ORV: 4D Occupancy-centric Robot Video Generation

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] **Xin Jin**[4] **Hang Zhao**[2] **Hao Zhao**[1,9]

[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] National University of Singapore
[7] ByteDance [8] Megvii Technology [9] AIR, Tsinghua University

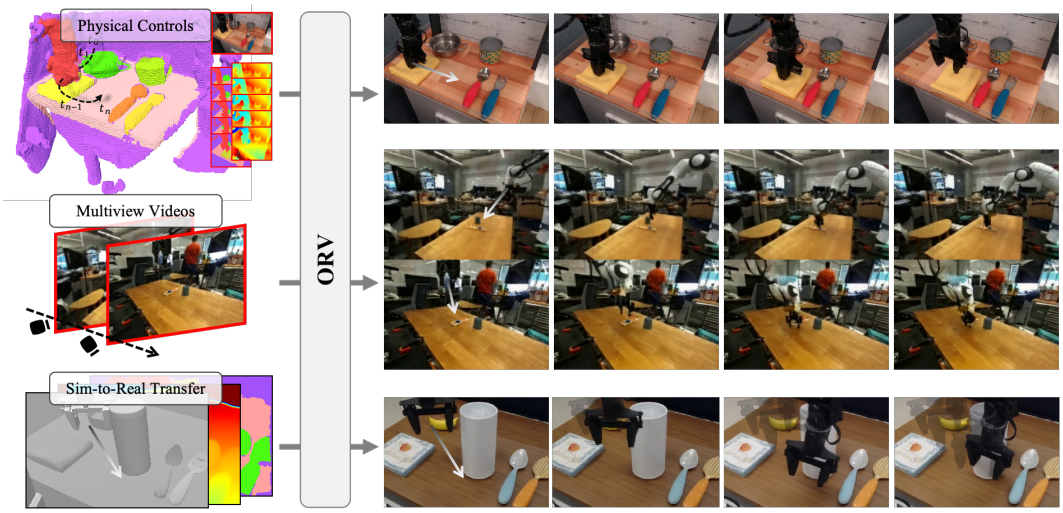

Figure 1: Our ORV generates action-conditioned robot manipulation videos under the guidance of the 4D occupancy (top) with higher control precision, performs multiview videos generation to build realistic 4D embodied world (middle) and conducts simulation-to-real videos transfer (bottom).

Abstract

Acquiring real-world robotic simulation data through teleoperation is notoriously time-consuming and labor-intensive. Recently, action-driven generative models have gained widespread adoption in robot learning and simulation, as they eliminate safety concerns and reduce maintenance efforts. However, the action sequences used in these methods often result in limited control precision and poor generalization due to their globally coarse alignment. To address these limitations, we propose **ORV**, an **O**ccupancy-centric **R**obot **V**ideo generation framework, which utilizes 4D semantic occupancy sequences as a fine-grained representation to provide more accurate semantic and geometric guidance for video generation. By leveraging occupancy-based representations, ORV enables seamless translation of simulation data into photorealistic robot videos, while ensuring high temporal consistency and precise controllability. Furthermore, our framework supports the simultaneous generation of multi-view videos of robot gripping operations—an important capability for downstream robotic learning tasks. Extensive experimental results demonstrate that ORV consistently outperforms existing baseline methods across various datasets and sub-tasks. Demo, Code & Model: `https://orangesodahub.github.io/ORV`

*Equal contribution

Preprint. Under review.

1 Introduction

Learning realistic simulators for robotic manipulation is essential for scaling robot learning [1–4], as they enable efficient data collection, safe policy development, and reproducible experiments without the cost and constraints of real-world interaction. Prior physics-based methods [5–9] have attempted to capture the full complexity of real-world scenes, exploring domain randomization, photorealistic rendering, and imitation learning. Yet these methods require extensive engineering efforts, struggle to scale across diverse tasks, and often provide limited visual fidelity—unrealistic textures, motions, and object dynamics that hinder policy generalization [10, 11].

Recent breakthroughs in generative models, *e.g.*, video generation models [12–14]—offer powerful foundation models for such simulators. They serve as strong visual priors and are capable of synthesizing high-fidelity manipulation videos and most importantly, are controllable through various external signals. Most recently, IRASim [15] and RoboMaster [16] generate realistic robot manipulation videos conditioned on 3D or 2D trajectories, while UniSim [17], EnerVerse [18] and TesserAct [19] employ multimodal commands or pure language inputs to condition future video prediction.

Despite their promising results, these methods often rely on high-level action sequences [15, 16] or task-level text prompts [19] as the controls, which suffer from limited alignment with the low-level visual content. This misalignment results in degraded motion accuracy and leads to lower video quality. Additionally, these global signals lack the spatial granularity required for fine-grained robot manipulation tasks, especially when precise physical interactions are needed.

To this end, we propose ORV, a 4D occupancy-centric framework for robot video generation that achieves high-fidelity video synthesis, with more precise controllability and strong generalizations (Fig. 1 top row), performs multiview robot video generation (Fig. 1 mid row) and conducts simulation-to-real dynamics transfer (Fig. 1 bottom row). The key idea is to leverage 4D semantic occupancy as intermediate representations and take the spatial-temporal-aligned guidance maps from renderings as the visual control signals. These 4D occupancy-derived visual signals preserve scene geometry and semantics, offering localized supervision that naturally guides the generation process. Moreover, recent advances in 3D semantic occupancy learning [20–22] have proven effective in representing structure in the field of self-driving and robotics.

Building on such an occupancy-centric pipeline, we further present ORV-MV, which simultaneously generates multiview-consistent robot videos following motion controls, since multiview observations can largely help with robot learning [23–25]; and ORV-S2R, which leverages occupancy as a bridge for sim-to-real adaptation, effectively narrowing the domain gap during inference. Some concurrent works [26, 27] also present some progress on transferring various high-level conditions to real-world RGB videos. Furthermore, to facilitate our training process, we also propose an efficient pipeline for curating occupancy data tailored to robot scenarios, leveraging the mainstream foundation models [28–31], as no high-quality public occupancy datasets are currently available. Together, these components form a scalable, fine-grained, and physically grounded system for advancing robot video generation.

The contributions of our framework can be summarized as follows:

- We propose an occupancy-centric pipeline for robot video generation, where 4D semantic occupancy sequences serve as efficient intermediate representations that enable high-quality and more precisely controllable generation.
- Facilitated with the occupancy representation, our framework seamlessly integrates physical simulators and generative models to enable realistic and scalable data synthesis.
- We curate a series of high-quality semantic occupancy datasets that accurately reflect 3D robot arm/gripper motions along with rich semantic and geometric information.
- Extensive experiments demonstrate the effectiveness of our method, particularly in terms of generation quality, motion precision, and transfer generalizability.

2 Related Work

2.1 Controllable Video Generation

Recent advances have greatly improved the realism of controllable video generation, particularly for applications in autonomous driving and embodied intelligence[32–41, 20, 42–53]. Early

methods like MAGE [32] aligned appearance and motion modalities using a Motion Anchor-based generator, while ControlVideo [33] introduced a training-free approach with cross-frame attention for text-to-video synthesis. Works such as DriveDreamer[54], MagicDrive [34], and Panacea[55] focus on temporal video generation, and frameworks like Drive-WM [40] and Vista[36] incorporate world models to enhance realism. UniScene [20] enables multi-modal scene generation via unified representations and hierarchical learning. In robotics, methods like Gen2Act[56] leverage video generation models to infer motion for robot policies, while This&That [57] ensures intent-aligned synthesis through language and gesture control. VidEgoThink[58] evaluates embodied AI systems using egocentric video understanding.

2.2 3D Occupancy Representation

Semantic occupancy is a key 3D scene representation for perception and generation tasks[59–62, 37, 63–65]. Methods like MonoScene [60] and FB-Occ[66] focus on monocular and Bird's Eye View (BEV) learning, while TPVFormer [62] uses a tri-perspective framework. SurroundOcc[67] improves estimation with multi-view inputs, and VPD [64] applies diffusion models for prediction. OccWorld[63] forecasts future states, and OccLlama [68] integrates Large Language Models (LLMs). Despite advances, frameworks like OccSora[65] still fall short of ground truth quality in temporal 3D generation. Occupancy anticipation methods infer unseen regions to enhance spatial awareness [69]. Generative approaches such as TRELLIS[70] support flexible 3D outputs, while object-centric methods refine predictions using 3D semantic Gaussians [71, 72, 21, 73]. GaussianFormer[71] refines Gaussians via deformable attention, and EmbodiedOcc [72] updates global representations online. This work introduces an occupancy-centric framework for robot video generation, leveraging 3D occupancy to bridge the sim-to-real gap and guide high-quality synthesis.

2.3 World Models for Embodied Intelligence

Recent advancements in simulating dynamic environments have fueled interest in world models for robotics and embodied intelligence [15, 17, 74, 18, 75, 56, 76–87, 26]. IRASim [15] generates realistic robot action videos from trajectories, enabling scalable learning. UniSim [17] integrates diverse datasets for high-fidelity training, while ReCamMaster [74] enhances scene synthesis using pre-trained models. TesserAct [19] produces temporally coherent 4D reconstructions, and EnerVerse [18] forecasts future spaces with a self-reinforcing pipeline. WorldSimBench [75] benchmarks perceptual fidelity and task consistency. Human-centric methods like Gen2Act [56] generalize policies to unseen tasks, and EVA [76] combines visual generation with language reasoning. However, most methods rely on coarse-grained guidance (*e.g.*, action sequences). In contrast, we propose fine-grained 3D occupancy representations to improve quality and precision.

