WORLDGYM: WORLD MODEL AS AN ENVIRONMENT FOR POLICY EVALUATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Evaluating robot control policies is difficult: real-world testing is costly, and handcrafted simulators require manual effort to improve in realism and generality. We propose a world-model-based policy evaluation environment (WorldGym), an autoregressive, action-conditioned video generation model which serves as a proxy to real world environments. Policies are evaluated via Monte Carlo rollouts in the world model, with a vision-language model providing rewards. We evaluate a set of VLA-based real-robot policies in the world model using only initial frames from real robots, and show that policy success rates within the world model highly correlate with real-world success rates. Moreoever, we show that WorldGym is able to preserve relative policy rankings across different policy versions, sizes, and training checkpoints. Due to requiring only a single start frame as input, the world model further enables efficient evaluation of robot policies' generalization ability on novel tasks and environments. We find that modern VLA-based robot policies still struggle to distinguish object shapes and can become distracted by adversarial facades of objects. While generating highly realistic object interaction remains challenging, WorldGym faithfully emulates robot motions and offers a practical starting point for safe and reproducible policy evaluation before deployment.[1]

1 INTRODUCTION

Robots can help humans in ways that range from home robots performing chores (Shafiullah et al., 2023; Liu et al., 2024) to hospital robots taking care of patients (Soljacic et al., 2024). One of the major road blocks in the development robots lies in evaluation — how should we ensure that these robots will work reliably without causing any physical damage when deployed in the real world? Traditionally, people have used *handcrafted* software simulators to develop and evaluate robot control policies (Tedrake et al., 2019; Todorov et al., 2012; Erez et al., 2015). However, handcrafted simulation based on our understanding of the physical world can be limited, especially when it comes to hardcoding complex dynamics with high degrees of freedom or complex interactions such as manipulating soft objects (Sünderhauf et al., 2018; Afzal et al., 2020; Choi et al., 2021). As a result, the sim-to-real gap has hindered progress in robotics (Zhao et al., 2020; Salvato et al., 2021; Dulac-Arnold et al., 2019).

With the development of generative models trained on large-scale video data (Ho et al., 2022; Villegas et al., 2022; Singer et al., 2022), recent work has shown that video world models can visually emulate interactions with the physical real world, by conditioning on control inputs in the form of text (Yang et al., 2023; Brooks et al., 2024) or keyboard strokes (Bruce et al., 2024). This brings up an interesting question — could video world models be used to emulate robot interactions with the real world, hence being used as an environment to evaluate robot policies in the world model before real-world testing or deployment?

Learning a dynamics model from past experience and performing rollouts in the learned dynamics model has been extensively studied in model-based reinforcement learning (RL) (Hafner et al., 2019; Fonteneau et al., 2013; Zhang et al., 2021; Kaiser et al., 2019; Yu et al., 2020). However, most of the existing work in model-based RL considers single-task settings, which puts itself at a disadvantage compared to model-free RL, since learning a dynamics model can be much harder than learning a policy in the single-task setting. Nevertheless, we make the important observation that

[1] See videos and code in supplementary materials.

*While there can be many tasks and policies, there is only **one physical world** in which we live that is governed by the **same set of physical laws**.*

This makes it possible to learn a single world model that, in principle, can be used as an interactive environment to evaluate any policies on any tasks.

Inspired by this observation, we propose a world-model-based policy evaluation environment (WorldGym), as shown in Figure 1. WorldGym first combines knowledge of the world across diverse environments by learning a *single* world model that generates videos conditioned on actions. To enable efficient rollouts of policies which predict different-length action chunks, WorldGym aligns its diffusion horizon length with policies' chunk sizes at inference time. With video rollouts from the world model, WorldGym then uses a vision-language model (VLM) to determine tasks' success from generated videos.

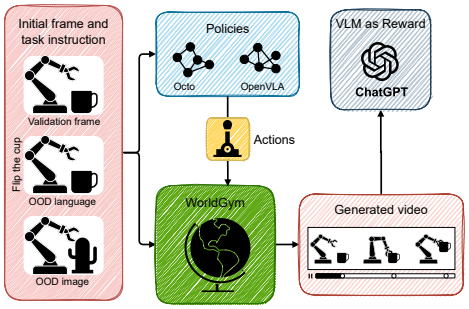


Figure 1: **Overview of WorldGym.** Given an initial frame and an action sequence predicted by a policy, WorldGym uses a world model to predict future frames, serving as a generative simulator. WorldGym then passes the generated rollout to a VLM which provides rewards. WorldGym can easily be used to test policies on OOD tasks and environments by changing the input language instruction or directly modifying the initial image.

Our experiments show that WorldGym can emulate end-effector controls across different control axes highly effectively for robots with different morphologies. We then use the world model to evaluate VLA-based robot policies by rolling out the policies in the world model starting from real initial frames, and compare their success rates (policy values) in WorldGym to those achieved in real-world experiments. Our result suggests that policy values in WorldGym are highly correlated with policy performance in the real world, and the relative rankings of different policies are preserved.

Furthermore, as WorldGym requires only a single initial frame as input, we show how we can easily design out-of-distribution (OOD) tasks and environments and then use WorldGym to evaluate robot policies within these newly "created" environments. We find that modern robot policies still struggle to distinguish some classes of objects by their shape, and can even be distracted by adversarial facades of objects.

Although simulating realistic object interactions remains challenging, we believe WorldGym can serve as a highly useful tool for sanity check and testing robot policies safely and reproducibly before deploying them on real robots. Key contributions of this paper include:

- We propose to use video world model to evaluate robot policies across different robot morphologies, and perform a comprehensive set of studies to understand its feasibility.
- We propose flexibly aligning diffusion horizon length with policies' action chunk sizes for efficient rollouts of a variety of policies over hundreds of interactive steps.
- We show a single world model learned on data from diverse tasks and environments can enable policy value estimates that highly correlate with real-world policy success rates.
- We demonstrate the ease of testing robot policies on OOD tasks and environments within an autoregressive video generation-based world model.

2 Problem Formulation

In this section, we define relevant notations and review the formulation of offline policy evaluation (OPE). We also situate OPE in practical settings with partially observable environments and image-based observations.

Multi-Task POMDP. 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 a state space, action space, observation space, goal space, reward function, transition function, emission function, and horizon length. A policy π interacts with the environment for a goal starting from an initial state $g, s_0 \sim G$, producing a distribution $\pi(\cdot|s_t, g)$ 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 $o_{t+1} \sim \mathcal{E}(s_{t+1})$. We consider the sparse reward setting

with $R(s_H, g) \in \{0, 1\}$ and $R(s_t, g) = 0, \forall t < H$, where g is a language goal that defines the task. Data is logged from previous interactions into an offline dataset $D = \{g, s_0, o_0, a_0, ..., s_H, o_H, r_H\}$. The value of a policy π can be defined as the total expected future reward:

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

Estimating the value of $\rho(\pi)$ from previously collected data D, known as offline policy evaluation (OPE) (Levine et al., 2020), has been extensively studied (Thomas & Brunskill, 2016; Jiang & Li, 2016; Fu et al., 2021; Yang et al., 2020; Thomas et al., 2015b). However, existing work in OPE mostly focuses on simulated settings that are less practical (e.g., assumptions about full observability, access to ground truth states).

Model-Based Evaluation. Motivated by characteristics of a real-robot system such as image based observations, high control frequencies, diverse offline data from different tasks/environments, and the lack of access to the ground truth state of the world, we consider the use of offline data to learn a *single* world model $\hat{T}(\cdot|\mathbf{o}, \mathbf{a})$, where $\mathbf{o}$ represents a sequence of previous image observations and $\mathbf{a}$ represents a sequence of next actions. A sequence of next observations can be sampled from the world model $\mathbf{o}' \sim \hat{T}(\mathbf{o}, \mathbf{a})$. With this world model, one can estimate the policy value $\rho(\pi)$ with Monte-Carlo sampling using stochastic rollouts from the policy and the world model:

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

where $\hat{R}$ is a learned reward function. Previously, model-free policy evaluation may be more preferable since in a single task setting, dynamics models are potentially harder to learn than policy values themselves, and doing rollouts in a dynamics model may lead to compounding errors (Xiao et al., 2019). However, we make the key observations that while there can be many tasks and many policies, there is only one physical world that is governed by the same set of physical laws. As a result, learning a world model can benefit from diverse data from different tasks and environments with different state spaces, goals, and reward functions. More importantly, a world model can be directly trained on image-based observations, which is often the perception modality of real-world robots.

3 Building and Evaluating the World Model

In this section, we first describe our implementation of world model training and inference. Then, we discuss how we validate our world model's performance prior to rolling out real robot policies within it in the next section.

3.1 Building the World Model

First, we describe the architecture and key implementation details, followed by our proposed inference scheme for policy rollouts.

3.1.1 World Model Training

We train a latent Diffusion Transformer (Peebles & Xie, 2023) on sequences of frames paired with actions, using Diffusion Forcing (Chen et al., 2024) to enable autoregressive frame generation. Per-frame robot action vectors are linearly projected to the model dimension and added elementwise to diffusion timestep embeddings, the result of which is used to condition the model through AdaLN-Zero modulation, similar to class conditioning in Peebles & Xie (2023). To ensure the world model is fully controllable by robot actions, we propose to randomly drop out actions for entire video clips, and use classifier-free guidance to improve the world model's adherence to action inputs. Conditioning on previous frames' latents is achieved via causal temporal attention blocks interleaved between spatial attention blocks, as in Bruce et al. (2024); Ma et al. (2025). See Appendix A for additional implementation details.

3.1.2 Rolling Out a Policy in the World Model

Our policy evaluation pipeline operates through an iterative loop between the robot policy and the world model. First, the world model is initialized with an initial observation o_0, which is then passed as input to a policy π which produces a chunk of actions $\mathbf{a}_{\text{pred}}$. The actions are passed back to the world model, which predicts a new frame for each action in $\mathbf{a}_{\text{pred}}$. The latest frame produced by the world model is then returned to the policy as its next input observation.

