t=0}^{H-1} \min\Big(r_{t,k}(\theta)\hat{A}_k, \text{clip}(r_{t,k}(\theta), 1-\epsilon_{low}, 1+\epsilon_{high})\hat{A}_k\Big)\Bigg], \tag{6}$$

where $r_{t,k}(\theta) = \frac{\pi_\theta(a_{t,k}|o_{t,k},g)}{\pi_{\theta_{old}}(a_{t,k}|o_{t,k},g)}$ denotes the probability ratio.

Following Li et al. (2025a), we adopt several tricks to stabilize training and improve exploration: (1) removing the KL penalty, (2) filtering groups with zero reward variance, (3) clipping higher in GRPO, and (4) sampling actions with higher temperature.

3.2 Diverse Training Scenarios in the World Model

A world model pretrained on diverse datasets allows us to generate diverse training configurations using only images and langauge instructions, avoiding the need for task-specific simulators. We explore varous possibilities in training a policy in diverse configurations enabled by World-Gymnast.

Training from Any Frame. We can train the policy with RL using any frames that are close enough to the world model's training distribution as the initial observation o_0, then rolling out the policy π, the world model $\hat{T}$, and the reward model $\hat{R}$ to provide learning signals on this initial configuration. This flexibility substantially increases the effective amount of training data available for RL in contrast to SFT which is bottlenecked by the amount of expert demonstrations. RL training from any frame also enables the policy to learn recovery behaviors, thereby improving the robustness of policies.

Training on Novel Language Instructions. Training can be further scaled by modifying language instructions associated with the same initial frame. For example, a VLM can propose plausible tasks for a given scene, which are then used as instructions for RL training. This enables the policy to be trained on new tasks and interact with objects previously present in the environment but not explicitly interacted with. Previous work in policy evaluation had shown that pretrained VLA policies often fail at following OOD langauge instructions (Quevedo et al., 2025). We can overcome these limitations of VLAs with RL post-training in a world model.

Training with Distractions. To improve policy's robustness to irrelevant visual clutter, we leverage image editing tools like Nano Banana (Google, 2025) to synthesize additional objects as distractors in the input image frames. Training the policy on diverse distractor objects encourages the policy to be more robust when such distractor objects are present during actual deployment and to have better performance in cluttered scenes. This can bridge the gap between robots that work in demos and robots that can work in anyone's messy household.

3.3 Test-Time Training from a Novel Frame

Because World-Gymnast allows a policy to rollout from just an initial frame, when a novel frame is presented to a policy at test time, the policy can trade-off compute for improved policy performance

by running RL training in the world model starting from the test frame. This allows rapid adaptation of the policy to novel scenes while avoiding the cost and risk of collecting real-world interaction data.

3.4 Iterative World Model and Policy Improvement

When the visual observations encountered during policy rollouts deviate too much from the original training distribution of the world model, directly rolling out the policy in the world model might lead to compounding modeling errors. Inspired by classical Dyna-style algorithms (Sutton, 1991), World-Gymnast allows an iterative training procedure in which the policy and world model are alternately refined. Specifically, the current policy can be rolled out (with inference-time scaling or test-time training using the world model as a reward function) to collect new environment interactions, which are then incorporated to further fine-tune the world model. The updated world model is subsequently used to generate improved imagined rollouts for policy optimization. This data flywheel enables the world model to progressively adapt to the policy-induced state distribution, while allowing the policy to benefit from increasingly accurate long-horizon predictions from the world model.

4 Experiments

We now evaluate the performance of policies trained in World-Gymnast. We explain the experimental setup in Section 4.1, followed by comparisons to policies trained with software simulators and SFT in Section 4.2. We then demonstrate the capabilities of World-Gymnast in supporting diverse training from images with distractors, novel language instructions, and scaling the number of RL tasks in Section 4.3. Finally, we evaluate test-time scaling and iterative policy and world model improvement in Section 4.4 and Section 4.5.

4.1 Experimental Setup

Tasks and Pipeline. We evaluate the efficacy of World-Gymnast using the curated evaluation dataset used in Kim et al. (2024). The dataset follows the BridgeData V2 (Walke et al., 2023) setup with the WidowX robot and is designed to test policy generalization across visual, motion, physical, and semantic variations, as well as language grounding, over 17 tasks (Appendix C).

We further leverage the data scaling capabilities enabled by World-Gymnast, such as image editing, language augmentation, and novel task setups, to improve generalization of the trained RL policy. To train with World-Gymnast, a task just requires an initial frame and language instruction. During training, World-Gymnast rolls out the policy for up to 40 steps in WorldGym and a binary task completion reward is assigned to the rollout by GPT-4o (Hurst et al., 2024).

Post training, we evaluate the policy on WorldGym (Quevedo et al., 2025) to estimate real robot performance and ensure policy is safe for testing, following its default configuration unless otherwise specified (Appendix A.4). We finally run the policy on AutoEval (Zhou et al., 2025), a real-robot evaluation framework that currently supports 4 tasks across 2 setups. AutoEval evaluates each policy–task pair over 10 trials; we repeat this evaluation 5 times to estimate the standard error.

Base Models. Successful RL finetuning requires a reasonably competent initial policy. To this end, we use OpenVLA-OFT (Kim et al., 2025) as our base model. OpenVLA-OFT provides an optimized finetuning recipe to build on top of OpenVLA (Kim et al., 2024) which was originally trained on Open X-Embodiment dataset (Vuong et al., 2023). We use the BridgeData V2 (Walke et al., 2023) to finetune our base model. Following the idea from Li et al. (2025a), we made several modifications to the official implementation of OpenVLA-OFT: 1) Disable the proprioception and secondary camera inputs to match the observation space used by our policy, 2) Use LLAMA-2 (Touvron et al., 2023) LM head as action head instead of the default L1 loss one to get action probabilities essential for RL. For WorldGym, we used a 600M parameter variant pretrained on Open X-Embodiment dataset.

Training Details. For RL training, we use 4 NVIDIA H200 GPUs (140GB each) for full-parameter finetuning over 1–2 days. We use the following training parameters: learning rate $5 \cdot 10^{-6}$,

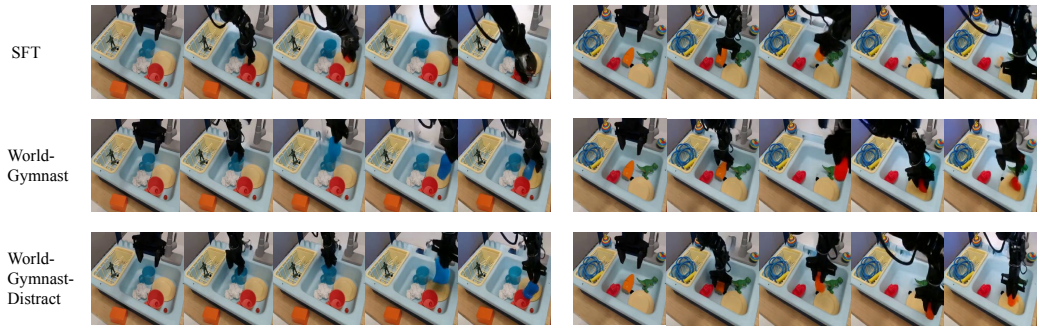

Figure 2: **Qualitative evaluation of policy rollouts in WorldGym with distractors.** We compare rollout quality among SFT, World-Gymnast and World-Gymnast-Distract under visual distractions. The task on the left is *put blue cup on plate* and the SFT policy clearly picks up the wrong cup, while both World-Gymnast variants are able to correctly execute the task. On the right task (*put carrot on plate*), we can see SFT struggle again and seems to grab the dinosaur along with the carrot. Both World-Gymnast variants are again successful but World-Gymnast-Distract has better grasping and placing movements. It is worth noting that even with the visual artifacts introduced by the imperfect world model, the policies transfer effectively to the real robot setting.

group size 8, size of training batch 20, length of action chunk 5, clip ratio ($\epsilon_{high} = 0.28$, $\epsilon_{low} = 0.2$), temperature 1.6. More detailed hyperparameter setup is available in Appendix A.4.

