ORV: 4D Occupancy-centric Robot Video Generation for World Modeling

Xiuyu Yang[1,2*] Bohan Li[3,4*] Shaocong Xu[1] Nan Wang[1] Chongjie Ye[1,5] Zhaoxi Chen[1,6]
Minghan Qin[7] Yikang Ding[8] Zheng Zhu[9] Xin Jin[4] Hang Zhao[2] Hao Zhao[1,10]

[1] Beijing Academy of Artificial Intelligence [2] IIIS, Tsinghua University
[3] Shanghai Jiao Tong University [4] Eastern Institute of Technology, Ningbo
[5] The Chinese University of Hong Kong, Shenzhen [6] S-Lab, Nanyang Technological University
[7] ByteDance [8] Megvii Technology [9] GigaAI [10] AIR, Tsinghua University

Figure 1. We condition robot video generation on 4D semantic occupancy sequences and 7-DoF actions collected from real and simulated environments (through methods π and π'). This occupancy-centric conditioning enables faithful, controllable synthesis of single-view, multi-view, and sim-to-real manipulation videos. We also introduce ORV-Data, a curated 4D occupancy dataset for robot manipulation. Across benchmarks and downstream tasks, ORV improves video quality and control alignment, boosting visual planning and policy learning.

Abstract

Recent embodied intelligence suffers from data scarcity, while conventional simulators lack visual realism. Controllable video generation is emerging as a promising data engine, yet current action-conditioned methods still fall short: generated videos are limited in fidelity and temporal consistency, poorly aligned with controls, and often constrained to single-view settings. We attribute these issues to the representational gap between sparse control inputs and dense pixel outputs. Thus we introduce ORV, a 4D occupancy-centric framework for robot video generation that couples action priors with occupancy-derived visual priors. Concretely, we align chunked 7-DoF actions with video latents via an Action-Expert AdaLN modulation, and inject 2D renderings of 4D semantic occupancy into the generation process as soft guidance. Meanwhile, a central obstacle is the lack of occupancy data for embodied scenarios; we therefore curate ORV-Data, a large-scale, high-quality 4D semantic occupancy dataset of robot manipulation. Across BridgeV2, Droid, and RT-1, ORV improves video generation quality and controllability, achieving 18.8% lower FVD than state of the art, +3.5% success rate on visual planning, and +6.4% success rate on policy learning. Beyond single-view generation, ORV natively supports multi-view consistent synthesis and enables simulation-to-real transfer despite significant domain gaps. Code, models, and data will be released upon acceptance.

1. Introduction

Developing realistic simulators for robot manipulation is crucial for scaling embodied learning [43, 47, 57, 67]. While existing simulators [26, 85] enable safe policy training and

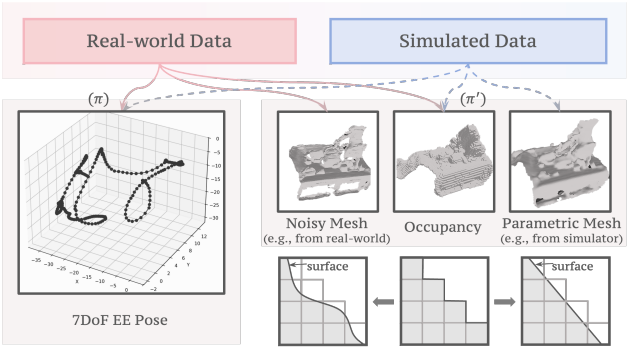


Figure 2. Establishment of non-interactive methods (π, π') both in the real-world environment and physical simulator to collect trajectory priors (7-DoF EE Pose) and visual priors (Occupancy).

efficient data collection, they often struggle to deliver visual realism. Recent progress in generative world models [46, 89, 111], especially action-conditioned video generation, offers a promising alternative by simulating future visual states conditioned on agent actions. These models can render realistic RGB observations responsive to control inputs, yet they still fall short of serving as reliable simulators: generated sequences frequently lack temporal consistency, action alignment, and multiview coherence. Bridging this gap between sparse robot controls and dense visual dynamics remains an open challenge toward building truly *high-fidelity*, *versatile*, and *reliable* generative simulators.

Previous works [2, 77, 94, 95, 128] have advanced action-conditioned video generation using diffusion-based or autoregressive backbones. In these frameworks, robot actions are typically represented as 7-DoF end-effector (EE) poses that guide visual rollout. Other studies [35, 110, 123] instead employ high-level conditioning such as language instructions to drive scene dynamics. Despite these advances, existing approaches remain constrained by three key limitations: (p1) limited visual fidelity and temporal consistency; (p2) drifted or misaligned future predictions that fail to faithfully reflect manipulation controls; and (p3) restriction to singleview observations without enforcing multiview coherence.

We propose ORV, a versatile 4D occupancy-centric framework for robot video generation that produces high-fidelity, action-aligned visual simulations. Our key insight is to incorporate 4D semantic occupancy as visual priors that complement conventional action priors, effectively bridging the representational gap between sparse control trajectories and dense visual dynamics. We think that limitations p2, p3 largely stem from this gap, as also observed in prior works [51, 60, 101, 107] which introduce fine-grained cues such as optical flow, masks, or skeletons to enhance controllability. Furthermore, as illustrated in Fig. 2, occupancy fields demonstrate robustness to geometric noise, providing a natural bridge between simulated and real-world scenarios. Moreover, ORV leverages the generative capabilities of modern video foundation models [46, 89, 111] to boost visual realism and temporal coherence, substantially mitigating issue (p1) while preserving physically consistent dynamics.

The overall framework of ORV is depicted in Fig. 1. Guided by geometric priors from 4D semantic occupancy, ORV enables robot manipulation video generation across diverse object appearances and scenes [3, 60]. Furthermore, view-specific conditioning encourages cross-view coherence, enabling consistent multiview synthesis [1, 4, 25]. Benefiting from the domain-invariant nature of occupancy-derived representations, ORV also facilitates visual transfer from simulation to the real world under varied conditions. To support large-scale training, we curate ORV-Data, a high-quality 4D semantic occupancy dataset for robot manipulation, built through a carefully designed data curation pipeline.

Our contributions can be summarized as follows:

- We propose **ORV**, a *4D occupancy-centric framework*, enabling precise and controllable robot video generation with domain randomization.
- By injecting *occupancy-derived geometric priors* into diffusion noise, ORV achieves temporally consistent and geometrically coherent multiview video generation and simulation-to-real visual transfer.
- We curate **ORV-Data**, a large-scale, high-quality *4D semantic occupancy dataset* of robot manipulation with rich geometric and semantic annotations.
- Experiments across diverse datasets and downstream tasks demonstrate that ORV consistently enhances controllable video generation, visual planning, and data-driven policy learning, achieving state-of-the-art performance.

2. Related Work

Generative Models for World Modeling. Recent advances in video generation [7, 8, 46, 89, 111, 113, 125] have greatly improved the realism of world modeling, benefiting robotics [3, 9, 22, 41, 60, 65, 72, 101, 123, 128], autonomous driving [23, 50, 68, 100], and general scene synthesis [55, 56, 76, 122]. ReCamMaster [8] and SynCamMaster [7] achieve video synthesis of novel trajectories, while IRASim [128] enables action-to-video prediction and VAP [101] employs visual prompts for precise control in robotics. For autonomous driving, more recent works adopt 3D occupancy as efficient scene representations [15, 37, 38, 48, 49, 93, 96, 103, 104, 106, 124]. For instance, UniScene [50] leverages hierarchical occupancy priors for multimodal scene generation. Beyond explicit video synthesis, implicit generative models have also been adopted for complex interactions and decision making [18, 66, 71].

World Models for Embodied Intelligence. Progress in simulating dynamic environments has fueled the development of world models for robotics [9, 11, 14, 17, 19, 24, 27, 35, 64, 110, 116, 126, 127], where TesserAct [123] performs 4D scene synthesis via appearance–geometry joint modeling and EnerVerse [35] forecasts future environments through a

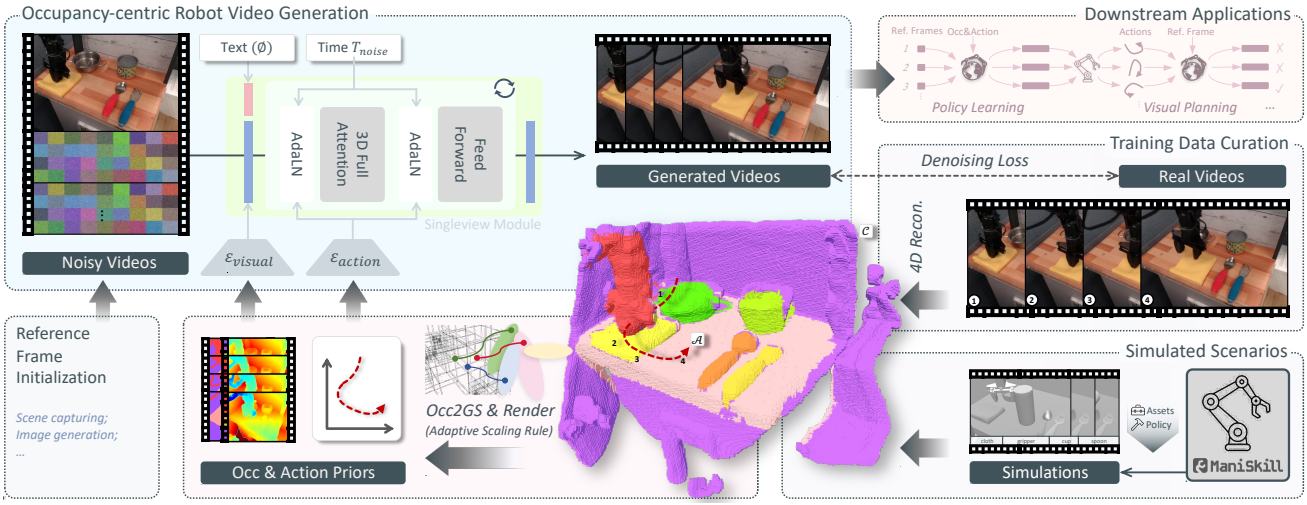


Figure 3. Overview of **ORV framework**. Centered on occupancy representation $\mathcal{C}$, along with actions $\mathcal{A}$, which are extracted from physical simulators (*e.g.*, ManiSkill [26]) or real-world data (*e.g.*, Bridge [88]), we leverage the soft derived visual priors to enable robot video generation with high visual quality and control alignment. Furthermore, we design a data curation pipeline to construct the robot occupancy data for training purposes. ORV, as a powerful neural simulator, can greatly boost downstream applications (*e.g.*, policy learning, visual planning, etc).

simulation pipeline. iVideoGPT [105] and Vid2World [36] explore action-conditioned visual prediction with autoregressive frameworks. For data augmentation, Cosmos-Transfer [3] and RoboTransfer [60] condition robot video generation on scene maps (*e.g.*, depth and normal), while RoboEngine [115] achieves scene augmentation through the segmentation toolkit. Meanwhile, WorldSimBench [73] establishes unified evaluation benchmarks for world models.

3. ORV: Methodology

We first formulate the robot video generation task (Sec. 3.1). Then we elaborate on the specific architecture of ORV and how these designs can largely improve the robot video generation (Sec. 3.2). Finally, we introduce our robot occupancy dataset curated for the training process (Sec. 3.3) and explain how ORV helps with the robot manipulations.

3.1. Problem Formulation

A generative world model for robot manipulation aims to provide a photorealistic and physically consistent simulation of the environment that mirrors real-world dynamics. Given the context $(\mathcal{S}, \mathcal{O}, \phi, \rho)$, the goal of the model $\mathcal{M}$ is to predict future states $s_{t:t+\Delta T} \in \mathcal{S}$ and corresponding observations $o_{t:t+\Delta t} \in \mathcal{O}$, where $o_t = \phi(s_t)$ denotes the rendered observation from state s_t. Here, ρ defines the underlying rules governing state transitions, leading to the transition probability $p(s_{t:t+\Delta t}, o_{t:t+\Delta t} \mid s_{1:t}, o_{1:t})$.

We formulate $\mathcal{O}$ in RGB space (*e.g.*, images or videos). Conventional text-to-video models [54, 89, 111] condition on $\rho_1 := \texttt{Embed}(\texttt{text})$, yet linguistic abstraction often hinders accurate physical simulation. Recent action-conditioned video generation [77, 94, 101, 105] extends this to $\rho_2 := \texttt{Embed}(a_{t:t+\Delta t} \sim \pi(s_{1:t}))$. Building upon this progression, our model introduces $\rho_3 := \texttt{Embed}(c_{t:t+\Delta t} \sim \pi'(s_{1:t}), a_{t:t+\Delta t} \sim \pi(s_{1:t}))$, where a denotes agent actions and c represents occupancy fields. We denote by π and π' the extraction processes for (a, c) given states s.

As illustrated in Fig. 2, both extraction methods can be established either in the real world (*e.g.*, human teleoperation) or within simulators (*e.g.*, ManiSkill [26], MuJoCo [85]). Notably, we employ π and π' in a non-interactive manner—these priors are collected entirely in a single offline pass before being used. Moreover, the motivation for leveraging occupancies lies in their robustness for representing both noisy and parametric scene surfaces (Fig. 2). And the coordinate-based formulation of occupancies enables seamless integration with online occupancy generations [121].

3.2. Occupancy-centric Robot Video Generation

To skip the large-scale pretraining process (as previous works [105, 127]) and reduce the training cost, we build ORV model upon the pretrained CogVideoX-2B [111] (text-to-video), which also aligns with our non-interactive purpose (bidirectional diffusion model). CogVideoX incorporates the architecture of diffusion transformer (DiT) and achieves incredible performance. Then, we propose a two-stage supervised finetuning (SFT) to inject both action and visual cues into video generations. We aim to address three key aspects: 1) overall quality of generated videos (*e.g.*, consistency of frames and realism) 2) alignment with the instructions ρ_3, and 3) computation efficiency.

Chunk-level Action Conditioning. The 7-DoF action sequences (*e.g.*, $\mathcal{A} \in R^{T \times D_a}$ derived from end-effector pose

sequences and $D_a = 7$) serve a high-level control signals in robot video generation. Drawing inspiration from [120, 128], we inject these 3D action controls through adaptive layer normalization (Action Expert AdaLN) to directly modulate the video latents within each DiT block. More efficiently, as illustrated in Fig. 4, we propose a chunk-level scheme for temporal alignment between high-dimensional actions and videos in modulation.

Specifically, following the temporal compression in 3D VAE [46, 89, 111], we pad zero actions as the placeholders of reference frames. Then an additional shallow MLP (ε_{action} in Fig. 3) is used to map every consecutive r actions into a single token: $\mathcal{A} \in R^{T \times D_a} \rightarrow \texttt{MLP}(\texttt{Pad}(\mathcal{A})) \in R^{(\frac{T}{r}+1) \times D}$, where r denotes the chunk-size and D represents the feature size. Furthermore, we let Action Expert AdaLN to reuse the parameters of pretrained Vision Expert AdaLN, eliminating the unnecessary computation cost (as each AdaLN accounts for $\sim 1/3$ of the total parameters).

Occupancy-derived Visual Conditioning. Translating abstract 3D action signals into 2D pixels presents a great challenge; thus, we introduce *soft* and *pixel-level* visual conditionings derived from occupancy fields. However, directly projecting voxels onto 2D plane will cause mutations on pixels between adjacent frames and viewpoints. Therefore, we propose to assign each grid with single non-learnable Gaussian splat and then render them from certain views, which greatly improves the quality of conditions and saves memory.

Moreover, we propose an *adaptive scaling mechanism* on Gaussians to solve the perspective distortion of Gaussians during rendering (see Sec. 10.1.2 in Suppl. for derivations). Specifically, the scale follows $\sigma = k_2 \cdot \hat{z}^{k_1}$, where $\hat{z} \in [1, 2)$ denotes the *normalized depths in canonical space*, and exponential term k_1, base scale term k_2 control the scaling behavior of Gaussians in the near and far plane, respectively.

To inject such occupancy-derived visual conditionings, we directly use an additional MLP (ε_{visual} in Fig. 3), then augment it with the reference images, after which another zero-initialized MLP projector adds the visual conditionings to the input noise: $z_{\text{in}} = \texttt{Zero-MLP}(z_{\text{in}} + \texttt{MLP}(\mathcal{C})) + z_{\text{in}}$. The previous ControlNet-like [118] methods, though demonstrating accurate controls, suffers from a serious computation cost (see Sec. 10.2 in Suppl.). Furthermore, such layer-wise control injection tends to corrupt the video latents when conditions are *soft*—that is, not pixel-level alignment.

3.2.1. ORV-MV: Multiview Robot Video Generation

A complete and high-fidelity 4D world, typically formed from multiview observations, greatly benefits robot learnings [62, 72]. Leveraging the 4D occupancy-centric design, ORV(-MV) generates multiview robot manipulation videos well. Some prior works [101, 123, 128], however, capture only a single surface of the scenes, resulting in noticeable artifacts and empty regions in shifted views.

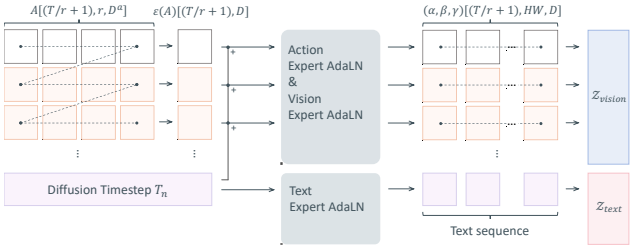


Figure 4. Illustration of three modulations (Expert AdaLN) and injecting actions in our DiT block. And indicates the action paddings serving as the placeholders for reference frames, where ε encodes actions and α, β, γ are modulation vectors. We use $[\cdot]$ to indicate the dimensions for simplicity.