3 ORV: Methodology

In this section, we focus on how we address the mentioned issues in robot video generations, including the control precision, generation quality and reducing the simulation-real gap. We first demonstrate the semantic occupancy curation pipeline 3.1, then introduce the use of 4D semantic occupancy priors as the intermediate representation, which efficiently facilitates high-quality and precise controllability in robot video generation 3.2. After that, our multiview videos generation comes to build 4D sequences of robot manipulations 3.3. Finally, we discuss the efforts to bridge simulation dynamics and real-world videos 3.4.

3.1 Semantic Occupancy Data Curation

Since there exists no publicly available high-quality 4D semantic occupancy data, we have designed an efficient data curation process (as Figure 3) to build pseudo occupancy ground-truth data upon existing popular robot manipulation video datasets (BridgeV2 [88], Droid [90]), RT-1 [89]). Some samples of curated data are shown in Figure 2.

Semantics Labeling. Semantics information plays a fundamental role in scene understanding, recognition, and generation tasks [20]. In robot manipulation scenarios, precise object recognition is crucial for executing text-instruction-driven operations. While action-conditioned tasks relax this requirement to some degree, however, physical-world semantic understanding remains essential.

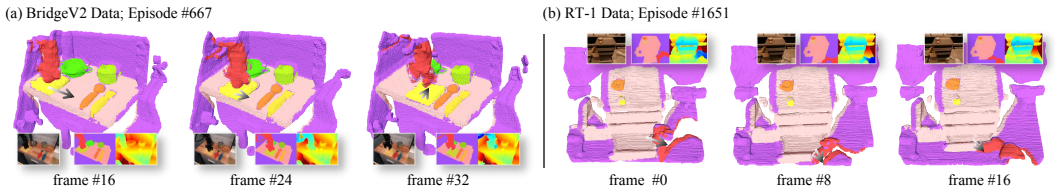

Figure 2: 3D Semantics Occupancy Samples of Dataset BridgeV2 [88] and RT-1 [89]. Better to zoom in. Refer to Supplementary Materials for more examples.

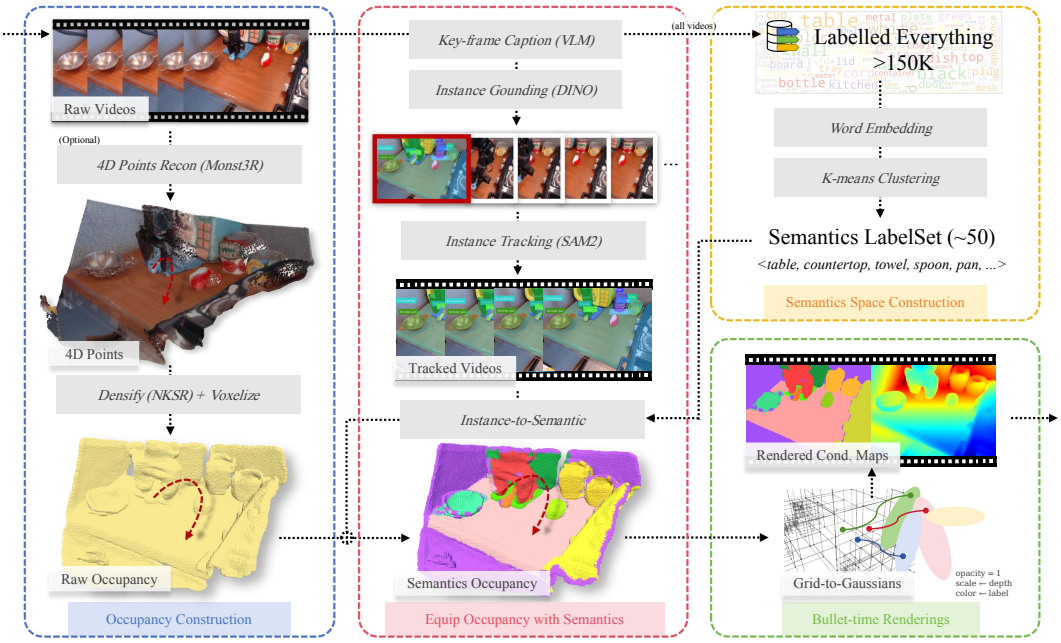

Figure 3: Overview of **Dataset Curation Pipeline,** which consists of four parts: semantics space construction, occupancy construction, equip occupancy with semantics, and bullet-time occupancy-to-gaussian renderings in practical usage.

When predicting subsequent frames, the model still needs object categories to accurately infer next-state dynamics — particularly for distinguishing between rigid bodies, articulated objects, and deformable materials, each exhibiting distinct physical behaviors.

As illustrated in Figure 3, we split this labeling process into two steps: (1) One-time semantics space construction upon the entire dataset; (2) Per-video instances association and semantics mapping. Starting from the raw videos in the dataset, we employ Vision-Language Model (VLM), such as Qwen-VL-Chat [31], to conduct key-frame captioning (in our case, we force the use of the first frame) on each video holistically and get the key objects through designed prompts. These captioned objects contribute to both steps above. For the overall semantics initialization, we perform efficient K-means clustering on the entire word embeddings of nearly 150K captioned objects. And get a comprehensive label-set (of size $\sim$50) as the dataset-level semantic labels, with the trade-off between expressiveness and cost. For each single video to be labeled, we utilize Grounding DINO [29] to extract initial object prompts (*e.g.*, bounding-box, segment mask) which are then input to SAM2 [30] to track the instances starting from the first frame. Having temporally consistent instance masks throughout the video, we then efficiently map these instances to semantic labels, using the instance-semantic correspondence and label-set from the first step.

4D Occupancy Generation. This process consists of two subsequent steps: (1) Occupancy construction; and (2) Equipping occupancy with semantics. We begin with reconstructing sparse 4D points using Monst3R [28], which is well-suited for robustly estimating 3D structure and camera motion from dynamic monocular videos. To overcome the inherent sparsity of the points from Monst3R, we adopt mesh reconstruction for denser points. In our framework, we choose NKSR [91], which can more effectively fill large holes and is robust to noise. After that, we perform volatilization

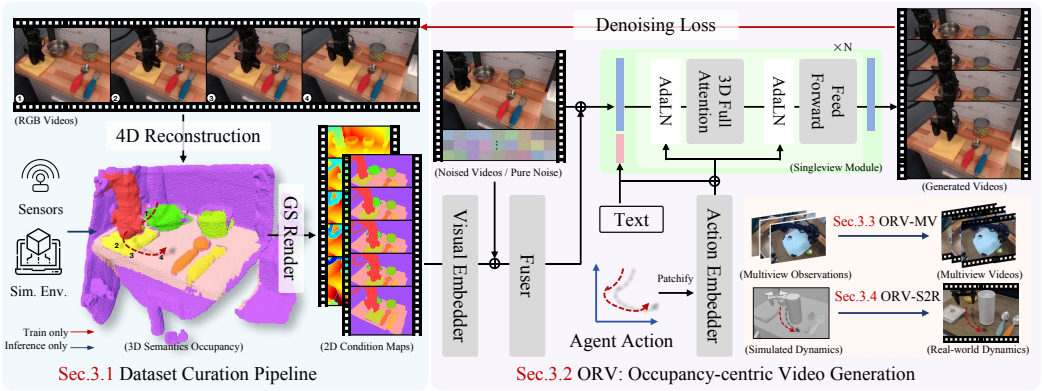

Figure 4: Overview of **ORV**. For training purposes, we start from Dataset Curation (Sec. 3.1) to produce high-quality semantic occupancy data. Leveraging pixel-level aligned condition maps from such 3D representation, we generate robot videos that precisely follow the motion instructions (Sec. 3.2). Furthermore, we introduce ORV-MV (Sec. 3.3) and ORV-S2R (Sec. 3.4), which simultaneously produce multi-view robot videos and effectively convert the simulation data to real-world videos.

on densified points to obtain the 4D occupancy in canonical space. To acquire an occupancy-wise semantics label, we project the points onto the image plane to extract the semantic labels from pixels, followed by majority voting for each voxel.

To render any 2D maps from 4D semantic occupancy, we directly associate each grid with a single non-learnable Gaussian to save memory and time cost. In this way, it yields a compact yet informative 2D representation that captures the real-world dynamics. Moreover, to enhance the rendering quality, we employ the adaptive scaling mechanism on the Gaussian primitive based on depth. Specifically, the size σ follows $\sigma = k \cdot (\hat{d})^{\alpha}$, where $\hat{d} \in (0, 1]$ denotes the normalized depth values in canonical space, and k, α control the scaling behavior of gaussians in the near and far plane.

3.2 Occupancy-centric Video Generation Model

We choose the pretrained CogVideox-2b [14] (text-to-video) as our foundation model, following the increasing trend of leveraging advances in scalable video generation for specialized subtasks [19, 92].

