

# IRASim: A Fine-Grained World Model for Robot Manipulation

Fangqi Zhu<sup>1,2</sup>, Hongtao Wu<sup>2†\*</sup>, Song Guo<sup>1\*</sup>, Yuxiao Liu<sup>2</sup>, Chilam Cheang<sup>2</sup>, Tao Kong<sup>2</sup>

<sup>1</sup>Hong Kong University of Science and Technology <sup>2</sup>ByteDance Seed

fangqi.zhu@connect.ust.hk, wuhongtao.123@bytedance.com, songguo@cse.ust.hk

## Abstract

World models allow autonomous agents to plan and explore by predicting the visual outcomes of different actions. However, for robot manipulation, it is challenging to accurately model the fine-grained robot-object interaction within the visual space using existing methods which overlooks precise alignment between each action and the corresponding frame. In this paper, we present IRASim, a novel world model capable of generating videos with fine-grained robot-object interaction details, conditioned on historical observations and robot action trajectories. We train a diffusion transformer and introduce a novel frame-level action-conditioning module within each transformer block to explicitly model and strengthen the action-frame alignment. Extensive experiments show that: (1) the quality of the videos generated by our method surpasses all the baseline methods and scales effectively with increased model size and computation; (2) policy evaluations using IRASim exhibit a strong correlation with those using the ground-truth simulator, highlighting its potential to accelerate real-world policy evaluation; (3) testing-time scaling through model-based planning with IRASim significantly enhances policy performance, as evidenced by an improvement in the IoU metric on the Push-T benchmark from 0.637 to 0.961; (4) IRASim provides flexible action controllability, allowing virtual robotic arms in datasets to be controlled via a keyboard or VR controller. Video and code are available at <https://gen-irasim.github.io/>.

## 1. Introduction

World models empower agents to foresee the outcomes of their actions by learning the fundamental dynamics of the world [1, 54]. This capability offers two key applications for robot manipulation. Firstly, it allows robots to improve autonomous policies by exploring various action proposals in the model and selecting the optimal one for executing. Secondly, world models offer the potential for scalable pol-

\*Corresponding authors.

†Project lead.

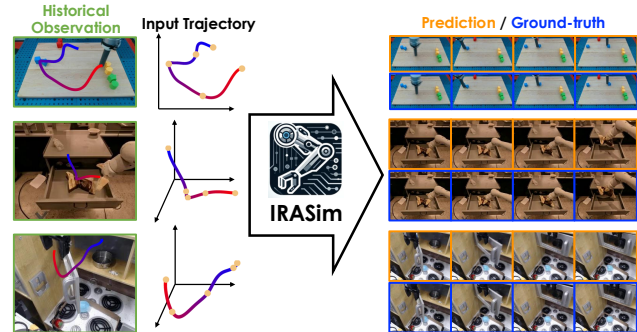


Figure 1. **Overview of IRASim.** IRASim is a fine-grained world model for robot manipulation. It generates high-fidelity videos that simulate accurate robot-object interactions of a robot executes an action trajectory, given historical observation.

icy evaluation – they can generate realistic and reasonable physical interactions, providing an efficient alternative to real-world evaluation [32].

When training a world model for robot manipulation, accurately simulating the intricate interactions between the robot, objects, and the surrounding environment remains a substantial challenge. Manipulation tasks are inherently delicate, where even subtle variations can result in task failure. Consequently, constructing a fine-grained world model is essential for faithfully capturing these precise interactions. Moreover, modern robotic manipulation policy leverages an action chunking technique [9, 57] which generates action trajectories rather than single actions to accomplish complex manipulation tasks. In this paper, we focus on building a world model that generates videos, with fine-grained robot-object interaction details, of a robot executing an action trajectory given historical observation (Fig. 1). We refer to this task as the trajectory-to-video task. Inspired by recent advances in text-to-video generation [6, 59], we leverage generative models to capture visual details that are essential for representing the intricate dynamics of the world (e.g., robot-object contact and object articulation). However, text-to-video models are trained to generate videos based on high-level textual descriptions that provide general contextual cues rather than detailed, frame-level instructions. This is different from the trajectory-to-video task in which each

action in the trajectory provides an exact description of the robot’s movement in each frame of the predicted video.

To bridge this gap, we introduce IRASim, a new world model trained with a diffusion transformer to capture complex environment dynamics. We incorporate a novel frame-level action-conditioning module within each transformer block, explicitly modeling and strengthening the alignment between each action and the corresponding frame. IRASim can generate high-fidelity videos to simulate fine-grained robot-object interactions, as shown in Fig. 1. To generate a long-horizon video that completes an entire task, IRASim can be rolled out in an autoregressive manner and maintain temporal consistency across each generated video clip.

We perform extensive experiments on four tasks to validate the effectiveness of the proposed method: 1) trajectory-conditioned video generation, 2) policy evaluation, 3) model-based planning, and 4) flexible action controllability. For trajectory-conditioned video generation, we validate IRASim on four real-robot manipulation datasets: RT-1 [5], Bridge [46], Language-Table [34], and RoboNet [10]. Results show that IRASim can generate high-quality videos of high resolution (up to  $288 \times 512$ ) and long horizon (more than 150 frames). It outperforms all the comparing baseline methods in all four datasets and is more preferable in human evaluation. In addition, it scales effectively with increased model size and computation. For policy evaluation, we evaluate autonomous policies in both IRASim and the LIBERO simulation environment [33]. The evaluation results from IRASim strongly correlate with those from the ground truth simulator, indicating great potential for scalable real-world policy evaluation. Moreover, we leverage IRASim as a visual dynamics model for model-based planning in both simulation and real-world settings. IRASim significantly improves the policy performance on accomplishing complex manipulation tasks in both settings by allowing the policy to explore various trajectory proposals and select the optimal one for execution. IRASim improves the performance (IoU metric) of a vanilla diffusion policy on the Push-T benchmark from 0.637 to 0.961. More importantly, the performance improvement scales well with increased test-time computation, highlighting a promising path towards *test-time scaling* [11] for robot manipulation. Finally, we demonstrate the flexible action controllability of IRASim by generating videos of controlling the virtual robots in the datasets via trajectories collected with a keyboard or VR controller. We recommend visiting the [project page](#) for full videos. To summarize, the contribution of this paper is threefold:

- We propose IRASim, a novel method that is capable of generating high-quality videos with fine-grained robot-object interaction details for the trajectory-to-video task. It achieves precise action-frame alignment via a novel frame-level action-conditioning module.
- We perform extensive experiments on trajectory-

conditioned video generation. Results show that IRASim outperforms all the comparing baseline methods in video generation and scales effectively with increased model size and computation.

- We showcase the usefulness of IRASim in robot manipulation through policy evaluation and policy improvement. We observe a strong correlation of evaluation results between evaluating in IRASim and the ground-truth simulator. When combined with model-based planning algorithm, IRASim improves the policy performance on accomplishing complex manipulation tasks in both simulation and the real world.