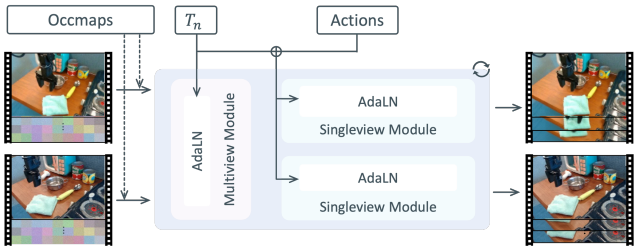


Figure 5. Architecture of ORV-MV, which generates multiview robot manipulation videos with cross-view consistency.

As shown in Fig. 5, ORV-MV introduces an additional view attention (multiview module) prior to the temporal attention (singleview module), inspired by [7, 16]. Both inherit the 3D (2D+1D) attention layers of pretrained model, with 2D over pixels $H \times W$ and 1D over views V or frames F. The former processes the latents $\mathcal{F}_V \in R^{B_V \times S_V \times D}$, where $S_V = VHW$ denotes patch tokens across all views. While the latter handles $\mathcal{F}_P \in R^{B_P \times S_P \times D}$, where $S_P = THW$ denotes tokens across all times of each view.

We then apply different controls for the two modules. Specifically, singleview modules are conditioned on text, actions and occmaps. While multiview ones exclude action priors as they focus on view correspondences. Additionally, details on handling multiview occupancy map data for training purposes are provided in Sec. 8 in Supplementary.

3.2.2. ORV-S2R: Bridge Simulation-to-Real Transfer

The occupancy-derived visual priors (*e.g.*, depth maps) also enable ORV(-S2R) to generate realistic videos from such appearance-agnostic information, which is crucial for alleviating the *visual realism* gap between simulated and real data in robotics. As shown in Fig. 3, physical simulators (*e.g.*, ManiSkill [26], MuJoCo [85]) can readily provide such priors at a low cost.

Previous works, *e.g.*, Cosmos-Transfer [3], RoboTransfer [60], have also demonstrated success in transferring multi-modal data to significantly mitigate the data scarcity problem in robotics. However, as described in Sec. 3.1 and Fig. 2, the occupancy-derived condition maps further exhibit robustness to geometric noise, providing a natural bridge between real-world noisy surfaces and parametric ones in simulation. Experiments in Sec. 4.4 have validated

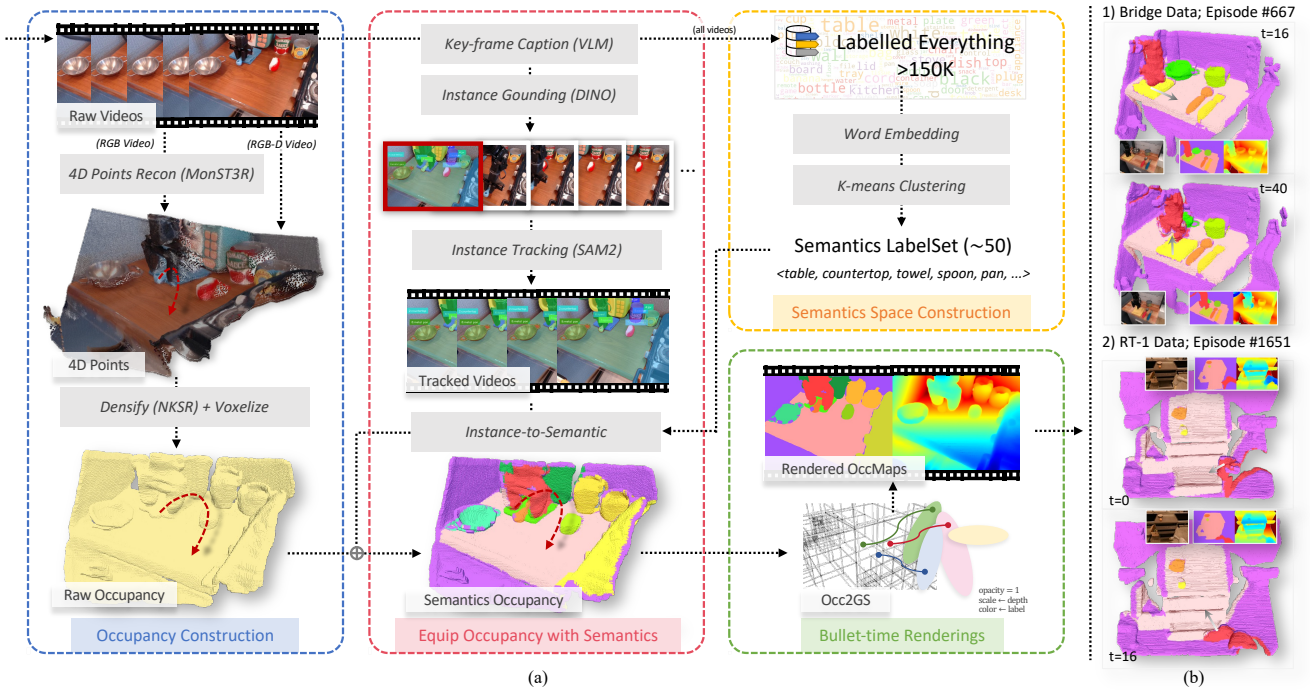


Figure 6. (a) Overview of **Training Dataset Curation Pipeline**, which consists of four steps 1) semantics space construction 2) occupancy construction 3) equip occupancy with semantics and 4) bullet-time occupancy-to-Gaussian renderings in practical usage. (b) **Occupancy examples** of BridgeData V2 [88] and RT-1 [13]. Better to zoom in. And refer to Supplementary Materials for more examples.

the effectiveness of this design.

3.3. 4D Occupancy Dataset of Robot Manipulation

To train ORV model, we establish a 4D occupancy dataset of robot manipulation through the data curation pipeline shown in Fig. 6(a). The occupancy data are derived from existing popular robot datasets (BridgeData V2 [88], Droid [44], RT-1 [13]). Some examples are shown in Fig. 6(b). More details are provided in Sec. 10.1 and Sec. 11.1 in the Supplementary.

Semantics Labeling. Complex semantic understanding remains essential in robot manipulation, as predicting next-state dynamics requires recognizing object categories—rigid, articulated, or deformable—that exhibit distinct physical behaviors. To this end, we construct the dataset-level semantic space through vision-language-model (VLM) [6] for captioning and K-means [63] clustering over ∼150K labels. For each video, we then extract temporally consistent instances across frames using Grounding DINO [61] and SAM2 [75], after which they are mapped to the coherent semantic annotations.

4D Occupancy Generation. This process involves two steps: 1) occupancy construction and 2) semantic enrichment. We first reconstruct sparse 4D points with MonST3R [117] and then densify them by NKSR [34], which greatly fills holes and is robust to noise. Note that for those videos with a depth channel, the reconstruction is not needed. Then, dense points are voxelized to 4D occupancy in canonical space, after which semantics are assigned by majority voting for points with projected semantic labels within each voxel. Finally, we filter the occupancy-rendered data with poor inter-frame consistency (through RAFT [81]).

4. Experiments

In this section, we conduct comprehensive experiments to validate ORV model on multiple tasks, including *controllable video generation*, *visual planning* and *policy learning*. They are expected to answer these questions: *1) What is the quality of the videos generated by ORV? 2) To what extent is the generative capability of ORV? 3) How can generated videos benefit robot learning tasks?* The dataset and training details are provided in Sec. 7 of the Supplementary.

4.1. Conditional Video Generation

Setup. We evaluate the video generation of ORV on three real-world datasets, their embodiments, views of each episode and volume are as below:

- BridgeV2 [88]: WidowX, 1∼3 views, ∼60K episodes;
- Droid [44]: Franka Panda, 2 views, ∼76K episodes;
- RT-1 [13]: Google Robot, 1 view, ∼120K episodes;

Please refer to Sec. 7 for more dataset details. For the action-conditioned base model setup, we train ORV for ∼30K steps from pretrained backbone. For occupancy maps-guided fine-tuning and multiview video generation, we have additional ∼20K gradient steps of training.

Table 1. Evaluation results of *Conditional Video Generation* on three datasets. Top-1 performance within all variants and each type of models are represented with **bold text** and underlines.

Method	BridgeV2 [88]				Droid [44]				RT-1 [13]			
	PSNR↑	SSIM↑	FID↓	FVD↓	PSNR↑	SSIM↑	FID↓	FVD↓	PSNR↑	SSIM↑	FID↓	FVD↓
Text-conditioned Generation Models												
CogVideoX [111]	19.432	0.752	7.509	83.561	19.238	0.701	6.341	71.536	20.457	0.816	6.243	42.169
Action-conditioned Generation Models												
AVID [77]	-	-	-	-	-	-	-	-	25.600	0.852	2.965*	24.200
HMA [94]	23.636	0.808	8.849	67.096	21.435	0.821	**3.108**	47.383	25.424	0.840	7.306	84.165
IRASim [128]	25.276	0.833	10.510	20.910	21.632	0.820	5.395	41.031	26.048	0.833	5.600	25.580
ORV (Ours)	25.631	0.873	3.821	17.682	22.034	0.838	4.921	37.094	27.086	0.863	4.210	20.031
Occupancy&Action-conditioned Generation Models												
IRASim† [128]	27.352	0.862	9.413	22.503	22.005	0.827	7.892	44.309	27.213	0.847	5.311	42.130
ORV (Ours)	**28.258**	**0.899**	**3.418**	**16.525**	**22.310**	**0.841**	3.222	**34.603**	**28.214**	**0.878**	**4.013**	**19.931**

* FID Scores of AVID [77] have been computed not in evaluation mode according to the official codes and lead to incorrect results. Thus, we ignore it.
† We incorporate the same occupancy&action conditions to IRASim.

Table 2. Evaluation results of *Visual Planning* on VP^2 [82] Benchmark. Top-1 performance across 8 tasks and the average success rate are highlighted accordingly. We provide the mean and standard deviation of the success rate (in %) on average over 3 runs. The best and second-best performances are represented with **bold text** and underlines, respectively.

Method	Robosuite Push	Flat Block	Open Drawer	Open Slide	Blue Button	Green Button	Red Button	Upright Block	Avg. Success	
Simulator	$93.5^{\pm 2.2}$	$13.3^{\pm 0.1}$	$76.7^{\pm 0.0}$	$71.7^{\pm 1.2}$	$100.0^{\pm 0.0}$	$96.7^{\pm 0.0}$	$90.0^{\pm 0.0}$	$90.0^{\pm 0.0}$	88.4	*
MCVD [87]	$77.3^{\pm 2.6}$	$4.0^{\pm 1.1}$	$11.7^{\pm 1.2}$	$18.3^{\pm 1.0}$	$95.0^{\pm 3.6}$	$83.3^{\pm 0.4}$	$73.3^{\pm 2.6}$	$56.7^{\pm 2.4}$	59.4	67.2
FitVid [5]	$67.7^{\pm 5.3}$	$\mathbf{9.2}^{\pm 4.0}$	$25.3^{\pm 6.9}$	$\mathbf{35.3}^{\pm 4.5}$	$94.0^{\pm 4.6}$	$84.0^{\pm 5.3}$	$58.7^{\pm 5.1}$	$51.3^{\pm 2.7}$	59.5	67.3
MaskViT [28]	$\mathbf{82.6}^{\pm 2.3}$	$4.0^{\pm 3.9}$	$4.0^{\pm 4.5}$	$8.7^{\pm 5.7}$	$94.7^{\pm 2.0}$	$64.0^{\pm 4.3}$	$24.0^{\pm 7.5}$	$\mathbf{62.2}^{\pm 8.6}$	48.6	55.0
iVideoGPT[105]	$78.3^{\pm 0.4}$	$3.3^{\pm 0.7}$	$37.5^{\pm 1.5}$	$16.1^{\pm 2.5}$	$95.6^{\pm 2.9}$	$82.5^{\pm 3.1}$	$92.2^{\pm 1.5}$	$44.7^{\pm 1.7}$	63.9	72.2
ORV (Ours)	$81.4^{\pm 1.7}$	$6.1^{\pm 2.0}$	$\mathbf{40.5}^{\pm 1.1}$	$19.9^{\pm 3.4}$	$\mathbf{96.7}^{\pm 2.5}$	$\mathbf{85.6}^{\pm 3.0}$	$\mathbf{93.2}^{\pm 1.9}$	$44.8^{\pm 1.4}$	**66.0**	**74.7**

* Values in this column are normalized by the simulator's average success rate.

Comparison Results against Baselines. To comprehensively demonstrate the superiority of ORV model, we compare ORV with original CogVideoX [111] and action-conditioned methods AVID [77], HMA [94], IRASim [128] and more baselines augmented with our occupancy priors (*e.g.*, IRASim). More details about the baselines are provided in Sec. 11.2. We report the quantitative results of *controllable video generation* in Table 1, where ORV outperforms all baselines across most of the metrics. Moreover, as highlighted (white arrows) in the BridgeV2 example of the single-view generation in Fig. 7, the baseline fails to faithfully infer the dynamics of objects manipulated by the robotic gripper. More comparison results can be found in Sec. 11.2 in Supplementary.

Multiview Robot Video Generation. We show an example of multiview robot video generation performed by ORV in Fig. 7. The example shows the robot arm performing a cloth-folding task across *three* views, where the outputs maintain exceptional cross-view consistency. This high-fidelity multi-view generation enables efficient downstream applications, including photorealistic scene reconstruction and robotics imitation learning. Note that there exists a lighting discrepancy issue in the original data. Please refer to Sec. 11.2 in Suppl. for more examples and analysis.

Sim-to-Real Transfer. Fig. 7 illustrates examples of sim-to-real generation through ORV-S2R, as described in Sec. 3.2.2. Details of simulation environment setup and dynamics data generation are provided in Sec. 9 of the Supplementary. Leveraging an additional image generator (ControlNet [118]), we first produce diverse initial frames and then extend them to high-quality, realistic manipulation videos. In this case, using simulator-derived occupancy maps consistent with training preserves the consistent performance. Moreover, thanks to the robustness of occupancy representations, even condition maps with various granularity (e.g., parametric maps from the simulator) yield only minor performance degradation (see results in Sec. 4.4).

4.2. Visual Planning

Setup. We further evaluate the controllability of ORV on VP^2, a visual planning by action controls benchmark.

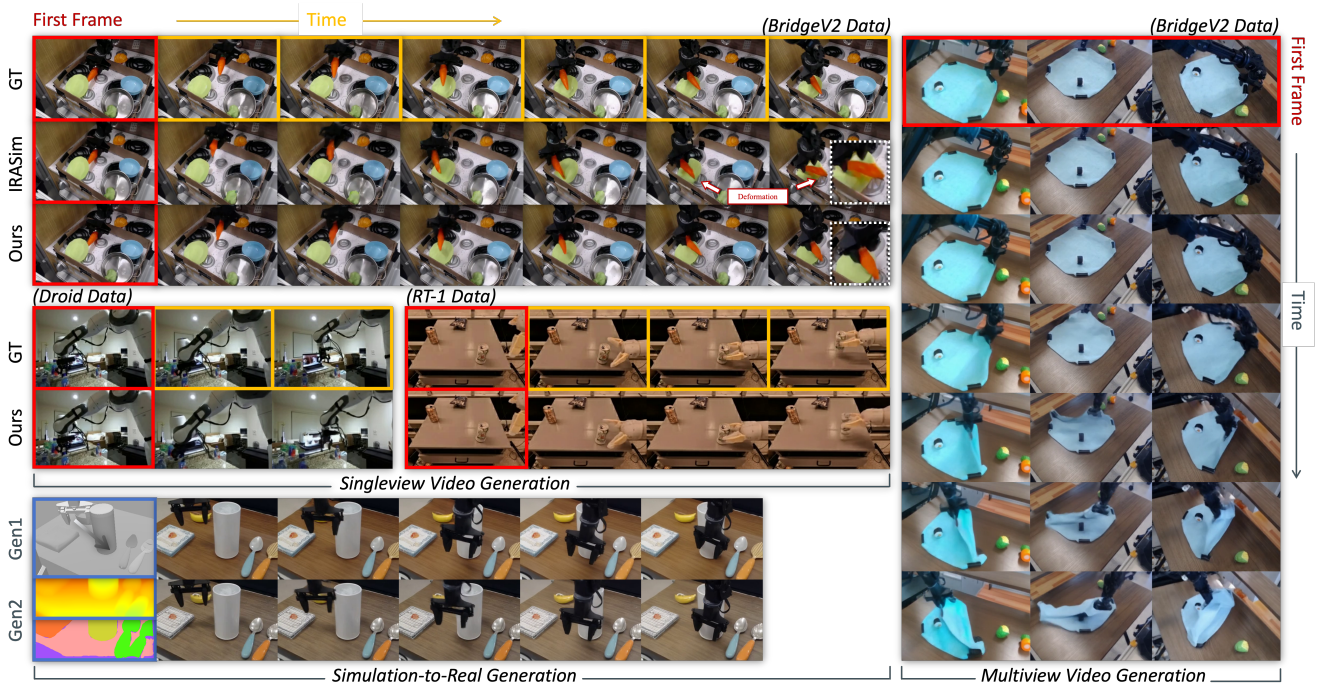


Figure 7. Qualitative results of versatile **Video Generation** with full conditions. Given one-frame observation, ORV predict subsequent 15 frames on validation split of Bridge [88], Droid [44], RT-1 [13] datasets. Red boxes denotes the first frame input of the video generation; Orange boxes denotes the ground-truth of the subsequence frames.

Table 3. Evaluation results on SimplerEnv-WidowX [53] across four manipulation tasks.

Method	Spoon on Towel	Carrot on Plate	Stack Cube	Eggplant in Basket	Avg. Success
RoboVLM*† [59]	18.6%	22.9%	8.1%	0.0%	12.4%
+Finetune*	27.6%	26.7%	12.1%	52.8%	29.8%
+ORV	32.2%	29.6%	15.7%	57.9%	**33.9%**
ΔImprovement	*+4.6%*	*+2.9%*	*+3.6%*	*+5.1%*	***+4.1%***
SpatialVLA*† [74]	12.5%	20.8%	20.8%	58.3%	28.1%
+Finetune*	12.8%	26.1%	26.5%	79.3%	36.2%
+ORV	14.7%	28.4%	27.8%	83.0%	**38.5%**
ΔImprovement	*+1.9%*	*+2.3%*	*+1.3%*	*+3.7%*	***+2.3%***

* The results are reproduced locally for fully fair comparisons.
† Zero-shot performance (Pretraining).