Action Conditioning. Following the most straightforward approach to controllable video generation in robotics manipulation and recent [15, 92, 93], we first directly take the 3D trajectory sequence (end-effector poses) or actions along with gripper states as a high-level control signals, *e.g.* $\mathcal{A} \in R^{T \times D_{action}}$, where D_{action} denotes the action dimension. Drawing inspiration from [15, 94], we inject these 3D action controls to AdaLN to directly modulate the video latents within each DiT block. More efficiently, we take a chunk-level integration scheme for better alignment between high-dimensional actions and videos in these extensive modulations. Specifically, we apply frame compression which strictly aligns with the videos operated by 3D VAE of CogVideoX, to produce $\mathcal{A}' \in R^{\frac{T}{r} \times r \cdot D_{action}}$, where r denotes the temporal compression rate. Then, an additional shallow MLP (as Action Embedder in Figure 4) is used to get action features $\phi(\mathcal{A}') \in R^{\frac{T}{r} \times D}$. It ensures latent-frame-level alignment between the actions and videos in the latent space. Notably, the action-, text-, and denoising step-AdaLN all share the same parameters, eliminating the potential explosion in model size (as the AdaLN accounts for over $1/3$ of the parameters of CogVideoX).

Visual Conditioning. While action conditioning provides direct commands for robotic motion, translating these high-dimensional control signals into consistent and physically plausible pixel-level transformations presents notable challenges. This is largely due to the complex and diverse object dynamics present in robot operation videos, including changes in viewpoint flickering, object deformation, and articulated movements often not fully captured by the action commands. These complexities make it difficult to reliably infer the underlying 3D spatial actions from 2D observations and accurately model the relationship between pixel changes and 3D physical motions, impeding precise generation and leading to inconsistencies and a lack of realism. Thus, we introduce additional visual conditioning that stems from 3D semantic occupancy.

Since our 2D visual control signals keep the same spatial resolution as the input observation frames, pixel-level alignment can be readily achieved. Combined with the frame-level alignment of action controls, it significantly improves the control accuracy. Specifically, as depicted in Figure 4, we employ an additional shallow MLP (as Visual Embedder) to learn the visual control features. And then augment it with the image conditions, after which another zero-initialized projector adds the visual control signals to the input noise.

Though ControlNet-like [95] offers stronger and more refined pixel-level control, it suffers from a heavy model size explosion. Moreover, our method prioritizes the control of the 3D actions, following the baseline methods, while introducing a *soft* visual control signal (from a hard render-based procedure) as a simple yet effective auxiliary control. Moreover, directly taking SFT (Supervised Fine-Tuning) will not undermine the generalization ability of our model, which is detailed in the Supplementary Materials, leading to comparable performance with ControlNet.

3.3 ORV-MV: Multiview Robot Videos Generation

A complete, high-fidelity 4D scene would provide significant benefits for robotic policy learning and other related tasks. Several concurrent works [19] have demonstrated the capability to generate high-quality 4D scenes. However, it only captures a single surface of the scenes, resulting in noticeable artifacts and empty regions when the viewpoint changes. While ORV can further showcase the ability to generate and construct diverse, comprehensive 4D RGB scenes with realistic visual fidelity.

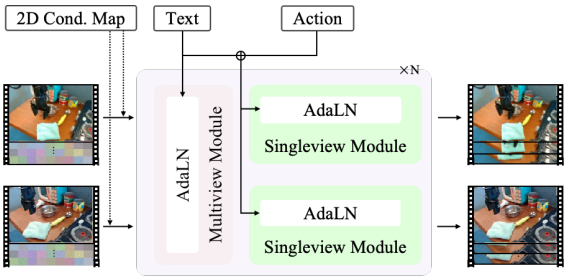

Figure 5: Overview of ORV-MV, which generates multi-view robot manipulation videos following various controls.

We extend our controllable single-view video generation model (Sec. 3.2) to ORV-MV, as depicted in Figure 5. Inspired by recent successes in multi-view content synthesis [96, 97], we integrate an additional view attention module into each DiT blocks, which deal with the input latents $\mathcal{F}_{\mathcal{V}} \in R^{B \times S_V \times D}$ (S_V denotes the tokens of the same patch-level across all views) to enable cross-view interaction. And the original frame attention (as the 'Singleview Module' presented in both Figure 5 and Figure 4) layers that process patch-level latents $\mathcal{F}_{\mathcal{P}} \in R^{B \times S_P \times D}$ (S_P denotes the view-independent patch tokens) will be frozen during this stage of training. We use the multi-view videos from the datasets as the supervision. Note that only the frame attentions take the 3D temporal controls (*e.g.* action sequences) as the inputs, while multiview images also fuse with 2D condition maps. In this way, the model infers the view poses according to multi-view observations (robot arms, or grippers), then jointly predicts multi-view pixel changes consistent with 3D controls. Please refer to the supplementary for more architectural details.

3.4 ORV-S2R: Bridge Sim-to-Real via Occupancy

Another extension of our work, ORV-S2R, will further take a small step towards addressing the significant *visual realism* gap between simulation data and real-world observations. While prior efforts [98, 99] have attempted to minimize this discrepancy, our approach offers a direct solution—we propose that combining physical simulators with expressive neural models presents a more viable solution. From the reusable geometry assets (*e.g.* meshes) in simulators, which can be readily converted to our 3D occupancy representations and then rendered to 2D condition maps, we can synthesize diverse photorealistic manipulation videos while preserving physical plausibility, leveraging our ORV model. It eliminates the need for laborious and performance-limited texture authoring of geometries. Notably, it also hinges on the generalizations of ORV-S2R: supporting *arbitrary visual observations* and *action inputs*, while producing high-quality videos precisely reflecting all the control signals.

The use of our occupancy also helps bridge the sim-to-real by mitigating the differences in conditioning data quality between simulated environments and the real world. For example, compared to depth signals from simulators or real-world sensors, our occupancy provides a more adaptable representation—this coarser yet strictly geometry-aligned format enables the efficient transformation

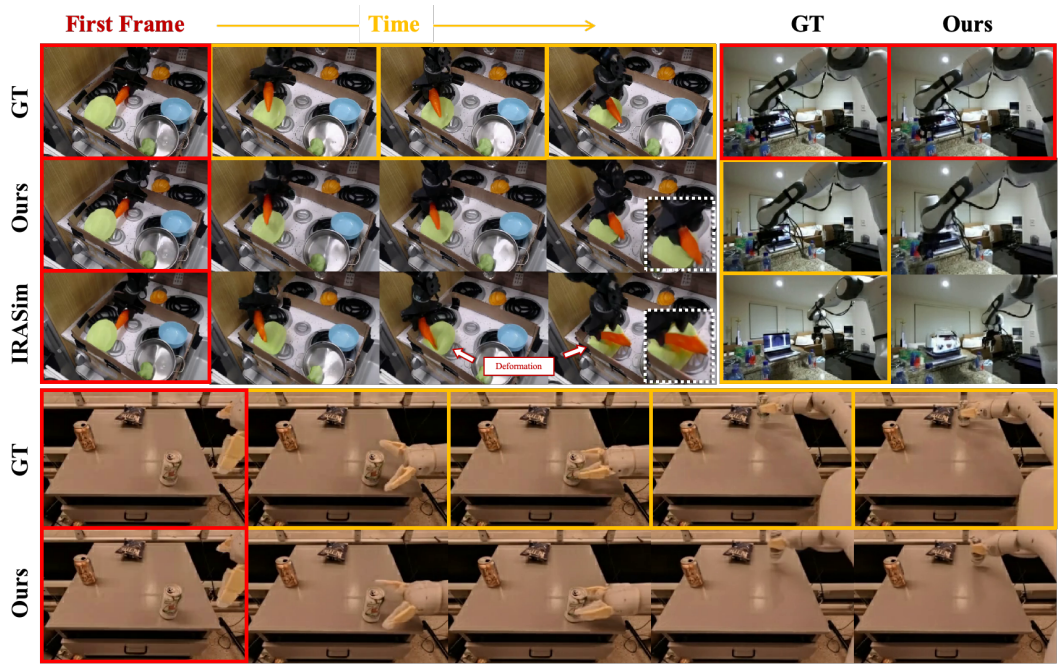

Figure 6: Qualitative Results of **Controllable Video Generation** with full conditions. Given one-frame observation, ORV predict subsequent 15 frames on validation split of Bridge [88], Droid [90], RT-1 [89] datasets. Red boxes denotes the first frame input of the video generation; Orange boxes denotes the ground-truth of the subsequence frames.

Figure 7: Qualitative Results of **Sim-to-Real Transfer.** Given raw dynamic data (*e.g.*, a tabletop manipulation scene, which consists of various mesh components) in the simulation environment, we can transfer them into real-world data, which possesses better visual quality and leads to higher efficiency than that in original physical simulators.

from both the sensors depth and the simulated depth to occupancy data. Which is particularly valuable given the significant gap that exists between these two—for instance, simulator depth suffers from an unstable physical engine, whereas sensor-derived depth contains varying degrees of noise. Therefore, though our model is trained on real-world data, it can be effectively applied to simulated dynamics and complete the sim-to-real transfer.