## 2. Related Work

**World Models.** Learning a world model (or dynamics model) [21, 31], which predicts future observations based on current observations and actions, has recently become increasingly popular [1, 7, 37]. In autonomous driving, world models have been used to infer future states of the environment for safe and robust driving [19, 30, 41]. World models are also leveraged as a promising approach for training safe and sample-efficient reinforcement learning agents in gaming [2, 45]. In robot manipulation, prior works [3, 20] train action-conditioned video prediction models for planning. Recently, iVideoGPT [50] proposes to train an autoregressive transformer for action-conditioned video prediction. VLP [13] and UniSim [54] use languages with action information to prompt text-to-video models for generating video. IRASim differs from these works in that it can generate high-resolution (up to  $288 \times 512$ ) and long-horizon (up to 150+ frames) videos given the initial observation and a robot trajectory, accurately capturing fine-grained robot-object interactions. It showcases strong capabilities in improving policy through model-based planning and potential for scalable policy evaluation.

**Video Models.** Video models generate video frames either unconditionally or with conditions including classes, initial frames, texts, strokes, and/or actions [4, 16, 35, 47]. Recently, diffusion models [27] are becoming more and more popular in video generation [6, 25, 28, 55, 59]. Sora [6] showcases extraordinary video generation capability with Diffusion Transformers [38]. IRASim also leverages Diffusion Transformers as the backbone. A relevant line of work is to control video synthesis with motions. These methods use either user-specified strokes [8, 56], bounding boxes [47], or human poses [48, 53] as conditions. In contrast, IRASim models complex 2D and 3D actions over timesteps via a novel frame-level action-conditioning module.

**Robot Learning with World Models.** World models hold the promise of allowing the robot to predict the ef-

fects of actions and plan solutions in complex environments [15, 18, 20, 43, 58]. For policy learning, prior works combine action-conditioned video prediction with model-predictive control for robot manipulation [14, 15, 58]. DreamerV3 [23] and DayDreamer [51] leverage recurrent state space model (RSSMs) [22] to learn a latent representation of states by modeling a world model for reinforcement learning. Recently, FLIP [18] proposed generating video plans that maximizes reward by leveraging flow prediction and then performing inverse dynamics to generate actions. This differs from the model-based planning we use in that we can predict the rewards of actions by predicting future videos, thereby selecting the optimal actions for execution. To facilitate scalable policy evaluation, recent work [32] shows a correlation between evaluation in a physical simulator and on real robots. In contrast to using a physical simulator, our work aims to leverage powerful generative models to simulate the rollouts of policies to evaluate their quality.

### 3. Methods

#### 3.1. Problem Statement

We define the trajectory-to-video task as predicting the video of a robot that executes a trajectory  $\mathbf{a}^{t:t+n}$  given the historical observation images  $\mathbf{I}^{t-h:t}$ :

$$\mathbf{I}^{t+1:t+n+1} = f(\mathbf{I}^{t-h:t}, \mathbf{a}^{t:t+n}) \quad (1)$$

where  $h$  denotes the number of historical frames;  $n$  denotes the number of actions in the video;  $\mathbf{a}^i \in \mathbb{R}^d$  denotes the action at the  $i$ -th timestep. In this paper, we focus on predicting videos for robot arms. A typical action space of a robot arm contains 7 degrees of freedom (DoFs), *i.e.*,  $\mathbf{a}^i \in \mathbb{R}^7$ , where 3 DoFs represent translation in the 3D space, 3 DoFs correspond to 3D rotation, and 1 DoF accounts for the gripper action. Additional details regarding the number of historical frames  $h$  and action space dimension  $d$  are provided in Appendix B.

#### 3.2. Preliminaries

Before delving into our method, we briefly review preliminaries of diffusion models [27, 42]. Diffusion models typically consist of a forward process and a reverse process. The forward process gradually adds Gaussian noises to data  $\mathbf{x}_0$  over  $T$  timesteps. It can be formulated as  $q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, 1 - \bar{\alpha}_t\mathbf{I})$ , where  $\mathbf{x}_t$  is the diffused data at the  $t$ -th diffusion timestep and  $\bar{\alpha}_t$  is a constant defined by a variance schedule. The reverse process starts from  $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$  and gradually remove noises to recover  $\mathbf{x}_0$ . It can be mathematically expressed as  $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \mu_\theta(\mathbf{x}_t, t), \Sigma_\theta(\mathbf{x}_t, t))$ , where  $\mu_\theta(\cdot)$  and  $\Sigma_\theta(\cdot)$  denote the mean and covariance functions, respectively, and can be parameterized via a neural network.

In the training phase, we sample a timestep  $t \in [1, T]$  and obtain  $\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon_t$  via the reparameterization trick [27] where  $\epsilon_t \in \mathcal{N}(\mathbf{0}, \mathbf{I})$ . We leverage the simplified training objective to train a noise prediction model  $\epsilon_\theta$  as in DDPM [27]:

$$\mathcal{L}_{\text{simple}}(\theta) = \|\epsilon_\theta(\mathbf{x}_t, t) - \epsilon_t\|^2 \quad (2)$$

In the inference phase, we generate  $\mathbf{x}_0$  by first sampling  $\mathbf{x}_T$  from  $\mathcal{N}(\mathbf{0}, \mathbf{I})$  and iteratively compute

$$\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left( \mathbf{x}_t - \frac{1 - \alpha_t}{\sqrt{1 - \bar{\alpha}_t}} \epsilon_\theta(\mathbf{x}_t, t) \right) \quad (3)$$

until  $t = 0$ . For conditional diffusion processes, the noise prediction model  $\epsilon_\theta$  can be parameterized as  $\epsilon_\theta(\mathbf{x}_t, t, \mathbf{c})$  where  $\mathbf{c}$  is the condition that controls the generation process. Throughout the paper, we use superscript and subscript to indicate the timestep of a frame in the input video and the diffusion timestep, respectively.

However, directly diffusing the entire video in the pixel space is time-consuming and requires substantial computation to generate long videos with high resolutions [28]. Inspired by Ma et al. [35], we perform the diffusion process in a low-dimension latent space  $\mathbf{z}$  instead of the pixel space for computation efficiency. Following He et al. [25], we leverage the pre-trained variational autoencoder (VAE) in SDXL [39] to compress each frame  $\mathbf{I}^t$  in the video to a latent representation with the VAE encoder  $\mathbf{z}^t = \text{Enc}(\mathbf{I}^t)$ . The latent representation can be decoded back to the pixel space with the VAE decoder  $\mathbf{I}^t = \text{Dec}(\mathbf{z}^t)$ .

#### 3.3. IRASim

IRASim is a conditional diffusion model operating in the latent space of the VAE introduced in Sec. 3.2. The condition  $\mathbf{c}$  consists of the latent representation of the historical frames,  $\mathbf{z}^{t-h:t} = \text{Enc}(\mathbf{I}^{t-h:t})$ , and an action trajectory,  $\mathbf{a}^{t:t+n}$ . The diffusion target is the latent representations of the subsequent  $n$  frames of the video in which the robot executes the action trajectory, *i.e.*  $\mathbf{x} = \mathbf{z}^{t+1:t+n+1}$ . Inspired by Sora’s remarkable capability of understanding the physical world [6], we similarly adopt Diffusion Transformers (DiT) [38] as the backbone of IRASim. In the design of IRASim, we aim to address three key aspects: 1) consistency with the given historical frames 2) adherence to the given action trajectory and 3) computation efficiency. In the following, we describe pivotal design choices to achieve the aforementioned objectives.