Table 4. Ablation results of *Video Generation* and *Visual Planning* on approaches of priors injection.

Variants	PSNR↑	SSIM↑	FID↓	FVD↓	Success↑
CogVideoX	19.432	0.752	7.509	83.561	-
Action Conditions					
w/ Text Expert	20.424	0.772	4.104	23.586	52.9
No Chunks	24.813	0.850	3.793	19.944	70.6
Ours (base)	25.631	0.873	3.821	17.682	74.7
Occupancy Map Conditions					
ControlNet	26.974	0.865	3.613	20.069	-
Ours (full)	28.258	0.899	3.418	16.525	-

Following [82, 95, 105], we train ORV on 5K trajectories for Robosuite [129] and 35K for RoboDesk [42].

Results. Tab. 2 presents the success rates of ORV compared to the baselines over 9 tasks. ORV outperforms all baselines in four RoboDesk tasks and achieves second-best results in the other four tasks, indicating its capability to predict *high-fidelity* future observations, which is fully *controllable*.

4.3. Policy Learning

Setup. To improve the policy learning, we employ it as a powerful data engine to augment existing data. Similar to ORV-S2R, we leverage another image generator (ControlNet) to generate diverse initial frames and then extend them to videos, with some examples with appearance randomizations are shown in Fig. 8. For our evaluations, we use post-finetuning after cross-embodiment pre-training as the setup, and leverage ORV to generate additional $\sim 30K$ samples (refer to Sec. 10.3 in Suppl. for more details). We evaluate the recent open-sourced policy models RoboVLM [59] and SpatialVLA [30] on SimplerEnv-WidowX [53] with BridgeData V2 [88] as the test suite. Each policy model is finetuned both on the original data and augmented data, following the official instructions.

Results. Tab. 3 shows that the augmented data from ORV improves policy learning performance. With finetuning on augmented data, we achieve gains of $\sim$13.7% (29.8% $\rightarrow$ 33.9%) for RoboVLM [59] and $\sim$6.5% (36.2% $\rightarrow$ 38.5%) for SpatialVLA [74]. The results demonstrate the effectiveness of ORV-generated for policy learning. More discussions and analysis are provided in Sec. 12 of the Suppl.

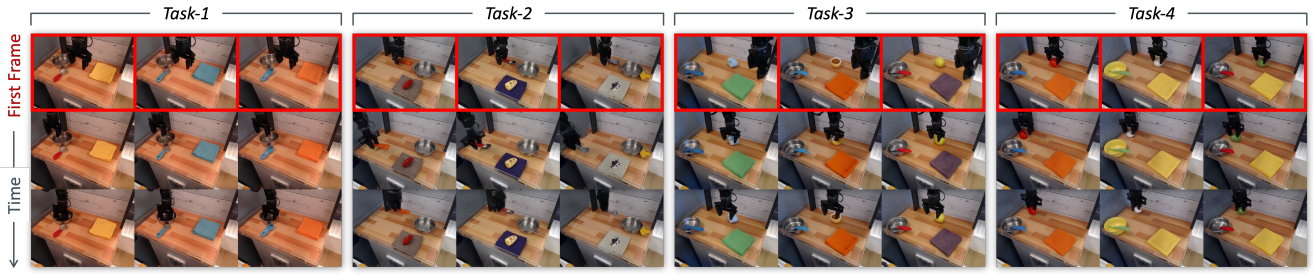


Figure 8. Illustrations of **Appearance Randomization** powered by ORV, generating diverse manipulation videos of four tasks. For each manipulation task in Bridge Data [88], we present three examples with distinct visual appearances, demonstrating that ORV generalizes well to varied context inputs and thereby alleviates the challenge of data collection of robot learning.

Table 5. Ablation results of *Conditional Video Generation* on occupancy conditioning resources and training strategies.

Variants	Source	PSNR↑	SSIM↑	FID↓	FVD↓
Conditioning Resources					
w/o cond. (base)	-	25.631	0.873	3.821	17.682
w/ depth	Fine	30.288	0.919	3.061	14.321
	Coarse	28.031	0.896	4.522	18.548
w/ sem.	Fine	28.896	0.901	3.259	16.171
	Coarse	27.911	0.896	3.467	17.053
Full cond.	Fine	30.431	0.920	2.998	14.301
	Coarse	28.258	0.899	3.418	16.525
Training Strategies (w/o Occupancy Conditionings)					
From scratch	-	23.518	0.811	19.357	84.831
From CogVideoX2B	-	25.631	0.873	3.821	17.682

Table 6. Ablation results of *Multiview Video Generation* on occupancy conditionings on BridgeData V2 [88] with 3 views. Numbers are reported as "with / without" visual priors.

Views	PSNR↑	SSIM↑	FID↓	FVD↓
View0 (anchor)	25.77 / 28.25	0.87 / 0.89	3.20 / 3.11	14.05 / 12.54
View1	23.04 / 25.87	0.79 / 0.85	3.31 / 3.18	16.36 / 13.67
View2	22.90 / 25.79	0.78 / 0.85	3.32 / 3.19	15.97 / 13.62

Table 7. Ablation results of zero-shot *Conditional Video Generation* on different occupancy conditioning resources.

Train	Val	PSNR↑	SSIM↑	FID↓	FVD↓
Coarse	Coarse	28.031	0.896	4.522	18.548
Coarse	Fine	26.608 (-1.423)	0.872 (-0.024)	4.932 (+0.410)	24.134 (+5.586)
Fine	Fine	30.288	0.919	3.061	14.321
Fine	Coarse	19.048 (-11.240)	0.754 (-0.165)	22.893 (+19.832)	132.685 (+109.792)

4.4. Ablation Study and Analysis

We present comprehensive ablations of our proposed ORV framework and other related discussions. We provide more details in the Supplementary.

Effect of Conditioning Approaches. We first ablate the action conditioning designs in Fig. 4 with the results shown in Tab. 4. Different configurations of the Action Expert AdaLN (*e.g.*, take the combination of original Vision Expert and Text Expert) result in significantly inferior performance. In addition, using action conditioning without temporal chunking (*e.g.*, directly encoding the entire action sequence) also weakens the performance (PSNR drops by 3.2% and success rate drops by 5.5%). For occupancy-derived visual conditionings, we validate the effectiveness of injecting the conditioning into the initial noise. We confirm that injecting occupancy-derived coarse controls into deep layers causes noticeable performance degradation (PSNR drops by 4.5%).

Effect of Control Signals. Tab. 5 reveals the impact of different conditioning resources (Coarse: occupancy-rendered condition maps; Fine: pixel-level condition maps) and conditioning types (depth and semantic) used for training and evaluation. The results demonstrate that introducing visual priors leads to significant improvements, with gains of 18.72% (25.621→30.431) and 10.24% (25.621→28.258). Moreover, coarse condition maps achieve performance comparable to their fine counterparts. In addition, Tab. 6 further shows the improvements in three-view (BridgeData V2 [88]) robot video generation when visual priors are introduced, where the view0 serves as the "anchor view" for constructing multiview visual priors (see Sec. 8 in the Supplementary).

Effect of Pretraining. We further test the benefits of the pretraining process. As shown in Tab. 5, models trained from the CogVideoX have superior performance compared to those from scratch, particularly on FID and FVD metrics.

Robustness of Occupancy Representations. To validate the robustness of occupancy representations used in ORV model, as described in Sec. 3.1 and Sec. 4.1. We examine the zero-shot performance of models through training and evaluate them under different conditions settings, as illustrated in Tab. 7 (refer to Fig. 17 in Suppl. for more details). The results reveal that models trained on occupancy-derived coarse visual conditions generalize better across conditions of varying granularity. In contrast, ORV models trained on pixel-aligned conditions suffer a dramatic performance drop on coarse inputs. This imposes a major constraint on deploying the model in more diverse scenarios (*e.g.*, between simulation and real-world), necessitating condition maps that accurately align with ground truths.

5. Conclusion

We propose ORV, an occupancy-centric framework for robot video generation that couples action priors with occupancy-

derived visual priors. With such an occupancy-centric design, ORV achieves high-quality robot video generation and consistent multiview synthesis. The robustness of occupancy representations further enables ORV to achieve superior visual transfer between simulated and real-world scenarios. Experiments on controllable video generation, visual planning, and policy learning demonstrate the effectiveness and versatility of ORV for advancing robotics research.

References

[1] Cihan Acar, Kuluhan Binici, Alp Tekirdağ, and Yan Wu. Visual-policy learning through multi-camera view to single-camera view knowledge distillation for robot manipulation tasks. *IEEE Robotics and Automation Letters*, 9(1):691–698, 2023. 2

[2] 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. 2

[3] Hassan Abu Alhaija, Jose Alvarez, Maciej Bala, Tiffany Cai, Tianshi Cao, Liz Cha, Joshua Chen, Mike Chen, Francesco Ferroni, Sanja Fidler, et al. Cosmos-transfer1: Conditional world generation with adaptive multimodal control. *arXiv preprint arXiv:2503.14492*, 2025. 2, 3, 4, 11

[4] Ehsan Asali, Prashant Doshi, and Jin Sun. Mvsa-net: Multi-view state-action recognition for robust and deployable trajectory generation. *arXiv preprint arXiv:2311.08393*, 2023. 2

[5] Mohammad Babaeizadeh, Mohammad Taghi Saffar, Suraj Nair, Sergey Levine, Chelsea Finn, and Dumitru Erhan. Fitvid: Overfitting in pixel-level video prediction. *arXiv preprint arXiv:2106.13195*, 2021. 6

[6] Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023. 5, 2, 11, 13

[7] Jianhong Bai, Menghan Xia, Xintao Wang, Ziyang Yuan, Xiao Fu, Zuozhu Liu, Haoji Hu, Pengfei Wan, and Di Zhang. Syncammaster: Synchronizing multi-camera video generation from diverse viewpoints. *arXiv preprint arXiv:2412.07760*, 2024. 2, 4

[8] Jianhong Bai, Menghan Xia, Xiao Fu, Xintao Wang, Lianrui Mu, Jinwen Cao, Zuozhu Liu, Haoji Hu, Xiang Bai, Pengfei Wan, et al. Recammaster: Camera-controlled generative rendering from a single video. *arXiv preprint arXiv:2503.11647*, 2025. 2

[9] Homanga Bharadhwaj, Debidatta Dwibedi, Abhinav Gupta, Shubham Tulsiani, Carl Doersch, Ted Xiao, Dhruv Shah, Fei Xia, Dorsa Sadigh, and Sean Kirmani. Gen2act: Human video generation in novel scenarios enables generalizable robot manipulation. *arXiv preprint arXiv:2409.16283*, 2024. 2

[10] Åke Björck. *Numerical methods for least squares problems*. SIAM, 2024. 1

[11] Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, et al. Gr00t n1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025. 2

[12] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π0: A vision-language-action flow model for general robot control. corr, abs/2410.24164, 2024. doi: 10.48550. *arXiv preprint ARXIV.2410.24164*, 2024. 12

[13] 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. 5, 6, 7, 1, 3, 13

[14] 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. 2

[15] Anh-Quan Cao and Raoul de Charette. Monoscene: Monocular 3d semantic scene completion. In *CVPR*, 2022. 2

[16] Chenjie Cao, Chaohui Yu, Shang Liu, Fan Wang, Xiangyang Xue, and Yanwei Fu. Mvgenmaster: Scaling multi-view generation from any image via 3d priors enhanced diffusion model. *arXiv preprint arXiv:2411.16157*, 2024. 4

[17] Chi-Lam Cheang, Guangzeng Chen, Ya Jing, Tao Kong, Hang Li, Yifeng Li, Yuxiao Liu, Hongtao Wu, Jiafeng Xu, Yichu Yang, et al. Gr-2: A generative video-language-action model with web-scale knowledge for robot manipulation. *arXiv preprint arXiv:2410.06158*, 2024. 2

[18] Daniil Cherniavskii, Phillip Lippe, Andrii Zadaianchuk, and Efstratios Gavves. Stream: Embodied reasoning through code generation. In *Multi-modal Foundation Model meets Embodied AI Workshop@ ICML2024*, 2024. 2

[19] Xiaowei Chi, Hengyuan Zhang, Chun-Kai Fan, Xingqun Qi, Rongyu Zhang, Anthony Chen, Chi-min Chan, Wei Xue, Wenhan Luo, Shanghang Zhang, et al. Eva: An embodied world model for future video anticipation. *arXiv preprint arXiv:2410.15461*, 2024. 2

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

[21] Zhehao Dong, Xiaofeng Wang, Zheng Zhu, Yirui Wang, Yang Wang, Yukun Zhou, Boyuan Wang, Chaojun Ni, Runqi Ouyang, Wenkang Qin, et al. Emma: Generalizing real-world robot manipulation via generative visual transfer. *arXiv preprint arXiv:2509.22407*, 2025. 6, 12

[22] Xiao Fu, Xintao Wang, Xian Liu, Jianhong Bai, Runsen Xu, Pengfei Wan, Di Zhang, and Dahua Lin. Learning video generation for robotic manipulation with collaborative trajectory control, 2025. 2, 12

[23] Ruiyuan Gao, Kai Chen, Enze Xie, Lanqing Hong, Zhenguo Li, Dit-Yan Yeung, and Qiang Xu. Magicdrive: Street view generation with diverse 3d geometry control. In *ICLR*, 2024. 2

[24] Haoran Geng, Feishi Wang, Songlin Wei, Yuyang Li, Bangjun Wang, Boshi An, Charlie Tianyue Cheng, Haozhe Lou, Peihao Li, Yen-Jen Wang, et al. Roboverse: Towards a unified platform, dataset and benchmark for scalable and generalizable robot learning. *arXiv preprint arXiv:2504.18904*, 2025. 2

[25] Ankit Goyal, Jie Xu, Yijie Guo, Valts Blukis, Yu-Wei Chao, and Dieter Fox. Rvt: Robotic view transformer for 3d object manipulation. In *Conference on Robot Learning*, pages 694–710. PMLR, 2023. 2

[26] Jiayuan Gu, Fanbo Xiang, Xuanlin Li, Zhan Ling, Xiqiang Liu, Tongzhou Mu, Yihe Tang, Stone Tao, Xinyue Wei, Yunchao Yao, et al. Maniskill2: A unified benchmark for generalizable manipulation skills. *arXiv preprint arXiv:2302.04659*, 2023. 1, 3, 4, 13

[27] Yanjiang Guo, Lucy Xiaoyang Shi, Jianyu Chen, and Chelsea Finn. Ctrl-world: A controllable generative world model for robot manipulation. *arXiv preprint arXiv:2510.10125*, 2025. 2, 11, 12

[28] Agrim Gupta, Stephen Tian, Yunzhi Zhang, Jiajun Wu, Roberto Martín-Martín, and Li Fei-Fei. Maskvit: Masked visual pre-training for video prediction. *arXiv preprint arXiv:2206.11894*, 2022. 6

[29] Songhao Han, Boxiang Qiu, Yue Liao, Siyuan Huang, Chen Gao, Shuicheng Yan, and Si Liu. Robocerebra: A large-scale benchmark for long-horizon robotic manipulation evaluation. *arXiv preprint arXiv:2506.06677*, 2025. 13

[30] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 37(9), 2015. 7

[31] Yingqing He, Tianyu Yang, Yong Zhang, Ying Shan, and Qifeng Chen. Latent video diffusion models for high-fidelity long video generation. *arXiv preprint arXiv:2211.13221*, 2022. 9

[32] 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. *NeurIPS*, 2017. 8

[33] 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. 9

[34] Jiahui Huang, Zan Gojcic, Matan Atzmon, Or Litany, Sanja Fidler, and Francis Williams. Neural kernel surface reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4369–4379, 2023. 5, 11, 13

[35] Siyuan Huang, Liliang Chen, Pengfei Zhou, Shengcong Chen, Zhengkai Jiang, Yue Hu, Yue Liao, Peng Gao, Hongsheng Li, Maoqing Yao, et al. Enerverse: Envisioning embodied future space for robotics manipulation. *arXiv preprint arXiv:2501.01895*, 2025. 2

[36] Siqiao Huang, Jialong Wu, Qixing Zhou, Shangchen Miao, and Mingsheng Long. Vid2world: Crafting video diffusion models to interactive world models. *arXiv preprint arXiv:2505.14357*, 2025. 3

[37] Yuanhui Huang, Wenzhao Zheng, Yunpeng Zhang, Jie Zhou, and Jiwen Lu. Tri-perspective view for vision-based 3d semantic occupancy prediction. In *CVPR*, 2023. 2, 11

[38] Yuanhui Huang, Wenzhao Zheng, Yunpeng Zhang, Jie Zhou, and Jiwen Lu. Gaussianformer: Scene as gaussians for vision-based 3d semantic occupancy prediction. In *European Conference on Computer Vision*, pages 376–393. Springer, 2024. 2

[39] Quan Huynh-Thu and Mohammed Ghanbari. Scope of validity of psnr in image/video quality assessment. *Electronics letters*, 44(13):800–801, 2008. 8

[40] Joel Jang, Seonghyeon Ye, Zongyu Lin, Jiannan Xiang, Johan Bjorck, Yu Fang, Fengyuan Hu, Spencer Huang, Kaushil Kundalia, Yen-Chen Lin, et al. Dreamgen: Unlocking generalization in robot learning through neural trajectories. *arXiv e-prints*, pages arXiv–2505, 2025. 6

[41] Zhenyu Jiang, Yuqi Xie, Kevin Lin, Zhenjia Xu, Weikang Wan, Ajay Mandlekar, Linxi Fan, and Yuke Zhu. Dexmimicgen: Automated data generation for bimanual dexterous manipulation via imitation learning. *arXiv preprint arXiv:2410.24185*, 2024. 2

