ManipDreamer3D: Synthesizing Plausible Robotic Manipulation Video with Occupancy-aware 3D Trajectory

Ying Li[1,2,3,4], Xiaobao Wei[1,4], Xiaowei Chi[3] *, Yuming Li[1,2], Zhongyu Zhao[1,2], Hao Wang[1], Ningning Ma[4], Ming Lu[1] †, Sirui Han[3] †,

[1]State Key Laboratory of Multimedia Information Processing, School of Computer Science, Peking University
[2]School of Software and Microelectronics, Peking University
[3]Hong Kong University of Science and Technology
[4]Autonomous Driving Development, NIO

Abstract

Data scarcity continues to be a critical bottleneck in the field of robotic manipulation, limiting the ability to train robust and generalizable models. While diffusion models provide a promising approach to synthesizing realistic robotic manipulation videos, their effectiveness hinges on the availability of precise and reasonable control instructions. Current methods primarily rely on 2D trajectories as instruction prompts, which inherently face issues with 3D spatial ambiguity. In this work, we present a novel framework named ManipDreamer3D for generating plausible 3D-aware robotic manipulation videos from the input image and the text instruction. Our method combines 3D trajectory planning with a reconstructed 3D occupancy map created from a third-person perspective, along with a novel trajectory-to-video diffusion model. Specifically, ManipDreamer3D first reconstructs the 3D occupancy representation from the input image and then computes an optimized 3D end-effector trajectory, minimizing path length, avoiding collisions and retiming. Next, we employ a latent editing technique to create video sequences from the initial image latent, text instruction and the optimized 3D trajectory. This process conditions our specially trained trajectory-to-video diffusion model to produce robotic pick-and-place videos. Our method significantly reduces human intervention requirements by autonomously planing plausible 3D trajectories. Experimental results demonstrate its superior visual quality and precision.

Introduction

Collecting real-world robot manipulation demonstrations is often time-consuming, labor-intensive, and constrained by hardware limitations (Park et al. 2024; An et al. 2025). These challenges hinder the scalability of robotic policy learning, where large and diverse datasets are critical for achieving robust generalization (Schmidt et al. 2018; Xu and Mannor 2012). Generating realistic demonstrations with safe and short trajectories becomes particularly important (Ding et al. 2020; Hanselmann et al. 2022), as it can reduce the dependency on extensive physical data collection while providing high-quality supervision for an effective robot policy model.

Recent works have sought to expand real-world robot manipulation datasets by replaying recorded trajectories with diverse visual augmentations, such as altering objects, robot embodiments, textures, backgrounds, distractors, lighting conditions, and camera viewpoints (Mandi et al. 2022; Chen et al. 2023; Fang et al. 2025; Yang et al. 2025a). Beyond augmentation-based replay, Re^{3}Sim (Han et al. 2025) reconstructs realistic scenes in simulation to synthesize plausible manipulation demonstrations, while ORV (Yang et al. 2025b) uses 4D semantic occupancy as an intermediate representation to generate realistic videos from either real-world or simulated data. Modeling robotic actions serves as another vital method for enriching realistic manipulation datasets. This&That (Wang et al. 2025a) conditions generation on object and target gestures, whereas RoboMaster (Fu et al. 2025) jointly models both robot and object trajectories to guide the generation process.

While these approaches have generated promising robotic videos, several limitations remain. First, existing methods often overlook that robotic actions are planned in a 3D space. As a result, the generated videos may contain trajectories that violate physical constraints, lack collision avoidance, or execution efficiency. These methods still rely heavily on manual object selection (Yang et al. 2025b; Wang et al. 2025a; Fu et al. 2025). Second, the generated scenes fail to maintain geometric and physical consistency with real-world environments (Wang et al. 2025a; Zhou et al. 2024; Li et al. 2025). Even when the generated 2D videos exhibit promising perceptual quality, inaccuracies in object size, placement, or contact state can lead to unrealistic interactions. These issues significantly limit the applicability of such videos for training generalizable robotic policies.

To overcome these limitations, we propose ManipDreamer3D, a novel method to automatically generate both realistic and physically-plausible robotic manipulation videos given a third-view observation image along with an instruction. The key idea is to first reconstruct a 3D occupancy representation of the scene and plan a physically valid, short manipulation trajectory with reasonable speed from the robot end-effector to the manipulated object, and then to the destination. Afterwards, we transforms the initial image latent to a temporally coherent latent video by masked replacement following the 3D trajectories of both end-effector and object, achieving precise spatial control, fi-