Standard transformer blocks apply Multi-Head Self-Attention (MHA) to all tokens in the input token sequence, resulting in quadratic computation cost. We thus leverage the memory-efficient spatial-temporal attention mechanism [7, 35, 52] in the transformer blocks of IRASim to reduce the computation cost (Fig. 2). The historical frame

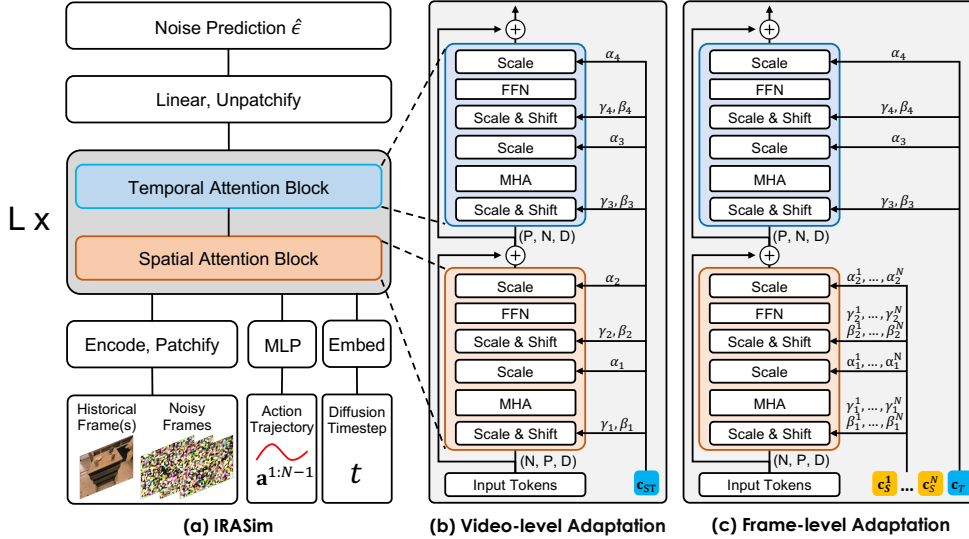


Figure 2. **Network Architecture of IRASim.** (a) shows the general diffusion transformer architecture of IRASim. The input to IRASim includes the historical frames and the given trajectory. (b) Video-level adaptation (Video-Ada). (c) Frame-level adaptation (Frame-Ada).

condition is achieved by treating the historical frames as the ground-truth portion in the input video sequence [6]. That is, during training, we only add noise to the tokens corresponding to the predicted frames  $\mathbf{z}^{t+1:t+n+1}$ , while keeping those of the historical frame  $\mathbf{z}^{t-h:t}$  intact as it does not need to be predicted (Fig. 2). And the diffusion loss is only computed upon the predicted frames. This condition approach ensures consistency with the historical frames by enabling the predicted frames to interact with them via attention mechanism.

To inject the trajectory condition into video generation, we follow Diffusion Transformers [38] and utilize adaptive layer normalization for conditioning. Below, we outline two methods for incorporating the trajectory condition.

- *Video-Level Condition.* Similar to using a text embedding to condition the generation of the entire video in the text-to-video task, we use a linear layer to encode the trajectory into a single embedding for condition. The embedding is then added to the embedding of the diffusion timestep  $t$  for generating the scale parameters  $\gamma$  and  $\alpha$  and the shift parameters  $\beta$  for each spatial and temporal attention block. The overall framework is shown in Figure 2(b). See Appendix C.1 for more details.
- *Frame-Level Condition.* Unlike the text-to-video task where the text describes the entire video, the trajectory in the trajectory-to-video task is a finer description. Each action in the trajectory defines how the robot should move in each frame. And thus, each generated frame must match with its corresponding action in the trajectory. To achieve this precise frame-level alignment, we condition the generation of each frame by its corresponding action. Instead of encoding the action trajectory into a single embedding, we use a linear layer to encode each action into an individual embedding. The diffusion timestep embedding is added

to each action embedding to generate the scale and shift parameters for each individual frame in the spatial block. The scale and shift parameters of the temporal block for all frames share the same conditioning embedding which is derived similarly as in video-level condition. The overall framework is shown in Figure 2(c). See Appendix C.2 for more details.

The output layer contains a linear layer which outputs the noise prediction  $\hat{\epsilon} = \epsilon_\theta(\mathbf{x}_t, t, \mathbf{c})$ .  $\hat{\epsilon}$  is used to compute the L2 loss with the ground-truth noise during training (Eq. 2). Note that the VAE is frozen during the whole training process. During inference, we sample  $\mathbf{x}^T$  from  $\mathcal{N}(\mathbf{0}, \mathbf{I})$  and gradually denoise it via Eq. 3 to obtain the latent representation of the predicted frames  $\hat{\mathbf{z}}^{t+1:t+n+1} = \mathbf{x}_0$ . The predicted frames can be decoded with the VAE decoder  $\hat{\mathbf{I}}^{t+1:t+n+1} = \text{Dec}(\hat{\mathbf{z}}^{t+1:t+n+1})$ .

## 4. Experiments

We perform extensive experiments to validate the effectiveness of IRASim. We aim to answer three main questions: 1) Is IRASim effective on modeling fine-grained robot-object interactions and solving the trajectory-to-video task on various real-robot datasets with different action spaces? 2) Can we leverage IRASim as a world model for policy evaluation on manipulation tasks? 3) Can we utilize IRASim for model-based planning and improve flat autonomous policies on manipulation tasks? We also perform extensive ablation studies to analyze the contribution of different components of the proposed method.