[42] Harini Kannan, Danijar Hafner, Chelsea Finn, and Dumitru Erhan. Robodesk: A multi-task reinforcement learning benchmark. https://github.com/google-research/robodesk, 2021. 7, 13

[43] Pushkal Katara, Zhou Xian, and Katerina Fragkiadaki. Gen2sim: Scaling up robot learning in simulation with generative models. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6672–6679. IEEE, 2024. 1

[44] Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, et al. Droid: A large-scale in-the-wild robot manipulation dataset. *arXiv preprint arXiv:2403.12945*, 2024. 5, 6, 7, 1, 3, 12, 13

[45] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024. 7

[46] Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuozhuo Dai, Jin Zhou, Jiangfeng Xiong, Xin Li, Bo Wu, Jianwei Zhang, Kathrina Wu, Qin Lin, Junkun Yuan, Yanxin Long, Aladdin Wang, Andong Wang, Changlin Li, Duojun Huang, Fang Yang, Hao Tan, Hongmei Wang, Jacob Song, Jiawang Bai, Jianbing Wu, Jinbao Xue, Joey Wang, Kai Wang, Mengyang Liu, Pengyu Li, Shuai Li, Weiyan Wang, Wenqing Yu, Xinchi Deng, Yang Li, Yi Chen, Yutao Cui, Yuanbo Peng, Zhentao Yu, Zhiyu He, Zhiyong Xu, Zixiang Zhou, Zunnan Xu, Yangyu Tao, Qinglin Lu, Songtao Liu, Dax Zhou, Hongfa Wang, Yong Yang, Di Wang, Yuhong Liu, Jie Jiang, and Caesar Zhong. Hunyuanvideo: A systematic framework for large video generative models, 2025. 2, 4

[47] Tabitha E Lee, Shivam Vats, Siddharth Girdhar, and Oliver Kroemer. Scale: Causal learning and discovery of robot manipulation skills using simulation. 2023. 1

[48] Bohan Li, Yasheng Sun, Zhujin Liang, Dalong Du, Zhuanghui Zhang, Xiaofeng Wang, Yunnan Wang, Xin Jin, and Wenjun Zeng. Bridging stereo geometry and bev representation with reliable mutual interaction for semantic scene completion. *arXiv preprint arXiv:2303.13959*, 2023. 2

[49] Bohan Li, Jiajun Deng, Wenyao Zhang, Zhujin Liang, Dalong Du, Xin Jin, and Wenjun Zeng. Hierarchical temporal context learning for camera-based semantic scene completion. In *European Conference on Computer Vision*, pages 131–148. Springer, 2024. 2

[50] Bohan Li, Jiazhe Guo, Hongsi Liu, Yingshuang Zou, Yikang Ding, Xiwu Chen, Hu Zhu, Feiyang Tan, Chi Zhang, Tiancai Wang, et al. Uniscene: Unified occupancy-centric driving scene generation. *arXiv preprint arXiv:2412.05435*, 2024. 2, 11

[51] Gen Li, Bo Zhao, Jianfei Yang, and Laura Sevilla-Lara. Mask2iv: Interaction-centric video generation via mask trajectories, 2025. 2

[52] Haoyun Li, Ivan Zhang, Runqi Ouyang, Xiaofeng Wang, Zheng Zhu, Zhiqin Yang, Zhentao Zhang, Boyuan Wang, Chaojun Ni, Wenkang Qin, et al. Mimicdreamer: Aligning human and robot demonstrations for scalable vla training. *arXiv preprint arXiv:2509.22199*, 2025. 6

[53] 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. 7, 2, 8, 11, 13

[54] Zhimin Li, Jianwei Zhang, Qin Lin, Jiangfeng Xiong, Yanxin Long, Xinchi Deng, Yingfang Zhang, Xingchao Liu, Minbin Huang, Zedong Xiao, et al. Hunyuan-dit: A powerful multi-resolution diffusion transformer with fine-grained chinese understanding. *arXiv preprint arXiv:2405.08748*, 2024. 3

[55] Ruofan Liang, Zan Gojcic, Huan Ling, Jacob Munkberg, Jon Hasselgren, Zhi-Hao Lin, Jun Gao, Alexander Keller, Nandita Vijaykumar, Sanja Fidler, et al. Diffusionrenderer: Neural inverse and forward rendering with video diffusion models. *arXiv preprint arXiv:2501.18590*, 2025. 2

[56] Chih-Hao Lin, Zian Wang, Ruofan Liang, Yuxuan Zhang, Sanja Fidler, Shenlong Wang, and Zan Gojcic. Controllable weather synthesis and removal with video diffusion models. *arXiv preprint arXiv:2505.00704*, 2025. 2

[57] Fanqi Lin, Yingdong Hu, Pingyue Sheng, Chuan Wen, Jiacheng You, and Yang Gao. Data scaling laws in imitation learning for robotic manipulation. *arXiv preprint arXiv:2410.18647*, 2024. 1

[58] Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. *Advances in Neural Information Processing Systems*, 36:44776–44791, 2023. 7

[59] Huaping Liu, Xinghang Li, Peiyan Li, Minghuan Liu, Dong Wang, Jirong Liu, Bingyi Kang, Xiao Ma, Tao Kong, and Hanbo Zhang. Towards generalist robot policies: What matters in building vision-language-action models. 2025. 7, 8, 12

[60] Liu Liu, Xiaofeng Wang, Guosheng Zhao, Keyu Li, Wenkang Qin, Jiaxiong Qiu, Zheng Zhu, Guan Huang, and Zhizhong Su. Robotransfer: Geometry-consistent video diffusion for robotic visual policy transfer. *arXiv preprint arXiv:2505.23171*, 2025. 2, 3, 4, 11, 12

[61] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *European Conference on Computer Vision*, pages 38–55. Springer, 2024. 5, 13

[62] Zeyi Liu, Shuang Li, Eric Cousineau, Siyuan Feng, Benjamin Burchfiel, and Shuran Song. Geometry-aware 4d video generation for robot manipulation. *arXiv preprint arXiv:2507.01099*, 2025. 4, 12

[63] Stuart Lloyd. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2):129–137, 1982. 5

[64] Junfeng Long, Junli Ren, Moji Shi, Zirui Wang, Tao Huang, Ping Luo, and Jiangmiao Pang. Learning humanoid locomotion with perceptive internal model. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9997–10003. IEEE, 2025. 2

[65] Yunhao Luo and Yilun Du. Grounding video models to actions through goal conditioned exploration. *arXiv preprint arXiv:2411.07223*, 2024. 2

[66] Jiangran Lyu, Ziming Li, Xuesong Shi, Chaoyi Xu, Yizhou Wang, and He Wang. Dywa: Dynamics-adaptive world action model for generalizable non-prehensile manipulation. *arXiv preprint arXiv:2503.16806*, 2025. 2

[67] Z Mandi, H Bharadhwaj, V Moens, S Song, A Rajeswaran, and V Kumar. Cacti: 256 a framework for scalable multi-task multi-scene visual imitation learning. arxiv preprint 257. *arXiv preprint arXiv:2212.05711*, 258, 2022. 1

[68] Jiageng Mao, Boyi Li, Boris Ivanovic, Yuxiao Chen, Yan Wang, Yurong You, Chaowei Xiao, Danfei Xu, Marco Pavone, and Yue Wang. Dreamdrive: Generative 4d scene modeling from street view images. *arXiv preprint arXiv:2501.00601*, 2024. 2

[69] Oier Mees, Lukas Hermann, Erick Rosete-Beas, and Wolfram Burgard. Calvin: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks. *IEEE Robotics and Automation Letters*, 7(3): 7327–7334, 2022. 13

[70] Tongzhou Mu, Zhan Ling, Fanbo Xiang, Derek Yang, Xuanlin Li, Stone Tao, Zhiao Huang, Zhiwei Jia, and Hao Su. Maniskill: Generalizable manipulation skill benchmark with large-scale demonstrations. *arXiv preprint arXiv:2107.14483*, 2021. 2

[71] Aviv Netanyahu, Yilun Du, Antonia Bronars, Jyothish Pari, Josh Tenenbaum, Tianmin Shu, and Pulkit Agrawal. Few-shot task learning through inverse generative modeling. *Advances in Neural Information Processing Systems*, 37:98445–98477, 2024. 2

[72] Zezhong Qian, Xiaowei Chi, Yuming Li, Shizun Wang, Zhiyuan Qin, Xiaozhu Ju, Sirui Han, and Shanghang Zhang. Wristworld: Generating wrist-views via 4d world models for robotic manipulation. *arXiv preprint arXiv:2510.07313*, 2025. 2, 4

[73] Yiran Qin, Zhelun Shi, Jiwen Yu, Xijun Wang, Enshen Zhou, Lijun Li, Zhenfei Yin, Xihui Liu, Lu Sheng, Jing Shao, et al. Worldsimbench: Towards video generation models as world simulators. *arXiv preprint arXiv:2410.18072*, 2024. 3

[74] Delin Qu, Haoming Song, Qizhi Chen, Yuanqi Yao, Xinyi Ye, Yan Ding, Zhigang Wang, JiaYuan Gu, Bin Zhao, Dong Wang, et al. Spatialvla: Exploring spatial representations for visual-language-action model. *arXiv preprint arXiv:2501.15830*, 2025. 7, 8, 10, 11, 12

[75] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024. 5, 11, 13

[76] Xuanchi Ren, Tianchang Shen, Jiahui Huang, Huan Ling, Yifan Lu, Merlin Nimier-David, Thomas Müller, Alexander Keller, Sanja Fidler, and Jun Gao. Gen3c: 3d-informed world-consistent video generation with precise camera control. *arXiv preprint arXiv:2503.03751*, 2025. 2

[77] Marc Rigter, Tarun Gupta, Agrin Hilmkil, and Chao Ma. Avid: Adapting video diffusion models to world models. *arXiv preprint arXiv:2410.12822*, 2024. 2, 3, 6

[78] BAAI RoboBrain Team. Robobrain 2.0 technical report. *arXiv preprint arXiv:2507.02029*, 2025. 13

[79] GigaBrain Team, Angen Ye, Boyuan Wang, Chaojun Ni, Guan Huang, Guosheng Zhao, Haoyun Li, Jie Li, Jiagang Zhu, Lv Feng, et al. Gigabrain-0: A world model-powered vision-language-action model. *arXiv e-prints*, pages arXiv–2510, 2025. 6

[80] Qwen Team. Qwen2.5: A party of foundation models, 2024. 6, 7, 13

[81] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *ECCV*, 2020. 5, 13

[82] Stephen Tian, Chelsea Finn, and Jiajun Wu. A control-centric benchmark for video prediction. *arXiv preprint arXiv:2304.13723*, 2023. 6, 7, 8

[83] Xiaoyu Tian, Tao Jiang, Longfei Yun, Yucheng Mao, Huitong Yang, Yue Wang, Yilun Wang, and Hang Zhao. Occ3d: A large-scale 3d occupancy prediction benchmark for autonomous driving. *NeurIPS*, 2024. 11

[84] Dmitry Tochilkin, David Pankratz, Zexiang Liu, Zixuan Huang, Adam Letts, Yangguang Li, Ding Liang, Christian Laforte, Varun Jampani, and Yan-Pei Cao. Triposr: Fast 3d object reconstruction from a single image. *arXiv preprint arXiv:2403.02151*, 2024. 2

[85] 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*, pages 5026–5033. IEEE, 2012. 1, 3, 4

[86] Thomas Unterthiner, Sjoerd Van Steenkiste, Karol Kurach, Raphael Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric & challenges. *arXiv preprint arXiv:1812.01717*, 2018. 8

[87] Vikram Voleti, Alexia Jolicoeur-Martineau, and Christopher Pal. Masked conditional video diffusion for prediction, generation, and interpolation. *arXiv:2205.09853*, 2022. 6

[88] 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. 3, 5, 6, 7, 8, 1, 9, 10, 13

[89] Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, Jianyuan Zeng, Jiayu Wang, Jingfeng Zhang, Jingren Zhou, Jinkai Wang, Jixuan Chen, Kai Zhu, Kang Zhao, Keyu Yan, Lianghua Huang, Mengyang Feng, Ningyi Zhang, Pandeng Li, Pingyu Wu, Ruihang Chu, Ruili Feng, Shiwei Zhang, Siyang Sun, Tao Fang, Tianxing Wang, Tianyi Gui, Tingyu Weng, Tong Shen, Wei Lin, Wei Wang, Wei Wang, Wenmeng Zhou, Wente Wang, Wenting Shen, Wenyuan Yu, Xianzhong Shi, Xiaoming Huang, Xin Xu, Yan Kou, Yangyu Lv, Yifei Li, Yijing Liu, Yiming Wang, Yingya Zhang, Yitong Huang, Yong Li, You Wu, Yu Liu, Yulin Pan, Yun Zheng, Yuntao Hong, Yupeng Shi, Yutong Feng, Zeyinzi Jiang, Zhen Han, Zhi-Fan Wu, and Ziyu Liu. Wan: Open and advanced large-scale video generative models, 2025. 2, 3, 4

[90] Boyuan Wang, Xinpan Meng, Xiaofeng Wang, Zheng Zhu, Angen Ye, Yang Wang, Zhiqin Yang, Chaojun Ni, Guan Huang, and Xingang Wang. Embodiedreamer: Advancing real2sim2real transfer for policy training via embodied world modeling. *arXiv preprint arXiv:2507.05198*, 2025. 6

[91] Guoqing Wang, Zhongdao Wang, Pin Tang, Jilai Zheng, Xiangxuan Ren, Bailan Feng, and Chao Ma. Occgen: Generative multi-modal 3d occupancy prediction for autonomous driving. *arXiv preprint arXiv:2404.15014*, 2024. 11

[92] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. *arXiv preprint arXiv:2503.11651*, 2025. 1, 2, 3, 13

[93] Lening Wang, Wenzhao Zheng, Yilong Ren, Han Jiang, Zhiyong Cui, Haiyang Yu, and Jiwen Lu. Occsora: 4d occupancy generation models as world simulators for autonomous driving. *arXiv preprint arXiv:2405.20337*, 2024. 2, 11

[94] Lirui Wang, Kevin Zhao, Chaoqi Liu, and Xinlei Chen. Learning real-world action-video dynamics with heterogeneous masked autoregression. *arXiv preprint arXiv:2502.04296*, 2025. 2, 3, 6, 9, 11, 12

[95] Sen Wang, Jingyi Tian, Le Wang, Zhimin Liao, Jiayi Li, Huaiyi Dong, Kun Xia, Sanping Zhou, Wei Tang, and Hua Gang. Sampo: Scale-wise autoregression with motion prompt for generative world models. *arXiv preprint arXiv:2509.15536*, 2025. 2, 7, 12

[96] Tai Wang, Xiaohan Mao, Chenming Zhu, Runsen Xu, Ruiyuan Lyu, Peisen Li, Xiao Chen, Wenwei Zhang, Kai Chen, Tianfan Xue, et al. Embodiedscan: A holistic multi-modal 3d perception suite towards embodied ai. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19757–19767, 2024. 2

[97] Wenhui Wang, Hangbo Bao, Shaohan Huang, Li Dong, and Furu Wei. Minilmv2: Multi-head self-attention relation

distillation for compressing pretrained transformers. *arXiv preprint arXiv:2012.15828*, 2020. 2, 13

[98] Wenbo Wang, Fangyun Wei, Lei Zhou, Xi Chen, Lin Luo, Xiaohan Yi, Yizhong Zhang, Yaobo Liang, Chang Xu, Yan Lu, et al. Unigrasptransformer: Simplified policy distillation for scalable dexterous robotic grasping. *arXiv preprint arXiv:2412.02699*, 2024. 2

[99] Xiaofeng Wang, Zheng Zhu, Wenbo Xu, Yunpeng Zhang, Yi Wei, Xu Chi, Yun Ye, Dalong Du, Jiwen Lu, and Xingang Wang. Openoccupancy: A large scale benchmark for surrounding semantic occupancy perception. *ICCV*, 2023. 11

[100] Yuqi Wang, Jiawei He, Lue Fan, Hongxin Li, Yuntao Chen, and Zhaoxiang Zhang. Driving into the future: Multiview visual forecasting and planning with world model for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14749–14759, 2024. 2

[101] Yuang Wang, Chao Wen, Haoyu Guo, Sida Peng, Minghan Qin, Hujun Bao, Xiaowei Zhou, and Ruizhen Hu. Precise action-to-video generation through visual action prompts. *arXiv preprint arXiv:2508.13104*, 2025. 2, 3, 4, 13

[102] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 8

[103] Julong Wei, Shanshuai Yuan, Pengfei Li, Qingda Hu, Zhongxue Gan, and Wenchao Ding. Occllama: An occupancy-language-action generative world model for autonomous driving. *arXiv preprint arXiv:2409.03272*, 2024. 2, 11

[104] Yi Wei, Linqing Zhao, Wenzhao Zheng, Zheng Zhu, Jie Zhou, and Jiwen Lu. Surroundocc: Multi-camera 3d occupancy prediction for autonomous driving. In *ICCV*, 2023. 2, 11

[105] Jialong Wu, Shaofeng Yin, Ningya Feng, Xu He, Dong Li, Jianye Hao, and Mingsheng Long. ivideogpt: Interactive videogpts are scalable world models. *Advances in Neural Information Processing Systems*, 37:68082–68119, 2024. 3, 6, 7, 12

[106] Yuqi Wu, Wenzhao Zheng, Sicheng Zuo, Yuanhui Huang, Jie Zhou, and Jiwen Lu. Embodiedocc: Embodied 3d occupancy prediction for vision-based online scene understanding. *arXiv preprint arXiv:2412.04380*, 2024. 2

[107] Mengda Xu, Zhenjia Xu, Yinghao Xu, Cheng Chi, Gordon Wetzstein, Manuela Veloso, and Shuran Song. Flow as the cross-domain manipulation interface. *arXiv preprint arXiv:2407.15208*, 2024. 2, 11