4 Experiments

In this section, we focus on demonstrating the generation quality of ORV, and compare our performance with publicly available methods quantitatively and qualitatively.

Datasets. We train and validate ORV on three real-world datasets: BridgeV2 [88], Droid [90] and RT-1 [89], with their basic statistics summarized in Table 1. We sampled video sequences at specified frame rates to construct approximately 120k training samples for each dataset, while randomly selecting around 2.6k samples

Table 1: Overview of Datasets used in ORV.

Dataset	Embodiment	Views	Episodes
BridgeV2 [88]	WidoxX	1∼3	60k
Droid [90]	Franka Panda	2	76k
RT-1 [89]	Google Robot	1	120k

Figure 8: **Multiview Videos Generation Results.** ORV-MV supports generating multiview videos with high cross-view consistency from initial frames. We illustrate here ORV-MV generates both three-view video and two-view video.

for evaluation. All datasets employ 7-DoF action representations, with respective video resolutions in our work of $16 \times 320 \times 480$ for BridgeV2 and RT-1, $24 \times 256 \times 384$ for Droid.

Models. For the development of our ORV, we start from pretrained CogVideoX2b (Text-to-Video) [14] as our base model. To support the image conditioning, we extend the input channels of the original CogVideoX2b model, retain the parameters that deal with text input unchanged, while letting image input channels be zero-initialized. For the action-conditioned base model setup, we train models on $8 \times$ H100 cluster for 30K steps. For depth-semantics guided finetuning and multi-view video extension, we have an additional 20K steps of training. In our training, we use a total batch size of 64, a learning rate of 1e-4, and AdamW Optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.95$.

4.1 Controllable Video Generation

Results. Table 2 presents the quantitative results of our controllable video generation across, demonstrating consistent outperformance over most of the baseline methods across various datasets. Figure 6 shows the qualitative comparison results. According to the highlighted (white) area of the first example, the baseline fails to faithfully infer the dynamics of objects manipulated by the robotic gripper—which presents a significant challenge in this task, as no descriptive conditions were provided for these objects, requiring reasoning based solely on the gripper's motion and understanding to physical world. In our results, the carrot's dynamics exhibits substantially smaller errors.

We provide the details about baselines in the Supplementary Materials. We also validate the generalization ability of ORV through in- and out-of-domain tests there.

4.2 Multi-view Videos Generation

Figure 8 depicts the multi-view video generation results of our framework. The first example demonstrates the robot arm performing a cloth-folding task across three distinct viewpoints, 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 due to lighting variations, there is a color discrepancy in the input data itself, so the lighting from the three views is not perfectly consistent.

Table 2: Evaluation results of video generation on three datasets. ('-' denotes model not available)

Method	BridgeV2 [88]				Droid [90]				RT-1 [89]			
	PSNR↑	SSIM↑	FID↓	FVD↓	PSNR↑	SSIM↑	FID↓	FVD↓	PSNR↑	SSIM↑	FID↓	FVD↓
CogVideoX [14]	19.432	0.752	7.509	83.561	19.238	0.701	6.341	71.536	20.457	0.816	6.243	42.169
AVID [100]	-	-	-	-	-	-	-	-	25.600	0.852	**2.965**	24.200
HMA [93]	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 [15]	25.276	0.833	10.510	20.910	-	-	-	-	26.048	0.833	5.600	25.580
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**

4.3 Sim-to-Real Transfer

As introduced in Sec. 3.4, our model effectively addresses the data quality challenges in sim-to-real transfer. Figure 7 demonstrates one of our attempts. From the simulated dynamics in the simulation environment, we first get its corresponding colored initial frame, and then extend it to video which is guided by the control signals from the simulation data (*e.g.*, 3D action sequence and rendered visual conditions from 3D occupancy). To obtain the initial observation frame from the untextured geometry environment in the physics simulator for video generation, we employ an additional ControlNet model alongside multiple visual conditions rendered from occupancy. By subsequently combining these visual condition sequences and action controls, we get realistic manipulation data that faithfully adheres to physical constraints.

4.4 Ablation Study

We conducted ablation studies to validate the effectiveness of our proposed occupancy-centric visual guidance. Specifically, we trained separate single-view video generation models under different configurations. We report the ablation results on conditioning types, source, and training strategy. We test all on the Bridge [88] dataset. We provide more details in the Supplementary.

Table 3: Ablation Results on Conditioning Types.

Variants	Source	PSNR↑	SSIM↑	FID↓	FVD↓
base	-	25.631	0.873	3.821	17.682
w/ depth	Recon.	30.288	0.919	3.061	14.321
	Render	28.031	0.896	4.522	18.548
w/ sem.	Recon.	28.896	0.901	3.259	16.171
	Render	27.911	0.896	3.467	17.053
full cond.	Recon.	30.431	0.920	2.998	14.301
	Render	28.258	0.899	3.418	16.525

Effect of the control signals. Table 3 reveals the effect of the conditioning types. The results show that incorporation of physical constraints leads to immediate and significant improvements in video generation quality and motion accuracy, with the PSNR increasing substantially from ~ 25 (base model) to ~ 28. Furthermore, we observe that the rendering-based conditions perform comparably to those from reconstruction (serving as pseudo ground truth), which effectively relaxes the stringent quality requirements for physical constraints in practical application (*e.g.* Simulation to real transfer).

Effect of the pretraining. We further test the benefits of the pretraining model. As shown in Table 7, based models trained from the pretrained CogVideoX have superior performance compared to from scratch, particularly on FID and FVD metrics.

Table 4: Ablations on Training Strategies.

Variants	PSNR↑	SSIM↑	FID↓	FVD↓
from scratch	23.518	0.811	19.357	84.831
from CogVideoX (T2V)	25.631	0.873	3.821	17.682

5 Conclusions

We introduce ORV, an Occupancy-centric Robot Video generation framework, which utilizes 4D semantic occupancy as additional control signals for more controllable robot video generation. With our extended ORV-MV and ORV-S2R, multiview video generations are enabled and will produce a potential high-quality 4D world, which effectively helps with robot learning. Furthermore, the simulation-to-real gaps can be reduced with the occupancy representation. Extensive experiments validated our framework. Overall, we provide a powerful and efficient foundation model that supports various control signals and expect it can enable advancements in other areas of embodied intelligence.

References

[1] 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.

[2] 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.

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

[4] 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.

[5] Yecheng Jason Ma, William Liang, Hung-Ju Wang, Sam Wang, Yuke Zhu, Linxi Fan, Osbert Bastani, and Dinesh Jayaraman. Dreureka: Language model guided sim-to-real transfer. *arXiv preprint arXiv:2406.01967*, 2024.

[6] Yifeng Jiang, Tingnan Zhang, Daniel Ho, Yunfei Bai, C Karen Liu, Sergey Levine, and Jie Tan. Simgan: Hybrid simulator identification for domain adaptation via adversarial reinforcement learning. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2884–2890. IEEE, 2021.

[7] Fabio Muratore, Fabio Ramos, Greg Turk, Wenhao Yu, Michael Gienger, and Jan Peters. Robot learning from randomized simulations: A review. *Frontiers in Robotics and AI*, 9:799893, 2022.

[8] Sergey Zakharov, Rareș Ambruș, Vitor Guizilini, Wadim Kehl, and Adrien Gaidon. Photo-realistic neural domain randomization. In *European Conference on Computer Vision*, pages 310–327. Springer, 2022.

[9] Ricardo Garcia, Robin Strudel, Shizhe Chen, Etienne Arlaud, Ivan Laptev, and Cordelia Schmid. Robust visual sim-to-real transfer for robotic manipulation. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 992–999. ieee, 2023.

[10] Annie Xie, Lisa Lee, Ted Xiao, and Chelsea Finn. Decomposing the generalization gap in imitation learning for visual robotic manipulation. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3153–3160. IEEE, 2024.

[11] 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.

[12] 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. URL `https://arxiv.org/abs/2412.03603`.

[13] 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. URL https://arxiv.org/abs/2503.20314.

[14] 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.

[15] 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.

[16] 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. URL https://arxiv.org/abs/2506.01943.

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

[18] 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.

[19] 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.

[20] 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.

[21] Hao Wang, Xiaobao Wei, Xiaoan Zhang, Jianing Li, Chengyu Bai, Ying Li, Ming Lu, Wenzhao Zheng, and Shanghang Zhang. Embodiedocc++: Boosting embodied 3d occupancy prediction with plane regularization and uncertainty sampler. *arXiv preprint arXiv:2504.09540*, 2025.

[22] Zhang Zhang, Qiang Zhang, Wei Cui, Shuai Shi, Yijie Guo, Gang Han, Wen Zhao, Jingkai Sun, Jiahang Cao, Jiaxu Wang, Hao Cheng, Xiaozhu Ju, Zhengping Che, Renjing Xu, and Jian Tang. Occupancy world model for robots, 2025. URL https://arxiv.org/abs/2505.05512.

[23] 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.

[24] 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.

[25] 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.

[26] 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.

[27] 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.

[28] 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.

[29] 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.

[30] 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.

[31] 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.