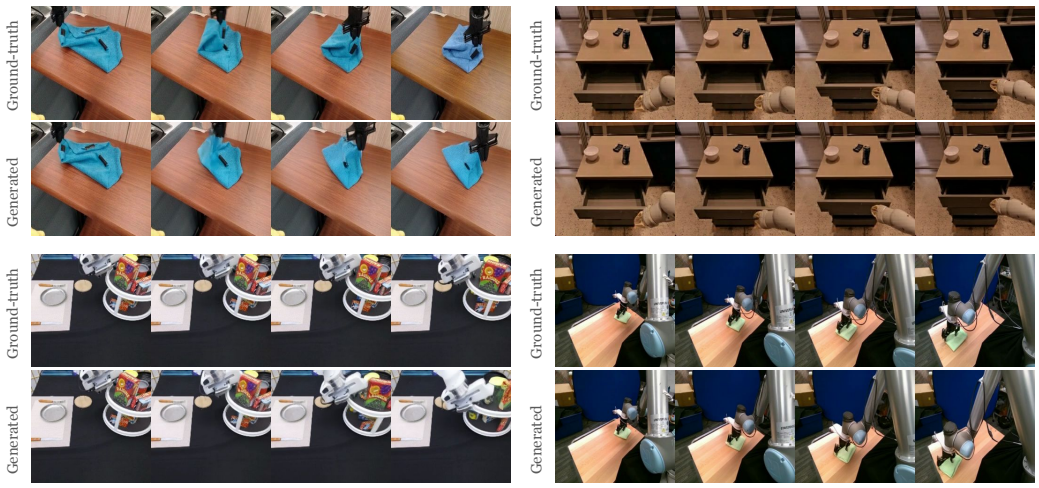


Figure 2: **Qualitative evaluation of the world model on Bridge, RT-1, VIOLA, and Berkeley UR5.** In each group, top row shows the ground truth video from the real robot. Bottom row shows the generated video from the world model conditioned on the *same* actions as the original video. The world model closely follows the true dynamics across different robot morphologies.

Since different robot policies output a different number of actions at once (Kim et al.; Brohan et al., 2022; Chi et al., 2023), WorldGym needs to support efficient prediction of a chunk of videos conditioned on a chunk of (variable length) actions. By virtue of being trained with Diffusion Forcing, as well as our usage of a causal temporal attention mask, we can flexibly control how many frames our world model denoises in parallel at inference time, i.e. its prediction *horizon length*. We propose setting the horizon equal to the policy's action chunk size, $|\mathbf{a}_{\text{pred}}|$. This has the benefit of efficient frame generation for policies with differing action chunk sizes, all from a single world model checkpoint. This contrasts with prior diffusion world models for robotics, such as Cosmos (NVIDIA et al., 2025), which, due to being trained with bidirectional attention and a fixed context length, must always denoise 16 latent frames in parallel. This constraint results in wasted compute for action chunk sizes less than the context length and unrealized parallelism for chunk sizes which are larger. On the other hand, our design allows parallelism to flexibly match the number of actions, thus utilizing hardware more effectively (see Appendix F.2).

3.1.3 VLM as Reward

We opt for GPT-4o (OpenAI et al., 2024) as a reward model, passing in the sequence of frames from the generated rollout and the language instruction (see the prompt for the VLM in Appendix B). In certain cases where both policies being evaluated fail to perform a task end-to-end, it is still helpful to get signals on which policy is closer to completing a task. We can specify these partial credit criteria to the VLM to further distinguish performance between different policies, which has been done manually using heuristics in prior work (Kim et al.). We validate the accuracy of VLM-predicted rewards in Appendix B.2.

3.2 Evaluating the World Model

Next, we describe how we validate the performance of our world model prior to policy evaluation, ensuring that it exhibits realistic robot movement and adheres to arbitrary action controls.

3.2.1 Agreement with Validation Split

First, we test the world model's ability to generate similar videos as running a robot in the real world. Specifically, we take the validation split of initial images from the Open-X Embodiment dataset, and predict videos conditioned on the *same* action sequences as in the original data. Figure 2 shows that the generated rollouts generally follow the real-robot rollouts across different initial observations and different robot morphologies.

3.2.2 End-Effector Control Sweeps

Next, we need a way to evaluate whether our world model can emulate arbitrary action sequences, beyond the kinds of action sequences present in the training data. We propose hard-coding a robot

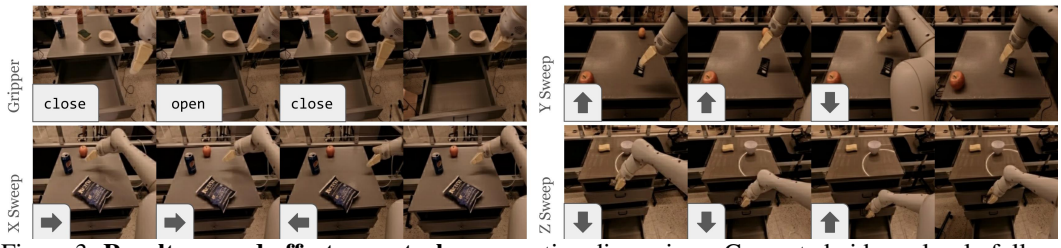
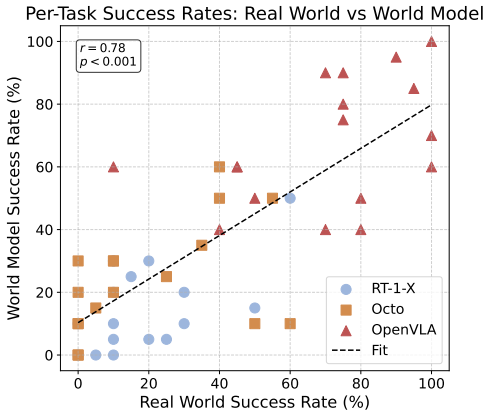
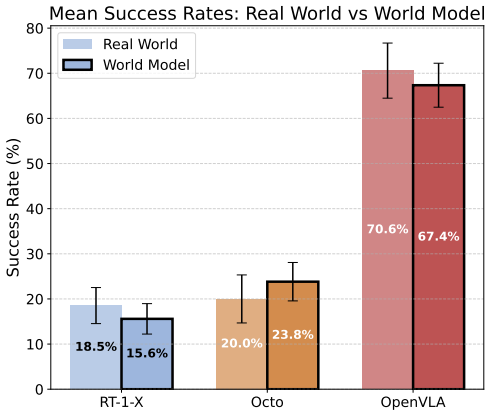


Figure 3: **Results on end-effector control** across action dimensions. Generated videos closely follow the gripper controls such as open and close the gripper as well as moving in different directions starting from any initial observation frame. Results for control sweeps on the Bridge robot can be found in Figure 16 in Appendix E.1.



(a) **Per-Task Task Success Rates.** Each point represents a task from Table 5, with different policies being represented by different shaped markers. There is a strong correlation ($r = 0.78$) between policy performance in our world model (y-axis) and within the real world (x-axis).



(b) **Mean Success Rates.** Robot policies' mean success rates in the world model differ by an average of only 3.3% between from the real world, near the standard error range for each policy. Relative performance rankings between RT-1-X, Octo, and OpenVLA are also preserved.

Figure 4: Success rates of modern VLAs, as evaluated within WorldGym and the real world.

control policy by only moving one action dimension at once (and keeping the other action dimensions as zeros). The robot is then expected to move along that one action dimension with non-zero input, corresponding to moving in different horizontal and vertical directions as well as open and close its gripper. Figure 3 shows that the generated videos faithfully follow the intended end-effector movement,[2] despite the fact that these particular sequences of controls are not present in the training data.

4 Evaluating Policies in WorldGym

Having established confidence in the world model's performance, we now use the world model to evaluate policies. We begin by rolling out three recent VLA policies in WorldGym and check whether WorldGym **reflects real-world success**. (Section 4.1). We then assess whether **relative policy performance** is preserved, comparing different versions, sizes, and training stages of the same models (Section 4.2). Finally, we explore WorldGym's potential to test policies on **out-of-distribution (OOD) tasks and environments** (Section 4.3), including novel instructions and altered visual contexts.

4.1 Correlation between Real-World and Simulated Policy Performance

Qualitative Evaluation. To ensure WorldGym is useful for policy evaluation, we test whether policy performance within the world model is similar to that of the real world. To do so, we perform a direct comparison with the Bridge evaluation trials from OpenVLA (Kim et al.). Specifically, the OpenVLA Bridge evaluation consists of 17 challenging tasks which are not present in the Bridge V2 (Walke et al., 2023) dataset. We use WorldGym to evaluate the three open-source policies evaluated

[2]Results are best viewed as videos in the supplementary material.

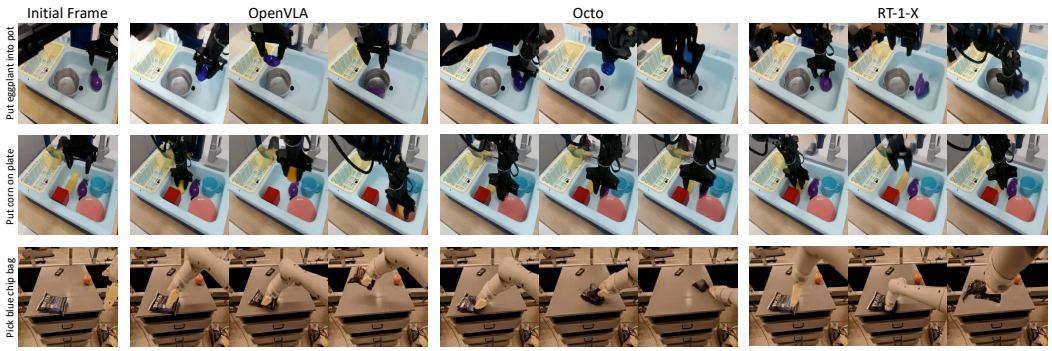


Figure 5: **Qualitative policy rollouts on Bridge and Google Robot** for RT-1-X, Octo, and OpenVLA. OpenVLA rollouts often lead to more visual successes than the other two policies across environments.