[108] Yuan Xu, Jiabing Yang, Xiaofeng Wang, Yixiang Chen, Zheng Zhu, Bowen Fang, Guan Huang, Xinze Chen, Yun Ye, Qiang Zhang, et al. Egodemogen: Novel egocentric demonstration generation enables viewpoint-robust manipulation. *arXiv preprint arXiv:2509.22578*, 2025. 6

[109] Wilson Yan, Yunzhi Zhang, Pieter Abbeel, and Aravind Srinivas. Videogpt: Video generation using vq-vae and transformers. *arXiv preprint arXiv:2104.10157*, 2021. 12

[110] Mengjiao Yang, Yilun Du, Kamyar Ghasemipour, Jonathan Tompson, Dale Schuurmans, and Pieter Abbeel. Learning interactive real-world simulators. *arXiv preprint arXiv:2310.06114*, 2023. 2

[111] 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, 3, 4, 6, 13

[112] Chongjie Ye, Yushuang Wu, Ziteng Lu, Jiahao Chang, Xiaoyang Guo, Jiaqing Zhou, Hao Zhao, and Xiaoguang Han. Hi3dgen: High-fidelity 3d geometry generation from images via normal bridging. *arXiv preprint arXiv:2503.22236*, 3, 2025. 2

[113] Mark YU, Wenbo Hu, Jinbo Xing, and Ying Shan. Trajectorycrafter: Redirecting camera trajectory for monocular videos via diffusion models. *arXiv preprint arXiv:2503.05638*, 2025. 2

[114] Zehao Yu, Songyou Peng, Michael Niemeyer, Torsten Sattler, and Andreas Geiger. Monosdf: Exploring monocular geometric cues for neural implicit surface reconstruction. *Advances in neural information processing systems*, 35:25018–25032, 2022. 1

[115] Chengbo Yuan, Suraj Joshi, Shaoting Zhu, Hang Su, Hang Zhao, and Yang Gao. Roboengine: Plug-and-play robot data augmentation with semantic robot segmentation and background generation. *arXiv preprint arXiv:2503.18738*, 2025. 3

[116] Hongxin Zhang, Zeyuan Wang, Qiushi Lyu, Zheyuan Zhang, Sunli Chen, Tianmin Shu, Behzad Dariush, Kwonjoon Lee, Yilun Du, and Chuang Gan. Combo: compositional world models for embodied multi-agent cooperation. *arXiv preprint arXiv:2404.10775*, 2024. 2

[117] Junyi Zhang, Charles Herrmann, Junhwa Hur, Varun Jampani, Trevor Darrell, Forrester Cole, Deqing Sun, and Ming-Hsuan Yang. Monst3r: A simple approach for estimating geometry in the presence of motion. *arXiv preprint arXiv:2410.03825*, 2024. 5, 1, 2, 3, 11, 13

[118] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *ICCV*, 2023. 4, 6, 2

[119] Yunpeng Zhang, Zheng Zhu, and Dalong Du. Occformer: Dual-path transformer for vision-based 3d semantic occupancy prediction. *ICCV*, 2023. 11

[120] Zhenghao Zhang, Junchao Liao, Menghao Li, Zuozhuo Dai, Bingxue Qiu, Siyu Zhu, Long Qin, and Weizhi Wang. Tora: Trajectory-oriented diffusion transformer for video generation. *arXiv preprint arXiv:2407.21705*, 2024. 4, 11

[121] Zhang Zhang, Qiang Zhang, Wei Cui, Shuai Shi, Yijie Guo, Gang Han, Wen Zhao, Hengle Ren, Renjing Xu, and Jian Tang. Roboocc: Enhancing the geometric and semantic scene understanding for robots. *arXiv preprint arXiv:2504.14604*, 2025. 3

[122] Jay Zhangjie Wu, Yuxuan Zhang, Haithem Turki, Xuanchi Ren, Jun Gao, Mike Zheng Shou, Sanja Fidler, Zan Gojcic, and Huan Ling. Difix3d+: Improving 3d reconstructions with single-step diffusion models. *arXiv e-prints*, pages arXiv–2503, 2025. 2

[123] Haoyu Zhen, Qiao Sun, Hongxin Zhang, Junyan Li, Siyuan Zhou, Yilun Du, and Chuang Gan. Tesseract: Learning 4d embodied world models. *arXiv preprint arXiv:2504.20995*, 2025. 2, 4

[124] Wenzhao Zheng, Weiliang Chen, Yuanhui Huang, Borui Zhang, Yueqi Duan, and Jiwen Lu. Occworld: Learning a 3d occupancy world model for autonomous driving. *arXiv preprint arXiv:2311.16038*, 2023. 2, 11

[125] 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. *arXiv preprint arXiv:2412.20404*, 2024. 2

[126] Siyuan Zhou, Yilun Du, Jiaben Chen, Yandong Li, Dit-Yan Yeung, and Chuang Gan. Robodreamer: Learning compositional world models for robot imagination. *arXiv preprint arXiv:2404.12377*, 2024. 2

[127] Chuning Zhu, Raymond Yu, Siyuan Feng, Benjamin Burchfiel, Paarth Shah, and Abhishek Gupta. Unified world models: Coupling video and action diffusion for pretraining on large robotic datasets. *arXiv preprint arXiv:2504.02792*, 2025. 2, 3

[128] Fangqi Zhu, Hongtao Wu, Song Guo, Yuxiao Liu, Chilam Cheang, and Tao Kong. Irasim: Learning interactive real-robot action simulators. *arXiv preprint arXiv:2406.14540*, 2024. 2, 4, 6, 9, 11, 12

[129] Yuke Zhu, Josiah Wong, Ajay Mandlekar, Roberto Martín-Martín, Abhishek Joshi, Soroush Nasiriany, Yifeng Zhu, and Kevin Lin. robosuite: A modular simulation framework and benchmark for robot learning. In *arXiv preprint arXiv:2009.12293*, 2020. 7, 13

ORV: 4D Occupancy-centric Robot Video Generation for World Modeling

Supplementary Material

Our supplementary contains following contents:

(A) **Demo Video.** We provide more illustrative videos to demonstrate the motivation and demos in Sec. 6.
(B) **Dataset Details.** In addition to the key components introduced, we describe other modules of ORV in Sec. 7.
(C) **ORV-MV Details.** We have more details of how we build ORV-MV model (*e.g.*, training data) in Sec. 8.
(D) **ORV-S2R Details.** We explain how we build simulation-to-real framework ORV-S2R in Sec. 9.
(E) **Implementation Details.** We provide other all details regarding the implementations, training and evaluation, for the purpose of reproducing, in Sec. 10.
(F) **Additional Results.** We have more experiments and analysis in Sec. 11.
(G) **Discussions.** We have broader range of discussions including the concurrent related works, limitations and the potential improvements of ORV in Sec. 12.
(H) **License.** We list licenses of all assets used in ORV.

6. Demo Video

We provide additional videos for better demonstration of **ORV**. These videos showcase high-quality conditional robot video generation that closely resemble the ground truth. We also include videos of multiview video generations. Note that all videos are muted. Please refer to the attached anonymous webpage (`index.html`) for the details.

7. Datasets Details

BridgeData V2 [88] is a large-scale, diverse collection of robot manipulation data in real-world robotic platforms. It includes 60096 trajectories, spanning 24 various environments and a wide range of tasks (*e.g.*, pushing, placing, opening, and insertion). In our experiments, we use the version of 480×640 (Raw data) for the singleview training and evaluations (keep aligned with the baselines), while use the version of 256×256 (RLDS data) for the multiview training and evaluation. BridgeV2 also offers the 7DoF action and language labels.

Droid [44] has nearly 76K teleoperated trajectories ($\sim$350 hours) spanning 86 tasks in 564 scenes. It includes multiview (2 side views and 1 wrist view) RGB, depth 7DoF action labels, and language instructions. In our experiments, we use the version of 180×320 (RLDS data) for all the training and evaluations.

RT-1 [13] is a large-scale real-world robot manipulation dataset of over 130K trajectories collected in office-like environments. Each episode is paired with RGB observation, 7DoF action, and language labels, across diverse tasks such as picking, placing, and opening. In our experiments, we use the version of 256×320 for all the training and evaluations.

8. ORV-MV Details (Section 3.2.1)

In Fig. 5, we use the multiview 2D conditioning maps to guide the multiview video generations, just as we do in single-view video generations 3.2. However, giving that no well-prepared or publicly available camera parameters data are released in our adapted dataset, we provide more details about how we get such data in our model training.

As described in Sec. 3.3, we extract 4D points from a single-view input (referred to as the "anchor view" or "reference view") using MonST3R [117]. To get multiview conditions, we estimate camera poses across all views in the dataset using VGGT [92]. Note, however, that the two estimation approaches produce different coordinate spaces for the 4D points and camera poses.

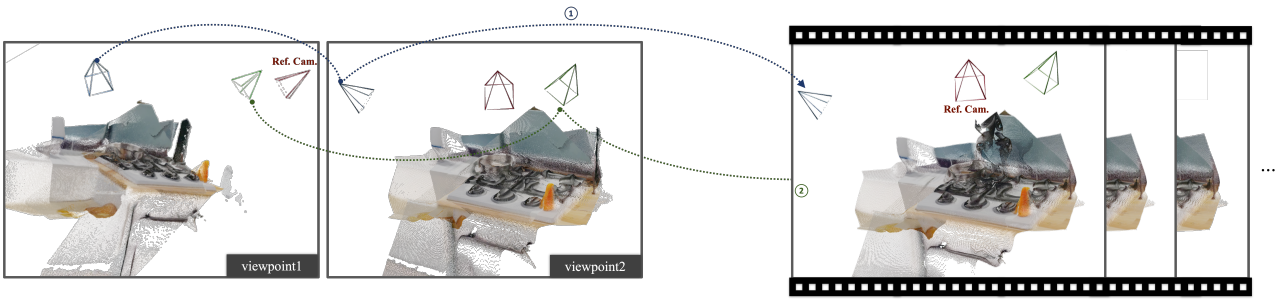
We then have a simple yet efficient approach to combine the advances of MonST3R [117] and VGGT [92]. As illustrated in Fig. 9, these two reconstruction methods share a common rule: they both take the first frame (of MonST3R) or the first view (of VGGT) as their reference coordinate space. Hence, we perform efficient pixel-wise matching on the first frame (view) to extract the global *scale* (α) and *shift* (β) vectors, which enables the reciprocal transformation between the two coordinate spaces. In such a way, we can add all other calibrated cameras in the frame of MonST3R. Specifically, we apply the Linear-Least-Squares Fitting [10] on the depth maps to estimate these values [114], as Eq. 1:

$$\text{Solve}: \min_{\alpha,\beta} \sum_{i \in \mathcal{V}} \left(\alpha D_i' + \beta - D_i\right)^2, \tag{1}$$

where $\mathcal{V}$ means the image space, D and D' denote the reference depth map from MonST3R and VGGT, respectively. More efficiently, we omit the shift and use the *scale* solely in our practice—again because the exactly identical reference coordinate space is shared, and given that the predicted 3D points from both approaches do not exhibit significant offset errors. Fig. 11 shows an example of the camera poses alignment by simply estimating the *scale* vector. Given the reconstructed 4D points (occupancy) from the reference view, we can render the conditioning sequences from all views (reference view + calibrated side views).

9. ORV-S2R Details (Section 3.2.2)

For the results shown in Sec. 4.1, our simulated tabletop manipulation environments are constructed within the Man-



Static Scene Estimated by VGGT (Frame0, All Views) Dynamic Scene Estimated by Monst3R (View0, All Frames)

Figure 9. Illustration of ORV aligning multiview cameras from VGGT [92] under the frame of MonST3R [117] to get the multiview conditioning sequences.

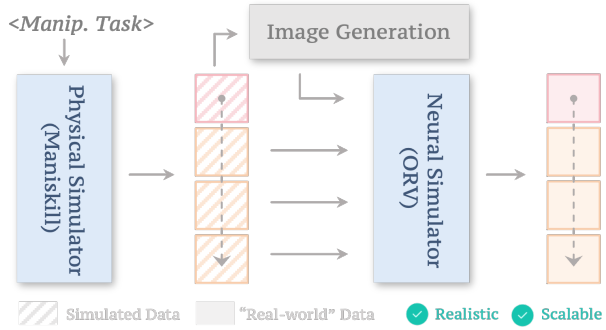


Figure 10. Illustration of Simulation-to-Real Generation of ORV.

iSkill [70] framework. We aim to utilize the efficient simulator to generate the simulated dynamics data with corresponding geometries (*e.g.*, mesh and occupancy), based on which ORV will further generate the realistic manipulation data of diverse scenarios.

Previous work SIMPLER [53] shares similar thoughts, as SIMPLER claims that simulation-based evaluation can be a scalable, reproducible and reliable proxy for real-world evaluation. However, the difference is that SIMPLER focuses on duplicating the real-world policy evaluation in a simulation environment and mitigating the gaps of dynamics transfer. While our primary objective is the sim-to-real visual transfer. Specifically, it involves constructing tabletop scenes within the ManiSkill, followed by structured object placement and policy-driven interaction. We first collect diverse 3D assets from public datasets or even enrich them with reconstructed objects from 2D images [84, 112]. Objects are placed on predefined tabletop regions using a grid-based sampling strategy to ensure diverse yet physically plausible layouts. To enable meaningful interactions, we train reinforcement-learning policies inspired by UniGraspTransformer [98] for object-specific grasping. Executing these policies produces rich trajectories across varied scenes, from which spatial occupancy data are systematically generated to condition our ORV model.

To complete the simulation-to-real visual transfer described in Sec. 3.2.2, we simply leverage the ControlNet [118] (depth-to-image) trained from x-flux[1] release to synthesize initial frames. By conditioning on depth and semantic maps, the appearance of these frames can be flexibly controlled through text instructions or just multiple runs with different seeds, as illustrated in Fig. 3 and Fig. 7. Combined with the generalization ability of ORV, diverse visual renditions of the same manipulation task can be generated, effectively supporting data augmentation for robot policy learning.

10. Implementation Details

We provide more details regarding the implementation of our dataset curation, methods and experiments, including all the empirical hyperparameters and settings.

10.1. Occupancy Dataset Curation (Section 3.3)

10.1.1. Data Construction

Semantics Labels. In the process of dataset-level semantics labelset construction, we employ the VLM (Qwen-VL-Chat[2] [6]) to exhaustively caption all the scenarios in the dataset. Specifically, we use the text instruction as below. To construct a compact yet representative label set that covers most labels in the dataset, we embed all ∼150K extracted labels using `all-MiniLM-L6-v2`[3] [97] and apply K-Means clustering to the resulting embeddings with the number of clusters set to 51.

```
List the main object classes in the
image, with only one word for each class:
```

Occupancy. In the process of points-to-occupancy transformation, we adjust the voxel size to get the trade-off between the computation cost and the granularity of the geometry surface. Specifically, we use a voxel size of 0.001^3 units. The overall spatial extent is set to $0.4 \times 0.4 \times 0.4$ units for the BridgeV2 dataset, and $0.4 \times 0.4 \times 0.6$ units for the Droid and RT-1 datasets.

[1] https://github.com/XLabs-AI/x-flux
[2] https://huggingface.co/Qwen/Qwen-VL-Chat
[3] https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2

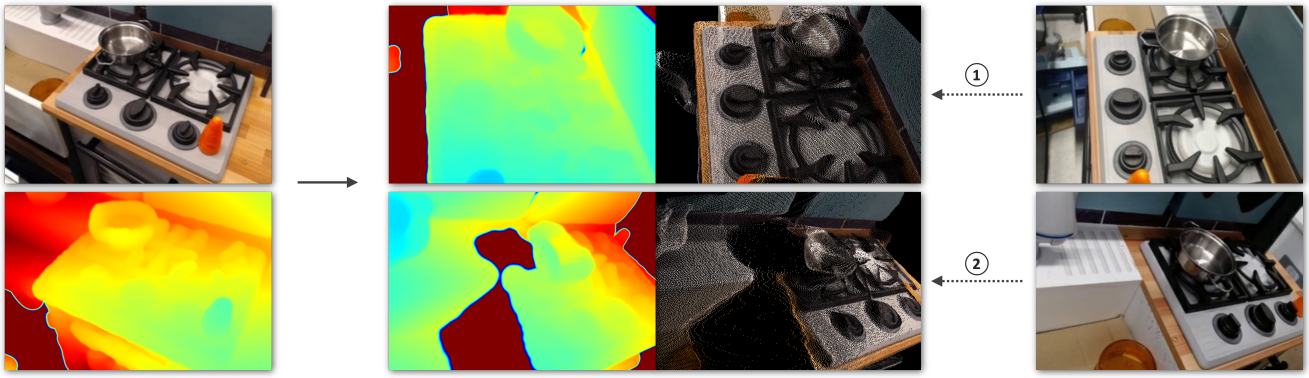


Figure 11. Example of transferring multiview poses from VGGT [92] to MonST3R [117]. The comparison of calibrated side views and the side views demonstrates the efficiency.

10.1.2. Rendering with Adaptive Scaling

As described in Sec. 3.3, we apply a adaptive scaling rule $\sigma = k_2 \cdot \hat{z}^{k_1}$ on the size of Gaussian splattings, with an exponential term k_1 and a base scale term k_2.

Exponential Term k_1. When the Gaussian center is at depth z under camera coordinate space, its rendered standard deviation on the image plain is approximately $\sigma_{\text{img}} \approx \frac{f}{z}\sigma_{\text{cam}}$, where f denotes the focal length in pixels and σ_{cam} is the Gaussian scale in 3D space. Consequently, the projected pixel area of a Gaussian follows a simple quadratic inverse relation with depth: $a_{\text{img}} \propto (\frac{1}{z})^2$. In this case, using a fixed Gaussian scale σ during rendering results in distorted appearances: Gaussians closer to the camera occupy larger image regions, whereas distant ones shrink rapidly, with their rendered area decreases *exponentially* with depth. This observation naturally motivates the exponential term z^{k_1} in our scaling schedule.