4.2 Evaluating RL with World-Gymnast

Comparing World-Gymnast to a Software Simulator. We compare World-Gymnast against traditional simulator-based RL. We select SIMPLER (Li et al., 2024), a real-to-sim policy evaluation framework, as our baseline since it provides the closest simulator-based approximation to the Bridge robot and tasks used in our evaluation. We train on all available SIMPLER tasks (Appendix B.2) and further include digital twins for the AutoEval setup. When initializing RL from the SFT policy, we observed low task success in SIMPLER for all tasks except *close the drawer*. Although the policy often moved in the correct direction, it failed to complete the task. Consequently, binary completion rewards led to collapsed reward variance, preventing effective policy updates. To overcome this problem, we define the reward for a rollout as the sum of partial rewards from each step. We share more details of reward design in Appendix B.2.

Task	SIMPLER	World-Gymnast
Open the drawer	$34 \pm 7\%$	$58 \pm 4\%$
Close the drawer	$74 \pm 5\%$	$62 \pm 6\%$
Put eggplant in blue sink	$32 \pm 10\%$	$72 \pm 10\%$
Put eggplant in yellow basket	$40 \pm 10\%$	$78 \pm 2\%$

Table 1: **Real-robot success rate from AutoEval** (Zhou et al., 2025) of World-Gymnast compared to running RL in a software simulator SIMPLER (Li et al., 2024). RL with a world model significantly outperforms RL in a simulator in terms of real-robot success for 3 out of the 4 tasks.

World-Gymnast outperformed training with SIMPLER on all tasks except *close the drawer*, as shown in Table 1. For *close the drawer* task, the base policy has already performed pretty well prior to RL. It is worth noting that the RL training set for World-Gymnast does not include these tasks in the training set, whereas the SIMPLER baseline was also trained on these exact environment-task setups (the digital twins) and yet the policy exhibited poor transfer to real world.

Comparing World-Gymnast to Supervised Learning. We also compare World-Gymnast with supervised fine-tuning methods. Following the recipe of OpenVLA-OFT Kim et al. (2025), we fine-tune a OpenVLA 7B policy Kim et al. (2024) on expert trajectories from the Bridge V2 dataset Walke et al. (2023) for 20k steps. This policy, denoted as *SFT* in Table 2, is also the base model on which we conduct RL training. Recent works like Ctrl-World Guo et al. (2025) further utilize the world model for policy improvement by generating synthesized roll-outs filtered by a reward model

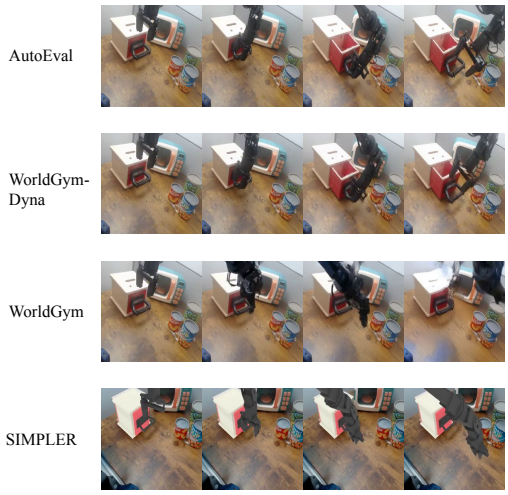

Figure 3: **Qualitative comparison** of rolling out the same action sequence on the real robot from AutoEval (Zhou et al., 2025), from software simulator SIMPLER (Li et al., 2024), from WorldGym (Quevedo et al., 2025), and from World-Gymnast with online world model updates. Rollouts from World-Gymnast adheres more closely to the real world than SIMPLER, suggesting Dyna (Sutton, 1991) style world model updates improve rollout quality.

Task	SFT	Iter-SFT	World-Gymnast
Open drawer	$40 \pm 10\%$	$30 \pm 10\%$	$58 \pm 4\%$
Close drawer	$62 \pm 6\%$	$60 \pm 8\%$	$62 \pm 6\%$
Eggplant $\rightarrow$ sink	$4 \pm 4\%$	$10 \pm 3\%$	$72 \pm 10\%$
Eggplant $\rightarrow$ basket	$8 \pm 4\%$	$17 \pm 17\%$	$78 \pm 2\%$

Table 2: **Real-robot task success rate** of World-Gymnast and supervised learning approaches. Standard errors are calculated between groups of 10 consecutive roll-outs.

Variant	Success Rate
SFT	$58 \pm 4\%$
World-Gymnast	$74 \pm 3\%$
World-Gymnast-Distract	$78 \pm 2\%$
World-Gymnast-Language	$81 \pm 1\%$
World-Gymnast-Scaled	$81 \pm 4\%$

Table 3: **Comparing RL on diverse settings.** Leveraging the diverse capabilities of world modeling, World-Gymnast allows significant improvement in the task success rates for multiple settings when evaluated in the world model. Evaluation task set is discussed in Appendix C

as additional supervision. Similar to Ctrl-World, we roll out the base *SFT* policy in our world model for the same amount of steps as World-Gymnast did for RL training, and filtered for successful trajectories on OpenVLA evaluation tasks using a VLM. We then conduct another iteration of supervised fine-tuning with a mixture of data from Bridge V2 plus the successful synthesized roll-outs with a $1:1$ sampling rate. The resulting policy, denoted *Iter-SFT* in Table 2, is then evaluated on real world held-out tasks through AutoEval.

As shown in Table 2, World-Gymnast achieves the best performance compared with supervised learning and Iter-SFT, with a 18-fold and nearly 10-fold improvements from the base policy on *Put the eggplant into the blue sink* and *Put the eggplant into the yellow basket*, respectively. Notably, Iter-SFT improves slightly on the harder tasks, but the performance degrades on the easier ones. One possible explanation is that RL, through active exploration and on-policy updates, learns more generalizable behaviors. In contrast, iterative SFT may overfit to synthetic experience and is more vulnerable to world model hallucinations and inaccurate VLM success judgments.

4.3 Evaluating Diverse Settings World-Gymnast Offers

Evaluating Training with Distractors. We use Nano Banana (Google, 2025) to generate a new dataset using the pre-existing frames from the OpenVLA Bridge task suite. The new dataset adds random objects to distract the policy from successfully achieving the given task. We then train a new policy with RL (World-Gymnast-Distract) by including these new frames in the training data. Next, we evaluate the performance of SFT, World-Gymnast and World-Gymnast-Distract on a held-out set of distractor frames. WorldGym evaluations show World-Gymnast-Distract is the most robust, while SFT is the easiest to distract (Figure 2). Additionally, we evaluate World-Gymnast-Distract on the original OpenVLA tasks in WorldGym and observe improved success rates (Table 3). This indicates that adding distractor-augmented data improves performance not only under visual perturbations but also on the original tasks. Qualitative rollout comparisons in Figure 2 further illustrate that under visual distractions, World-Gymnast-Distract executes more reliable grasping and placement

behaviors than SFT and World-Gymnast, while maintaining correct object grounding despite visual artifacts from the world model.