[32] Yaosi Hu, Chong Luo, and Zhenzhong Chen. Make it move: controllable image-to-video generation with text descriptions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18219–18228, 2022.

[33] Yabo Zhang, Yuxiang Wei, Dongsheng Jiang, Xiaopeng Zhang, Wangmeng Zuo, and Qi Tian. Controlvideo: Training-free controllable text-to-video generation. *arXiv preprint arXiv:2305.13077*, 2023.

[34] 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.

[35] 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*.

[36] Shenyuan Gao, Jiazhi Yang, Li Chen, Kashyap Chitta, Yihang Qiu, Andreas Geiger, Jun Zhang, and Hongyang Li. Vista: A generalizable driving world model with high fidelity and versatile controllability. *Advances in Neural Information Processing Systems*, 37:91560–91596, 2025.

[37] 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.

[38] Ruiyuan Gao, Kai Chen, Bo Xiao, Lanqing Hong, Zhenguo Li, and Qiang Xu. Magicdrivedit: High-resolution long video generation for autonomous driving with adaptive control. *arXiv preprint arXiv:2411.13807*, 2024.

[39] 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.

[40] 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.

[41] 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.

[42] Lening Wang, Wenzhao Zheng, Dalong Du, Yunpeng Zhang, Yilong Ren, Han Jiang, Zhiyong Cui, Haiyang Yu, Jie Zhou, Jiwen Lu, et al. Stag-1: Towards realistic 4d driving simulation with video generation model. *arXiv preprint arXiv:2412.05280*, 2024.

[43] Kiwhan Song, Boyuan Chen, Max Simchowitz, Yilun Du, Russ Tedrake, and Vincent Sitzmann. History-guided video diffusion. *arXiv preprint arXiv:2502.06764*, 2025.

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

[45] 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.

[46] Ruiyuan Gao, Kai Chen, Zhihao Li, Lanqing Hong, Zhenguo Li, and Qiang Xu. Magicdrive3d: Controllable 3d generation for any-view rendering in street scenes. *arXiv preprint arXiv:2405.14475*, 2024.

[47] Long Chen, Oleg Sinavski, Jan Hünermann, Alice Karnsund, Andrew James Willmott, Danny Birch, Daniel Maund, and Jamie Shotton. Driving with llms: Fusing object-level vector modality for explainable autonomous driving. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 14093–14100. IEEE, 2024.

[48] Yunzhi Yan, Zhen Xu, Haotong Lin, Haian Jin, Haoyu Guo, Yida Wang, Kun Zhan, Xianpeng Lang, Hujun Bao, Xiaowei Zhou, et al. Streetcrafter: Street view synthesis with controllable video diffusion models. *arXiv preprint arXiv:2412.13188*, 2024.

[49] 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.

[50] 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.

[51] 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.

[52] 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.

[53] 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.

[54] Xiaofeng Wang, Zheng Zhu, Guan Huang, Xinze Chen, and Jiwen Lu. Drivedreamer: Towards real-world-driven world models for autonomous driving. *ECCV*, 2024.

[55] Yuqing Wen, Yucheng Zhao, Yingfei Liu, Fan Jia, Yanhui Wang, Chong Luo, Chi Zhang, Tiancai Wang, Xiaoyan Sun, and Xiangyu Zhang. Panacea: Panoramic and controllable video generation for autonomous driving, 2023.

[56] 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.

[57] Boyang Wang, Nikhil Sridhar, Chao Feng, Mark Van der Merwe, Adam Fishman, Nima Fazeli, and Jeong Joon Park. This&that: Language-gesture controlled video generation for robot planning. *arXiv preprint arXiv:2407.05530*, 2024.

[58] Sijie Cheng, Kechen Fang, Yangyang Yu, Sicheng Zhou, Bohao Li, Ye Tian, Tingguang Li, Lei Han, and Yang Liu. Videgothink: Assessing egocentric video understanding capabilities for embodied ai. *arXiv preprint arXiv:2410.11623*, 2024.

[59] Antonin Vobecky, Oriane Siméoni, David Hurych, Spyridon Gidaris, Andrei Bursuc, Patrick Pérez, and Josef Sivic. Pop-3d: Open-vocabulary 3d occupancy prediction from images. *Advances in Neural Information Processing Systems*, 36:50545–50557, 2023.

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

[61] 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. In *IJCAI*, 2024.

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

[63] 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.

[64] Bohan Li, Yasheng Sun, Jingxin Dong, Zheng Zhu, Jinming Liu, Xin Jin, and Wenjun Zeng. One at a time: Progressive multi-step volumetric probability learning for reliable 3d scene perception. In *AAAI*, 2024.

[65] 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.

[66] Zhiqi Li, Zhiding Yu, David Austin, Mingsheng Fang, Shiyi Lan, Jan Kautz, and Jose M Alvarez. Fb-occ: 3d occupancy prediction based on forward-backward view transformation. *arXiv preprint arXiv:2307.01492*, 2023.

[67] 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.

[68] 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.

[69] Santhosh K Ramakrishnan, Ziad Al-Halah, and Kristen Grauman. Occupancy anticipation for efficient exploration and navigation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part V 16*, pages 400–418. Springer, 2020.

[70] Jianfeng Xiang, Zelong Lv, Sicheng Xu, Yu Deng, Ruicheng Wang, Bowen Zhang, Dong Chen, Xin Tong, and Jiaolong Yang. Structured 3d latents for scalable and versatile 3d generation. *arXiv preprint arXiv:2412.01506*, 2024.

[71] 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.

[72] 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.

[73] 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.

[74] 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.

[75] 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.

[76] 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.

[77] 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.

[78] 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.

[79] Zihui Sherry Xue, Romy Luo, Changan Chen, and Kristen Grauman. Hoi-swap: Swapping objects in videos with hand-object interaction awareness. *Advances in Neural Information Processing Systems*, 37:77132–77164, 2024.

[80] 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.

[81] 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.

[82] 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.

[83] Anthony Hu, Lloyd Russell, Hudson Yeo, Zak Murez, George Fedoseev, Alex Kendall, Jamie Shotton, and Gianluca Corrado. Gaia-1: A generative world model for autonomous driving. *arXiv preprint arXiv:2309.17080*, 2023.

[84] Lloyd Russell, Anthony Hu, Lorenzo Bertoni, George Fedoseev, Jamie Shotton, Elahe Arani, and Gianluca Corrado. Gaia-2: A controllable multi-view generative world model for autonomous driving. *arXiv preprint arXiv:2503.20523*, 2025.

[85] 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.

[86] 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.

[87] 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.

[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.

[89] 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.

[90] 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.

[91] 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.

[92] 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.

[93] 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.

[94] 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.

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

[96] 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.

[97] 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.

[98] Korrawe Karunratanakul, Jinlong Yang, Yan Zhang, Michael J Black, Krikamol Muandet, and Siyu Tang. Grasping field: Learning implicit representations for human grasps. In *2020 International Conference on 3D Vision (3DV)*, pages 333–344. IEEE, 2020.

[99] Mohamed Hassan, Yunrong Guo, Tingwu Wang, Michael Black, Sanja Fidler, and Xue Bin Peng. Synthesizing physical character-scene interactions. In *ACM SIGGRAPH 2023 Conference Proceedings*, pages 1–9, 2023.

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

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

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

[103] 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.

[104] 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.

[105] 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.

[106] 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.

[107] 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.

[108] 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.

[109] 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.

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

[111] 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.

[112] 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.

[113] 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.

ORV: 4D Occupancy-centric Robot Video Generation

Supplementary Materials

This supplementary document provides additional analysis and technical details regarding our proposed **ORV**. We begin with the detailed introduction of all datasets used in our work in Sec. A. We additionally explain more about our ORV-MV and ORV-S2R in Sec. B and Sec. C. And the additional experiments and analysis in Sec. D further demonstrate the superiority of our work. After that, we describe more detailed implementations of our model for any reproduce purpose in Sec. E. Finally, we showcase additional qualitative results of ORV in Sec. F and have another discussions in Sec. G. Our demo, code and models are publicly available at `https://orangesodahub.github.io/ORV`

A Datasets Details

BridgeV2 [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 [90] 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 [89] 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.

All datasets used in our work (BridgeV2 [88], Droid [90], RT-1 [89]) are maintained under CC-BY-4.0 License.

B ORV-MV Details

In Figure 5, we use the multiview 2D conditioning maps to enhance the multiview videos generation quality, just as we do in single-view video generation 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.

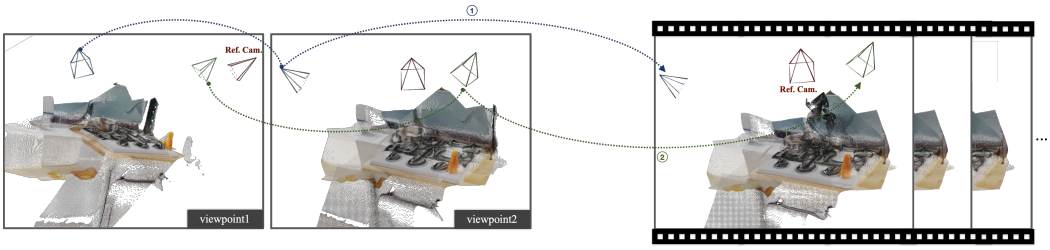

Figure A: Illustration of ORV aligning multiview cameras from VGGT [101] under the frame of Monst3R [28] to get the multiview conditioning sequences.