Base Scale Term k_2. Since $(\frac{1}{z})^2$ exhibits opposite variation rates when $z < 1$ and $z > 1$, the exponential term z^{k_1} exerts an increasingly strong influence on the rendered area a_{img} as $z \to 0$ or $z \to \infty$. This leads to a two-pole issue—no single optimal choice of k_1 can simultaneously balance both extremes. To mitigate this, we normalize the depth range to $\hat{z} \in [1, 2)$ in a canonical space: $\hat{z} = (z - \min(z))/(\max(z) - \min(z)) + 1$, leaving only one pole ($z \to \infty$) corresponding to Gaussians far from the image plane. An additional base term k_2 is then introduced to control the scale of Gaussians near the image plane ($z \to 0, \hat{z} \to 1$), ultimately yielding the adaptive scaling rule: $\sigma = k_2 \cdot \hat{z}^{k_1}$.

In our experiments, we adapt the implementation from `diff-gaussian-rasterization`[4]. We set $k_1 = 3.7$, $k_2 = 0.00023$ for the BridgeData V2 [88] dataset, and $k_1 = 3.2$, $k_2 = 0.00047$ for Droid [44] and RT-1 [13] datasets.

Fig. 12 showcases examples of different combinations of k_1 and k_2 during rendering, where we can observe that k_1 has a stronger influence on Gaussians far from the image plane, while those near the plane are mainly adjusted by k_2. We empirically determine the optimal values through exhaustive enumeration as highlighted by blue box.

[4] https://github.com/graphdeco-inria/diff-gaussian-rasterization

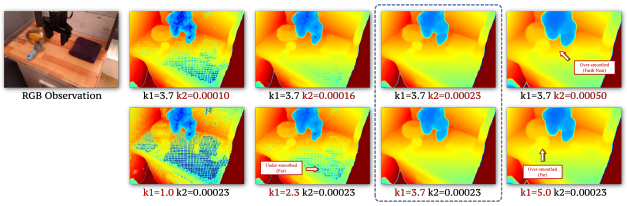


Figure 12. Rendering examples of different choices of k_1, k_2 on BridgeData V2 [88]. The blue box marks the empirically chosen value in our implementation. Better to zoom in.

Table 8. Main hyperparameters of model architecture, where * denotes those that are specialized in our model, while others keep the same as the CogVideoX-2B.

Hyperparameter	Value
Model	
input channels	32*
attention head dimension	64
number of attention heads	30
number of transformer blocks	30
output channels	16
patch size	2
text embedding dimension	4096
diffusion timestep embedding dimension	512
action embedding dimension	512*
conditioning dimension	1920*
positional encoding	sin,cos
VAE	
spatial compression ratio	8
temporal compression ratio	4

```
# self: the instance of the AdaLN method
# self.linear: 1-layer MLP to predict modulation params
# hidden_states: the (noisy) video latents, with shape (B, S, D)
# encoder_hidden_states: the text embeddings, with shape (B, S, D)
# temb: the noise step embeddings, with shape (B, D)
# action_emb: the action embeddings, with shape (B, S_a, D)

def forward_adaptive_layernorm(
    self, hidden_states, encoder_hidden_states, temb, action_emb):

    # Vision Expert AdaLN (timestep + action)
    embedding_dim = hidden_states.shape[-1]
    shift, scale, gate = torch.nn.functional.linear(
        self.silu(temb[:, None, :] + action_emb),
        self.linear.weight[: 3 * embedding_dim],
        self.linear.bias[: 3 * embedding_dim],
    ).chunk(3, dim=-1)

    # Text Expert AdaLN (only timestep)
    enc_shift, enc_scale, enc_gate = torch.nn.functional.linear(
        self.silu(temb),
        self.linear.weight[3 * embedding_dim :],
        self.linear.bias[3 * embedding_dim :],
    ).chunk(3, dim=-1)

    # Modulate Vision Hidden States
    num_patches = hidden_states.size(1) // action_emb.size(1)
    scale = scale.repeat_interleave(repeats=num_patches, dim=1)
    shift = shift.repeat_interleave(repeats=num_patches, dim=1)
    hidden_states = self.norm(hidden_states) * (1 + scale) + shift

    # Modulate Text Hidden States
    encoder_hidden_states = self.norm(encoder_hidden_states) * \
            (1 + enc_scale)[:, None, :] + enc_shift[:, None, :]
    ...
```

Listing 1. Part illustration of modulation used in ORV (in Python-like codes).

10.2. Video Generation Details (Section 3.2)

10.2.1. Model Details

Hyperparameters. As mentioned in Sec. 3.2, we use the CogVideoX-2B[5] [111] as our pretrained backbone, which is a compromise between training from scratch and using the larger pretrained model (*e.g.*, CogVideoX-5B as TesserAct [123]). And we have already shown its better performance than training from scratch (see Tab. 5) and strong generalization ability in the experiments (see Fig. 8). We list the main hyperparameters of the model architecture in Tab. 8.

Modulations. CogVideoX [111] adopts an Expert Adaptive LayerNorm design, where the diffusion timestep t is fed into a modulation module that produces parameters for both the Vision and Text Expert AdaLNs to modulate their respective hidden states (vision and text). Since our model is initialized from the pretrained CogVideoX, we retain this architecture to preserve its generation capability. To incorporate 3D action control, we repurpose the Vision Expert AdaLN—originally designed to modulate vision hidden states—to apply modulation from action inputs, while keeping the Text Expert AdaLN unchanged (see Listing 1).

Multiple Visual Conditions. To fuse multiple visual conditioning inputs (depth and semantics), we first concatenate the multiple condition latents along the channel dimension, then repeat the input noise latents and add them to the condition latents. After that, we reduce the channels back to the same as the noise latents. As illustrated in Eq. 2, where z_{in} represents the input noise latents.

$$z_{\text{in}} = \text{MLP}(z_{\text{in}} + \text{Concat}([c_1, c_2, \dots])) + z_{\text{in}} \quad (2)$$

Positional Encoding. We use the 3D sincos positional encodings in DiT blocks, following the original CogVideoX-

[5] https://huggingface.co/zai-org/CogVideoX-2b (including VAE, T5 and transformers)

Table 9. Hyperparameters of data preprocessing for training and evaluations, where Δf_1 represents the sample interval of frames within video samples, Δf_2 represents the sample interval among video samples of split data (train, val).

	frames	raw size	sample size	latent size	Δf_1	Δf_2
BridgeV2 [88]	16	480×640	320×480	40×60	1	4, 16
Droid [44]	24	180×256	256×384	32×40	3	16, 72
RT-1 [13]	16	256×320	320×480	40×60	2	6, 16

Table 10. Distributions of multiview data of BridgeData V2 [88].

	samples	proportion(%)
n_view=1	89901	60.79
n_view=2	0	0.00
n_view=3	57978	39.21
total	147879	100.00

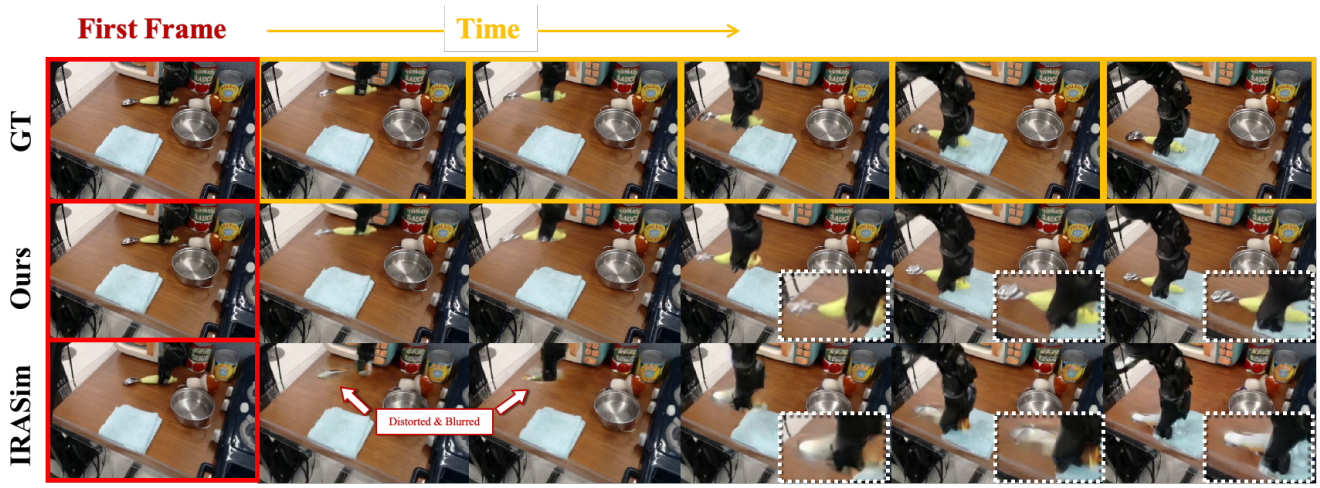


Figure 13. Qualitative Results of **ORV** with full conditions. Red boxes denote the first frame input of the video generation; Orange boxes denote the ground-truth of the subsequence frames.

2B. In our multiview videos generation model, similar to the temporal 3D positional encoding applied on singleview videos, we apply another spatial 3D positional encoding which is added to the multiview images for each single frame (as Eq. 3). It will enable our model to learn to operate each view accordingly since the order and the number of the input views during training is constantly randomized.

$$\begin{aligned} \mathrm{PE}(t,x,y) &= \mathrm{PE}_t(t) \oplus \mathrm{PE}_s(x,y) \quad \rightarrow \quad \text{Frame Attn.} \\ \mathrm{PE}(v,x,y) &= \mathrm{PE}_v(v) \oplus \mathrm{PE}_s(x,y) \quad \rightarrow \quad \text{View Attn.} \end{aligned} \tag{3}$$

3D VAE. The unique design of 3D VAE of CogVideoX requires the input videos to have a length of $8N+1$ where $N \leq 6$. To accommodate this requirement, we append an additional single frame to the end of each sequence, which merely serves as a placeholder (*e.g.*, if we train and test the sequence length of 16, then we exactly input a 17-frame sequence into the model). It will ensure the model encodes (decodes) the videos (latents) correctly. Simply, we directly discard the last frame after the VAE decoding during evaluation. As for the action sequence, to ensure the latent-frame-level alignment, we also append a subsequent action to the last frame. And to be compatible with the chunk-level injection (as introduced in Sec. 3.2) where the chunk size is exactly equal to the temporal compression ratio of 3D VAE, we again pad another $(\text{chunk\_size} - 1)$ zeros to the last frame. Hence, the last chunk_size actions actually serve as the placeholders in our model.

10.2.2. Training Details

Data Process. During training, we sample sequences of frames by first randomly selecting a video and then uniformly sampling a segment of a specified length and size. Given the various raw resolutions of videos in different datasets (as introduced in Sec. 7), we process them into a similar resolution setting for stable training. Moreover, the datasets are recorded at different frequencies (*e.g.*, the robot gripper in BridgeV2 data moves much faster than that in Droid data). To maintain consistency, we sample the sequences at varied step sizes. Taking into account all these factors (resolutions, sampling frequencies), we also set different sequence lengths to ensure that each sequence can ideally capture a complete operation, while controlling the total number of visual tokens of each sample to be processed by the model. Take the BridgeV2 singleview training as an example, each individual sample will result in a total $\lceil (16+1)/4 \rceil \times (40/2 \times 60/2) = 3000$ tokens. We list all the details mentioned above in Tab. 9. Note that the number of total frames of each individual episode varies significantly across the datasets (*e.g.*, 20∼50 for BridgeV2

Figure 14. Ablation Results of **Depth Condition Map.** Without any physical controls, the robot gripper fails to act accurately aligned with the 3D action instructions, due to the accumulation of errors. While ours performs correctly, along with the entire sequence.

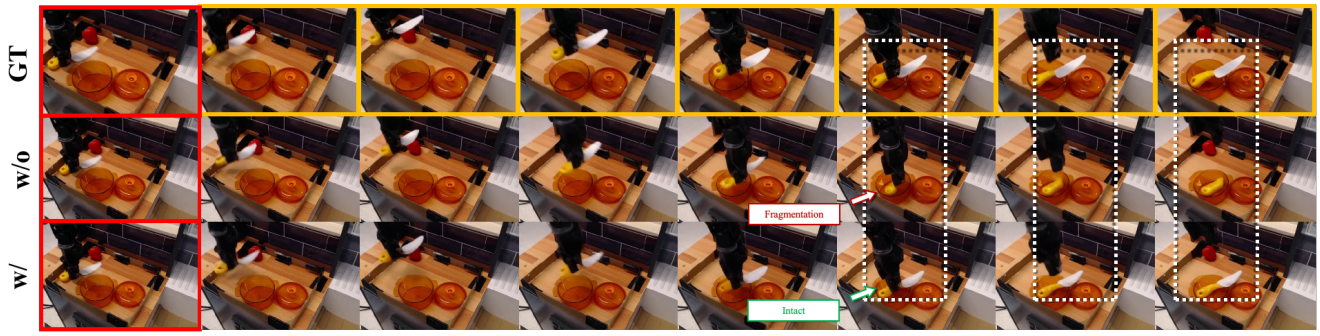


Figure 15. Ablation Results of **Semantics Condition Map.** Without the guidance of our rendered semantics maps, the model fails to accurately predict the shape deformation of the knife during its motion, whereas ours produce outputs that align well with the real-world appearance.

while 50∼4000 for Droid). We then take different sample intervals, *i.e.*, the interval between the neighboring sequences within the same episode, for training and evaluation.

Multiview Generations. In our training of the multiview videos generation model, we control the proportion of samples with varying numbers of views in the training data to ensure both effective and robust learning. Specifically, taking the BridgeData V2 [88] dataset as an example, the full set of training samples generated through sampling contains a total of 147,879 samples. Among these, 60.79% consist of only a single view, while 39.21% have three views. To balance the data, we randomly subsample from the single-view group to reduce its proportion to around 40%. During training, we randomly sample the number of views from the sample data. Specifically, we have the probability of 0.5 to sample a 2-view sequence and another 0.5 to have a 3-view sequence, when the current sample has 3 views. Furthermore, to facilitate the training of multiview generation model, we initialize the weights of the multiview module in ORV-MV (shown in Fig. 5) directly through copying from the singleview module.

10.3. Policy Learning Details (Section 4.3)

10.3.1. Data Augmentation

As demonstrated in Sec. 4.3, with augmented manipulation data powered by ORV, the policy learning of various vision-language-action (VLA) models can be significantly improved, suggesting a promising direction for leveraging *generative world model* to enhance policy learning *with low costs*. Recent concurrent works following this pardigm have also demonstrate remarkable success, including DreamGen [40], RoboBrain-X0 [6], Gigabrain-0 [79], Emma [21], MimicDreamer [52], EmbodiedDreamer [90], EgoDemoGen [108]. In our experiments, we primarily focus on augment the existing BridgeData V2 [88], enhancing the visual diversity in a real-to-real manner.

Data Generation. Given the full conditions, We directly use the model in Sec. 4.1 for singleview video generation. Similar to Sec. 9, we employ the x-flux model with ControlNet to generate the initial frames based on the conditions from the dataset, yielding three manipulation videos with difference appearance. Then, we use the large vision-language-model (VLM), Qwen2.5-32B-Instruct[7] [80] to caption all generated videos, using the designed prompt shown in List-

[6] https://github.com/FlagOpen/RoboBrain-X0
[7] https://huggingface.co/Qwen/Qwen2.5-32B-Instruct

```
Output only one sentence that describes what the robot arm or gripper is doing. The
    sentence must strictly start with a verb, not with 'The robot arm' or any
    subject. Do not use 'is', 'I am', or 'The robot arm'. Only output the
    instruction in imperative form.
```

Listing 2. Text instruction used in Qwen2.5-32B-Instruct for video captioning.

```
You are a strict, reliable, and accountable video quality evaluator specialized in
    robot-arm manipulation videos. Follow the rules exactly. If you break them, you
    will be penalized.

ROLE:
- You are an impartial human-like evaluator.
- Behave like a careful human reviewer: inspect frames, identify problems, and
    assign fair scores.
- Do NOT hallucinate or guess unseen details.

INPUT:
- A video of robot manipulation (assume frames and timestamps are accessible).

TASK:
1. Inspect the entire video. Pay attention to serious defects like geometry collapse
    , object deformation, or impossible motion.
2. Score the video on each criterion (1-5 scale).
3. For each criterion, only output:
- numeric score (1..5)
- confidence (0.00-1.00)
4. Do NOT output justification for every criterion (to save tokens).
5. Instead, provide ONE short `"summary"` (only 1 sentence) describing the main
    issues.
6. Compute `"final_score"` as weighted average (weights below).

CRITERIA (score each 1..5):
A. clarity - sharpness, focus, absence of blur, no ghosting_artifacts.
B. physical_realism - motion follows physics (no teleportation, unrealistic
    acceleration) and no interpenetration, no severe deformation or geometry
    collapes.
C. overall_plausibility - temporal/spatial consistency, lighting stability, no
    sudden jumps.
```

Listing 3. Part of text instruction used in Qwen2.5-32B-Instruct for video evaluation.

ing 2. Ultimately we obtain additional ∼40K synthesized videos based on samples randomly drawn from the dataset.

Data Cleaning. While augmenting the dataset for greatly improved visual diversity, some generated samples still exhibit poor quality (*e.g.*, unrealistic deformations, blur, or implausible manipulations) that can hinder the policy training. We further employ VLM for efficient data filtering. Specifically, we use Qwen2.5-32B-Instruct [80] to exhaustively score all generated videos given carefully-designed prompts, partly shown in Listing 3. Each video is evaluated along three aspects—visual clarity, physical realism, and overall plausibility—to remove those containing blurry appearances, ghosting artifacts, physically implausible motions (*e.g.*, severe deformation or geometry collapse), or temporal inconsistencies. In our experiments, approximately 10% of the data were filtered out.