Evaluating Training with Novel Language Instructions. Another approach to scaling data is augmenting language instructions in pre-existing tasks. We test this approach by creating 4 new tasks involving new interactions with objects already present in the scene. We combine this new data with the OpenVLA dataset and train a new policy, World-Gymnast-Language. Next we evaluate the performance of World-Gymnast-Language on the held-out split from OpenVLA data and observe that World-Gymnast-Language has improved performance over World-Gymnast (Table 3). This suggests that creating more tasks by introducing novel language instructions on existing frames can further improve the generalization performance of VLA policies.

Scaling the Number of Training Tasks. One advantage of World-Gymnast is its ability to train on a diverse set of tasks starting from any initial frame. To scale up data, we randomly selected 5 additional tasks from the Bridge dataset (Walke et al., 2023). We then train on these tasks in addition to the OpenVLA tasks and call this variant World-Gymnast-Scaled. We evaluate the performance of World-Gymnast-Scaled in WorldGym on the OpenVLA held-out task split and observed improvement compared to World-Gymnast, as shown in Table 3. These results suggest that World-Gymnast can effectively leverage additional training tasks to improve performance.

4.4 Evaluating Test-Time Optimization

Pretrained policies often struggle at generalizing well to novel real world scenarios. While online data collection followed by finetuning can address this gap, it is prohibitively expensive in terms of time and effort. With a pretrained world model, we show that World-Gymnast improves the performance of a base policy through test-time training without real world roll-outs. Specifically, provided only with the initial observation and the task instructions of the 4 scenarios from AutoEval (Zhou et al., 2025), we fine-tune our base policy with RL (details in Appendix A.4) using imagined roll-outs generated by the world model in a zero-shot manner from the testing frame. Test time training significantly improves the performance and robustness of *close the drawer* in the real world, improving the success rate from $62 \pm 6\%$ success rate to $100 \pm 0\%$ for the *Close the drawer* task.

4.5 Evaluating Iterative World and Policy Improvement

A unique advantage of World-Gymnast is that the world model can be iteratively updated with real world roll-outs for initial frames that are out-of-distribution of the pretrained world model. During policy evaluation with AutoEval (Zhou et al., 2025), we save the task instructions, observations, and action sequences to improve our world model. Iteratively, we collect around 100 trajectories per task, on which we finetune the world model for a total of 120k steps. Figure 3 provides a qualitative result demonstrating the improvement of the world model in playing a sequence of actions for *Open the drawer* task, which shows much less sim-to-real gap than SIMPLER (Li et al., 2024) and than WorldGym without online updates. Using the test-time-adapted Dyna style world model as an RL environment, World-Gymnast further improves its vanilla version for *close the drawer* in AutoEval by 53%, with a 95% success rate over 40 roll-outs.

5 Related Work

Model-Based Reinforcement Learning. Model-based RL has long been studied in the RL literature, which learns a dynamics model from previously collected data and rolling out the learned dynamics model for policy evaluation and improvement (Sutton, 1991; Tani, 1996; Ljung & Glad, 1994; Liu et al., 2019; Zhang et al., 2021; Yu et al., 2020; Hafner et al., 2020). Much of the model-based RL research has been focusing on learning one dynamics model per system in the lower dimensional state space as opposed to in the pixel space (Ferns et al., 2004; Achille & Soatto, 2018; Lesort et al., 2018; Castro, 2020), which, despite being a simpler modeling problem, limits knowledge sharing across systems. With large transformer architectures, learning image-based world models followed by RL has become plausible (Hafner et al., 2020; Chen et al., 2022; Seo et al., 2022; Micheli et al., 2022; Wu et al., 2022; Hafner et al., 2023), but mostly in games or simulated domains with visually simplistic and abundant data. Our work differs from existing model-based RL

in that we focus on using a single world model learned on broad data from many policies and tasks to train a generalist VLA policy on novel language instructions and scenes. We also directly tackle the sim-to-real gap by comparing against RL in traditional simulators, further showing the value of model-based RL where the model is learned from real-world data.

Sim-to-Real RL. Reinforcement learning in physics-based simulations has been widely adopted to overcome the sample inefficiency of real-world training. To bridge the reality gap, prior works have relied on domain randomization (Tobin et al., 2017; Peng et al., 2018), which varies visual and physical parameters to cover real-world distributions, or domain adaptation (Bousmalis et al., 2018; Rao et al., 2020). These strategies have achieved notable success in locomotion (Tan et al., 2018; Rudin et al., 2022) and rigid-body manipulation (OpenAI et al., 2019; Handa et al., 2024). However, traditional simulators (Todorov et al., 2012; Makoviychuk et al., 2021) face a scalability bottleneck for generalist manipulation: they require explicit object modeling, manual scene engineering, and struggle to faithfully render the diverse visual textures and deformable dynamics of the real world. Unlike these approaches, World-Gymnast leverages a world model learned directly from real-world data, effectively bypassing the need for manual asset creation and physics parameter tuning.

Video Generation for Robot Learning. Video-based learning for robotics (Nair et al., 2022; Bahl et al., 2022; Shao et al., 2021; Chen et al., 2021; Pari et al., 2022) has enabled visual representation learning, goal extraction, planning (Finn & Levine, 2017; Kurutach et al., 2018), and imitation from expert actions (Fang et al., 2019; Wang et al., 2023; Mani et al., 2024). Recent works reframe decision-making as a text conditioned video generation task, enabling policy learning from video predictions (Du et al., 2024; Ko et al., 2023; Wen et al., 2023; Du et al., 2023; Ajay et al., 2024), and use generative models to simulate agent-environment interactions (Yang et al., 2023). Most of these works use generated video plans as visual actions and train separate inverse dynamics to extract robot actions from generated videos. While text-to-video generation can be effective for long-horizon planning, it is less clear how to self-improve these models that use video-generation as policies. We study the problem of using video generation solely as environment and using RL with generated rollouts to improve policy performance. Notably, World-Gymnast can in principle improve policies beyond VLAs, including those parameterized by video generation.

Policy Evaluation using World Models. Recent work have shown that world models learned from real-robot data can approximate real-robot execution outcomes, and hence be used to evaluate robot policies (Quevedo et al., 2025; Guo et al., 2025; Tseng et al., 2025; Li et al., 2025b). While evaluating policies in a world model is by all means an important application of world models, we focus on the end-to-end problem of improving policy performance using a world model, the effect of which can be tested on real robot.

RL with a Video Based World Model. Our work is the most similar to Zhu et al. (2025) which uses RL to improve a policy in a video world model, but we focus on real-world evaluation with easily accessible evaluation settings using a open-source VLA policy (OpenVLA), world model (WorldGym), and evaluation platform (AutoEval). We also focus on exploring the emergent capabilities of RL in a world model, including training from any initial frame with novel language instructions, test-time training, and iterative world model and policy improvement.

6 Conclusion and Limitations

We have presented World-Gymnast, an RL framework for fine-tuning VLA policies using a learned world model. We show that existing training paradigms, including SFT and software-simulator-based RL, are expensive, restrictive, and often produce policies with limited generalization. In contrast, using a world model to rollout policies and VLM for reward is cheap, scalable and more robust to OOD scenarios.

A key advantage of world-model-based training is the ability to generate diverse training data from minimal inputs. Since World-Gymnast requires only an initial scene and a task description, it naturally supports data augmentation through image editing, language variation, and the reuse of scenes from novel environments. This flexibility further enables test-time training, planning, and iterative improvement of both the policy and the world model.