*Project Leader
†Corresponding Authors

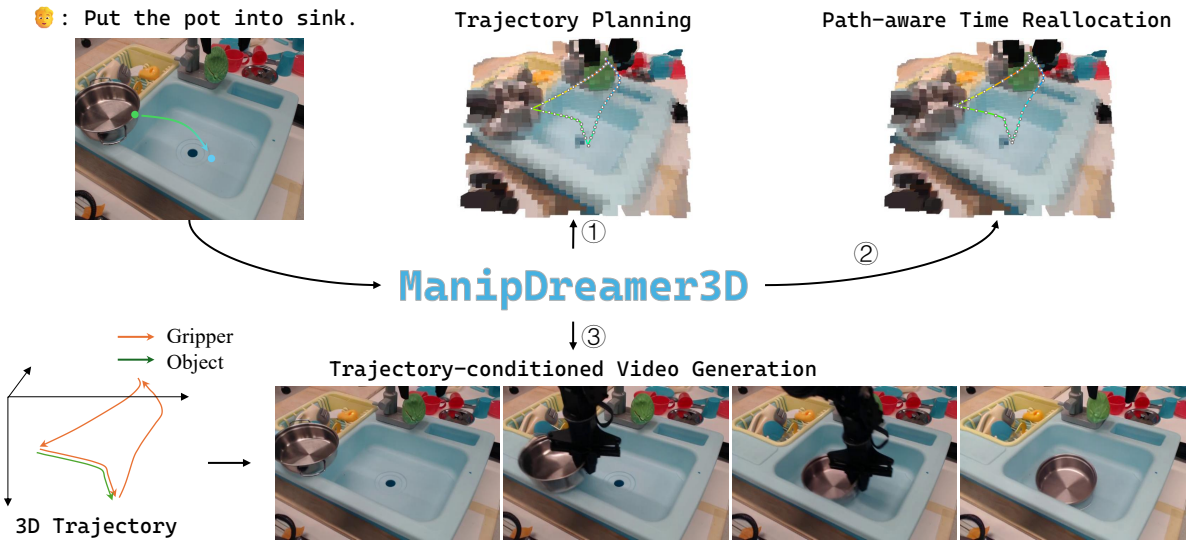

Figure 1: Overview of our proposed method ManipDreamer3D. Given a user-specified third-view image, an instruction, and gestures, ManipDreamer3D first **constructs an occupancy grid** and **initializes and optimizes sub-trajectories** within the grid space. The **time intervals are re-allocated** based on the sub-trajectory path lengths and predefined velocity profiles. Finally, ManipDreamer3D **synthesizes the output video** conditioned on the trajectories of both the robot end-effector and the object.

nally, the manipulation video is synthesized with diffusion model conditioned on the latent video.

In conclusion, our key contributions are threefold.

- We propose a novel occupancy-aware 3D trajectory planner that yields collision-free, length-efficient end-effector trajectories with reasonable velocities.
- We introduce a simple yet efficient trajectory-to-video synthesis scheme, which seamlessly plugs into diffusion-based models without any auxiliary modules.
- Our comprehensive experiments across diverse manipulation scenarios demonstrate that ManipDreamer3D achieves SOTA performance across multiple video quality metrics in robotic trajectory-conditioned video generation, while maintaining precise trajectory control.

Related Works

Robotic Trajectory Planning

Trajectory planning is a critical component in robotics and autonoumous driving, including vision-language navigation (VLN) (Zhang et al. 2024; Wei et al. 2025; Chen et al. 2025a,b; Huang et al. 2024a; Wei et al. 2024) and vision-language-action (VLA) tasks (Kim et al. 2024; Cao et al. 2025). A common paradigm of path planning in VLN involves constructing a global representation of the environment followed by subsequent path planning for mobile robots. For instance, (Wang et al. 2025c) proposes a method that samples path proposals via a random walk strategy and scores them using a multi-modal transformer. Alternatively, some studies leverage large language models (LLMs) for reasoning and planning based on human instructions and visual observations. NavGPT (Zhou, Hong, and Wu 2024) utilizes a supportive interaction and memory tracking framework, while (Chen et al. 2024) employs a global topological node map to enhance spatial reasoning for LLMs.

In contrast to VLN's global planning focus, the VLA field prioritizes real-time action prediction in a target-approaching manner, often generating short-term action chunks(Zitkovich et al. 2023; Kim et al. 2024; Black et al. 2024). For instance, OpenVLA processes the current observation and outputs the next 16 (8) delta actions chunk under 15 Hz(5 Hz) control, while π_0(Black et al. 2024) employs a 50-step action horizon. These predicted actions are subsequently refined into executable subtle executable trajectories by low-level controllers using numerical algorithms, like RRT(LaValle 1998) and CHOMP(Ratliff et al. 2009).