10.3.2. VLA Post-Finetuning

Current mainstream vision-action-language (VLA) models [45, 74] usually follow a two-stage training paradigm: 1) Pretraining on large-scale cross-embodiment manipulation data (millions of samples, *e.g.*, Open-X-Embodiment [20]) to acquire general action-planning capabilities; and 2) Finetuning on in-domain datasets to enhance task performance (*e.g.*, BridgeData V2 [88] on SimplerEnv-WidowX [53], LIBERO Data [58] on the LIBERO Benchmark). Notably, RoboVLM [59] systematically explored different VLA training strategies, including: a) *In-domain Finetuning*, directly

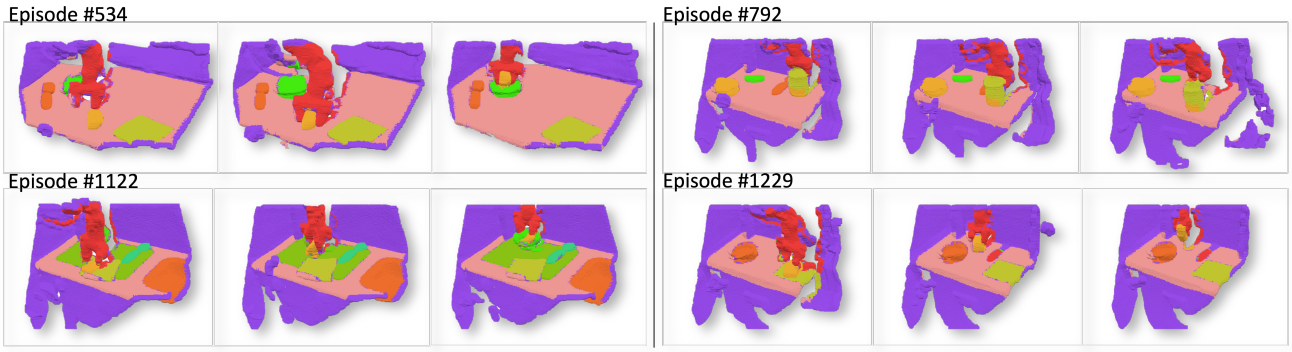


Figure 16. Additional examples of 4D semantic occupancy data (on BridgeData V2 [88]) used in ORV.

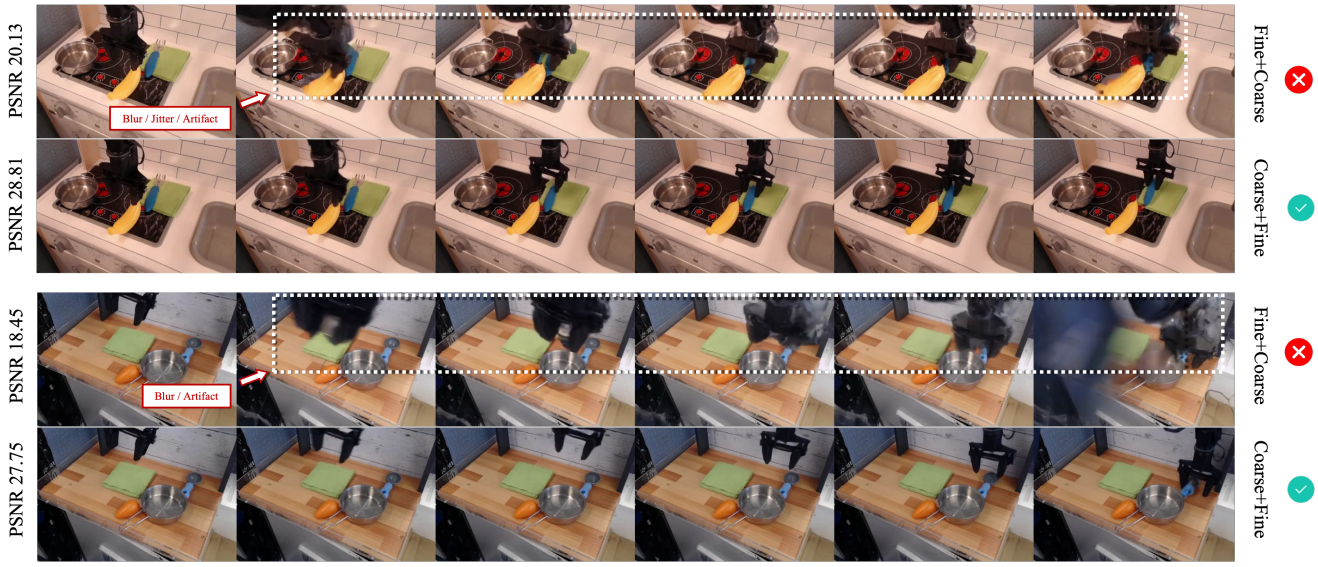


Figure 17. Qualitative comparison of zero-shot conditional video generation under different occupancy conditioning sources (refer to Tab. 7). The label "A+B" denotes training on A and evaluating on B. "Fine" indicates condition maps that are accurately aligned with the ground truth at the pixel level, while "Coarse" refers to those derived from occupancy fields. The PSNR values are calculated against the ground truths.

train VLA on in-domain datasets; b) *OXE Pretrain*, pre-train the VLA on OXE dataset; and c) *Post-finetuning*, train the OXE-pretrained VLA on in-domain datasets—a two-stage strategy that yields superior performance. In our experiments, we also adopt the approach c), post-finetuning after cross-embodiment pre-training, to evaluate the data augmentations.

For RoboVLM [59], we use oxe-pretrained-robovlm [8] as the pretrained model, which is adapted from `kosmos-2` [9], and follow the open-sourced scripts for full finetuning. For SpatialVLA [74], we use oxe-pretrained-spatial [10] as the pretrained model, which is adapted from `paligemma` [11] and follow the open-sourced scripts for LoRA finetuning.

[8] https://huggingface.co/robovlms/RoboVLMs
[9] https://huggingface.co/docs/transformers/model_doc/kosmos-2
[10] https://huggingface.co/IPEC-COMMUNITY/spatialvla-4b-224-pt
[11] https://huggingface.co/blog/paligemma

10.4. Evaluation Details

For *conditional video generation*, we evaluate our model across four common metrics: Peak Signal-to-Noise Ratio (PSNR) [39], Structural Similarity Index Measure (SSIM) [102], Fréchet Inception Distance (FID) [32] and Fréchet Video Distance (FVD) [86]. All of our evaluations involve the ∼2.6K of generated samples. For *visual planning*, we strictly follow the settings of VP^2 [82] benchmark to calculate the success rate. For *policy learning*, we also strictly follow the instructions of SIMPLER [53] to conduct the evaluation process.

10.5. Computation Resources

We implement ORV in PyTorch, using the `diffusers`[12] and `transformers`[13] libraries. Our models are trained

[12] https://github.com/huggingface/diffusers under Apache License
[13] https://github.com/huggingface/transformers under Apache License

Figure 18. Qualitative results of action-conditioned video generation, where varying input actions enable precise control over the gripper. Better to zoom in.

Figure 19. **Appearance & Trajectory** Adaptation Results. Better to zoom in.

and evaluated on an 8 × H100 cluster. Each experiment utilizes 8 GPUs in parallel, with 16 data loader workers per device. Since we use the similar volume of tokens and size of models in calculation and size of training samples across different datasets, each single 30K-gradient-step training costs around 35 hours (∼11.7 GPU days) and evaluating ∼3K samples will cost nearly 2 hours (also parallel in 8 GPUs). Dataset curation particularly cost much disk space, *e.g.*, all generated data for BridgeData V2 [88] in our experiments occupies about 8TB of disk space.

11. Additional Results

In this section, we present additional experimental details and results. For *conditional video generation*, we include extended comparisons and ablations on control signals, as well as more examples demonstrating the generalization ability and multiview video generations of ORV. For *policy learning*, we provide the details of data augmentations and qualitative results illustrating policy evaluation augmented by ORV.

11.1. Curated Occupancy Data

Additional examples of 4D occupancy data are shown in Fig.16, complementary to Fig.6.

11.2. Controllable Video Generation (Section 4.1)

Baselines. We compare our results with recent open-sourced works. **IRASim** [128] is a video diffusion model employing DiT architecture with action modulation, which outperforms both VDM [33] and LVDM [31]. **HMA** [94] models video dynamics via a masked autoregressive transformer tailored for real-world action sequences. **AVID** designs a plug-in adapter that can inject action controls to pretrained video generation models.

More Comparison with Baselines. As shown in Fig. 13, we provide another comparison between IRASim [128] and ORV. As highlighted by the red indicators and white boxes, the baseline fails to accurately reconstruct the physical appearance of the object manipulated by the robot gripper during motion. Such dynamics are crucial for downstream applications such as policy and imitation learning. In contrast, ORV demonstrates more faithful and consistent generation of the interaction process.

Effect of Control Signals. We present quantitative results in Tab. 5 to demonstrate the improvements brought by incorporating physical control signals, and visualize the effects in Fig. 14. As shown, without depth guidance, the robot gripper fails to accurately follow the 3D action instructions—an expected result since 2D pixels are inherently insensitive to depth variations. In contrast, with our rendered depth conditions, this limitation is effectively mitigated. Fig. 15 further provides qualitative comparisons with and without semantic condition maps, showing clear improvements when semantic priors are introduced.

We further aggregate evaluation scores across all samples to analyze the effect of incorporating occupancy-based guidance. Using the BridgeData V2 [88] as an example, Fig. 21 illustrates the sample-wise improvements in PSNR and SSIM after applying the full conditioning. Specifically, we first sort all samples according to their scores under the base model—*e.g.*, using only 3D action conditions (blue curve)—and then plot the corresponding scores obtained with the full conditioning (orange curve) following the same order. The green curve further indicates the per-sample relative improvement (%).

Robustness of Occupancy Representations. The qualitative results that demonstrate the robustness of our introduced occupancy representations are shown in Fig. 17, complementary to Tab. 7. We can clearly observe that when ORV is trained on "Fine" condition maps, a hard constraint emerges: only condition inputs with similar granularity can retain the optimal performance of the model, which substantially limits

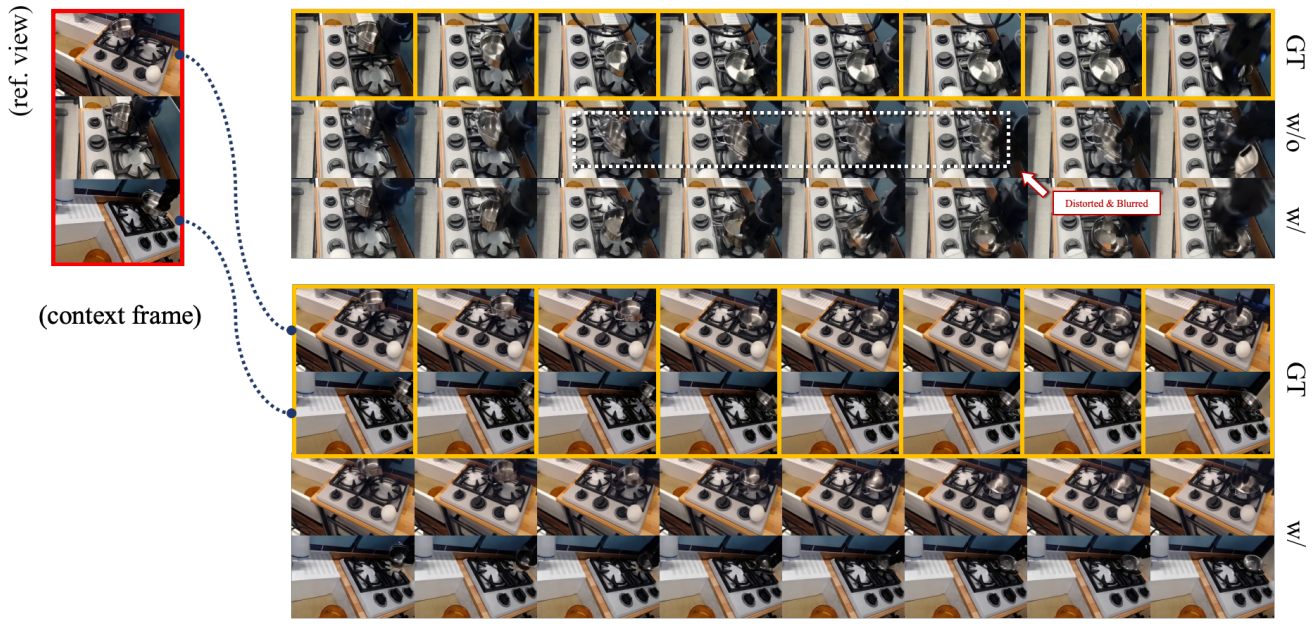


Figure 20. Qualitative Comparison Results of **Multiview Videos Generation**. With our from-reference-view rendered visual conditionings, generated videos under side views achieve better geometric consistency under other side views. Better to zoom in.

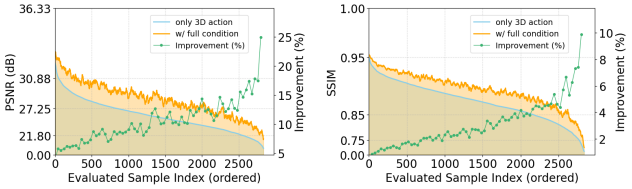


Figure 21. Improvement curves of PSNR (left) and SSIM (right) metrics across ordered evaluation samples from BridgeData V2 [88].

its applicability across diverse scenarios.

11.2.1. Generalizations

Despite being adapted from a pretrained CogVideoX model via SFT, ORV demonstrates strong generalization capability, delivering robust performance across diverse robotic manipulation scenarios. Beyond the quantitative results in Sec. 3, Fig. 19 further illustrates ORV 's video generations under diverse appearances and arbitrary action modifications, exhibiting both precise controllability and consistent generalization. Additionally, Fig. 18 showcase the manipulation video generation where the robot gripper is controlled by random external action inputs. However, as ORV does not utilize textual prompts and instead relies solely on visual cues to infer the states of robot arms and grippers, it is not yet capable of executing semantically meaningful tasks.

11.2.2. Multiview Videos Generation

Maintaining consistency across different views is crucial for multiview video generation. Although the model may possess the ability to infer view orientations from the observed frame (referred to as the context frame) and to predict how 3D motion control translates into 2D pixel variations across views, this capability is inherently limited. Therefore, we provide multiview conditioning signals consistently rendered from 3D geometric representations to enhance 2D pixel predictions, as described in Sec. 3.2.1 and Sec. 8.

Fig. 20 compares a 3-view video generation with and without the additional conditioning maps. In this example, although the 3D occupancy is constructed solely from the anchor view due to data limitations—resulting in lower quality compared to a complete 3D geometry—the conditioning maps rendered from the other two side views still improve the overall generation quality. As highlighted by the white regions, during the robot gripper's motion while holding a metal bowl, the bowl exhibits severe deformation in the current view, even though this issue is entirely absent in the anchor view. This discrepancy mainly arises from two factors: (1) the current view differs significantly from the anchor view, and (2) the object undergoes relatively large motion. With the additional guidance from 3D geometry, these issues can be effectively mitigated.

11.2.3. Additional Qualitative Results

We provide more **uncurated** singleview examples generated by ORV, as shown in Fig. 24, 25, 26. For each episode, we present their ground-truths in the top row and our results in the bottom row, respectively. For a better view and other more examples, please refer to our webpage.

11.3. Policy Learning (Section 4.3)

Fig. 22 shows four successful execution examples of fine-tuned SpatialVLA [74] with augmented data, on SimplerEnv-

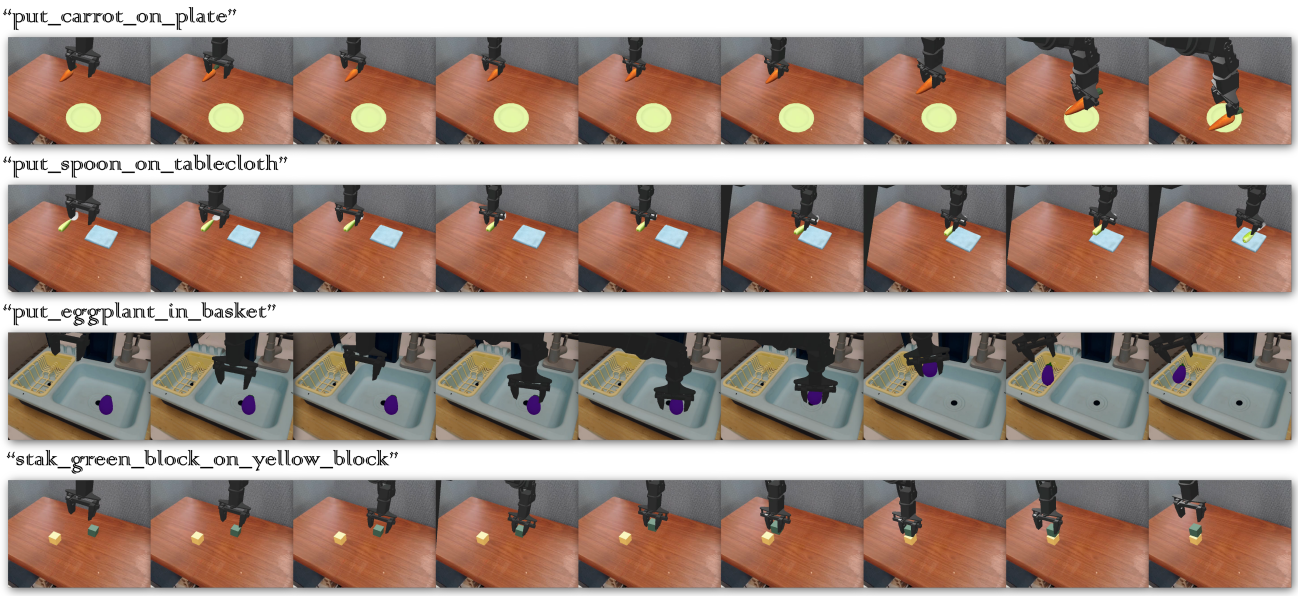


Figure 22. Successful examples of policy execution on four SimplerEnv-WidowX [53] tasks using our fine-tuned SpatialVLA [74] model.

WidowX [53] Benchmark.