Limitations. One limitation of World-Gymnast is that it cannot generalize to an arbitrary initial frame if the initial frame is far from the world model's training distribution. This calls for future research of pretraining robot world models on broad robot datasets. Another limitation of World-Gymnast is the reliance on a pretrained VLM for task success, which can produce hallucinations that leads to suboptimal RL training. Exploring ways to improve the reward model, such as training reward models (Lee et al., 2026), is an important future direction. Furthermore, utilizing dense rewards from VLMs and preventing reward hacking are also promising directions for future work.

REFERENCES

Alessandro Achille and Stefano Soatto. A separation principle for control in the age of deep learning. *Annual Review of Control, Robotics, and Autonomous Systems*, 1:287–307, 2018.

Anurag Ajay, Seungwook Han, Yilun Du, Shuang Li, Abhi Gupta, Tommi Jaakkola, Josh Tenenbaum, Leslie Kaelbling, Akash Srivastava, and Pulkit Agrawal. Compositional foundation models for hierarchical planning. *Advances in Neural Information Processing Systems*, 36, 2024.

Shikhar Bahl, Abhinav Gupta, and Deepak Pathak. Human-to-robot imitation in the wild. In *Robotics: Science and Systems*, 2022.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. $\pi 0$: A vision-language-action flow model for general robot control. corr, abs/2410.24164, 2024. doi: 10.48550. *arXiv preprint ARXIV.2410.24164*.

Konstantinos Bousmalis, Alex Irpan, Paul Wohlhart, Yunfei Bai, Matthew Kelcey, Mrinal Kalakrishnan, Laura Downs, Julian Ibarz, Peter Pastor, Kurt Konolige, et al. Using simulation and domain adaptation to improve efficiency of deep robotic grasping. In *2018 IEEE international conference on robotics and automation (ICRA)*, pp. 4243–4250. IEEE, 2018.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022.

Lukas Brunke, Melissa Greeff, Adam W Hall, Zhaocong Yuan, Siqi Zhou, Jacopo Panerati, and Angela P Schoellig. Safe learning in robotics: From learning-based control to safe reinforcement learning. *Annual Review of Control, Robotics, and Autonomous Systems*, 5(1):411–444, 2022.

Pablo Samuel Castro. Scalable methods for computing state similarity in deterministic markov decision processes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 10069–10076, 2020.

Annie S Chen, Suraj Nair, and Chelsea Finn. Learning Generalizable Robotic Reward Functions from "In-The-Wild" Human Videos. In *Robotics: Science and Systems*, 2021.

Chang Chen, Yi-Fu Wu, Jaesik Yoon, and Sungjin Ahn. Transdreamer: Reinforcement learning with transformer world models. *arXiv preprint arXiv:2202.09481*, 2022.

Kenji Doya, Kazuyuki Samejima, Ken-ichi Katagiri, and Mitsuo Kawato. Multiple model-based reinforcement learning. *Neural computation*, 14(6):1347–1369, 2002.

Yilun Du, Mengjiao Yang, Pete Florence, Fei Xia, Ayzaan Wahid, Brian Ichter, Pierre Sermanet, Tianhe Yu, Pieter Abbeel, Joshua B Tenenbaum, et al. Video language planning. *arXiv preprint arXiv:2310.10625*, 2023.

Yilun Du, Sherry Yang, Bo Dai, Hanjun Dai, Ofir Nachum, Josh Tenenbaum, Dale Schuurmans, and Pieter Abbeel. Learning universal policies via text-guided video generation. *Advances in Neural Information Processing Systems*, 36, 2024.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024.

Bin Fang, Shidong Jia, Di Guo, Muhua Xu, Shuhuan Wen, and Fuchun Sun. Survey of Imitation Learning for Robotic Manipulation. *International Journal of Intelligent Robotics and Applications*, 2019.

Norm Ferns, Prakash Panangaden, and Doina Precup. Metrics for finite markov decision processes. In *UAI*, volume 4, pp. 162–169, 2004.

Chelsea Finn and Sergey Levine. Deep Visual Foresight for Planning Robot Motion. In *IEEE International Conference on Robotics and Automation*, 2017.

Google. Image editing in gemini just got a major upgrade. Blog post on "The Keyword", Google, August 26 2025. URL https://blog.google/products/gemini/updated-image-editing-model/. Multimodal Generation Lead, Gemini Apps; Gemini Image Product Lead, Google DeepMind.

Yanjiang Guo, Lucy Xiaoyang Shi, Jianyu Chen, and Chelsea Finn. Ctrl-world: A controllable generative world model for robot manipulation. *arXiv preprint arXiv:2510.10125*, 2025.

Danijar Hafner, Timothy Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering atari with discrete world models. *arXiv preprint arXiv:2010.02193*, 2020.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*, 2023.

Ankur Handa, Arthur Allshire, Viktor Makoviychuk, Aleksei Petrenko, Ritvik Singh, Jingzhou Liu, Denys Makoviichuk, Karl Van Wyk, Alexander Zhurkevich, Balakumar Sundaralingam, Yashraj Narang, Jean-Francois Lafleche, Dieter Fox, and Gavriel State. Dextreme: Transfer of agile in-hand manipulation from simulation to reality, 2024. URL https://arxiv.org/abs/2210.13702.

Yingdong Hu, Fanqi Lin, Pingyue Sheng, Chuan Wen, Jiacheng You, and Yang Gao. Data scaling laws in imitation learning for robotic manipulation. *arXiv preprint arXiv:2410.18647*, 2024.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

Leslie Pack Kaelbling, Michael L Littman, and Anthony R Cassandra. Partially observable markov decision processes for artificial intelligence. In *International Workshop on Reasoning with Uncertainty in Robotics*, pp. 146–163. Springer, 1995.

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

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

Po-Chen Ko, Jiayuan Mao, Yilun Du, Shao-Hua Sun, and Joshua B Tenenbaum. Learning to act from actionless videos through dense correspondences. *arXiv preprint arXiv:2310.08576*, 2023.

Petar Kormushev, Sylvain Calinon, and Darwin G Caldwell. Reinforcement learning in robotics: Applications and real-world challenges. *Robotics*, 2(3):122–148, 2013.

Thanard Kurutach, Aviv Tamar, Ge Yang, Stuart J Russell, and Pieter Abbeel. Learning Plannable Representations with Causal InfoGAN. In *Neural Information Processing Systems*, 2018.

Tony Lee, Andrew Wagenmaker, Karl Pertsch, Percy Liang, Sergey Levine, and Chelsea Finn. Roboreward: General-purpose vision-language reward models for robotics, 2026. URL https://arxiv.org/abs/2601.00675.

Timothée Lesort, Natalia Díaz-Rodríguez, Jean-Franois Goudou, and David Filliat. State representation learning for control: An overview. *Neural Networks*, 108:379–392, 2018.

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

Xuanlin Li, Kyle Hsu, Jiayuan Gu, Karl Pertsch, Oier Mees, Homer Rich Walke, Chuyuan Fu, Ishikaa Lunawat, Isabel Sieh, Sean Kirmani, et al. Evaluating real-world robot manipulation policies in simulation. *arXiv preprint arXiv:2405.05941*, 2024.

Yaxuan Li, Yichen Zhu, Junjie Wen, Chaomin Shen, and Yi Xu. Worldeval: World model as real-world robot policies evaluator. *arXiv preprint arXiv:2505.19017*, 2025b.

Jingbin Liu, Xinyang Gu, and Shuai Liu. Reinforcement learning with world model. *arXiv preprint arXiv:1908.11494*, 2019.

Lennart Ljung and Torkel Glad. *Modeling of dynamic systems*. Prentice-Hall, Inc., 1994.