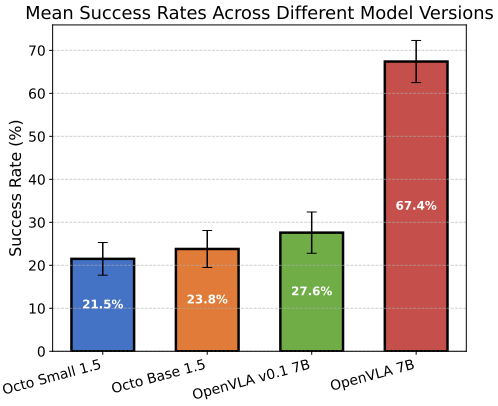


Figure 6: **Success Rates of different model versions in WorldGym.** We evaluate different generations of Octo and OpenVLA in the world model, showing that WorldGym assigns higher score to larger and more recent versions.

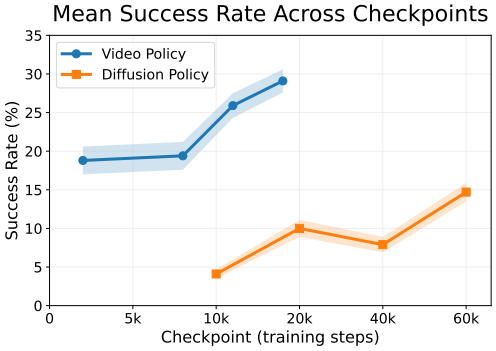


Figure 7: **Success Rate within WorldGym throughout training.** We train a video-based policy and a diffusion policy from scratch and evaluate it within our world model as it trains. We see that mean task success rate within the world model increases with additional training steps.

in Kim et al.: RT-1-X (O'Neill et al., 2023), Octo (Octo Model Team et al., 2024), and OpenVLA (Kim et al.). For each task and each policy, Kim et al. perform 10 trials, each with randomized initial object locations. We obtain the first frame of the recorded rollouts for all trials of all tasks. We then simulate each of the 10 real-world trials by using the original initial frame to roll out the policy within the world model as described in Section 3.1.2. We show qualitative rollouts in WorldGym from different policies in Figure 5, which shows that rollouts from OpenVLA generally perform better than rollouts from RT-1-X and Octo on the Bridge robot (top two rows). We further show that WorldGym can be easily used to perform rollouts in other environments with other robots, such as the Google Robot (bottom row in Figure 5).

Quantitative Evaluation. Using the simulated rollouts from WorldGym, we then compute the average task success rate similar to Kim et al., and plot the success rate for each task for each policy in Figure 4a. We find that real-world task performance is strongly correlated with the task performance reported by the world model, achieving a Pearson correlation of $r = 0.78$. While per-task policy success rates within WorldGym still differ slightly from those in the real world (see Table 5), the mean success rates achieved by these policies within WorldGym are quite close to the their real-world values, as shown in Figure 4b. The success rates differ by an average of only 3.3%, with RT-1-X achieving 18.5% in the real world vs 15.5% in the world model, Octo achieving 20.0% vs 23.82%, and OpenVLA achieving 70.6% vs 67.4%, respectively. See quantitative results of evaluating the three policies on the Google Robot in Appendix E.2

4.2 Policy Ranking within a World Model

Now we test whether WorldGym can preserve policy rankings known a priori. We evaluated policies across different versions, sizes, and training stages within WorldGym on the OpenVLA Bridge

evaluation task suite, and found their in-world-model performance rankings to be consistent with prior knowledge of their relative performance.

Different VLAs with Known Ranking. First, we average success rates across all 17 tasks and find that the relative performance rankings between RT-1-X, Octo, and OpenVLA are the same (Figure 4b) within both WorldGym and the real-world results reported in OpenVLA (Kim et al.).

Same Policies across Versions and Sizes. We further examine whether WorldGym preserves rankings between different versions and sizes of the same policy. In particular, we compare Octo-Small 1.5 against Octo-Base 1.5, and OpenVLA v0.1 7B, an undertrained development model, against OpenVLA 7B. As shown in Figure 6, the larger and more recent models outperform their smaller or earlier counterparts within WorldGym, consistent with the findings of real-world experiments performed in Octo Model Team et al. (2024) and Kim et al.. This provides additional evidence that WorldGym faithfully maintains relative rankings even across model upgrades.

Same Policy across Training Steps. To examine whether WorldGym provides meaningful signals for policy training, hyperparameter tuning, and checkpoint selections, we train two robot policies from scratch. Building on prior evidence of WorldGym's effectiveness in evaluating VLA-based policies, we extend our study to two additional families: a video prediction–based policy (UniPi) (Du et al., 2023a) and a diffusion-based policy (DexVLA) (Wen et al., 2025), both trained on the Bridge V2 dataset (see Appendix C and Appendix D). We evaluate checkpoints of the video prediction policy at 2K, 8K, 12K, and 18K steps, and the diffusion policy at 10K, 20K, 40K, and 60K steps.

As shown in Figure 7, WorldGym tends to assign higher success rates to checkpoints as they increase in training steps, consistent with the lower mean squared error these policies achieve on their validation splits. This demonstrates WorldGym's ability to preserve policy rankings across models with different amounts of training compute.

Thus, we have shown how WorldGym can be used to obtain reasonable policy rankings. In particular, for the VLA-based policies we evaluate, we arrive at the same conclusions as real-world experiments about their relative performances. Notably, this is achieved all **without the manual effort** of setting up real robot evaluation environments and monitoring policy rollouts. While real-world evaluation can sometimes take days to complete, all WorldGym rollouts reported here can be completed in under an hour on a single GPU and require only initial images for each trial.

4.3 Out-of-Distribution Inputs

In this section, use WorldGym to explore policies' performance on both OOD input images and OOD language instructions.

OOD Image Input. Using modern image generation models like Nano Banana (Google, 2025), we can easily generate new input images to initialize our world model with. We evaluate robot policies under three OOD settings: unseen object interaction, distractor objects, and object classification (see detailed results in Table 6).

- **Unseen Objects:** We edit a scene to contain both a carrot and an orange, asking the policy to pick up the orange (Figure 9). OpenVLA grabs whichever object is closer until we edit the carrot's color to be red, after which it always grabs the orange correctly. This suggests that it struggles to distinguish carrots and oranges by their shape.
- **Distractor Objects:** We use the image editing model to add a computer displaying an image of a carrot (Figure 10, left). We see that OpenVLA mistakenly to grabs the carrot on the computer screen in 15% of trials, suggesting limited 3D/2D object distinction.
- **Classification:** We add a piece of paper on each side of a desk. We first color one paper red and the other blue and instruct the model to "pick red"/"pick blue" (Figure 8). OpenVLA achieves a perfect score, always moving towards the

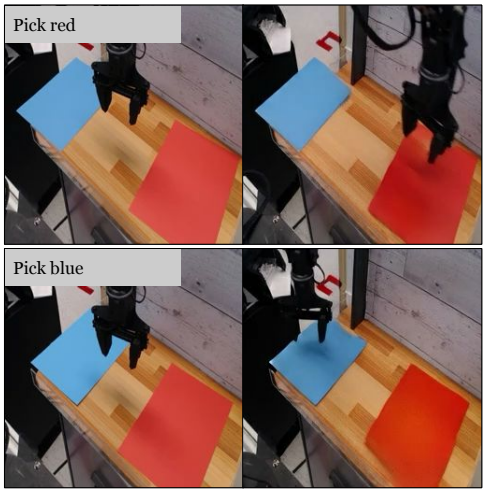


Figure 8: **OOD: Color Classification.** We add place red and blue pieces of paper on a table, and ask the policies to "pick red" or "pick blue" (OOD image and language). OpenVLA excels, picking the correct colored paper in all trials, whereas all other policies score near chance.

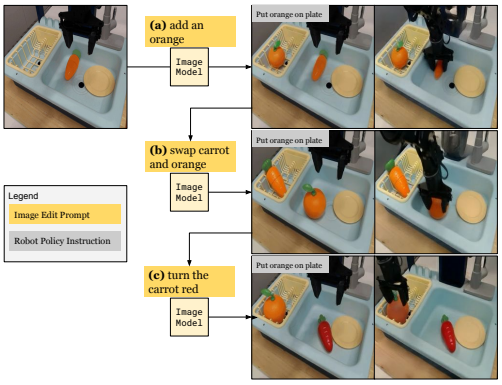


Figure 9: **OOD: Unseen object.** We use Nano Banana (Google, 2025) to add an orange to the world model's initial frame. When both the orange and the carrot are present, (a-b) OpenVLA grabs whichever is closer. After (c) editing the carrot's color to red, however, the orange is correctly picked up.

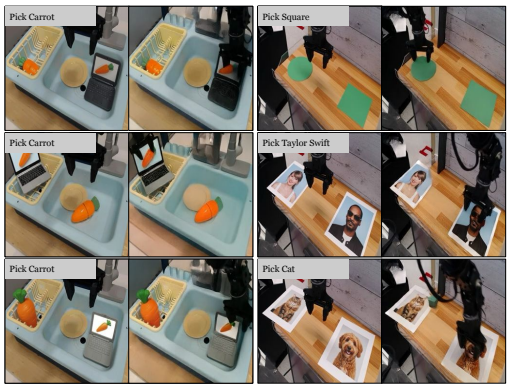


Figure 10: **OOD: Failure modes.** *Left:* We add a laptop to the scene, which displays an image of a carrot. In 15% trials, OpenVLA grabs the laptop instead of the real carrot. *Right:* We test the ability distinguish to between squares and circles, celebrity faces, and cats and dogs, with all policies scoring near-chance.