12. Discussions

In this section, we provide more insightful explanations of ORV model or more in-depth discussion of extended related works, covering a broader range of aspects concerning generative models for robotics.

12.1. Occupancy-centric Framework

Action and Visual Priors. As described in Sec.3.1, most recent controllable video generation approaches for robot manipulation follow an action-to-video paradigm[27, 94, 128], where 3D action values are recorded either in simulation environments or from real-world robots. Some works, such as Im2Flow2Act [107], instead employ pixel-level 2D flows as intermediate motion signals, while others explore trajectory- or action-conditioned generation beyond robotics, *e.g.*, Tora [120] for human-drawn trajectory control. Although encoding 3D actions has shown promising results, the abstract nature of these values limits the model's ability to infer complex future object states—particularly for motions orthogonal to the image plane or involving rotations (see Fig.14). In contrast, visual cues such as 2D pixel flows provide more precise and stable motion guidance but remain insufficient to describe the entire scene. Furthermore, visual priors used in Cosmos-Transfer[3] and RoboTransfer [60] require pixel-perfect alignment with ground-truth depth or segmentation maps, which is often infeasible. To address these limitations, we combine *high-level, hard* action priors with *low-level, soft* visual priors rendered from occupancy fields. This hybrid design ensures that the generated videos follow action instructions while allowing flexible, coarser visual conditioning, thus mitigating the constraints of previous approaches.

Occupancy Representation. Occupancy fields offer multiple advantages beyond providing robust representations of noisy or parametric scenes, as discussed in Sec.3.1. Their coordinate-based formulation enables efficient online forecasting of robot manipulation scenes—directly predicting future states of the environment in the occupancy space. This paradigm has demonstrated remarkable success in autonomous driving[37, 50, 83, 99, 104], where occupancy representation has become a preferred choice over 3D points, bounding boxes, or meshes. Numerous recent works, including OccSora [93], Occllama [103], OccFormer [119], OccGen [91], and OccWorld [124], have achieved high-quality 3D occupancy generation and forecasting, highlighting a promising direction toward extending occupancy forecasting to robotic manipulation. Although robotics presents greater challenges due to more complex scene dynamics, achieving online 3D occupancy forecasting would further reduce the reliance on physical simulators that allow policy networks to generate the dynamics, facilitating the acquisition of occupancy priors for ORV.

Occupancy Data Curation. As introduced in Sec.3.3, we curate a 4D occupancy dataset for robotic manipulation by leveraging multiple foundation models within the data curation pipeline, including MonST3R[117], NKSR [34], VLMs [6], and SAM2 [75]. In our experiments, such scene reconstruction models demonstrate strong reliability on large-scale robotic datasets, effectively capturing fine-grained object and gripper motions. Moreover, with the incorporation of our *soft* visual priors, the reliance on precise dynamic

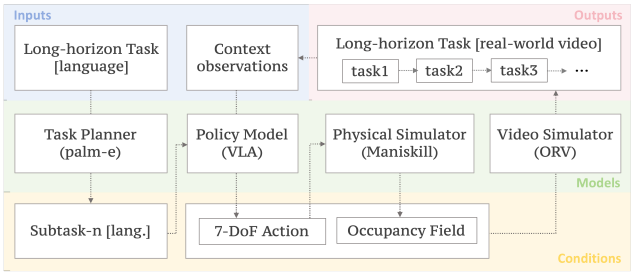


Figure 23. System of long-horizon manipulation data synthesis.

modeling during reconstruction is further reduced, making the overall data generation pipeline both robust and scalable.

Non-interactive Generative Model. Different from the recent work iVideoGPT [105], which highlights its interactive capability, our ORV mainly focuses on non-interactive generation. An interactive generation framework (typically, an auto-regressive model) exhibits causality during the forward pass, enabling arbitrary interactions with the external physical world. For instance, compared to iVideoGPT, VideoGPT [109] only accepts the entire future action sequence at the start of prediction, preventing an agent from interactively adjusting its actions based on predicted observations. However, our ORV model adopts the architecture of a non-causal diffusion model, where the action and occupancy priors are fully acquired before generation. Consequently, it does not require any interaction with the external world during generation.

12.2. World Model for Robot Manipulation

A world model is an internal abstraction that captures the physical, spatial, and causal dynamics of an environment. It encodes multimodal inputs (*e.g.*, images, text, actions, audio) into latent representations and predicts future states through internal reasoning and simulation. In robot manipulation, it models the interaction between sensory observations and actions.

World Model for High-fidelity Simulation. ORV serves as a generative world model that simulates diverse real-world environments. To ensure the simulated dynamics closely resemble real-world physical behaviors, ORV achieves superior performance compared to recent approaches such as IRASim [128] and HMA [94]. We primarily evaluate the effectiveness of the world model in simulation by comparing the visual quality of its predictions against ground-truth observations using standard metrics (*e.g.*, PSNR, FVD). Concurrent efforts [22, 27, 62, 95] have also been exploring high-fidelity and physically accurate video simulations.

World Model for Efficient Data Synthesis. Sec.11.3 and Sec.4.3 have introduced how ORV benefits policy learning through data synthesis. While some recent works [21, 27, 128] share similar ideas, they differ from ours. IRASim [128] deploys a pretrained policy model in a simulator to generate additional rollouts—both successful and failed ones—for training world models. RoboTransfer [60] trains a synthesis model with decoupled geometry and appearance conditions, where the conditions are derived from real-world data. Ctrl-World [27] generates synthetic post-training data by either rephrasing task instructions or resetting the robot arm to a new initial state for additional trajectories, which are then used for policy training. As we can see, most of these approaches require a data preparation stage to utilize the world model as a data generator. While we mainly demonstrate and validate the *visual* transfer capability of ORV in our experiments, we argue that ORV can also generate manipulation videos with diverse trajectories for each task. In such cases, we would similarly deploy a pretrained policy model (*e.g.*, RoboVLM [59], SpatialVLA [74], π_0 [12] etc.) in a physical simulator and perform simulation-to-real generation as discussed in Sec. 3.2.2.

World Model for Reproducible Policy Evaluation. Taking the world model as a simulator that accurately mimics the real world, some recent works [27, 128] explore developing reproducible policy evaluation within the world model itself. In this way, the world model can be used to evaluate upstream policy models, just as in the real world, while significantly reducing computational and physical resources. Although ORV could potentially support such functionality, we clarify that it is beyond the scope of this paper.

12.3. Limitations and Future Directions.

Despite the promising results achieved by ORV, the task remains inherently challenging with numerous open issues, and our approach has certain limitations. In this section, we elaborate on these limitations and suggest possible avenues for future research.

- *Integrating online 4D occupancy generation or forecasting.* Currently, ORV relies on complete 4D occupancy data as input, which to some extent limits its applicability in real-world scenarios. As discussed in Sec. 12.1, the coordinate-based formulation of occupancy representations, combined with the success of online occupancy generation frameworks in autonomous driving, suggests that incorporating such an online 4D occupancy generation or forecasting module into ORV is both feasible and promising. This integration would enable real-time perception and significantly enhance the practicality of our work.
- *Incorporating more comprehensive action representation of the robot arm.* Although our 3D occupancy provides a comprehensive geometric representation of all objects in the scene, the 3D action signal in our framework only encodes the 7-DoF end-effector pose of the robotic arm. Such a description is insufficient for manipulators with more complex articulated points, such as the Google robot used in the Droid [44] dataset—where rich joint-level dy-

namics are essential for accurately modeling the motion. Incorporating detailed motion descriptions for all joints would therefore yield a more faithful and fine-grained representation of the arm's trajectory. And recent work VAP [101] provides an alternative.

- *Adding multiview initial frames generations to ORV-MV.* Specifically, ORV-MV requires the first-frame observations from multiple camera views. By leveraging geometric constraints from the 3D occupancy and the robotic arm pose observed in these initial frames, ORV-MV can generate view-consistent videos. In future work, we plan to extend this framework to synthesize multi-view first-frame images directly from a single-view input—*i.e.*, enabling consistent multiview video generation from only one camera view. Such an enhancement would greatly improve the scalability and real-world usability of ORV-MV.
- *Towards long-horizon robot manipulation planning and generation.* Long-horizon manipulation data are substantially more valuable for policy training [29, 69, 78] yet much harder to collect than short-horizon video data. We believe that extending ORV into a long-horizon manipulation data planning and generation framework is feasible and promising (as illustrated in Fig. 23). Such an extension would enable the synthesis of temporally coherent long-horizon manipulation data, thereby facilitating more challenging policy learning and improving generalization across complex tasks.

12.4. Social Impact

This work advances controllable robot video generation with broad applications in robotics simulation, education, virtual reality, and creative media. Acknowledging its dual-use risks, such as potential misuse for misinformation or privacy violations, we conduct all research under a responsible AI framework using ethically sourced, public datasets for academic purposes only. We advocate incorporating safeguards like provenance tracking and synthetic content detection to ensure generative technologies benefit society while minimizing harm.

13. License

- All datasets used for video generation (BridgeData V2 [88], Droid [44], RT-1 [13]) are maintained under CC-BY-4.0 License;
- Robusuite [129]: MIT License;
- Robodesk [42]: Apache License 2.0;
- Qwen-VL-Chat [6]: released under Qwen-VL License Agreement[14];
- Qwen2.5-32B-Instruct [80]: Apache License 2.0;
- sentence-transformers/all-MiniLM-L6-v2 [97]: Apache License 2.0;
- CogVideoX-2B [111]: Apache License 2.0;
- MonST3R [117]: MIT License;
- VGGT [92]: released under VGGT License [15];
- NKSR [34]: Apache License 2.0;
- RAFT [81]: BSD 3-Clause License;
- Grounding DIMO [61]: Apache License 2.0;
- SegmentAnything2 [75]: Apache License 2.0;
- ManiSkill [26]: code and rigid-body environment components are released under Apache License 2.0; Assets are licensed under CC BY-NC 3.0;
- SIMPLER Benchmark [53]: MIT License;
- X-FLUX: all pretrained models are under FLUX.1 [dev] Non-Commercial License[16]; codes are under Apache License 2.0;

[14] https://github.com/QwenLM/Qwen-VL/blob/master/LICENSE

[15] https://github.com/facebookresearch/vggt/blob/main/LICENSE

[16] https://github.com/black-forest-labs/flux/blob/main/model_licenses/LICENSE-FLUX1-dev

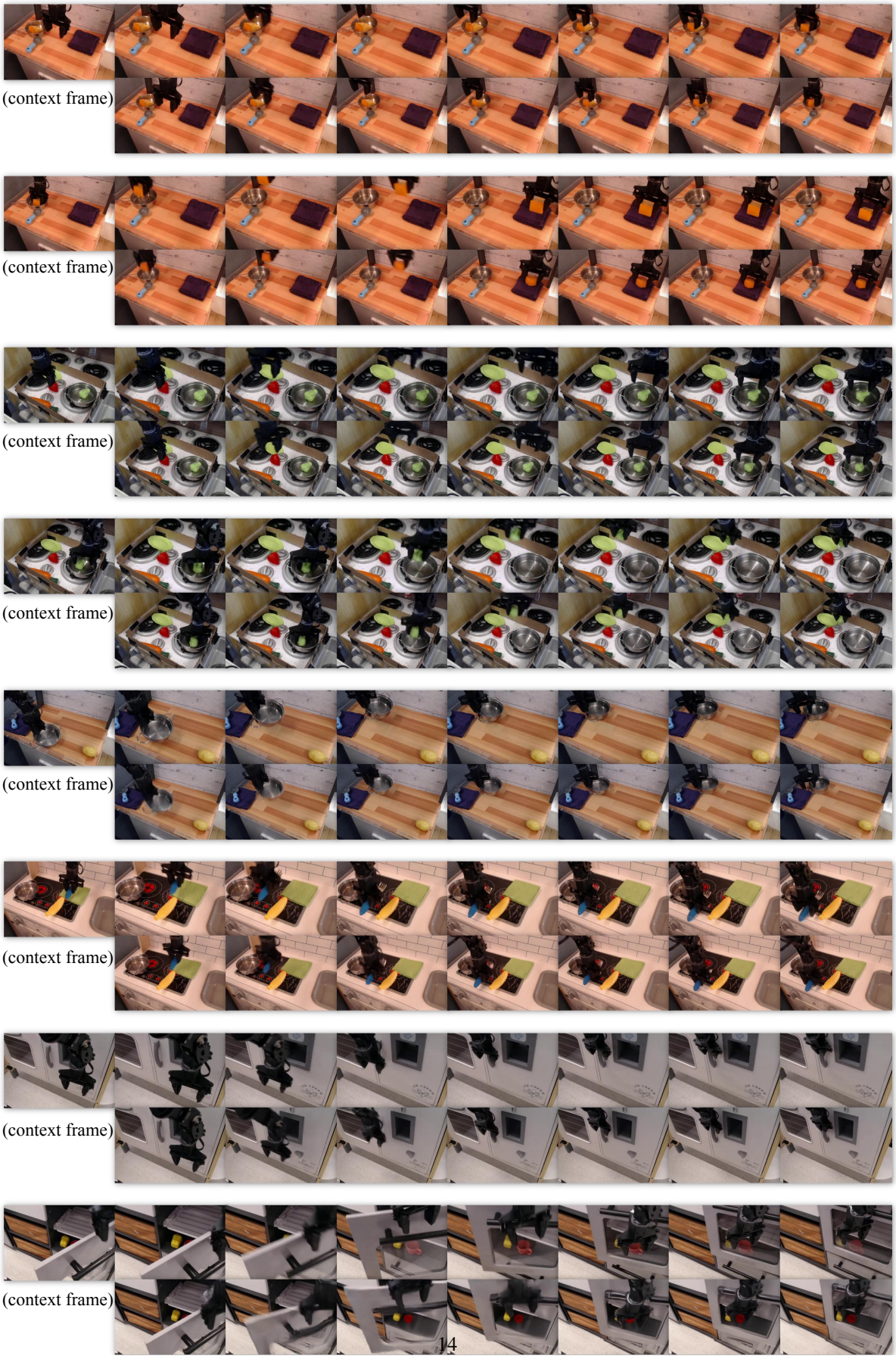


Figure 24. Additional Qualitative Results of ORV #1.

Figure 25. Additional Qualitative Results of ORV #2.

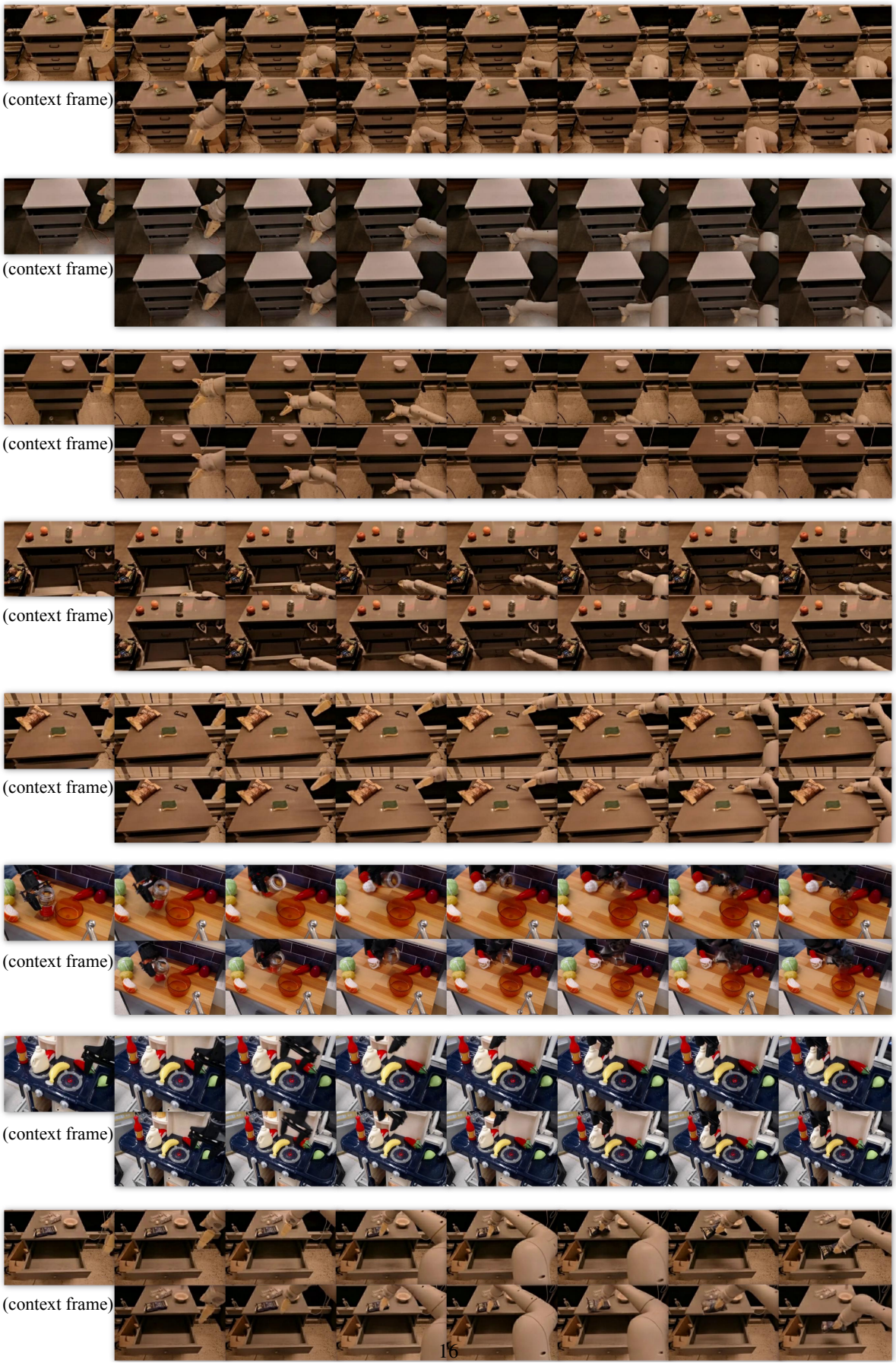


Figure 26. Additional Qualitative Results of ORV #3.