### 4.1. Trajectory-Conditioned Video Prediction

**Experiment Setup** We conduct experiments on four real-robot manipulation datasets: RT-1 [5], Bridge [46],

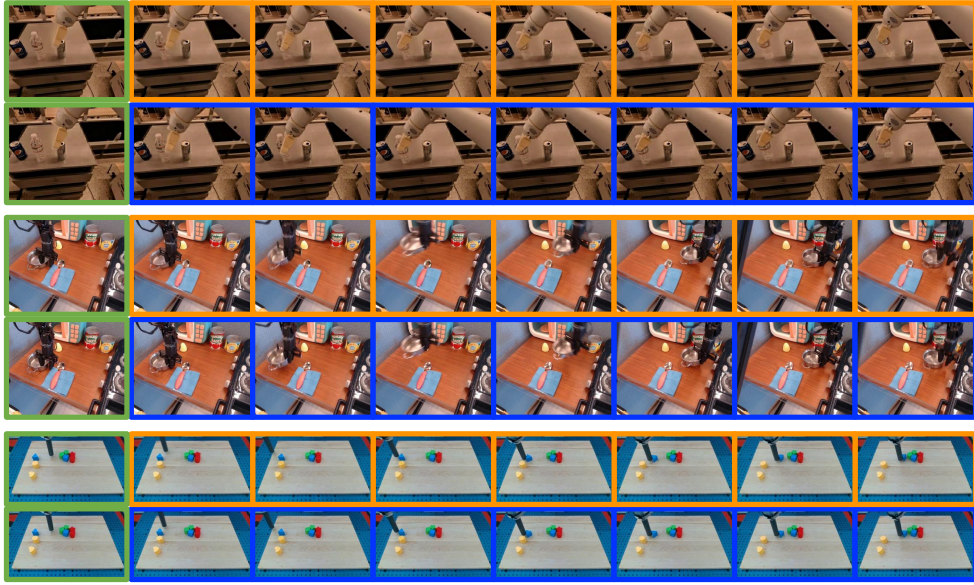


Figure 3. **Qualitative Results of Short-Trajectory Video Generation.** We show video generation of IRASim with short trajectories on the test set of RT-1, Bridge, and Language-Table. Ground-truths are in blue boxes. Predictions are in orange boxes. Initial ground-truth video frames are in green boxes. Please see our [project page](#) for videos.

Language-Table [34], and RoboNet [10]. The action space varies across datasets, with RT-1 and Bridge using 7 DoF, Language-Table 2 DoF, and RoboNet up to 5 DoF. Details of each dataset are provided in Appendix B. For RT-1, Bridge, and Language-Table, we use 1 historical frame and 15 actions as context to predict the next 15 frames. For RoboNet, we follow iVideoGPT [50] and use 2 historical frames and 10 actions to predict the next 10 frames. Videos are resized to  $256 \times 320$  for RT-1 and Bridge,  $288 \times 512$  for Language-Table, and  $256 \times 256$  for RoboNet. We evaluate video generation on both short and long trajectories. Short trajectories contains  $n$  actions and the videos can be generated in a single generation pass. Long trajectories consists of more actions and the videos are generated autoregressively over multiple passes. The final generated frame from the previous pass serves as the conditional historical frame for the current one. We denote video-level and frame-level adaptation as IRASim-Video-Ada and IRASim-Frame-Ada, respectively, and refer to them as Frame-Ada and Video-Ada for brevity. Training details can be found in Appendix E.

**Baselines.** We compare IRASim with two state-of-the-art methods, *i.e.*, VDM [28] and LVDM [25]. Both methods are diffusion models based on a U-Net architecture. This is in contrast to IRASim, which employs a Transformer architecture. LVDM diffuses videos in a latent space, while VDM operates in the pixel space. To impose trajectory conditions on video generation, we encode the trajectory into an embedding to condition the diffusion process for both methods. This is similar to the text embedding used for text-to-video generation in the original papers [25, 28]. Additionally, we

Dataset	Method	Computation-based		Model-based		
		PSNR $\uparrow$	SSIM $\uparrow$	Latent L2 $\downarrow$	FID $\downarrow$	FVD $\downarrow$
RT-1	VDM	13.762	0.554	0.4983	41.23	371.13
	LVDM	25.041	0.815	0.2244	<b>4.26</b>	30.72
	Video-Ada	<u>25.446</u>	<u>0.823</u>	0.2191	<u>4.34</u>	<u>29.27</u>
	<b>Frame-Ada (Ours)</b>	<b>26.048</b>	<b>0.833</b>	<b>0.2099</b>	5.60	<b>25.58</b>
Bridge	VDM	18.520	0.741	0.3709	39.82	127.25
	LVDM	23.546	0.810	0.2155	10.59	35.06
	Video-Ada	<u>24.733</u>	<u>0.827</u>	<u>0.2021</u>	<b>10.30</b>	<u>23.03</u>
	<b>Frame-Ada (Ours)</b>	<b>25.275</b>	<b>0.833</b>	<b>0.1947</b>	10.51	<b>20.91</b>
Language-Table	VDM	23.067	0.857	0.3204	64.63	136.56
	LVDM	<u>28.254</u>	<b>0.889</b>	0.1704	6.85	<b>24.34</b>
	Video-Ada	23.893	0.859	0.2028	7.05	73.84
	<b>Frame-Ada (Ours)</b>	<b>28.818</b>	<u>0.888</u>	<b>0.1660</b>	<b>6.38</b>	48.49

Table 1. **Quantitative results for short-trajectory video generation.** We prioritize Latent L2 and PSNR as the primary evaluation metrics. Video-Ada and Frame-Ada are variants of IRASim.

compare with two state-of-the-art non-diffusion methods, iVideoGPT [50] and MaskViT [20], on the RoboNet dataset. iVideoGPT autoregressively predicts the next visual token; MaskViT generates visual tokens via an iterative refinement process. More details about baselines can be found in Appendix D.

**Metrics.** Following [53], we evaluate the performance with two types of metrics: computation-based (PSNR [29], SSIM [49]) and model-based (Latent L2 loss, FID [26], and FVD [44]). Unlike the text-to-video task, where various videos may satisfy a single text condition, the trajectory-to-video task has much less variation: the robot in the predicted video must strictly follow the input trajectory. Therefore, we use video reconstruction metrics, Latent L2 loss and PSNR, as the primary evaluation metrics. In Appendix H, we show-case that Latent L2 loss and PSNR best align with human

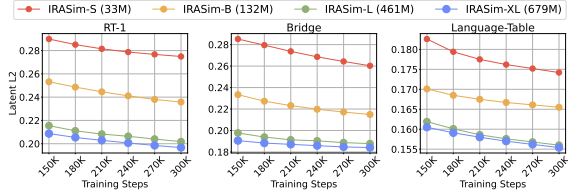


Figure 4. **Scaling.** IRASim scales effectively with the increase of model sizes and training steps.

Method	PSNR $\uparrow$	SSIM $\uparrow$
MaskViT [20]*	20.4	67.1
iVideoGPT [50]*	23.8	80.8
<b>IRASim (Ours)</b>	<b>24.6</b>	<b>81.1</b>

Table 2. **Quantitative results for video generation on RoboNet dataset.** \* indicates that the result is derived from [50].

preferences among all the evaluated metrics. More details about evaluation can be found in Appendix F.

**Results.** The qualitative results of short-trajectory video generation are shown in Fig. 3. And the quantitative results are shown in Tab. 1 & 2. The qualitative and quantitative results of long-trajectory video generation are shown in Appendix A.2. IRASim-Frame-Ada effectively models fine-grained robot-object interactions and generates high-quality videos that closely align with the ground truth. It surpasses all the comparing baseline methods in our primary evaluation metrics, Latent L2 and PSNR, as well as the human evaluation in Appendix H. As illustrated in Appendix A.1 & A.2, baseline methods struggle to guide the robot arm along the given trajectory and fail to realistically simulate interactions between the robot and the objects.

**Scaling.** We follow [38] and train IRASim-Frame-Ada with different model sizes, ranging from 33M to 679M. Detailed parameters of these models are shown Appendix E. Results are shown in Fig. 4. Across all three test datasets, IRASim scales effectively with larger model size and more training steps, highlighting its strong potential for further performance gains through increased computation.