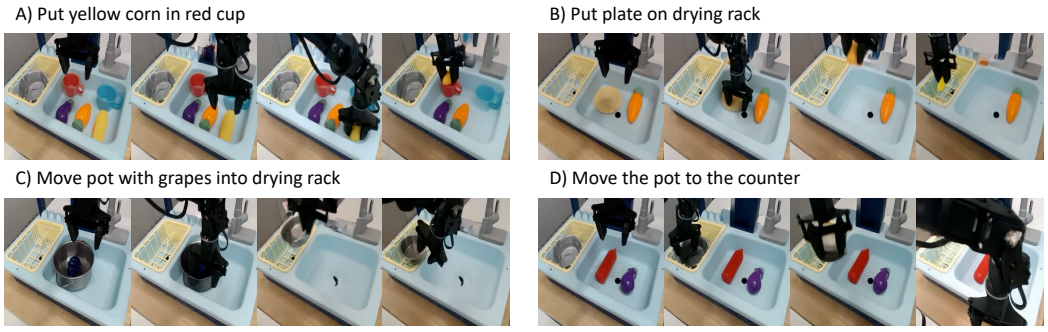


Figure 11: **OOD Language Instructions.** We pick a set of tasks from the OpenVLA Bridge evaluation suite and modify the language instruction, e.g. changing the the target object and/or its goal destination.

correct color. Octo and RT-1, on the other hand, typically move towards whichever paper is closer, scoring no better than chance. We also try more advanced classification tasks, (Figure 10, right), but find that the policies all score near-chance.

For a more quantitative study, we modify all the initial frames of the OpenVLA's Bridge evaluation task suite to include random OOD distractor items (see Figure 12), keeping the language instructions the same. We then repeat the rollout procedure from Section 4.1 in order to measure the degree to which the addition of unrelated objects affects policy performance. We find that all the tested VLAs degrade in performance, with OpenVLA being the most robust of the three (Figure 13).

OOD Language. Additionally, even without access to an image editing model, we demonstrate that WorldGym can be used to evaluate policies' performance on OOD language instructions. Starting from a set of initial frames from the tasks listed in Table 5, we modify each task's language instruction, e.g. changing the target object and/or its goal state. Figure 11 shows rollouts from OpenVLA for these OOD language tasks. We can then easily obtain success rates for these unseen tasks by rolling them out within WorldGym, finding that OpenVLA generalizes best (see Table 1). Policies struggle across the board on the "Move the pot to the counter" task, with only OpenVLA achieving a single success. We suspect that OpenVLA consistently outperforms Octo and RT-1-X on OOD language tasks due to its strong VLM backbone and richer robot pretraining dataset (Kim et al.).

The ability to use WorldGym to quickly design and evaluate policies within OOD tasks and environments thus leads us to new findings about policies' strengths and weaknesses. Future research could be prioritized to address these issues, all without spending extra effort to set up additional experiments in the real world or within handcrafted simulators.

Task	RT-1-X	Octo	OpenVLA
Move Pot Into Drying Rack	3	0	7
Move The Pot To The Counter	0	0	1
Put Plate On Drying Rack	4	2	8
Put Yellow Corn In Red Cup	1	2	3

Table 1: **Policy Evaluations Results on Bridge OOD Language Tasks.** "Move the pot to the counter" is perhaps the most challenging because the Bridge dataset does not contain trajectories which move objects outside of the sink basin. OpenVLA has the strongest performance, which we attribute to its more powerful language model backbone.

Figure 12: **OOD Distraction Examples.** We use Nano Banana (Google, 2025) to add distractions to every image of the OpenVLA Bridge task suite. The resulting change in mean success rates can be seen in Figure 13.

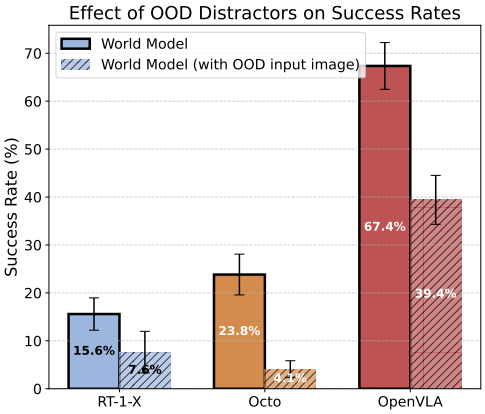


Figure 13: **Effect of OOD Distractors.** We use an image editing model to add distractor objects to the Bridge evaluation suite, finding that RT-1-X drops in performance by 51%, Octo by 83%, and OpenVLA by 41.5%, making OpenVLA the most robust to distractors. See Table 7 for details.

5 Related Work

Action-Conditioned Video Generation. Previous work has shown that video generation can simulate real-world interactions (Yang et al., 2023; Brooks et al., 2024), robotic plans (Du et al., 2024; 2023b), and games (AI et al., 2024; Bruce et al., 2024; Valevski et al., 2024; Alonso et al., 2024) when conditioned on text or keyboard controls. Prior work (NVIDIA et al., 2025) has begun to explore applying video generation to simulating complex robotic controls. We take this a step further by using video-based world models to quantitatively estimate robot policy success rates. WorldGym draws architectural inspirations from prior work on video generation such as Diffusion Forcing (Chen et al., 2024) and Diffusion Transformers (Peebles & Xie, 2023), but experiments with variable horizon lengths to support efficient long-horizon rollouts for policies with a variety of action chunk sizes.

Policy Evaluation. Off-policy and offline policy evaluation has long been studied in the RL literature (Farajtabar et al., 2018; Jiang & Li, 2015; Kallus & Uehara, 2019; Munos et al., 2016; Precup et al., 2000; Thomas et al., 2015a). Some of these approaches are model-based, learning a dynamics model from previously collected data and rolling out the learned dynamics model for policy evaluation (Fonteneau et al., 2013; Zhang et al., 2021; Yu et al., 2020; Hafner et al., 2020). Since learning a dynamics model is challenging and subject to accumulation of error, a broader set of work has focused on model-free policy evaluation, which works by estimating the value function (Le et al., 2019; Duan & Wang, 2020; Sutton et al., 2009; 2016) or policy correction (Kanamori et al., 2009; Nguyen et al., 2010; Nachum et al., 2019). WorldGym performs model-based policy evaluation, but proposes to learn a single world model on image-based observation that can be used to evaluate different policies on different tasks. SIMPLER (Li et al., 2024) aims to evaluate realistic policies by constructing software-based simulators from natural images and showed highly correlated curves between simulated evaluation and real-robot execution, but it is hard to evaluate OOD language and image input in SIMPLER without significant hand engineering of the software simulator. Li et al. (2025) proposes to evaluate robot policies in a world model in a specific bi-manual manipulation setup, whereas WorldGym focuses on evaluating policies across diverse environments and robot morphologies while enabling testing OOD language and image inputs.

6 Conclusion

We have presented WorldGym, a world-model-based environment for evaluating robot policies. WorldGym emulates realistic robot interactions and shows strong correlations between simulated evaluation and real-world policy outcomes. WorldGym further provides the flexibility for evaluating OOD language instructions and performing tasks with an OOD initial frame. While not all interactions emulated by WorldGym are fully realistic, WorldGym serves as an important step towards safe and reproducible policy evaluation before deployment.

REFERENCES

Afsoon Afzal, Deborah S Katz, Claire Le Goues, and Christopher S Timperley. A study on the challenges of using robotics simulators for testing. *arXiv preprint arXiv:2004.07368*, 2020.

Decart AI, Julian Quevedo, Quinn McIntyre, Spruce Campbell, Xinlei Chen, and Robert Wachen. Oasis: A universe in a transformer. 2024. URL https://oasis-model.github.io/.

Eloi Alonso, Adam Jelley, Vincent Micheli, Anssi Kanervisto, Amos Storkey, Tim Pearce, and François Fleuret. Diffusion for world modeling: Visual details matter in atari. *arXiv preprint arXiv:2405.12399*, 2024.

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.

Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, et al. Video generation models as world simulators. 2024. *URL https://openai. com/research/video-generation-models-as-world-simulators*, 3, 2024.

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

Boyuan Chen, Diego Marti Monso, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion forcing: Next-token prediction meets full-sequence diffusion. *arXiv preprint arXiv:2407.01392*, 2024.

Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, pp. 02783649241273668, 2023.

HeeSun Choi, Cindy Crump, Christian Duriez, Asher Elmquist, Gregory Hager, David Han, Frank Hearl, Jessica Hodgins, Abhinandan Jain, Frederick Leve, et al. On the use of simulation in robotics: Opportunities, challenges, and suggestions for moving forward. *Proceedings of the National Academy of Sciences*, 118(1):e1907856118, 2021.

Hyung Won Chung, Noah Constant, Xavier Garcia, Adam Roberts, Yi Tay, Sharan Narang, and Orhan Firat. Unimax: Fairer and more effective language sampling for large-scale multilingual pretraining, 2023. URL https://arxiv.org/abs/2304.09151.

Yilun Du, Mengjiao Yang, Bo Dai, Hanjun Dai, Ofir Nachum, Joshua B. Tenenbaum, Dale Schuurmans, and Pieter Abbeel. Learning universal policies via text-guided video generation, 2023a. URL https://arxiv.org/abs/2302.00111.

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*, 2023b.

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.

Yaqi Duan and Mengdi Wang. Minimax-optimal off-policy evaluation with linear function approximation, 2020. arXiv:2002.09516.

Gabriel Dulac-Arnold, Daniel Mankowitz, and Todd Hester. Challenges of real-world reinforcement learning. *arXiv preprint arXiv:1904.12901*, 2019.

Frederik Ebert, Yanlai Yang, Karl Schmeckpeper, Bernadette Bucher, Georgios Georgakis, Kostas Daniilidis, Chelsea Finn, and Sergey Levine. Bridge data: Boosting generalization of robotic skills with cross-domain datasets. *arXiv preprint arXiv:2109.13396*, 2021.

Tom Erez, Yuval Tassa, and Emanuel Todorov. Simulation tools for model-based robotics: Comparison of bullet, havok, mujoco, ode and physx. In *2015 IEEE international conference on robotics and automation (ICRA)*, pp. 4397–4404. IEEE, 2015.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, Kyle Lacey, Alex Goodwin, Yannik Marek, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis, 2024. URL https://arxiv.org/abs/2403.03206.

Mehrdad Farajtabar, Yinlam Chow, and Mohammad Ghavamzadeh. More robust doubly robust off-policy evaluation. *arXiv preprint arXiv:1802.03493*, 2018.

Raphael Fonteneau, Susan A. Murphy, Louis Wehenkel, and Damien Ernst. Batch mode reinforcement learning based on the synthesis of artificial trajectories. *Annals of Operations Research*, 208(1): 383–416, 2013.