As described in Sec. 3.1, we extract 4D points from a single-view input (referred to as the anchor view) using Monst3R [28]. To get multiview conditions, we estimate camera poses across all views in the dataset using VGGT [101]. Note, however, that these two approaches produce different coordinate spaces.

We then have a simple yet efficient approach to combine the advances of Monst3R [28] and VGGT [101]. As illustrated in Figure A, 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 [102] on the depth maps to estimate these values [103], 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. Figure B 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).

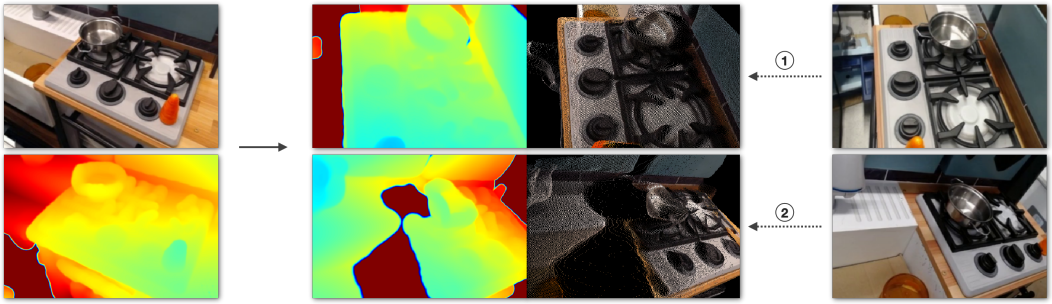

Figure B: Example of transferring multiview poses from VGGT [101] to Monst3R [28]. The comparison of calibrated side views and the side views demonstrates the efficiency.

C ORV-S2R Details

As depicted in Figure C, our simulated tabletop manipulation environments are constructed within the Maniskill [104] framework. The primary objective is to generate comprehensive occupancy data from diverse manipulation scenarios. This occupancy data serves as a crucial conditional input for a subsequent model designed to synthesize high-fidelity images. The generation process involves careful scene construction, strategic object placement, and the development of a capable grasping policy to interact with objects and thereby produce the necessary spatial occupancy information.

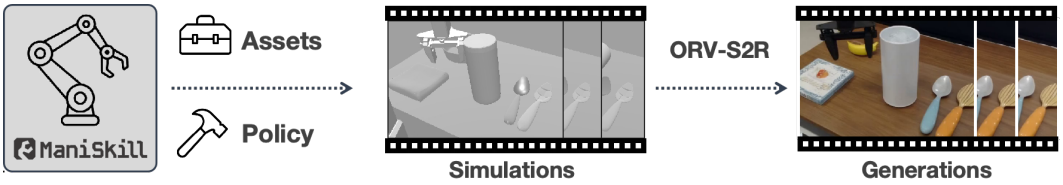

Figure C: Illustration of our simulation-to-real pipeline. We build simulated dynamics in popular simulation tools (*e.g.*, ManiSkill [104], and produce plausible geometries with generated motions. After that, with ORV-S2R, we transforme them into real-world videos.

We populate our scene layouts by first collecting a wide array of 3D object assets from established publicly available libraries. To further expand the variety of objects and introduce novel geometries,

we also employ image-to-3D reconstruction techniques [105, 106] to generate new assets from 2D images. Within each scene, we pre-define specific plausible regions on tabletop surfaces where objects can be placed. The final placement positions for these objects are then determined using a grid sampling strategy over these pre-defined areas. This ensures a structured yet varied distribution of objects, leading to a wide range of interaction possibilities and, consequently, diverse occupancy data.

To acquire the manipulation capabilities necessary for generating the required occupancy data, we employ a two-stage process. First, inspired by the initial phase of UniGraspTransformer [107], we train dedicated policies for individual objects or object categories using reinforcement learning (RL) with a two-finger parallel jaw grasper. These object-specific policies are optimized to generate successful grasp trajectories and interact effectively with their designated objects. Second, the successful interaction trajectories generated by these dedicated policies, encompassing various objects and initial poses, are collected. These dedicated policies are then directly utilized to perform the interactions within our simulated environments, and the resulting trajectories provide the basis for our occupancy data. This approach allows us to systematically generate the rich interaction data from these specialized policies, which is needed for creating the occupancy grids that condition our ORV model.

D Additional Experiments

In this section, we first give the detailed introductions of the baselines we compared in our work (presented in Table 2 and Figure 6). Then we provide additional comparison results and ablation on the control signals of controllable video generations of ORV. Moreover, we showcase the generalization ability of ORV, which plays a crucial role in the practical use of our model. After that, we have more analysis on multiview video generations of ORV.

Baselines. We compare our results with recent works. **IRASim** [15] is a video diffusion model employing DiT architecture with action modulation, which outperforms both VDM [108] and LVDM [109]. **HMA** [93] models video dynamics via a masked autoregressive transformer tailored for real-world action sequences. **AVID** designs a plug-in adapter which can inject action controls to pretrained video generation models. We also compare with the original text-to-video CogVideoX [14] model.

D.1 Controllable Video Generation

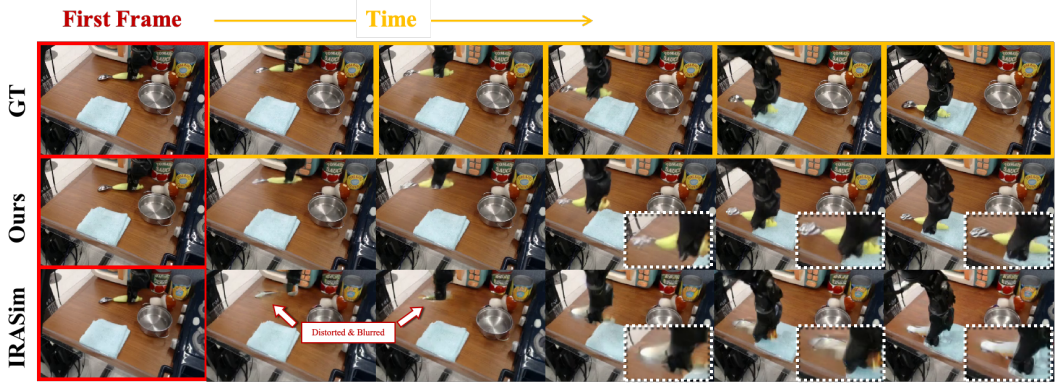

Figure D: Qualitative Results of **ORV** with full conditions. Red boxes denote the first frame input of the video generation; Orange boxes denote the ground-truth of the subsequence frames.

More Comparison with Baselines. In Figure D, we have another example to demonstrate the superiority of ORV. As highlighted by the red indicators and white boxes, the baseline [15] fails to correctly infer the physical appearance of the object handled by the robot gripper during the motion. However, this part of the dynamics is particularly essential to the downstream usage of our generated videos—such as policy learning, imitation learning. While ORV performs better.

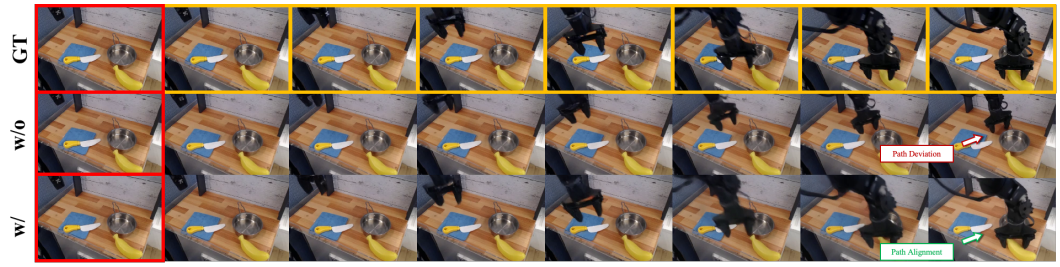

Figure E: Ablation Results of **Depth Condition Map.** Without any physical controls, the robot gripper fails to act accurately aligned with the 3D action instructions, due to the accumulation of errors. While ours performs correctly, along with the entire sequence.

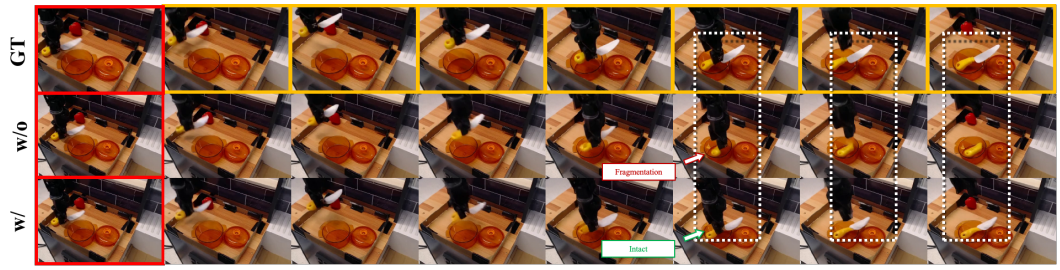

Figure G: Ablation Results of **Semantics Condition Map.** Without the guidance of our rendered semantics maps, the model fails to accurately predict the shape deformation of the knife during its motion, whereas ours produce outputs that align well with the real-world appearance.