## 4.2. Policy Evaluation

In this section, we showcase that we can use IRASim as a simulator for policy evaluation. We use the LIBERO simulation benchmark [33] as a controlled environment for this experiment. In particular, we evaluate a diffusion policy [9, 12] in IRASim and compare the evaluation results against those with the ground-truth simulator. We train the diffusion policy on expert trajectories provided by the benchmark. An evaluator must be able to simulate both successful and failed rollouts. And the world model needs to learn from a broader set of data than the expert demonstrations, which contain only successful rollouts, in order to simulate both successes and failures accurately. Thus, we deploy the trained policy

in the simulator to gather additional rollouts which contains both successes and failures. We refer to these rollouts as *post-trained rollouts*. The post-trained rollouts, along with the expert demonstrations, are used for training IRASim. Given the limited amount of training data, we initialize IRASim with the pre-trained weight of OpenSora [59] to expedite the training process. We incorporate our frame-level condition method (Sec. 3.3) to inject the trajectory condition into the model for trajectory-conditioned video generation.

We train the diffusion policy with four different steps on the task of "pick up the black bowl between the plate and the ramekin and place it on the plate", resulting in four different individual models. We then evaluate the performance of these four models in both the Mujoco simulator of the LIBERO benchmark and IRASim. The Mujoco simulator serves as a ground truth for comparison. We evaluate each model in both IRASim and the ground-truth simulator for 50 runs each. The rollouts generated by IRASim were assessed by humans to determine their success or failure.

Fig. 5 shows successful and failed rollouts generated by IRASim. Notably, IRASim successfully simulates scenarios where the bowl slips from the gripper, demonstrating strong capabilities to model fine-grained robot-object interaction. Tab. 3 reports the success rates of different models evaluated with the ground-truth Mujoco simulator and IRASim. The Pearson correlation coefficient between the two evaluation results is 0.99, indicating a strong correlation between evaluating in the ground-truth Mujoco simulator and IRASim. This result showcases the potential of leveraging IRASim as a world model for scalable real-world policy evaluation.

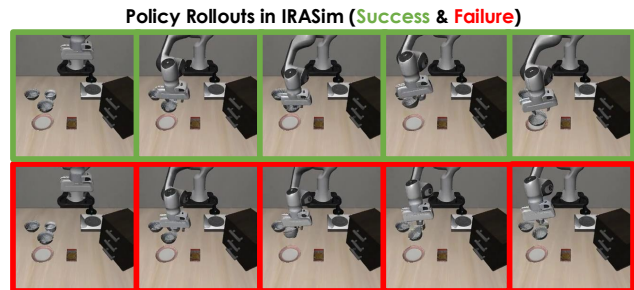


Figure 5. **Policy Evaluation with IRASim.** IRASim can simulate both successful and failed rollouts. Notably, it is able to simulate a bowl slipping from the gripper.

Evaluator	1	2	3	4
Ground-Truth Simulator	0.18	0.50	0.80	1.00
<b>IRASim (Ours)</b>	<b>0.28</b>	<b>0.48</b>	<b>0.74</b>	<b>0.96</b>

Table 3. Success rates of four different models evaluated in the two evaluators. We observe a strong correlation between the evaluation results from the ground-truth Mujoco simulator and IRASim. The Pearson correlation coefficient between the two evaluations is **0.99**.

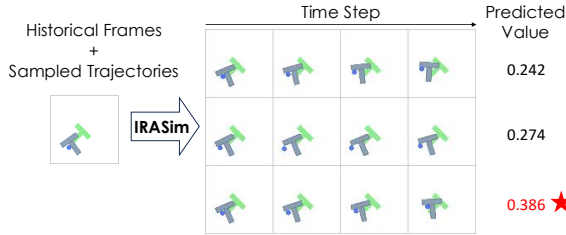


Figure 6. IRASim simulates the visual outcomes of different trajectories sampled from the policy and predicts the value of the final frame with a reward model. By selecting and executing the trajectory with the highest predicted value, we enhance the existing policy by spending more time thinking (test-time compute).

	$P$	$K = 1$	$K = 5$	$K = 10$	$K = 50$
GPC-RANK	N/A*	0.642	-	-	0.698
GPC-RANK+OPT	N/A*	0.642	0.824	0.882	-
<b>IRASim (Ours)</b>	0	0.637	0.679	0.572	0.418
	100	0.637	0.847	0.878	0.888
	200	0.637	0.866	0.916	0.912
	500	0.637	<b>0.907</b>	0.906	0.938
	1000	0.637	0.886	<b>0.945</b>	<b>0.961</b>

Table 4. **Results on Push-T Benchmark.**  $K$  denotes the number of sampled trajectory.  $P$  denotes the number of post-trained rollouts used for training IRASim. We report the average IoU over 100 trials. \*GPC also uses additional rollouts beyond expert demonstrations to train the world model, but the number of these rollouts is not available in the paper [40].

### 4.3. Model-based Planning for Policy Improvement

In this section, we perform experiments in both simulation and real-world settings to show that IRASim can be used as a world model for model-based planning to improve vanilla policies without planning on accomplishing complex manipulation tasks. Specifically, we adapt a simple ranking algorithm for model-based planning: 1) samples  $K$  trajectories from the policy, 2) unroll each trajectory in IRASim, and 3) select the trajectory with the highest value for executing.

**Push-T Simulation.** In this experiment, we use the Push-T simulation benchmark from [17] for evaluation. The robot is tasked to push a T-shaped block (gray) to a target (green) with a circular end-effector (blue) (Fig. 6). In order to perform effective model-based planning, a challenge is that the world model need to accurately predict the complex dynamics of robot-block contact. We first train a diffusion policy with 200 expert demonstrations. Similar to Sec. 4.2, we then collect post-trained rollouts, which contains both successful and failed rollouts, with the trained policy. We use intersection over union (IoU) between the block and the target as the value function for model-based planning. To predict the IoU of a given observation, we train a ResNet50 model [24] using the post-trained rollouts. Similar to the experiments in

Sec. 4.2, we initialize IRASim with the pre-trained weights of OpenSora and train it on both post-trained rollouts and expert demonstrations. We perform ablation studies to analyze the effect of varying the number of post-trained rollouts (denoted as  $P$ ) on overall performance.

We compare with a recent state-of-the-art method, generative predictive control (GPC) [40]. GPC perform autoregressive next-frame prediction via diffusion to generate a video. This contrasts with our trajectory-to-video approach, which generates all frames for a trajectory simultaneously. Similar to IRASim, GPC also enhances its video prediction with additional rollouts beyond expert demonstrations. And it also uses a diffusion policy to generate action proposals. Specifically, we compare with two variants of GPC introduced in [40], *i.e.*, GPC-RANK and GPC-RANK+OPT. GPC-RANK uses a similar ranking-based planning algorithm as our method. GPC-RANK+OPT utilizes a differentiable reward model to optimize action proposals via gradient optimization. In Tab. 4,  $M$  denotes the number of gradient optimization steps, and GPC-RANK+OPT represents the approach that incorporates both the RANK method and gradient optimization.