Justin Fu, Mohammad Norouzi, Ofir Nachum, George Tucker, Ziyu Wang, Alexander Novikov, Mengjiao Yang, Michael R Zhang, Yutian Chen, Aviral Kumar, et al. Benchmarks for deep off-policy evaluation. *arXiv preprint arXiv:2103.16596*, 2021.

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.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. *arXiv preprint arXiv:1912.01603*, 2019.

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

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015. URL https://arxiv.org/abs/1512.03385.

Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P Kingma, Ben Poole, Mohammad Norouzi, David J Fleet, et al. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022.

Nan Jiang and Lihong Li. Doubly robust off-policy value evaluation for reinforcement learning. *arXiv preprint arXiv:1511.03722*, 2015.

Nan Jiang and Lihong Li. Doubly robust off-policy value evaluation for reinforcement learning. In *International conference on machine learning*, pp. 652–661. PMLR, 2016.

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.

Lukasz Kaiser, Mohammad Babaeizadeh, Piotr Milos, Blazej Osinski, Roy H Campbell, Konrad Czechowski, Dumitru Erhan, Chelsea Finn, Piotr Kozakowski, Sergey Levine, et al. Model-based reinforcement learning for atari. *arXiv preprint arXiv:1903.00374*, 2019.

Nathan Kallus and Masatoshi Uehara. Double reinforcement learning for efficient off-policy evaluation in Markov decision processes. *arXiv preprint arXiv:1908.08526*, 2019.

Takafumi Kanamori, Shohei Hido, and Masashi Sugiyama. A least-squares approach to direct importance estimation. *The Journal of Machine Learning Research*, 10:1391–1445, 2009.

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, 2024. *URL https://arxiv. org/abs/2406.09246*.

Hoang Le, Cameron Voloshin, and Yisong Yue. Batch policy learning under constraints. In *International Conference on Machine Learning*, pp. 3703–3712. PMLR, 2019.

Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020.

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*, 2025.

Peiqi Liu, Yaswanth Orru, Jay Vakil, Chris Paxton, Nur Muhammad Mahi Shafiullah, and Lerrel Pinto. Ok-robot: What really matters in integrating open-knowledge models for robotics. *arXiv preprint arXiv:2401.12202*, 2024.

Xin Ma, Yaohui Wang, Xinyuan Chen, Gengyun Jia, Ziwei Liu, Yuan-Fang Li, Cunjian Chen, and Yu Qiao. Latte: Latent diffusion transformer for video generation, 2025. URL https://arxiv.org/abs/2401.03048.

R. Munos, T. Stepleton, A. Harutyunyan, and M. Bellemare. Safe and efficient off-policy reinforcement learning. In *Advances in Neural Information Processing Systems*, pp. 1054–1062, 2016.

Ofir Nachum, Yinlam Chow, Bo Dai, and Lihong Li. Dualdice: Behavior-agnostic estimation of discounted stationary distribution corrections. *Advances in neural information processing systems*, 32, 2019.

XuanLong Nguyen, Martin J Wainwright, and Michael I Jordan. Estimating divergence functionals and the likelihood ratio by convex risk minimization. *IEEE Transactions on Information Theory*, 56(11):5847–5861, 2010.

NVIDIA, :, Niket Agarwal, Arslan Ali, Maciej Bala, Yogesh Balaji, Erik Barker, Tiffany Cai, Prithvijit Chattopadhyay, Yongxin Chen, Yin Cui, Yifan Ding, Daniel Dworakowski, Jiaojiao Fan, Michele Fenzi, Francesco Ferroni, Sanja Fidler, Dieter Fox, Songwei Ge, Yunhao Ge, Jinwei Gu, Siddharth Gururani, Ethan He, Jiahui Huang, Jacob Huffman, Pooya Jannaty, Jingyi Jin, Seung Wook Kim, Gergely Klár, Grace Lam, Shiyi Lan, Laura Leal-Taixe, Anqi Li, Zhaoshuo Li, Chen-Hsuan Lin, Tsung-Yi Lin, Huan Ling, Ming-Yu Liu, Xian Liu, Alice Luo, Qianli Ma, Hanzi Mao, Kaichun Mo, Arsalan Mousavian, Seungjun Nah, Sriharsha Niverty, David Page, Despoina Paschalidou, Zeeshan Patel, Lindsey Pavao, Morteza Ramezanali, Fitsum Reda, Xiaowei Ren, Vasanth Rao Naik Sabavat, Ed Schmerling, Stella Shi, Bartosz Stefaniak, Shitao Tang, Lyne Tchapmi, Przemek Tredak, Wei-Cheng Tseng, Jibin Varghese, Hao Wang, Haoxiang Wang, Heng Wang, Ting-Chun Wang, Fangyin Wei, Xinyue Wei, Jay Zhangjie Wu, Jiashu Xu, Wei Yang, Lin Yen-Chen, Xiaohui Zeng, Yu Zeng, Jing Zhang, Qinsheng Zhang, Yuxuan Zhang, Qingqing Zhao, and Artur Zolkowski. Cosmos world foundation model platform for physical ai, 2025. URL https://arxiv.org/abs/2501.03575.

Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Charles Xu, Jianlan Luo, Tobias Kreiman, You Liang Tan, Lawrence Yunliang Chen, Pannag Sanketi, Quan Vuong, Ted Xiao, Dorsa Sadigh, Chelsea Finn, and Sergey Levine. Octo: An open-source generalist robot policy. In *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, 2024.

Abby O'Neill, Abdul Rehman, Abhinav Gupta, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, et al. Open x-embodiment: Robotic learning datasets and rt-x models. *arXiv preprint arXiv:2310.08864*, 2023.

OpenAI, :, Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, Aleksander Mądry, Alex Baker-Whitcomb, Alex Beutel, Alex Borzunov, Alex Carney, Alex Chow, Alex Kirillov, Alex Nichol, Alex Paino, Alex Renzin, Alex Tachard Passos, Alexander Kirillov, Alexi Christakis, Alexis Conneau,

Ali Kamali, Allan Jabri, Allison Moyer, Allison Tam, Amadou Crookes, Amin Tootoochian, Amin Tootoonchian, Ananya Kumar, Andrea Vallone, Andrej Karpathy, Andrew Braunstein, Andrew Cann, Andrew Codispoti, Andrew Galu, Andrew Kondrich, Andrew Tulloch, Andrey Mishchenko, Angela Baek, Angela Jiang, Antoine Pelisse, Antonia Woodford, Anuj Gosalia, Arka Dhar, Ashley Pantuliano, Avi Nayak, Avital Oliver, Barret Zoph, Behrooz Ghorbani, Ben Leimberger, Ben Rossen, Ben Sokolowsky, Ben Wang, Benjamin Zweig, Beth Hoover, Blake Samic, Bob McGrew, Bobby Spero, Bogo Giertler, Bowen Cheng, Brad Lightcap, Brandon Walkin, Brendan Quinn, Brian Guarraci, Brian Hsu, Bright Kellogg, Brydon Eastman, Camillo Lugaresi, Carroll Wainwright, Cary Bassin, Cary Hudson, Casey Chu, Chad Nelson, Chak Li, Chan Jun Shern, Channing Conger, Charlotte Barette, Chelsea Voss, Chen Ding, Cheng Lu, Chong Zhang, Chris Beaumont, Chris Hallacy, Chris Koch, Christian Gibson, Christina Kim, Christine Choi, Christine McLeavey, Christopher Hesse, Claudia Fischer, Clemens Winter, Coley Czarnecki, Colin Jarvis, Colin Wei, Constantin Koumouzelis, Dane Sherburn, Daniel Kappler, Daniel Levin, Daniel Levy, David Carr, David Farhi, David Mely, David Robinson, David Sasaki, Denny Jin, Dev Valladares, Dimitris Tsipras, Doug Li, Duc Phong Nguyen, Duncan Findlay, Edede Oiwoh, Edmund Wong, Ehsan Asdar, Elizabeth Proehl, Elizabeth Yang, Eric Antonow, Eric Kramer, Eric Peterson, Eric Sigler, Eric Wallace, Eugene Brevdo, Evan Mays, Farzad Khorasani, Felipe Petroski Such, Filippo Raso, Francis Zhang, Fred von Lohmann, Freddie Sulit, Gabriel Goh, Gene Oden, Geoff Salmon, Giulio Starace, Greg Brockman, Hadi Salman, Haiming Bao, Haitang Hu, Hannah Wong, Haoyu Wang, Heather Schmidt, Heather Whitney, Heewoo Jun, Hendrik Kirchner, Henrique Ponde de Oliveira Pinto, Hongyu Ren, Huiwen Chang, Hyung Won Chung, Ian Kivlichan, Ian O'Connell, Ian O'Connell, Ian Osband, Ian Silber, Ian Sohl, Ibrahim Okuyucu, Ikai Lan, Ilya Kostrikov, Ilya Sutskever, Ingmar Kanitscheider, Ishaan Gulrajani, Jacob Coxon, Jacob Menick, Jakub Pachocki, James Aung, James Betker, James Crooks, James Lennon, Jamie Kiros, Jan Leike, Jane Park, Jason Kwon, Jason Phang, Jason Teplitz, Jason Wei, Jason Wolfe, Jay Chen, Jeff Harris, Jenia Varavva, Jessica Gan Lee, Jessica Shieh, Ji Lin, Jiahui Yu, Jiayi Weng, Jie Tang, Jieqi Yu, Joanne Jang, Joaquin Quinonero Candela, Joe Beutler, Joe Landers, Joel Parish, Johannes Heidecke, John Schulman, Jonathan Lachman, Jonathan McKay, Jonathan Uesato, Jonathan Ward, Jong Wook Kim, Joost Huizinga, Jordan Sitkin, Jos Kraaijeveld, Josh Gross, Josh Kaplan, Josh Snyder, Joshua Achiam, Joy Jiao, Joyce Lee, Juntang Zhuang, Justyn Harriman, Kai Fricke, Kai Hayashi, Karan Singhal, Katy Shi, Kavin Karthik, Kayla Wood, Kendra Rimbach, Kenny Hsu, Kenny Nguyen, Keren Gu-Lemberg, Kevin Button, Kevin Liu, Kiel Howe, Krithika Muthukumar, Kyle Luther, Lama Ahmad, Larry Kai, Lauren Itow, Lauren Workman, Leher Pathak, Leo Chen, Li Jing, Lia Guy, Liam Fedus, Liang Zhou, Lien Mamitsuka, Lilian Weng, Lindsay McCallum, Lindsey Held, Long Ouyang, Louis Feuvrier, Lu Zhang, Lukas Kondraciuk, Lukasz Kaiser, Luke Hewitt, Luke Metz, Lyric Doshi, Mada Aflak, Maddie Simens, Madelaine Boyd, Madeleine Thompson, Marat Dukhan, Mark Chen, Mark Gray, Mark Hudnall, Marvin Zhang, Marwan Aljubeh, Mateusz Litwin, Matthew Zeng, Max Johnson, Maya Shetty, Mayank Gupta, Meghan Shah, Mehmet Yatbaz, Meng Jia Yang, Mengchao Zhong, Mia Glaese, Mianna Chen, Michael Janner, Michael Lampe, Michael Petrov, Michael Wu, Michele Wang, Michelle Fradin, Michelle Pokrass, Miguel Castro, Miguel Oom Temudo de Castro, Mikhail Pavlov, Miles Brundage, Miles Wang, Minal Khan, Mira Murati, Mo Bavarian, Molly Lin, Murat Yesildal, Nacho Soto, Natalia Gimelshein, Natalie Cone, Natalie Staudacher, Natalie Summers, Natan LaFontaine, Neil Chowdhury, Nick Ryder, Nick Stathas, Nick Turley, Nik Tezak, Niko Felix, Nithanth Kudige, Nitish Keskar, Noah Deutsch, Noel Bundick, Nora Puckett, Ofir Nachum, Ola Okelola, Oleg Boiko, Oleg Murk, Oliver Jaffe, Olivia Watkins, Olivier Godement, Owen Campbell-Moore, Patrick Chao, Paul McMillan, Pavel Belov, Peng Su, Peter Bak, Peter Bakkum, Peter Deng, Peter Dolan, Peter Hoeschele, Peter Welinder, Phil Tillet, Philip Pronin, Philippe Tillet, Prafulla Dhariwal, Qiming Yuan, Rachel Dias, Rachel Lim, Rahul Arora, Rajan Troll, Randall Lin, Rapha Gontijo Lopes, Raul Puri, Reah Miyara, Reimar Leike, Renaud Gaubert, Reza Zamani, Ricky Wang, Rob Donnelly, Rob Honsby, Rocky Smith, Rohan Sahai, Rohit Ramchandani, Romain Huet, Rory Carmichael, Rowan Zellers, Roy Chen, Ruby Chen, Ruslan Nigmatullin, Ryan Cheu, Saachi Jain, Sam Altman, Sam Schoenholz, Sam Toizer, Samuel Miserendino, Sandhini Agarwal, Sara Culver, Scott Ethersmith, Scott Gray, Sean Grove, Sean Metzger, Shamez Hermani, Shantanu Jain, Shengjia Zhao, Sherwin Wu, Shino Jomoto, Shirong Wu, Shuaiqi, Xia, Sonia Phene, Spencer Papay, Srinivas Narayanan, Steve Coffey, Steve Lee, Stewart Hall, Suchir Balaji, Tal Broda, Tal Stramer, Tao Xu, Tarun Gogineni, Taya Christianson, Ted Sanders, Tejal Patwardhan, Thomas Cunninghman, Thomas Degry, Thomas Dimson, Thomas Raoux, Thomas Shadwell, Tianhao Zheng, Todd Underwood, Todor Markov, Toki Sherbakov, Tom Rubin, Tom Stasi, Tomer Kaftan,