Cong Lu, Philip J Ball, Tim GJ Rudner, Jack Parker-Holder, Michael A Osborne, and Yee Whye Teh. Challenges and opportunities in offline reinforcement learning from visual observations. *arXiv preprint arXiv:2206.04779*, 2022.

Viktor Makoviychuk, Lukasz Wawrzyniak, Yunrong Guo, Michelle Lu, Kier Storey, Miles Macklin, David Hoeller, Nikita Rudin, Arthur Allshire, Ankur Handa, and Gavriel State. Isaac gym: High performance gpu-based physics simulation for robot learning, 2021. URL https://arxiv.org/abs/2108.10470.

Sabariswaran Mani, Sreyas Venkataraman, Abhranil Chandra, Adyan Rizvi, Yash Sirvi, Soumojit Bhattacharya, and Aritra Hazra. Diffclone: Enhanced behaviour cloning in robotics with diffusion-driven policy learning. *arXiv:2401.09243*, 2024.

Jan Matas, Stephen James, and Andrew J Davison. Sim-to-real reinforcement learning for deformable object manipulation. In *Conference on Robot Learning*, pp. 734–743. PMLR, 2018.

Vincent Micheli, Eloi Alonso, and François Fleuret. Transformers are sample efficient world models. *arXiv preprint arXiv:2209.00588*, 2022.

Suraj Nair, Aravind Rajeswaran, Vikash Kumar, Chelsea Finn, and Abhinav Gupta. R3M: A Universal Visual Representation for Robot Manipulation. In *Conference on Robot Learning*, 2022.

OpenAI, Ilge Akkaya, Marcin Andrychowicz, Maciek Chociej, Mateusz Litwin, Bob McGrew, Arthur Petron, Alex Paino, Matthias Plappert, Glenn Powell, Raphael Ribas, Jonas Schneider, Nikolas Tezak, Jerry Tworek, Peter Welinder, Lilian Weng, Qiming Yuan, Wojciech Zaremba, and Lei Zhang. Solving rubik's cube with a robot hand, 2019. URL https://arxiv.org/abs/1910.07113.

Jyothish Pari, Nur Muhammad Shafiullah, Sridhar Pandian Arunachalam, and Lerrel Pinto. The Surprising Effectiveness of Representation Learning for Visual Imitation. In *Robotics: Science and Systems*, 2022.

Xue Bin Peng, Marcin Andrychowicz, Wojciech Zaremba, and Pieter Abbeel. Sim-to-real transfer of robotic control with dynamics randomization. In *2018 IEEE international conference on robotics and automation (ICRA)*, pp. 3803–3810. IEEE, 2018.

Reiner Pope, Sholto Douglas, Aakanksha Chowdhery, Jacob Devlin, James Bradbury, Jonathan Heek, Kefan Xiao, Shivani Agrawal, and Jeff Dean. Efficiently scaling transformer inference. *Proceedings of machine learning and systems*, 5:606–624, 2023.

Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

Julian Quevedo, Ansh Kumar Sharma, Yixiang Sun, Varad Suryavanshi, Percy Liang, and Sherry Yang. Worldgym: World model as an environment for policy evaluation, 2025. URL https://arxiv.org/abs/2506.00613.

Kanishka Rao, Chris Harris, Alex Irpan, Sergey Levine, Julian Ibarz, and Mohi Khansari. Rl-cyclegan: Reinforcement learning aware simulation-to-real. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11157–11166, 2020.

Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 627–635. JMLR Workshop and Conference Proceedings, 2011.

Nikita Rudin, David Hoeller, Philipp Reist, and Marco Hutter. Learning to walk in minutes using massively parallel deep reinforcement learning, 2022. URL https://arxiv.org/abs/2109.11978.

Erica Salvato, Gianfranco Fenu, Eric Medvet, and Felice Andrea Pellegrino. Crossing the reality gap: A survey on sim-to-real transferability of robot controllers in reinforcement learning. *IEEE Access*, 9:153171–153187, 2021.

Stefan Schaal. Is imitation learning the route to humanoid robots? *Trends in cognitive sciences*, 3 (6):233–242, 1999.

Younggyo Seo, Kimin Lee, Stephen L James, and Pieter Abbeel. Reinforcement learning with action-free pre-training from videos. In *International Conference on Machine Learning*, pp. 19561–19579. PMLR, 2022.

Lin Shao, Toki Migimatsu, Qiang Zhang, Karen Yang, and Jeannette Bohg. Concept2Robot: Learning Manipulation Concepts from Instructions and Human Demonstrations. *IJRR*, 2021.

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

Richard S Sutton. Dyna, an integrated architecture for learning, planning, and reacting. *ACM Sigart Bulletin*, 2(4):160–163, 1991.

Richard S Sutton, Andrew G Barto, et al. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.

Jie Tan, Tingnan Zhang, Erwin Coumans, Atil Iscen, Yunfei Bai, Danijar Hafner, Steven Bohez, and Vincent Vanhoucke. Sim-to-real: Learning agile locomotion for quadruped robots, 2018. URL https://arxiv.org/abs/1804.10332.

Jun Tani. Model-based learning for mobile robot navigation from the dynamical systems perspective. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 26(3):421–436, 1996.

Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pp. 23–30. IEEE, 2017.

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *IROS* Todorov et al. (2012), pp. 5026–5033. ISBN 978-1-4673-1737-5. URL http://dblp.uni-trier.de/db/conf/iros/iros2012.html#TodorovET12.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Wei-Cheng Tseng, Jinwei Gu, Qinsheng Zhang, Hanzi Mao, Ming-Yu Liu, Florian Shkurti, and Lin Yen-Chen. Scalable policy evaluation with video world models. *arXiv preprint arXiv:2511.11520*, 2025.

Quan Vuong, Sergey Levine, Homer Rich Walke, Karl Pertsch, Anikait Singh, Ria Doshi, Charles Xu, Jianlan Luo, Liam Tan, Dhruv Shah, et al. Open x-embodiment: Robotic learning datasets and rt-x models. In *Towards Generalist Robots: Learning Paradigms for Scalable Skill Acquisition@ CoRL2023*, 2023.

Homer Rich Walke, Kevin Black, Tony Z Zhao, Quan Vuong, Chongyi Zheng, Philippe Hansen-Estruch, Andre Wang He, Vivek Myers, Moo Jin Kim, Max Du, et al. Bridgedata v2: A dataset for robot learning at scale. In *Conference on Robot Learning*, pp. 1723–1736. PMLR, 2023.

Hsiang-Chun Wang, Shang-Fu Chen, and Shao-Hua Sun. Diffusion Model-Augmented Behavioral Cloning. *arXiv:2302.13335*, 2023.

Chuan Wen, Xingyu Lin, John So, Kai Chen, Qi Dou, Yang Gao, and Pieter Abbeel. Any-point trajectory modeling for policy learning. *arXiv preprint arXiv:2401.00025*, 2023.

Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3):229–256, 1992.

Ziyi Wu, Nikita Dvornik, Klaus Greff, Thomas Kipf, and Animesh Garg. Slotformer: Unsupervised visual dynamics simulation with object-centric models. *arXiv preprint arXiv:2210.05861*, 2022.

Mengjiao Yang, Yilun Du, Kamyar Ghasemipour, Jonathan Tompson, Dale Schuurmans, and Pieter Abbeel. Learning interactive real-world simulators. *arXiv preprint arXiv:2310.06114*, 2023.

Sherry Yang, Jacob Walker, Jack Parker-Holder, Yilun Du, Jake Bruce, Andre Barreto, Pieter Abbeel, and Dale Schuurmans. Video as the new language for real-world decision making. *arXiv preprint arXiv:2402.17139*, 2024.