Results are shown in Tab. 4. The  $K = 1$  column show the performance of the vanilla diffusion policy without model-based planning. To ensure a fair comparison, we train our diffusion policy such that its IoU performance matches with that of reported in the GPC paper [40], *i.e.*, 0.637 v.s. 0.642. Using 200 post-trained rollouts, IRASim outperforms the two GPC variants. And the advantage grows as more post-trained rollouts are used. In addition, when  $K = 50, P = 1000$ , IRASim improves the IoU of the vanilla policy from 0.637 to 0.961. We further explore the effect of varying  $K$  and  $P$ . When  $P > 0$ , the policy performance consistently improves as the number of sampled trajectories  $K$  increases. This highlights the importance of including post-trained rollouts in training the world model. More importantly, this result indicates we can robustly improve policy performance by scaling up the number of sampled trajectories for ranking, highlighting a promising path toward *test-time scaling* [11] for robot manipulation. With the increase of  $P$ , the performance consistently improves for larger  $K$ . For smaller  $K$ , the performance initially improves and then reaches a plateau when  $P = 1000$ . This results indicate that data size and test-time computation should scale simultaneously.

**Real-Robot Experiments.** We train IRASim on a real-robot dataset and perform experiments on three different tasks in the training dataset. We leverage a goal-conditioned method which specifies the task via a goal image. We use the similarity between the final image of the predicted video and the goal image as the value function for model-based planning. We use a simple policy which samples 50 individual points from a sphere centered on the current end-effector

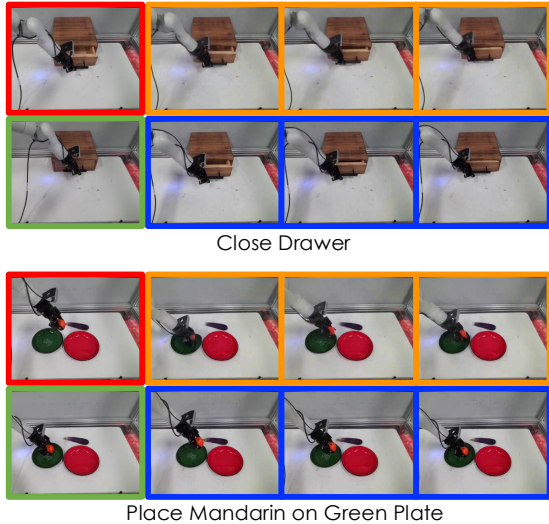


Figure 7. **Qualitative Results on Real-Robot Model-based Planning.** Historical frames are highlighted in red boxes, goal images in green boxes, real-robot rollouts in blue boxes, and videos generated by IRASim are shown in orange boxes. IRASim can generate videos, that faithfully matches with the ground truth in tasks involving object transportation and articulated object manipulation, enabling effective model-based planning.

Method	Close Drawer	Place Mandarin on Green Plate	Place Mandarin on Red Plate
Random	0.20	0.07	0.13
IRASim (ResNet50)	0.60	0.73	0.60
IRASim (MSE)	<b>0.87</b>	<b>0.80</b>	<b>0.87</b>

Table 5. **Quantitative Results on Real-Robot Model-based Planning.**

position and generates a trajectory between the current position and each sampled point, resulting in  $K = 50$  different sampled trajectories.

Qualitative results are shown in Fig. 7. Quantitative results are shown in Tab. 5. We experiment with two functions for similarity comparison: 1) mean squared error (MSE) and 2) cosine similarity of the feature extracted from ResNet50. We observe that the MSE for value functions significantly outperformed the ResNet counterpart. And both variants significantly outperform the policy without planning which randomly selects a trajectory for rollout. These results demonstrate the potential of IRASim as a real-world manipulation world model for model-based planning. More details and discussion can be found in Appendix G.

#### 4.4. Flexible Action Controllability

In this section, we perform qualitative experiments in which we “control” the virtual robot in two datasets, Language-Table [34] and RT-1 [5], using trajectories collected with two distinct input sources: a keyboard and a VR controller. Notably, the trajectories collected through these input sources

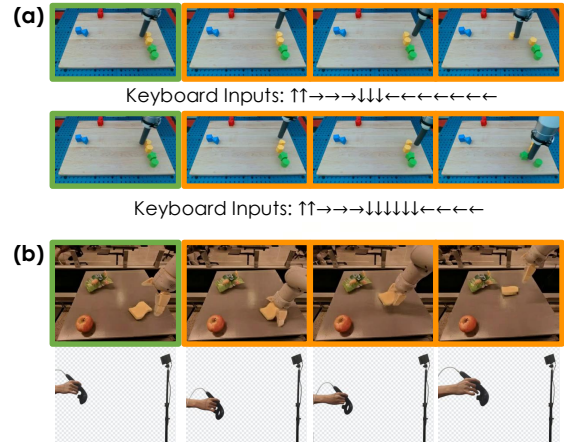


Figure 8. **Flexible Action Controllability.** We showcase controlling (a) the virtual robot in Language-Table with arrow keys on a keyboard and (b) the robot in RT-1 with a VR controller. Predictions are in orange boxes; initial frames are in green boxes.

exhibit distributions that deviate from those in the original dataset. For Language-Table with a 2D translation action space, we use the arrow keys on the keyboard to input action trajectories. For RT-1 with a 3D action space, we use a VR controller to collect action trajectories as input. Specifically, we prompt IRASim with an image from each dataset and a trajectory collected with the keyboard or VR controller. Fig. 8 shows the video generated by IRASim. IRASim is able to follow trajectories collected with different input sources and simulate robot-object interaction in a realistic and reasonable way. More importantly, it is able to robustly handle multimodality in generation. Fig. 8(a) shows videos generated with an identical initial frame but different trajectories. In Appendix A.4, we demonstrate that IRASim can also handle physically implausible trajectories robustly.

## 5. Conclusion, Limitation and Future Work

In this paper, we present IRASim, a novel world model that generates videos, with fine-grained robot-object interaction details, of a robot executing an action trajectory given historical observation. We achieve precise alignments between actions and video frames via a novel frame-level action-conditioning module. Extensive experiments show the videos quality generated by IRASim is able to generate long-horizon and high-resolution videos that accurately simulate the robot trajectory rollouts. Additionally, we showcase that IRASim can be leveraged as a simulator for policy evaluation and a dynamics model for model-based planning to improve policy performance. Similar to many other generative models, a limitation of IRASim is video generation is not real-time. In the future, we plan to explore leveraging diffusion distillation [36] to accelerate generation speed. In addition, we plan to investigate utilizing IRASim as a dynamics model and improve robot policies within the world model via reinforcement learning [2].

**Acknowledgments.** We are grateful to Dr. Zicong Hong for helping revise the paper and for the constructive feedback. This research was supported by fundings from the Hong Kong RGC General Research Fund (152169/22E, 152228/23E, 162161/24E), Research Impact Fund (No. R5060-19, No. R5011-23), Collaborative Research Fund (No. C1042-23GF), NSFC/RGC Collaborative Research Scheme (Grant No. 62461160332 & CRS\_HKUST602/24), Areas of Excellence Scheme (AoE/E-601/22-R), and the InnoHK (HKGAI).