Tristan Heywood, Troy Peterson, Tyce Walters, Tyna Eloundou, Valerie Qi, Veit Moeller, Vinnie Monaco, Vishal Kuo, Vlad Fomenko, Wayne Chang, Weiyi Zheng, Wenda Zhou, Wesam Manassra, Will Sheu, Wojciech Zaremba, Yash Patil, Yilei Qian, Yongjik Kim, Youlong Cheng, Yu Zhang, Yuchen He, Yuchen Zhang, Yujia Jin, Yunxing Dai, and Yury Malkov. Gpt-4o system card, 2024. URL https://arxiv.org/abs/2410.21276.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4195–4205, 2023.

Doina Precup, Richard S. Sutton, and Satinder P. Singh. Eligibility traces for off-policy policy evaluation. In *Proceedings of the 17th International Conference on Machine Learning*, pp. 759–766, 2000.

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

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.

Nur Muhammad Mahi Shafiullah, Anant Rai, Haritheja Etukuru, Yiqian Liu, Ishan Misra, Soumith Chintala, and Lerrel Pinto. On bringing robots home. *arXiv preprint arXiv:2311.16098*, 2023.

Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, et al. Make-a-video: Text-to-video generation without text-video data. *arXiv preprint arXiv:2209.14792*, 2022.

Fran Soljacic, Theresa Law, Meia Chita-Tegmark, and Matthias Scheutz. Robots in healthcare as envisioned by care professionals. *Intelligent Service Robotics*, pp. 1–17, 2024.

Niko Sünderhauf, Oliver Brock, Walter Scheirer, Raia Hadsell, Dieter Fox, Jürgen Leitner, Ben Upcroft, Pieter Abbeel, Wolfram Burgard, Michael Milford, et al. The limits and potentials of deep learning for robotics. *The International journal of robotics research*, 37(4-5):405–420, 2018.

Richard S Sutton, Hamid Reza Maei, Doina Precup, Shalabh Bhatnagar, David Silver, Csaba Szepesvári, and Eric Wiewiora. Fast gradient-descent methods for temporal-difference learning with linear function approximation. In *Proceedings of the 26th annual international conference on machine learning*, pp. 993–1000, 2009.

Richard S Sutton, A Rupam Mahmood, and Martha White. An emphatic approach to the problem of off-policy temporal-difference learning. *Journal of Machine Learning Research*, 17(73):1–29, 2016.

Russ Tedrake et al. Drake: Model-based design and verification for robotics. 2019.

P. Thomas, G. Theocharous, and M. Ghavamzadeh. High confidence off-policy evaluation. In *Proceedings of the 29th Conference on Artificial Intelligence*, 2015a.

Philip Thomas and Emma Brunskill. Data-efficient off-policy policy evaluation for reinforcement learning. In *International Conference on Machine Learning*, pp. 2139–2148. PMLR, 2016.

Philip Thomas, Georgios Theocharous, and Mohammad Ghavamzadeh. High-confidence off-policy evaluation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29, 2015b.

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pp. 5026–5033. IEEE, 2012.

Dani Valevski, Yaniv Leviathan, Moab Arar, and Shlomi Fruchter. Diffusion models are real-time game engines. *arXiv preprint arXiv:2408.14837*, 2024.

Ruben Villegas, Mohammad Babaeizadeh, Pieter-Jan Kindermans, Hernan Moraldo, Han Zhang, Mohammad Taghi Saffar, Santiago Castro, Julius Kunze, and Dumitru Erhan. Phenaki: Variable length video generation from open domain textual description. *arXiv preprint arXiv:2210.02399*, 2022.

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.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution, 2024. URL https://arxiv.org/abs/2409.12191.

Junjie Wen, Yichen Zhu, Jinming Li, Zhibin Tang, Chaomin Shen, and Feifei Feng. Dexvla: Vision-language model with plug-in diffusion expert for general robot control, 2025. URL https://arxiv.org/abs/2502.05855.

Chenjun Xiao, Yifan Wu, Chen Ma, Dale Schuurmans, and Martin Müller. Learning to combat compounding-error in model-based reinforcement learning. *arXiv preprint arXiv:1912.11206*, 2019.

Mengjiao Yang, Ofir Nachum, Bo Dai, Lihong Li, and Dale Schuurmans. Off-policy evaluation via the regularized lagrangian. *Advances in Neural Information Processing Systems*, 33:6551–6561, 2020.

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

Tianhe Yu, Garrett Thomas, Lantao Yu, Stefano Ermon, James Y Zou, Sergey Levine, Chelsea Finn, and Tengyu Ma. Mopo: Model-based offline policy optimization. *Advances in Neural Information Processing Systems*, 33:14129–14142, 2020.

Michael R Zhang, Tom Le Paine, Ofir Nachum, Cosmin Paduraru, George Tucker, Ziyu Wang, and Mohammad Norouzi. Autoregressive dynamics models for offline policy evaluation and optimization. *arXiv preprint arXiv:2104.13877*, 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.

Appendix

A Additional Details of the Autoregressive Diffusion Transformer

Implementation details: We use the VAE from Stable Diffusion 3 Esser et al. (2024) to independently encode 256×256 image frames into latent space. We employ a 16-layer transformer with 1024 hidden dimensions and 16 attention heads. We train the world model on a diverse set of data sources, including 9 of the robot datasets from Open-X Embodiment whose action spaces can be unified, such as Bridge V2 (Walke et al., 2023) and RT-1 (Brohan et al., 2022). We encode actions from a 7-dimensional vector, using the 6-dimensional end-effector position and binary gripper state as our action space. Action spaces from different robots are aligned by normalizing each component's 10^{th}- and 90^{th}-percentile values to those of the RT-1 dataset. We train with a context length of 20 frames; for longer rollouts, we condition on a sliding window of the last 20 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	7
Training hardware	2xA100 80GB
Training steps	300k
Diffusion noise schedule	sigmoid
Sampling timesteps	10
Prediction target	v

Table 2: Hyperparameters for training WorldGym's video prediction model.

Algorithm 1 WorldGym policy evaluation loop.