Tianhe Yu, Garrett Thomas, Lantao Yu, Stefano Ermon, James Zou, Sergey Levine, Chelsea Finn, and Tengyu Ma. MOPO: Model-based offline policy optimization. In *Advances in Neural Information Processing Systems*, 2020.

Michael R Zhang, Thomas Paine, Ofir Nachum, Cosmin Paduraru, George Tucker, ziyu wang, and Mohammad Norouzi. Autoregressive dynamics models for offline policy evaluation and optimization. In *International Conference on Learning Representations*, 2021.

Wenshuai Zhao, Jorge Peña Queralta, and Tomi Westerlund. Sim-to-real transfer in deep reinforcement learning for robotics: a survey. In *2020 IEEE symposium series on computational intelligence (SSCI)*, pp. 737–744. IEEE, 2020.

Zhiyuan Zhou, Pranav Atreya, You Liang Tan, Karl Pertsch, and Sergey Levine. Autoeval: Autonomous evaluation of generalist robot manipulation policies in the real world. *arXiv preprint arXiv:2503.24278*, 2025.

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

Algorithm 1 WorldGym inference with temporal KV caching.

```
Init:
    x ← x_0, curr_frame ← 0, chunk = 1, max_frames                ▷ key variables
    wm ← WorldModel(checkpoint, use_kv_cache = True)
    wm.reset(x_0)                                                  ▷ clears KV cache
for each action a_t do
    wm.clear_kv_cache()                                            ▷ per-chunk cache
    append noisy latent to history
    start ← max(0, curr_frame + chunk − max_frames)
    for each diffusion step do
        cache_idx ← boundary_of_clean_frames(start)                ▷ t=0 window
        v_cond ← DiT(x, t, a_t, cache_idx, start, cache = ``cond'')
        v_null ← DiT(x, t, null(a_t), cache_idx, start, cache = ``null'')
        v ← cfg(v_cond, v_null)
        x ← ddim_update(x, v, maybe_cache_suffix)
    end for
    decode latest clean frame(s)
    curr_frame ← curr_frame + chunk
end for
```

A Additional Details of The Models and Training

A.1 Fast Inference on World Model

A major bottleneck in the RL training pipeline is the world model inference latency. Without optimization, temporal attention layers recompute attention over the entire frame history at every step. We modify the WorldGym Quevedo et al. (2025) architecture for fast inference by introducing Key-Value (KV) caching (Pope et al., 2023) as described in Algorithm 1. This change substantially reduces per-step latency especially with long context, enabling long-horizon RL training that would otherwise be infeasible. In practice, we observe a 10× reduction in rollout time when generating 20 trajectories in parallel over 40 frames.

A.2 OpenVLA-OFT as a Base Model

For all experiments, we use OpenVLA-OFT (Kim et al., 2025) as our base model. OpenVLA-OFT builds on OpenVLA Kim et al. (2024) by providing a finetuning recipe designed to improve training efficiency and performance. In particular, it supports parallel decoding, action chunking, and a continuous action representation, resulting in faster and more performant inference and learning. We finetune OpenVLA-OFT starting from the OpenVLA checkpoint pretrained on the Open X-Embodiment dataset Vuong et al. (2023), using the Bridge Dataset V2 Walke et al. (2023).

A.3 DETAILS OF VLM AS REWARD

A.3.1 PROMPT FOR VLM AS REWARD

Prompt GPT-4o (Hurst et al., 2024) as Reward $\hat{R}$.

```
Here is a sequence of frames from a robot policy which has been
rolled out in a video-generation-based world model.  I need your help
determining whether the policy is successful.  How successfully does
the robot complete the following task?
Instruction:
{instruction}
{rubric.strip()}
Provide brief reasoning (2--3 sentences).  Then output EXACTLY one
final line:
Final Score:  X
Where X is { 'one of '0 or 1' }.
No extra numbers after that line.

Note:  Since this video was generated by a video prediction model
(conditioned on robot actions), it may contain some artifacts due to
the video model capacity.
```

The rubric is:

```
Score rubric:
0 = Failure:  instruction "{instruction}" not completed.
1 = Success:  instruction completed.
```

For each policy evaluation, we perform 5 independent rollouts in the world model. To reduce token usage and mitigate redundancy in adjacent frames, each rollout video is temporally downsampled with a stride of 3 before being sent to the VLM for scoring.

A.4 DETAILS OF HYPERPARAMETERS FOR WORLD MODEL AND RL TRAINING

World Model Implementation Details. WorldGym (Quevedo et al., 2025) serves as the world-model-based simulator throughout our experiments. It encodes 256×256 RGB image frames into a latent space using a pretrained VAE from Stable Diffusion 3 (Esser et al., 2024). The underlying world model in WorldGym is a 16-layer transformer with a hidden dimension of 1024 and 16 attention heads. Unless otherwise specified, we follow the default WorldGym configuration for visual encoding, world model architecture, and rollout settings.We use a world model initialized from a pretrained checkpoint trained on the Open-X Embodiment dataset (Vuong et al., 2023).

WorldGym represents actions as 10-dimensional vectors. In our experiments, we use the first 7 dimensions to match the OpenVLA (Kim et al., 2024) action space, consisting of a 6-dimensional end-effector pose and a binary gripper state. WorldGym is trained with a fixed context length of 20 frames; during longer rollouts, it conditions on a sliding window of the most 20 recent frames.

Hyperparameter	**Value**
Total parameters	609 M
Image Resolution	256×256
DiT Patch Size	2
Input Channels	16
Hidden Size	1024
Layers	16
Attention Heads	16
MLP Ratio	4
Optimizer	AdamW (weight decay = 0.002, $\beta_1 = 0.9, \beta_2 = 0.99$)
Learning rate	8e-5
Batch size	16
Action dimension	10
Training hardware	2xA100 80GB
Training steps	580k
Diffusion noise schedule	sigmoid
Sampling timesteps	10
Prediction target	v

Table 4: Hyperparameters for training World-Gymnast's video world model.

RL Training Implementation Details. We train policies using Group Relative Policy Optimization (GRPO) (Shao et al., 2024) within WorldGym (Quevedo et al., 2025). For each task, training is initialized from a single observation frame and a language instruction. Policies are rolled out for up to 40 steps in the world model, and a binary task completion reward is assigned to each rollout using GPT-4o (Hurst et al., 2024). Actions are generated in chunks of length 5, and policy updates follow a clipped policy gradient objective. All RL trainings are conducted entirely within WorldGym.

Hyperparameter	**Value**
RL algorithm	GRPO
World model environment	WorldGym
Base policy	OpenVLA-OFT
Max rollout length	40 steps
Action chunk length	5
Reward type	Binary task completion
Reward source	GPT-4o
Group size	8
Batch size	20
Learning rate	5×10^{-6}
Clip ratio	$\epsilon_{\text{high}} = 0.28,\ \epsilon_{\text{low}} = 0.2$
Temperature	1.6
Entropy coefficient	0.0
Gradient clipping	1.0
Optimizer	AdamW
Training hardware	4 $\times$ NVIDIA H200 (140GB)
Training duration	1–2 days

Table 5: Hyperparameters for RL training in WorldGymnast.

B BASELINES

B.1 SUPERVISED FINE-TUNING

Supervised fine-tuning is an alternative for reinforcement learning for policy improvement due to its sample efficiency. As our first baseline, we follow the recipe of OpenVLA-OFT Kim et al. (2025) to fine-tune a OpenVLA-7B model Kim et al. (2024) on expert trajectories from the Bridge V2 dataset Walke et al. (2023).