## References

- [1] Niket Agarwal, Arslan Ali, Maciej Bala, Yogesh Balaji, Erik Barker, Tiffany Cai, Prithvijit Chattopadhyay, Yongxin Chen, Yin Cui, Yifan Ding, et al. Cosmos world foundation model platform for physical ai. *arXiv preprint arXiv:2501.03575*, 2025. 1, 2
- [2] Eloi Alonso, Adam Jelley, Vincent Micheli, Anssi Kanervisto, Amos J Storkey, Tim Pearce, and François Fleuret. Diffusion for world modeling: Visual details matter in atari. *Advances in Neural Information Processing Systems*, 37:58757–58791, 2025. 2, 8
- [3] Mohammad Babaeizadeh, Mohammad Taghi Saffar, Suraj Nair, Sergey Levine, Chelsea Finn, and Dumitru Erhan. Fitvid: Overfitting in pixel-level video prediction, 2021. 2
- [4] Fan Bao, Chendong Xiang, Gang Yue, Guande He, Hongzhou Zhu, Kaiwen Zheng, Min Zhao, Shilong Liu, Yaole Wang, and Jun Zhu. Vidu: a highly consistent, dynamic and skilled text-to-video generator with diffusion models, 2024. 2
- [5] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Tomas Jackson, Sally Jesmonth, Nikhil J Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Kuang-Huei Lee, Sergey Levine, Yao Lu, Utsav Malla, Deeksha Manjunath, Igor Mordatch, Ofir Nachum, Carolina Parada, Jodilyn Peralta, Emily Perez, Karl Pertsch, Jornell Quiambao, Kanishka Rao, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Kevin Sayed, Jaspier Singh, Sumedh Sontakke, Austin Stone, Clayton Tan, Huong Tran, Vincent Vanhoucke, Steve Vega, Quan Vuong, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. Rt-1: Robotics transformer for real-world control at scale, 2023. 2, 4, 8
- [6] Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, Clarence Ng, Ricky Wang, and Aditya Ramesh. Video generation models as world simulators. 2024. 1, 2, 3, 4
- [7] Jake Bruce, Michael Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, et al. Genie: Generative interactive environments. *arXiv preprint arXiv:2402.15391*, 2024. 2, 3
- [8] Tsai-Shien Chen, Chieh Hubert Lin, Hung-Yu Tseng, Tsung-Yi Lin, and Ming-Hsuan Yang. Motion-conditioned diffusion model for controllable video synthesis. *arXiv preprint arXiv:2304.14404*, 2023. 2
- [9] Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Proceedings of Robotics: Science and Systems (RSS)*, 2023. 1, 6
- [10] Sudeep Dasari, Frederik Ebert, Stephen Tian, Suraj Nair, Bernadette Bucher, Karl Schmeckpeper, Siddharth Singh, Sergey Levine, and Chelsea Finn. Robonet: Large-scale multi-robot learning. In *Proceedings of the Conference on Robot Learning*, pages 885–897. PMLR, 2020. 2, 5
- [11] DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Shengfeng Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanjia Zhao, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yao-hui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanhong Xu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025. 2, 7

- [12] Zibin Dong, Yifu Yuan, Jianye HAO, Fei Ni, Yi Ma, Pengyi Li, and YAN ZHENG. Cleandiffuser: An easy-to-use modularized library for diffusion models in decision making. In The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track, 2024. 6
- [13] Yilun Du, Sherry Yang, Pete Florence, Fei Xia, Ayzaan Wahid, brian ichter, Pierre Sermanet, Tianhe Yu, Pieter Abbeel, Joshua B. Tenenbaum, Leslie Pack Kaelbling, Andy Zeng, and Jonathan Tompson. Video language planning. In The Twelfth International Conference on Learning Representations, 2024. 2
- [14] Frederik Ebert, Chelsea Finn, Sudeep Dasari, Annie Xie, Alex Lee, and Sergey Levine. Visual foresight: Model-based deep reinforcement learning for vision-based robotic control. arXiv preprint arXiv:1812.00568, 2018. 3
- [15] Chelsea Finn and Sergey Levine. Deep visual foresight for planning robot motion. In 2017 IEEE International Conference on Robotics and Automation (ICRA), pages 2786–2793. IEEE, 2017. 3
- [16] Chelsea Finn, Ian Goodfellow, and Sergey Levine. Unsupervised learning for physical interaction through video prediction. Advances in neural information processing systems, 29, 2016. 2
- [17] Pete Florence, Corey Lynch, Andy Zeng, Oscar A Ramirez, Ayzaan Wahid, Laura Downs, Adrian Wong, Johnny Lee, Igor Mordatch, and Jonathan Tompson. Implicit behavioral cloning. In Conference on robot learning, pages 158–168. PMLR, 2022. 7
- [18] Chongkai Gao, Haozhuo Zhang, Zhixuan Xu, Zhehao Cai, and Lin Shao. Flip: Flow-centric generative planning for general-purpose manipulation tasks. arXiv preprint arXiv:2412.08261, 2024. 3
- [19] Shenyan Gao, Jiazhi Yang, Li Chen, Kashyap Chitta, Yihang Qiu, Andreas Geiger, Jun Zhang, and Hongyang Li. Vista: A generalizable driving world model with high fidelity and versatile controllability. Advances in Neural Information Processing Systems, 37:91560–91596, 2025. 2
- [20] Agrim Gupta, Stephen Tian, Yunzhi Zhang, Jiajun Wu, Roberto Martín-Martín, and Li Fei-Fei. Maskvit: Masked visual pre-training for video prediction. In ICLR, 2023. 2, 3, 5, 6
- [21] David Ha and Jürgen Schmidhuber. World models. arXiv preprint arXiv:1803.10122, 2018. 2
- [22] Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In International conference on machine learning, pages 2555–2565. PMLR, 2019. 3
- [23] Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models. arXiv preprint arXiv:2301.04104, 2023. 3
- [24] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 770–778, 2016. 7
- [25] Yingqing He, Tianyu Yang, Yong Zhang, Ying Shan, and Qifeng Chen. Latent video diffusion models for high-fidelity long video generation, 2023. 2, 3, 5
- [26] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In Proceedings of the 31st International Conference on Neural Information Processing Systems, page 6629–6640, Red Hook, NY, USA, 2017. Curran Associates Inc. 5
- [27] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. Advances in neural information processing systems, 33:6840–6851, 2020. 2, 3
- [28] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. Advances in Neural Information Processing Systems, 35: 8633–8646, 2022. 2, 3, 5
- [29] Alain Horé and Djemel Ziou. Image quality metrics: Psnr vs. ssim. In 2010 20th International Conference on Pattern Recognition, pages 2366–2369, 2010. 5
- [30] Anthony Hu, Lloyd Russell, Hudson Yeo, Zak Murez, George Fedoseev, Alex Kendall, Jamie Shotton, and Gianluca Corrado. Gaia-1: A generative world model for autonomous driving. arXiv preprint arXiv:2309.17080, 2023. 2
- [31] Yann LeCun. A path towards autonomous machine intelligence version 0.9. 2, 2022-06-27. Open Review, 62(1), 2022. 2
- [32] Xuanlin Li, Kyle Hsu, Jiayuan Gu, Karl Pertsch, Oier Mees, Homer Rich Walke, Chuyuan Fu, Ishikaa Lunawat, Isabel Sieh, Sean Kirmani, Sergey Levine, Jiajun Wu, Chelsea Finn, Hao Su, Quan Vuong, and Ted Xiao. Evaluating real-world robot manipulation policies in simulation. arXiv preprint arXiv:2405.05941, 2024. 1, 3
- [33] Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, qiang liu, Yuke Zhu, and Peter Stone. LIBERO: Benchmarking knowledge transfer for lifelong robot learning. In Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track, 2023. 2, 6
- [34] Corey Lynch, Ayzaan Wahid, Jonathan Tompson, Tianli Ding, James Betker, Robert Baruch, Travis Armstrong, and Pete Florence. Interactive language: Talking to robots in real time. IEEE Robotics and Automation Letters, 2023. 2, 5, 8
- [35] Xin Ma, Yaohui Wang, Gengyun Jia, Xinyuan Chen, Ziwei Liu, Yuan-Fang Li, Cunjian Chen, and Yu Qiao. Latte: Latent diffusion transformer for video generation. arXiv preprint arXiv:2401.03048, 2024. 2, 3
- [36] Chenlin Meng, Robin Rombach, Ruiqi Gao, Diederik Kingma, Stefano Ermon, Jonathan Ho, and Tim Salimans. On distillation of guided diffusion models. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 14297–14306, 2023. 8
- [37] Jack Parker-Holder, Philip Ball, Jake Bruce, Vibhavari Dasagi, Kristian Holsheimer, Christos Kaplanis, Alexandre Moufarek, Guy Scully, Jeremy Shar, Jimmy Shi, Stephen Spencer, Jessica Yung, Michael Dennis, Sultan Kenjeyev, Shangbang Long, Vlad Mnih, Harris Chan, Maxime Gazeau, Bonnie Li, Fabio Pardo, Luyu Wang, Lei Zhang, Frederic Besse, Tim Harley, Anna Mitenkova, Jane Wang, Jeff Clune, Demis Hassabis, Raia Hadsell, Adrian Bolton, Satinder Singh, and Tim Rocktäschel. Genie 2: A large-scale foundation world model. 2024. 2