Require: World model $\hat{T}$ with training context length N_{train} and prediction horizon h, rollout length N_{rollout}, policy π with action chunk size $|\mathbf{a}_{\text{pred}}|$, reward model $\hat{R}$, initial observation o_0, goal g

$\mathbf{o} \leftarrow [o_0]$
$\mathbf{a} \leftarrow [a_{\text{null}}]$
$n = 0$
while $n \leq N_{\text{rollout}}$ **do**
  $\mathbf{a}_{\text{pred}} \leftarrow \pi(\mathbf{o}_n, g)$
  for $i = 0$ to $\lceil |\mathbf{a}_{\text{pred}}|/h \rceil - 1$ **do**
    $\mathbf{a}_{\text{ctx}} \leftarrow \mathbf{a}_{-N_{\text{train}}:}$
    $\mathbf{o}_{\text{ctx}} \leftarrow \mathbf{o}_{-N_{\text{train}}:}$
    $\mathbf{o}_{\text{pred}} \leftarrow \hat{T}(\mathbf{o}_{\text{ctx}}, \mathbf{a}_{\text{ctx}} || \mathbf{a}_{\text{pred},h\cdot i:h\cdot(i+1)})$ ▷ predict a block of h frames in parallel
    $\mathbf{o} \leftarrow \mathbf{o} || \mathbf{o}_{\text{pred}}$ ▷ concatenate generated block of observation frames with observation history
    $\mathbf{a} \leftarrow \mathbf{a} || \mathbf{a}_{\text{pred},h\cdot i:h\cdot(i+1)}$
  end for
  $n \leftarrow n + n_{\text{chunk}}$
end while
$r \leftarrow \hat{R}(\mathbf{o})$

Algorithm 1 shows the detailed algorithm for performing a sliding window rollout of a policy with action chunk size $|\mathbf{a}_{\text{pred}}|$ and a world model with prediction horizon h. Note that in practice we always choose $h = |\mathbf{a}_{\text{pred}}|$ at inference time.

B Details of VLM as Reward

B.1 Prompt for VLM as Reward

Prompt GPT-4o as Reward $\hat{R}$. Note that `has_partial` is `True` if the chosen task has a partial credit criteria, which is the case for some tasks used in OpenVLA (Kim et al.).

```
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, 0.5, or 1' if has_partial else '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.
```

If there is a partial credit criteria, the rubric is:

```
0   = Failure: little or no progress toward: "{instruction}"
0.5 = Partial: "{partial_desc}" achieved BUT the instruction
    not fully completed
1   = Success: Instruction fully completed (counts even if
    partial also true)
```

Otherwise, the rubric is:

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

B.2 Validating VLM Success Predictions

To determine whether a VLM can serve as a reliable reward function, we pass rollout videos from the RT-1 dataset, along with the prompts constructed from the templates above, as inputs to query GPT-4o. We use whether the task is successful according to the RT-1 data (validation split) as the ground truth. Table 3 shows that GPT-4o achieves high true positive and true negative rate for real

Table 3: **Performance of VLM as reward** (mean and standard error across 4 runs) on videos from RT-1 (Brohan et al., 2022) using ground truth task success labels. GPT-4o achieves high true positives and true negatives. Notably, GPT-4o as reward has very low false positive rate, which is especially important for not over-estimating a policy value.

	RT-1 Success	RT-1 Fail
VLM Success	0.81 ± 0.14 (TP)	0.03 ± 0.05 (FP)
VLM Fail	0.19 ± 0.14 (FN)	0.97 ± 0.05 (TN)

videos, indicating that it is an effective evaluator of task success. Notably, GPT-4o achieves very low false positives (i.e., the rollout is a failure but the VLM thinks it is a success), which is highly useful in policy evaluation.

C Architecture and Training Details of Video Based Policy

Our video-based policy follows the framework of UniPi Du et al. (2023a), combining a language-conditioned video prediction model with an inverse dynamics model.

The video prediction module shares the same architecture as our world model, but replaces the conditioning on robot actions at each timestep with language instructions. For language conditioning, we employ the pretrained and frozen UMT5-xxl encoder Chung et al. (2023) to obtain token-level embeddings. These embeddings are aggregated via mean pooling to form a 4096-dimensional instruction representation. This representation is projected to match the model dimensionality and is used to modulate the diffusion transformer through adaptive layer normalization (adaLN-Zero). In this way, task semantics are directly integrated into the video prediction process. We train our video generation model for 180k steps on Bridge V2 (Walke et al., 2023). The visualization of the video generation policy on validation scenes can be seen in Figure 14.

The inverse dynamics model predicts the action sequence given a short video clip of 10 frames. Each frame is encoded with a ResNet 50 backbone He et al. (2015), producing per-frame features f_t. To capture motion, we compute both f_t and temporal differences $\Delta f_t = f_{t+1} - f_t$, concatenate them, and flatten across the clip. The resulting representation is passed through an MLP to predict $10 \times d_a$ outputs, corresponding to the action dimension d_a at each timestep. Input images are normalized with ImageNet statistics, and the model is trained with mean squared error on ground-truth actions. The inverse dynamics model is trained independently on the Bridge V2 dataset for 200k steps.

C.1 Validation Visualization of Language Conditioned Video Generation Model

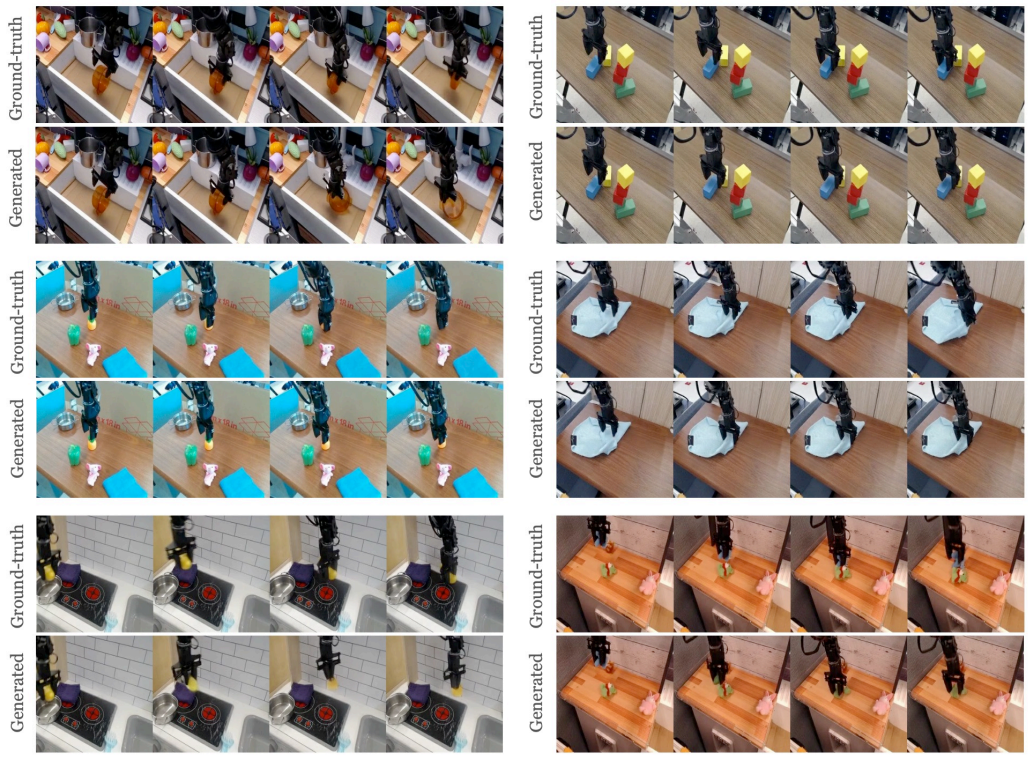


Figure 14: **Validation Visualization** of the language-conditioned video generation model on Bridge-V2. At inference, the model takes a UMT5-xxl instruction encoding and an initial frame, then predicts the next nine frames to complete the task.

D Architecture and Training Details of Diffusion Policy

We followed the recipe of DexVLA (Wen et al., 2025) for training the diffusion policy. We load the Qwen2-VL-2B (Wang et al., 2024) backbone and the pre-trained control head, and perform an adaptation on BridgeV2 using LoRA (Hu et al., 2022), which inserts low-rank adapter matrices inside the backbone's attention and feed-forward blocks. These matrices along with the policy head are the only trainable modules in the adaptation stage. This preserves the backbone's general vision-language competence, and makes adaptation compute and memory efficient. We fine-tune the model with adapters on Bridge-V2 for 60k steps. During training, we rescale the actions to $(-1, 1)$ to match the diffusion target range.

DexVLA's policy head is trained as a denoising diffusion model with the standard ϵ-prediction DDPM objective, i.e. at each update Gaussian noise is added to ground-truth action sequences at a randomly sampled diffusion step and the network is trained to predict that noise using an MSE loss. At inference, actions are generated with DDIM in a small number of steps, progressively denoising from a Gaussian initialization to a trajectory. We choose AdamW for the optimizer, using standard decoupled weight decay which applies decay to linear/attention weights, but exclude biases and LayerNorm parameters.

E Additional Experimental Results

E.1 Additional Results on Real-Robot Videos

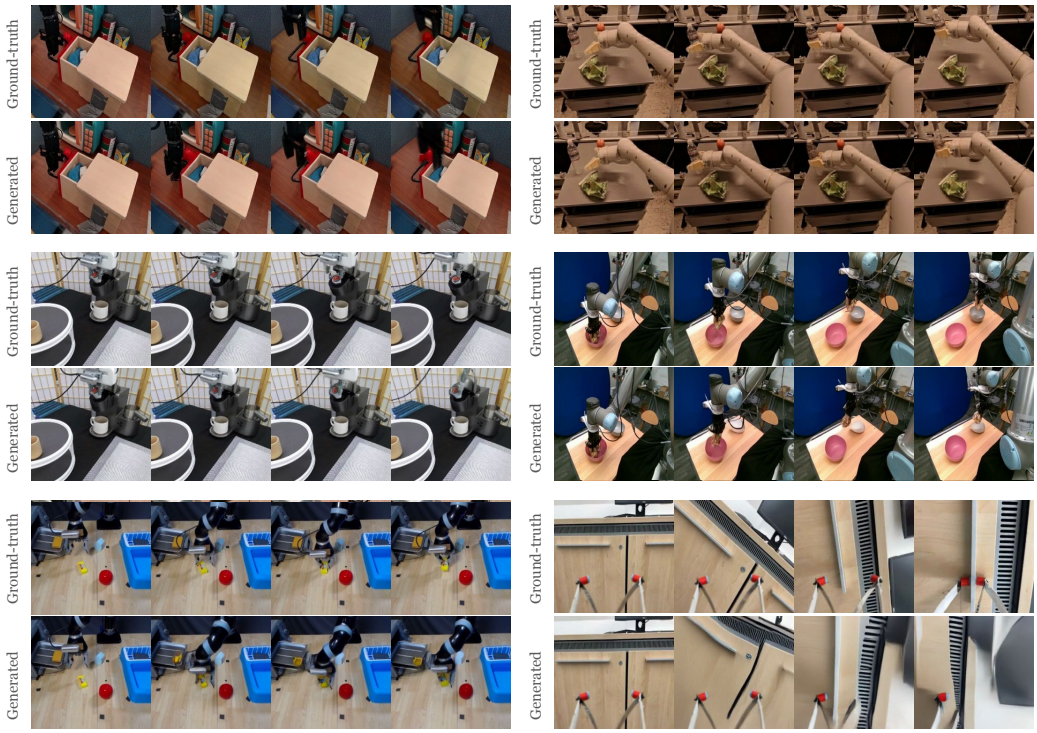


Figure 15: **Additional Qualitative Evaluation** of simulating actions from different robots. The world model generally generates the video that look very similar to the original video conditioned on the same actions that produced the original video in the real world.