Methods like Ctrl-World Guo et al. (2025) explore iterative supervised fine-tuning with synthetic data generated from the world model. Following their approach, we pick the same held-out scenarios used in RL training, and roll out the base SFT policy with our world model pretrained on OpenX Embodiment dataset Vuong et al. (2023). To ensure fair comparison, the roll-out steps for iterative SFT is the same as total roll-out steps during RL fine-tuning. We then prompt a VLM to filter for successful trajectories, which are mixed with Bridge V2 dataset for the next iteration of fine-tuning, with a $1:1$ sampling rate.

B.2 SIMPLER

To compare World-Gymnast with simulator-based RL, we select the SIMPLER simulator Li et al. (2024) which provides a digital twin of the Bridge robot setup. SIMPLER includes the following tasks: 1) *put spoon on table cloth*, 2) *put carrot on plate*, 3) *stack green cube on yellow cube*, 4) *put eggplant in basket*, 5) *put eggplant in sink*, 6) *open drawer* and 7) *close drawer*. The last 4 tasks are digital twins of AutoEval setup Zhou et al. (2025).

The base SFT policy showed poor performance on SIMPLER tasks, resulting in low reward variance and unstable training with GRPO Shao et al. (2024). To address this, we define trajectory reward as sum of step rewards. Each step is assigned partial credit using a dense reward scheme: a reward of 0.1 was awarded for each of the following conditions: *is source object grasped*, *is grasped*, *consecutive grasp*, *lifted object significantly* and *lifted object*. Success earned reward of value 1.

C DATASETS

C.1 OPENVLA EVALUATION TASK SET

We evaluate different methods on a curated set of tabletop manipulation tasks originally introduced in OpenVLA (Kim et al., 2024). This task suite is designed to assess policy generalization across multiple axes, including visual, motion, physical, and semantic variations, as well as language grounding in scenes with multiple objects. For our RL training, the task suite is split into an 80/20 train–evaluation split, where both splits share the same set of tasks but differ in initial frames.

Tasks	**Generalization Types**
Put Eggplant into Pot (Easy Version)	Visual Generalization
Put Eggplant into Pot	Visual Generalization
Put Cup from Counter into Sink	Visual Generalization
Put Eggplant into Pot (w/ Clutter)	Visual Generalization
Put Yellow Corn on Pink Plate	Visual Generalization
Lift Eggplant	Motion Generalization
Put Carrot on Plate (w/ Height Change)	Motion Generalization
Put Carrot on Plate	Physical Generalization
Flip Pot Upright	Physical Generalization
Lift AAA Battery	Physical Generalization
Move Skull into Drying Rack	Semantic Generalization
Lift White Tape	Semantic Generalization
Take Purple Grapes out of Pot	Semantic Generalization
Stack Blue Cup on Pink Cup	Semantic Generalization
Put Eggplant, Red Bottle into Pot	Language Grounding
Lift Cheese, Red Chili Pepper	Language Grounding
Put Blue Cup, Pink Cup on Plate	Language Grounding

Table 6: Tasks and corresponding Generalization Types for OpenVLA Evaluation Task Set.

C.2 VISUAL DISTRACTORS

To evaluate robustness to visual perturbations, we construct a distractor-augmented training distribution by modifying initial frames from the OpenVLA evaluation task set (Kim et al., 2024). Using Nano Banana (Google, 2025), we insert visually diverse but task-irrelevant objects into the scene while keeping the underlying task, goal specification, and robot configuration unchanged. The distractor dataset is constructed from the training split of OpenVLA evaluation task set. Specifically, 90% of training episodes use the original initial frames, while 10% are initialized from distractor-augmented frames.

For evaluation, we measure performance both on a held-out set of distractor-augmented initial frames and on the original OpenVLA evaluation tasks in WorldGym (Quevedo et al., 2025), to assess robustness under visual perturbations. Qualitative rollout examples under visual distractors are included in Figure 2

Prompt for Nano Banana to edit initial frames.

```
This is an initial observation for a robotics task [TASK_INSTRUCTION].
Modify this image by adding distraction objects in the scene in a
natural way without moving or changing any objects in the original
scene.
Requirements:
    • The robotic arm should be visible.
    • With the distractors, the task [TASK_INSTRUCTION] should remain
      achievable.
```

C.3 OUT-OF-DISTRIBUTION LANGUAGES

To evaluate robustness to language distribution shift, we construct an out-of-distribution (OOD) language training distribution by modifying task instructions for a subset of OpenVLA tasks (Kim et al., 2024). The underlying scenes and objects remain unchanged; only the natural language instructions are altered to describe new interactions.

The OOD language dataset consists of four newly defined language variants applied to existing OpenVLA tasks: **(a)**: *move pot with grapes into the drying rack*, **(b)**: *pick up the pot from the drying rack and place it outside the sink on the counter*, **(c)**: *put plate on drying rack*, and **(d)**: *put yellow corn in red cup*. During RL training, we combine this OOD language data with the training split of OpenVLA evaluation task set using a 50/50 mixture over initial frames between the original and OOD language instructions. Qualitative rollout examples under OOD language instructions are included in Figure 6.

C.4 EVALUATING DIVERSE SETTINGS WORLD-GYMNAST OFFERS

C.5 SCALED TASK SET

To study the effect of increasing task diversity during RL training, we construct a scaled training distribution by augmenting the training split of the OpenVLA evaluation task set with five additional manipulation tasks randomly selected from the Bridge Dataset V2 (Walke et al., 2023): **(a)**: *close cabinet*, **(b)**: *flip orange pot upright in sink*, **(c)**: *fold the cloth from right to left*, **(d)**: *opened the drawer*, and **(e)**: *take spatula off plate sink*. These additional tasks are not present in the OpenVLA evaluation task suite and are used only during RL training, while the training procedure remains unchanged.

C.6 REAL-ROBOT EVALUATION DATASET

To evaluate real-world transfer, we report performance on AutoEval (Zhou et al., 2025), an automated real-robot evaluation framework built on tasks from the Bridge Dataset V2 distribution. In our experiments, we evaluate policies on three manipulation tasks supported by AutoEval across two scenes using a WidowX robot arm: two drawer manipulation tasks (*open the drawer*, *close the drawer*) and one pick-and-place tasks in the sink scene (*put the eggplant in the blue sink*). We do not evaluate on the cloth manipulation task supported by AutoEval. Qualitative rollout examples under scaled task set are included in Fig 5.

D EVALUATION

D.1 WORLDGYM EVALUATION

We evaluate learned policies in WorldGym (Quevedo et al., 2025), a world-model-based simulator designed to approximate real-robot execution.

Rollout protocol. For each task, the policy is rolled out in WorldGym for up to 40 steps starting from a given initial frame and language instruction. Each rollout produces a sequence of predicted future frames, which are used to assess task completion.

Success evaluation. Following the WorldGym evaluation protocol, task success is determined using a vision-language model (VLM) that compares the rollout outcome against the task instruction. To reduce variance in VLM-based scoring, we obtain 5 independent VLM judgments per rollout and assign a binary success label by majority vote.

D.2 AUTOEVAL EVALUATION

AutoEval Zhou et al. (2025) provides a convenient and reliable platform for policy evaluation on real world robots. There are two setups available, each equipped with a WidowX robotic arm (Figure 5). Each setup can run two tasks which perform the opposite operation to one another (e.g. open and close drawer). When evaluating a policy on a task, there is a reset policy to reset the conditions, allowing the policy being evaluated to conduct next trial.