Effect of control signals. We present quantitative comparisons in Table 3 to demonstrate the improvements enabled by the physical control signals. Furthermore, we highlight this in Figure E. As shown, without depth guidance, the robot gripper fails to accurately execute the 3D action instructions—an expected outcome, as 2D pixels are inherently insensitive to depth variations. In contrast, with the amendment by our rendered depth conditions, this limitation is effectively resolved. And Figure G shows the qualitative comparison between with and without the semantics condition maps, where we can see the obvious improvement from this kind of conditions.

We further collect the evaluation scores across all the samples and analyze the effect of guidance on occupancy conditions. Taking the BridgeV2 data as an example, Figure F illustrates the sample-wise improvement in PSNR and SSIM metrics after applying the full condition. We first sort the evaluation samples based on the scores obtained with the base model, namely only the 3D action condition (blue curve), and then, following this order, we plot the scores of each sample with the full condition (orange curve). Additionally, the green line indicates the improvement (%) for each sample.

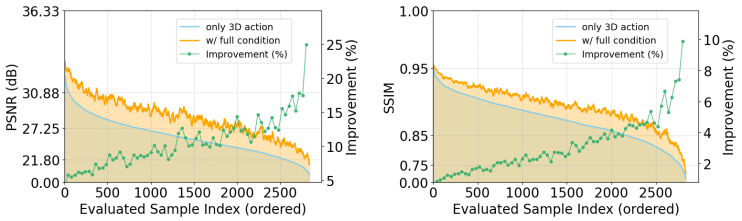

Figure F: Improvement curves of PSNR (left) and SSIM (right) metrics across ordered evaluation samples from BridgeV2 [88].

D.2 Generalizations

Despite employing SFT to adapt a pretrained CogVideoX model as our base model, ORV retains strong generalization capabilities, enabling robust performance across diverse scenarios in the robot manipulation task. Figure H demonstrates our model's video generations under varying appearances and arbitrary action control modifications, exhibiting both precise controllability and effective generalization. Furthermore, ORV also maintain the out-of-domain generalization, namely operating

on the in-the-wild observation inputs. However, since ORV takes no texts as the prompt and it relies on visual clues to infer the state of the robot arms or grippers, it cannot complete a meaningful task yet.

D.3 Multi-view Videos Generation

Maintaining consistency across different views in multiview video generation is crucial. Although the model may possess the ability to infer view orientations from the observations frame (referred to as 'context frame') and predict how 3D motion control translates into 2D pixel changes across views, this capability is inherently limited. Hence, we provide the multiview conditionings which are consistently rendered from the 3D geometry representations, to enhance the 2D pixel predictions (as introduced in Sec. 3.3 and Sec. B).

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

Figure I shows the comparison of a 3-view video generation with and without the additional conditioning maps. In this example, although we construct the 3D occupancy solely from the reference view due to the constraint of data resource, which has lower quality than a complete 3D geometry, our conditioning maps rendered under the other two side views help to improve the generation quality to some extent. As highlighted by the white area, during the motion of the robotic gripper while holding a metal bowl, the bowl undergoes severe deformation in the current view—even though this issue is entirely absent in the reference view. This is primarily due to two reasons: first—and most importantly—the current view differs significantly from the reference view; second, the object has relatively intense motion. With additional guidance from 3D geometry, all these can be addressed readily.

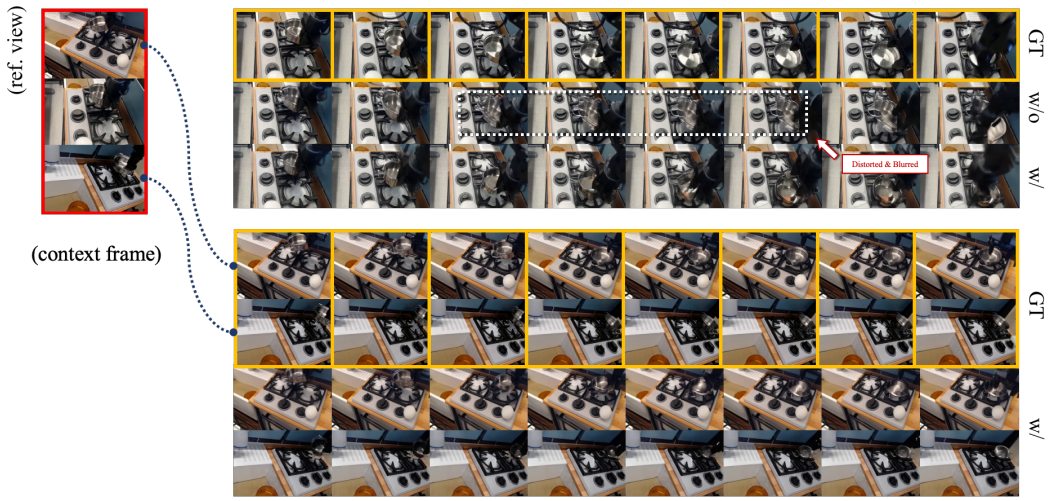

Figure I: Qualitative Comparison Results of **Multiview Videos Generation**. With our from-reference-view rendered visual conditionings, generated videos under side views achieve better geometric consistency under other side views. Better to zoom in.

E Implementation Details

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

E.1 Dataset Curation

In the process of dataset-level semantics labelset construction, we employ the VLM (QWen-VL-Chat [31]) to exhaustively caption all the scenarios in the dataset. Specifically, we use the text instruction as below:

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

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. In the process of Gaussian renderings, as described in Sec. 3.1, we apply a scaling schedule on the size of Gaussian splats, to more accurately represent the geometric surface. Specifically, we set $\alpha = 0.00023$, $\beta = 3.7$ for the BridgeV2 dataset, and $\alpha = 0.00047$, $\beta = 3.2$ for Droid and RT-1 datasets.

E.2 Model Architecture Details

Hyperparameters. As mentioned in Sec. 3.2, we use the CogVideoX-2B [14] as our pretrained backbone, which is a compromise between training from scratch and using the 5B pretrained model (as TesserAct [19]). And we have already shown its better performance than training from scratch 7 and strong generalization ability in the experiments. We list the main hyperparameters of the model architecture in Table 5, where * denotes those that are specialized in our model, while others keep the same as the CogVideoX-2B.

Table 5: Main hyperparameters of model architecture.

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

Modulation. CogVideoX [14] uses a design of Expert Adaptive Layernorm: It uses the timestep t of the diffusion process as the input to the modulation module. Then the Vision Expert Adaptive Layernorm (Vision Expert AdaLN) and Text Expert Adaptive Layernorm (Text Expert AdaLN) apply this modulation to the vision hidden states and text hidden states, respectively. Since we adapt the pretrained parameters from CogVideoX, we strictly keep this architecture. Moreover, to inject our 3D actions control, we reuse the Vision Expert AdaLN (aiming to modulate the vision hidden states) to apply such modulations from actions while keep the Text Expert AdaLN unchanged:

```
def forward(
    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, :]

...
```

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.

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

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

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

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.

E.3 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. A), 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 Table 6.

Table 6: Hyperparameters of data preprocessing for training and evaluations.

	seq. length	raw size	sample size	latent size	step size	sample interval
BridgeV2 [88]	16	480×640	320×480	40×60	1	4, 16
Droid [90]	24	180×256	256×384	32×40	3	16, 72
RT-1 [89]	16	256×320	320×480	40×60	2	6, 16

Note that the number of total frames of each individual episode varies significantly across the datasets (*e.g.*, 20∼50 for BridgeV2 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 BridgeV2 [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.

Table 7: Distributions of multiview data samples of BridgeV2 [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

E.4 Evaluation Details

We evaluate our model across four common metrics: Peak Signal-to-Noise Ratio (PSNR) [110], Structural Similarity Index Measure (SSIM) [111], Fréchet Inception Distance (FID) [112] and Fréchet Video Distance (FVD) [113]. All of our evaluations involve the ∼2.6K of generated samples.

E.5 Computation Resources

We implement ORV in PyTorch, using the `diffusers`[2] and `transformers`[3] libraries. Our models are trained and evaluated on an 8 × H100 cluster. Each experiment utilize 8 GPUs in parallel, with 16 data loader workers per device. Since we use the similar volume of tokens and size of in models 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 BridgeV2 [88] in our experiments occupies about 8TB of disk space.

F Additional Qualitative Results

In this section, we provide more **uncurated** singleview examples generated by ORV, as shown in Figure J, K, L. 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.

[2]https://github.com/huggingface/diffusers under Apache License

[3]https://github.com/huggingface/transformers under Apache License

G Limitations and Future Work

Although we have achieved promising results in , it remains a challenging task with many unsolved problems, and our method is limited in some aspects. This section provides a detailed discussion of the limitations and outlines potential future directions.