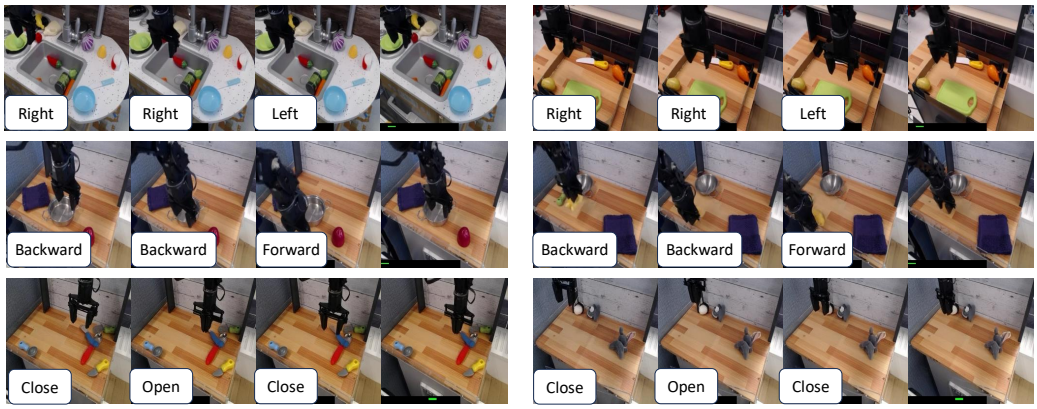


Figure 16: **Additional End-Effector Control Sweep on Bridge.** We simulate different gripper controls along different action dimensions corresponding to left-right, forward-backward, and gripper open-close. The world model generally generates videos that follow the actions.

E.2 Additional Results on Google Robot

To assess the generalizability of WorldGym, we performed rollouts with different policies on Google Robot. For our analysis, we chose a subset of tasks from the RT-1 dataset (Brohan et al., 2022). A partial score of 0.5 was assigned to a rollout if the robot attempted to reach the target location. OpenVLA again outperformed Octo and RT-1-X (see Table 4). However, in this environment Octo and RT-1-X are narrowly behind. The strong performance of RT-1-X might be due to a higher proportion of Google Robot trajectories than WidowX in its pretraining mix.

Table 4: **Policy rollouts on Google Robot** (RT-1 subset). OpenVLA outperforms RT-1-X and Octo, but by a smaller margin than on the Bridge dataset.

Task	# Trials	RT-1-X # Successes	Octo # Successes	OpenVLA # Successes
Close Bottom Drawer	10	9	8.5	6
Open Left Fridge Door	10	4.5	3.5	4
Pick Blue Chip Bag	10	5	5.5	9
Place Redbull Can Upright	10	1.5	3	3.5

E.3 Detailed Results on the OpenVLA Bridge evaluation tasks

Table 5: **Detailed Bridge Evaluation Results** comparing RT-1-X (O'Neill et al., 2023), Octo (Octo Model Team et al., 2024), and OpenVLA (Kim et al.) on the Bridge evaluation suite of tasks from Kim et al.. Real-world task success rates are taken directly from (Kim et al.), WorldGym success rates are from rolling out policies within our world model.

Task	# Trials	RT-1-X		Octo		OpenVLA	
		Real-world # Successes	WorldGym # Successes	Real-world # Successes	WorldGym # Successes	Real-world # Successes	WorldGym # Successes
Put Eggplant into Pot (Easy Version)	10	1	1	5	1	10	7
Put Eggplant into Pot	10	0	0	1	2	10	6
Put Cup from Counter into Sink	10	1	3	1	3	7	9
Put Eggplant into Pot (w/ Clutter)	10	1	0.5	3.5	3.5	7.5	8
Put Yellow Corn on Pink Plate	10	1	3	4	6	9	9.5
Lift Eggplant	10	3	2	0.5	1.5	7.5	7.5
Put Carrot on Plate (w/ Height Change)	10	2	0.5	1	3	4.5	6
Put Carrot on Plate	10	1	0	0	1	8	4
Flip Pot Upright	10	2	3	6	1	8	5
Lift AAA Battery	10	0	1	0	0	7	4
Move Skull into Drying Rack	10	1	2	0	3	5	5
Lift White Tape	10	3	1	0	1	1	6
Take Purple Grapes out of Pot	10	6	5	0	2	4	4
Stack Blue Cup on Pink Cup	10	0.5	0	0	0	4.5	6
Put {Eggplant, Red Bottle} into Pot	10	2.5	0.5	4	5	7.5	9
Lift {Cheese, Red Chili Pepper}	10	1.5	2.5	2.5	2.5	10	10
Put {Blue Cup, Pink Cup} on Plate	10	5	1.5	5.5	5	9.5	8.5
Mean Success Rate		18.5±4.0%	15.5±3.4%	20.0±5.3%	23.82±4.3%	70.6±6.1%	67.4±4.9%

We report the mean success rate across tasks with standard error (SE) computed as

$$\text{SE} = \frac{\text{sd}(r_1, \ldots, r_T)}{\sqrt{T}},$$

where r_i is the per-task success rate and T is the number of tasks.

E.4 Detailed Results on OOD Image Evaluation Tasks

Table 6: **Detailed Bridge OOD Image task results.** OpenVLA appears to be more robust across the different OOD settings of object generalization, distractions and classification.

Category	Task	# Trials	RT-1-X # Successes	Octo # Successes	OpenVLA # Successes
Object Generalization	Pick up Orange (Carrot closer to Gripper)	10	1	1	4
Object Generalization	Pick up Orange (Orange closer to Gripper)	10	3	4	9
Object Generalization	Pick up Orange (Replace Carrot with Radish)	10	1	4	10
Distractor Robustness	Pick up Carrot (With Computer on side)	10	6	7	9
Distractor Robustness	Pick up Carrot (Computer closer to gripper)	10	3	1	8
Classification	Pick {Red, Blue}	20	8	10	20
Classification	Pick {Circle, Square}	20	8	10	12
Classification	Pick {Taylor Swift, Snoop Dogg}	20	7	10	11

Table 7: **Policy rollout performance comparison in the presence of unrelated distractions.** OpenVLA is more robust to distractions over RT-1-X and Octo. However, all policies suffer significant performance drop in the presence of distractors.

Task	# Trials	RT-1-X # Successes	Octo # Successes	OpenVLA # Successes
Put Eggplant into Pot (Easy Version)	10	1	1	3
Put Eggplant into Pot	10	0	0	6
Put Cup from Counter into Sink	10	0	1	8
Put Eggplant into Pot (w/ Clutter)	10	0	0	4
Put Yellow Corn on Pink Plate	10	0	0	2
Lift Eggplant	10	0	1	7
Put Carrot on Plate (w/ Height Change)	10	0	0	2
Put Carrot on Plate	10	0	2	4
Flip Pot Upright	10	0	0	0
Lift AAA Battery	10	1	0	1
Move Skull into Drying Rack	10	3	0	5
Lift White Tape	10	0	0	4
Take Purple Grapes out of Pot	10	7	0	3
Stack Blue Cup on Pink Cup	10	0	0	4
Put {Eggplant, Red Bottle} into Pot	10	0	0	5
Lift {Cheese, Red Chili Pepper}	10	1	2	6
Put {Blue Cup, Pink Cup} on Plate	10	0	0	3
Mean Success Rate		7.6±4.3%	4.1±1.7%	39.4±5.1%

F Ablation Studies

F.1 Dataset Size Analysis

Table 8: **Dataset ablation.** Larger training dataset improves all three metrics comparing generated videos and ground-truth validation videos. ↑ means higher the better.

	Subset (Bridge V1)	Full (Bridge V2)
MSE ↓	0.015	0.010
LPIPS ↓	0.131	0.073
SSIM ↑	0.735	0.827

We measure MSE, LPIPS, and SSIM on generated videos from a model that is trained on less video data (Bridge V1 (Ebert et al., 2021)) and compare with a model that is trained on more data (Bridge V2 (Walke et al., 2023)). Table 8 shows that the model trained on more data leads to improvements in all three metrics.

F.2 Parallelism Efficiency Analysis

Table 9: **Parallelism efficiency comparison.** Inference time for generating 40-frame video rollouts on an A100 GPU with different horizon lengths, demonstrating the efficiency gains from parallel frame denoising.

Prediction Horizon	Time (s)
$h = 1$	93
$h = 4$	33

Increasing the horizon from 1 to 4 frames achieves a 2.8× speedup. This is particularly useful for evaluating robot policies with differing action chunk sizes. For instance, OpenVLA (Kim et al.) predicts just a single action per frame, while Octo (Octo Model Team et al., 2024) predicts 4 actions per frame. Using the same world model checkpoint, we can improve the efficiency of rollout generations by matching the horizon to the policy's chunk size at inference time.