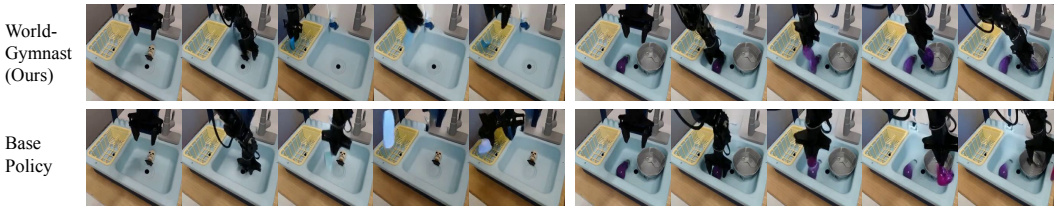

Figure 4: **Qualitative evaluation of policy rollouts in WorldGym.** We compare the World-Gymnast policy fine-tuned with RL and the base policy before performing RL. **Left:** *lift skull*; **Right:** *put eggplant in pot*.

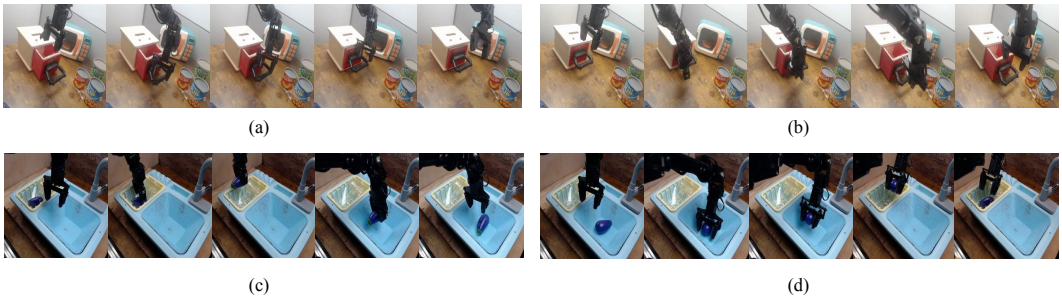

Figure 5: **Qualitative evaluation of policy rollouts in AutoEval. (a):** *close the drawer*; **(b):** *open the drawer*; **(c):** *put the eggplant in the blue sink*; **(d):** *put the eggplant in the yellow basket*.

E QUALITATIVE EXAMPLES UNDER DIFFERENCE SCENARIOS

In this section, we provide additional qualitative results and visualization to supplement our reported results.

Rollout visualization. In Figure 4, we visualize a few rollout samples in the learned world model WorldGym. We also present a comparison between the base policy and the RL-finetuned policy from World-Gymnast, illustrating the effectiveness of RL with a world model. Figure 5 shows some examples of the AutoEval evaluation settings. On the other hand, Figure 9 displays how WorldGym performs with the slightly out-of-distribution AutoEval image frames.

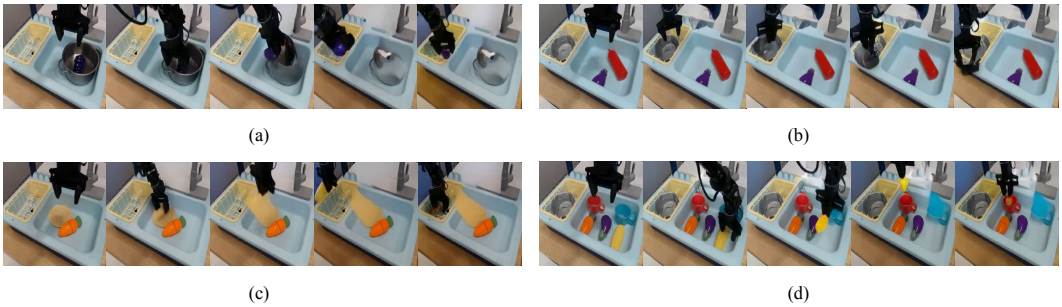

Figure 6: **Qualitative evaluation of policy rollouts with out-of-distribution language descriptions. (a):** *move pot with grapes into the drying rack*; **(b):** *pick up the pot from the drying rack and place it outside the sink on the counter*; **(c):** *put plate on drying rack*; **(d):** *put yellow corn in red cup*.

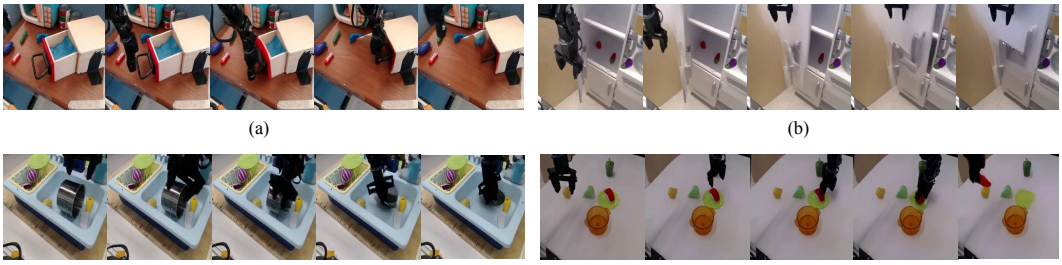

Figure 7: **Qualitative evaluation of policy rollouts given novel initial image frames. (a):** *close the drawer*; **(b):** *close fridge*; **(c):** *flip pot upright which is in sink*; **(d):** *take sushi off plate*.

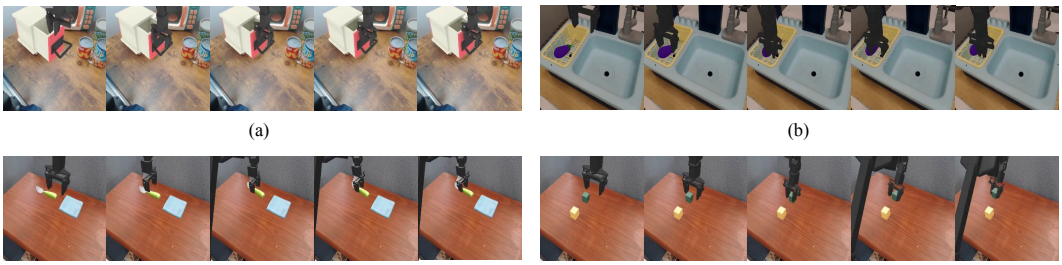

Figure 8: **Qualitative evaluation of the policy trained in the SIMPLER simulator. (a):** *close drawer*; **(b):** *put the eggplant in sink*; **(c):** *put spoon on table cloth*; **(d):** *stack green cube on yellow cube*.

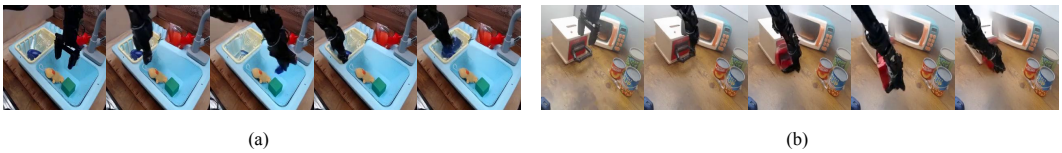

Figure 9: **Qualitative evaluation of policy rollouts in WorldGym with frames from AutoEval. (a):** *put the eggplant in the blue sink*; **(b):** *open the drawer*. WorldGym was not trained on these image observations from AutoEval.

Diverse training scenarios for World-Gymnast. We provide additional qualitative results on the discussed diverse training settings enabled by World-Gymnast in Figure 6. Figure 7 shows the policy behavior when trained on unseen scenes in the world model without expert data.

Comparison with SIMPLER. Here we show in Figure 8 additional qualitative evaluations of a policy trained with RL in the SIMPLER simulator, as discussed in Section 4.2. Training in such a simulator proves to be less desirable for real-life deployment.