- [38] William Peebles and Saining Xie. Scalable diffusion models with transformers. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 4195–4205, 2023. [2](#), [3](#), [4](#), [6](#)
- [39] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis, 2023. [3](#)
- [40] Han Qi, Haocheng Yin, Yilun Du, and Heng Yang. Strengthening generative robot policies through predictive world modeling, 2025. [7](#)
- [41] Eder Santana and George Hotz. Learning a driving simulator. arXiv preprint arXiv:1608.01230, 2016. [2](#)
- [42] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In International conference on machine learning, pages 2256–2265. PMLR, 2015. [3](#)
- [43] Stephen Tian, Chelsea Finn, and Jiajun Wu. A control-centric benchmark for video prediction. In The Eleventh International Conference on Learning Representations, 2023. [3](#)
- [44] Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphael Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric & challenges, 2019. [5](#)
- [45] Dani Valevski, Yaniv Leviathan, Moab Arar, and Shlomi Fruchter. Diffusion models are real-time game engines. arXiv preprint arXiv:2408.14837, 2024. [2](#)
- [46] 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, pages 1723–1736. PMLR, 2023. [2](#), [4](#)
- [47] Jiawei Wang, Yuchen Zhang, Jiaxin Zou, Yan Zeng, Guoqiang Wei, Liping Yuan, and Hang Li. Boximator: Generating rich and controllable motions for video synthesis. arXiv preprint arXiv:2402.01566, 2024. [2](#)
- [48] Tan Wang, Linjie Li, Kevin Lin, Chung-Ching Lin, Zhengyuan Yang, Hanwang Zhang, Zicheng Liu, and Lijuan Wang. Disco: Disentangled control for referring human dance generation in real world. arXiv preprint arXiv:2307.00040, 2023. [2](#)
- [49] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. IEEE Transactions on Image Processing, 13(4): 600–612, 2004. [5](#)
- [50] Jialong Wu, Shaofeng Yin, Ningya Feng, Xu He, Dong Li, Jianye Hao, and Mingsheng Long. ivideoopt: Interactive videoopt are scalable world models, 2024. [2](#), [5](#), [6](#)
- [51] Philipp Wu, Alejandro Escontrela, Danijar Hafner, Pieter Abbeel, and Ken Goldberg. Daydreamer: World models for physical robot learning. In Conference on Robot Learning, pages 2226–2240. PMLR, 2023. [3](#)
- [52] Mingxing Xu, Wenrui Dai, Chunmiao Liu, Xing Gao, Weiyao Lin, Guo-Jun Qi, and Hongkai Xiong. Spatial-temporal transformer networks for traffic flow forecasting. arXiv preprint arXiv:2001.02908, 2020. [3](#)
- [53] Zhongcong Xu, Jianfeng Zhang, Jun Hao Liew, Hanshu Yan, Jia-Wei Liu, Chenxu Zhang, Jiashi Feng, and Mike Zheng Shou. Magicanimate: Temporally consistent human image animation using diffusion model, 2023. [2](#), [5](#)
- [54] Sherry Yang, Yilun Du, Seyed Kamyar Seyed Ghasemipour, Jonathan Tompson, Leslie Pack Kaelbling, Dale Schuurmans, and Pieter Abbeel. Learning interactive real-world simulators. In The Twelfth International Conference on Learning Representations, 2024. [1](#), [2](#)
- [55] Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. Cogvideox: Text-to-video diffusion models with an expert transformer. arXiv preprint arXiv:2408.06072, 2024. [2](#)
- [56] Shengming Yin, Chenfei Wu, Jian Liang, Jie Shi, Houqiang Li, Gong Ming, and Nan Duan. Dragnuwa: Fine-grained control in video generation by integrating text, image, and trajectory, 2023. [2](#)
- [57] Tony Z. Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning Fine-Grained Bimanual Manipulation with Low-Cost Hardware. In Proceedings of Robotics: Science and Systems, Daegu, Republic of Korea, 2023. [1](#)
- [58] Wentao Zhao, Jiaming Chen, Ziyu Meng, Donghui Mao, Ran Song, and Wei Zhang. Vlmpe: Vision-language model predictive control for robotic manipulation, 2024. [3](#)
- [59] Zangwei Zheng, Xiangyu Peng, Tianji Yang, Chenhui Shen, Shenggui Li, Hongxin Liu, Yukun Zhou, Tianyi Li, and Yang You. Open-sora: Democratizing efficient video production for all, 2024. [1](#), [2](#), [6](#)