(1) In our work, although our 3D occupancy provides geometry representation for all objects in the scene, the 3D action signal only describes the end-effector pose of the robotic arm. This description is insufficient for arms with more complex articulated joints—such as the Google robot used in the Droid [90] dataset. Incorporating precise motion descriptions of all the joints would yield a more accurate representation of the arm's trajectory.

(2) Our current ORV-MV requires the inputs of first-frame observations from multiple views. By leveraging geometric constraints from the 3D occupancy and the robotic arm pose observed in the initial frames, ORV-MV is able to generate view-consistent videos. In the future, we plan to include the generation of multi-view first-frame images within this framework—*i.e.*, generating consistent multiview videos from only a singleview first-frame input. This enhancement would significantly improve the usability and practicality of ORV-MV.

H Social Impact

This work advances the field of controllable robot video generation, which has broad potential applications in areas such as robotics simulation, education, virtual reality, and creative media production. However, we recognize the dual-use nature of generative video models, particularly the risk of misuse in creating misleading or deceptive content (e.g., deepfakes) that could contribute to misinformation or privacy violations. To mitigate these concerns, our research is conducted under a responsible AI framework: we use publicly available, ethically sourced datasets, and our models are intended strictly for academic research. We encourage future work to incorporate safeguards such as provenance tracking and synthetic content detection alongside model development, ensuring that the societal benefits of generative technologies are realized while minimizing their potential harms.

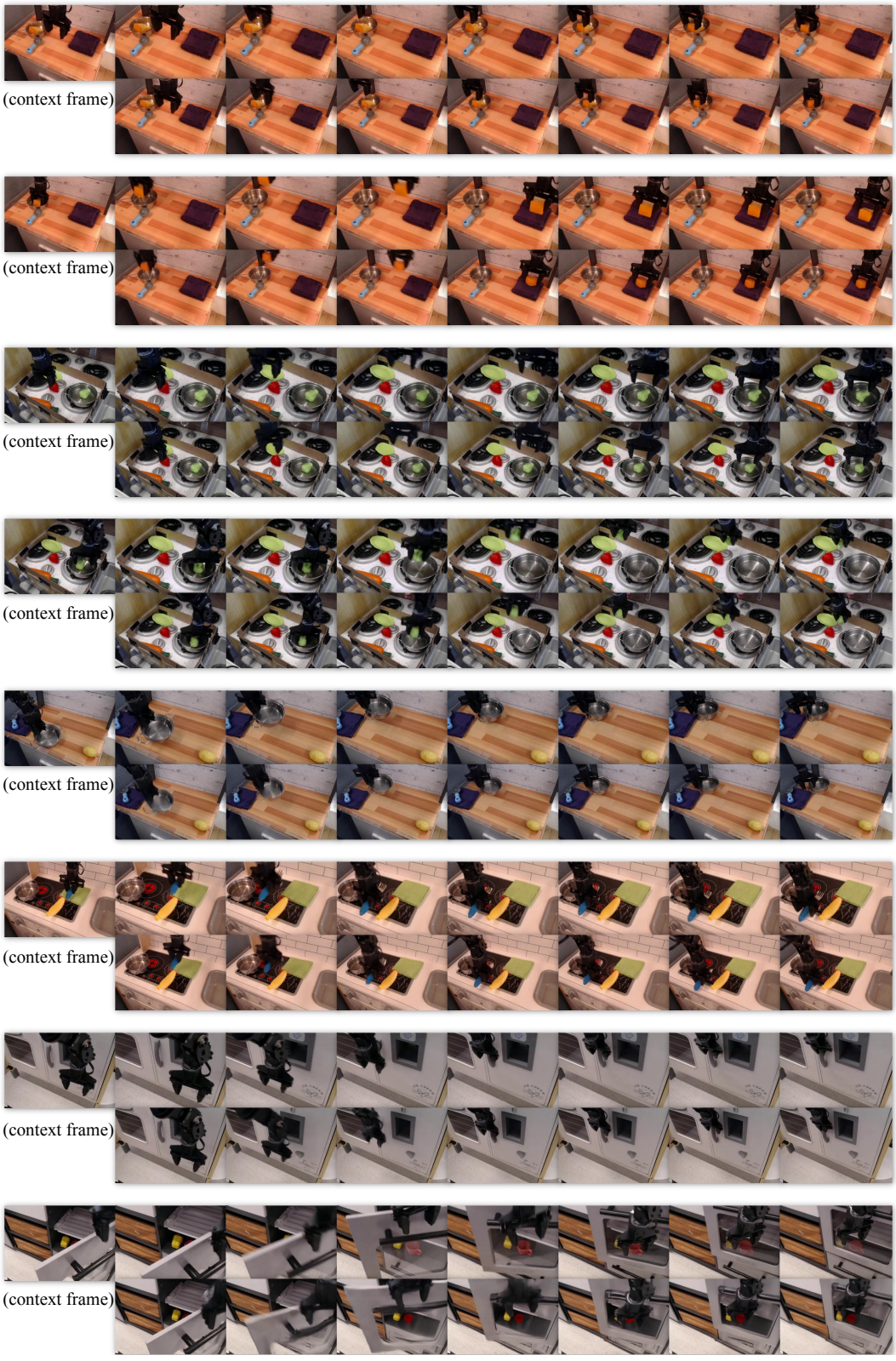

Figure J: Additional Qualitative Results of ORV #1.

Figure K: Additional Qualitative Results of ORV #2.

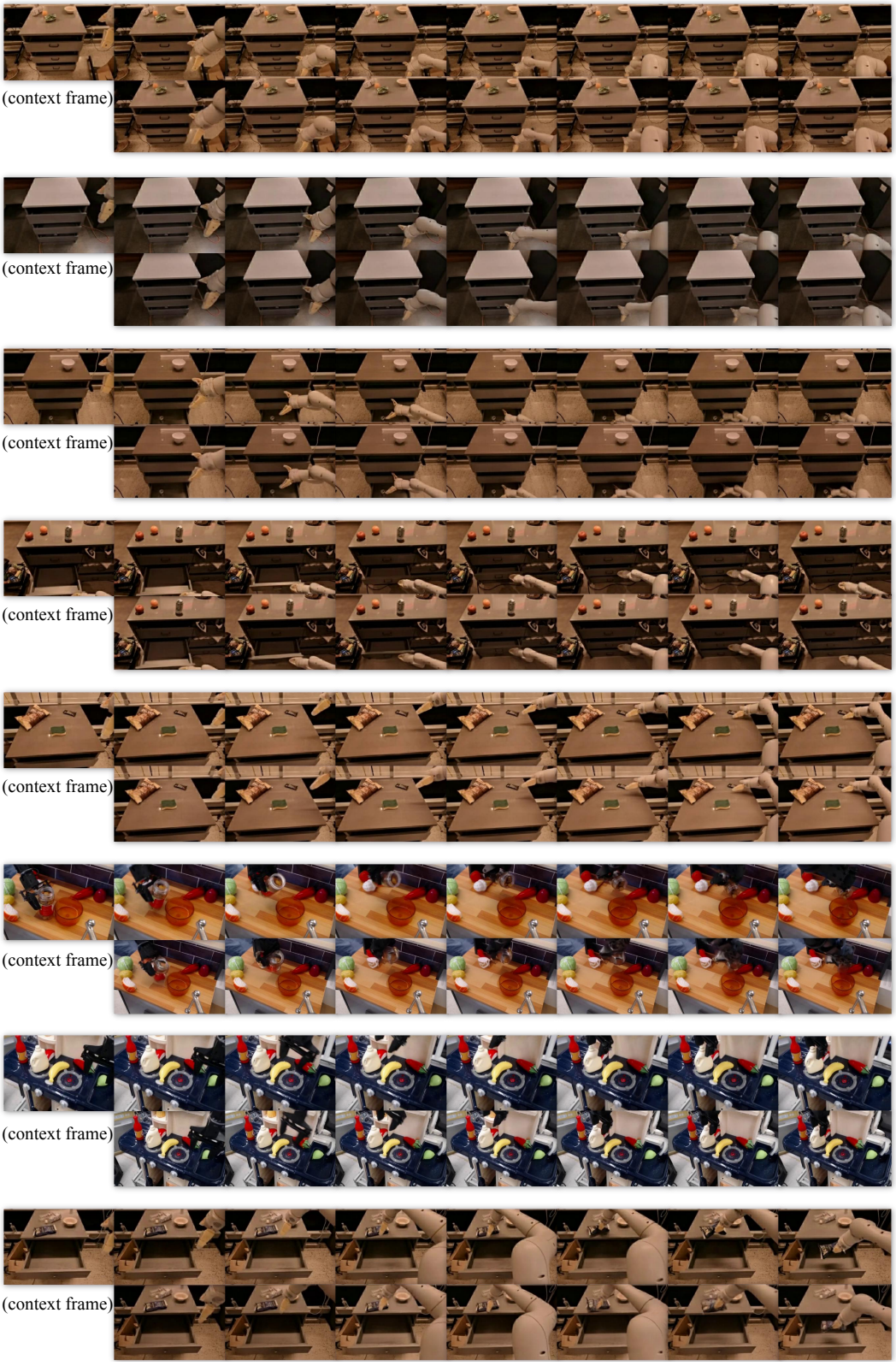

Figure L: Additional Qualitative Results of ORV #3.