Unlike VLA models that predict conditional action distributions from observation history and robot states, robotic manipulation video generation models benefit from being provided with globally optimal paths that unlocks holistic semantic understanding of a task. To realize this, we propose planning complete 3D trajectories in occupancy space, optimized via a CHOMP-inspired method. We detail this trajectory planning approach in the path planning section.

Trajectory Controlled Video Generation

Recent advances in motion-controlled diffusion video generation have demonstrated remarkable progress. TrailBlazer (Ma, Lewis, and Kleijn 2024) introduces sparse keyframe bounding boxes to guide object motion. DragAnything (Wu et al. 2024) employs 2D Gaussian maps and entity feature maps extracted by a pretrained model for generation guidance. MotionCtrl (Wang et al. 2024) control camera and object motion through Camera Motion Control Module (CMCM) and Object Motion Control Module (OMCM) respectively, while a recent work, DaS (Gu et al. 2025), pro-

Method	Keypoint Control	Full Trajectory Control	Affordance Control
This&That	✓	✗	✗
RoboMaster	✓	✓	✗
ManipDreamer3D	✓	✓	✓

Table 1: Control capabilities of existing methods. Ours ManipDreamer3D is the most fine-grained by supporting keypoint, full-trajectory, and **affordance control**.

poses a unified 3D-tracking representation that unified camera and object motion by tracking 3D point movements.

These methods are extended to robotic video generation where the core challenge lies in jointly modeling robot and object motion as cameras are typically fixed. This&That (Wang et al. 2025a) places one gaussian at the object's initial position on grasping frame and another at its target on placement frame of a 2D gaussian map video, controlling generation via ControlNet. RoboMaster (Fu et al. 2025) divides the whole process into three phases based on the dominate moving agents, edits latents according to agents' movements and conditions with spatial-temporal convolution injection modules.

Method

Problem Formulation

Given a single third-view observation image $I_0 \in \mathbb{R}^{3\times H\times W}$ and corresponding instruction t, our goal is to generate a 3D-aware robot manipulation video. The pipeline consists of four main components: 1) We first construct a 3D occupancy map $O \in \mathbb{R}^{h\times w\times d}$. This discrete volumetric representation captures the spatial distribution of objects and the scene. 2) Upon the occupancy map, we plan an initial 3D trajectory $P^3_{init} \in \mathbb{R}^{N\times 3}$ via the A^* algorithm and optimize it through gradient descent to obtain a shorter and smoother gripper trajectory P^3_g. The object trajectory P^3_o is then acquired by relatively static assumptions. 3) These 3D trajectories are projected to 2D mask maps of object and gripper $M^{obj}, M^{grip} \in \mathbb{R}^{N\times 3\times H\times W}$ and guides the construction process of video latent $\mathcal{Z}$. 4) The output video $V \in \mathbb{R}^{N\times 3\times H\times W}$ is conditionally generated with video latent $\mathcal{Z}$. Our method offers the most fine-grained control compared to baselines, as in Tab. 1 and showcases in Fig. 7.

Occupancy-aware path planning

3D Occupancy Map Reconstruction To enable effective robot motion planning, we first establish an accurate 3D representation of the scene. We decide to use occupancy to represent the matters in the scene, as it is able to represent local structure in the scene and provides an regular scene representation that benefits path searching. We apply three steps to construct an occupancy grid given a single-view observation. Firstly, we leverage the 3D scene understanding capabilities of VGGT (Wang et al. 2025b) to generate an initial point cloud in camera coordinates. However, this raw point cloud exhibits discontinuity in regions of occlusion. To address this, we employ a neural surface reconstruction technique (Huang et al. 2023) that is able to recover a continuous surface from sparse points. From this reconstructed surface, we uniformly resample points to create a more complete point cloud. Finally, we discretize the 3D space by converting the processed point cloud into a $64 \times 64 \times 64$ occupancy grid, where each voxel indicates the presence or absence of matter, balancing efficiency and precision. Fig. 2 (a) illustrates this multi-stage 3D reconstruction pipeline.

Optimal Trajectory Planning We propose a three-stage method to find a locally optimal manipulation path in the 3D occupancy space. First, we generate an initial trajectory using the A^* algorithm applied three times for different stages:

- **Approaching stage.** The path P_1 is planned from the end-effector's 3D position to the object's position.
- **Manipulating stage.** P_2 moves while grasping the object from its initial position to the target position.
- **Back-idle stage.** The last path P_3 returns the end effector to the start position.

While A^* provides a heuristic solution, the trajectories needs to be further optimized for safety and smoothness. Therefore these sub-trajectories P_1, P_2, P_3 are optimized for safety and smoothness respectively. Inspired by the widely used CHOMP algorithm, we jointly optimize the trajectory with multiple objectives to find a plausible, short, and smooth path as shown in Fig.2(b). Note each the ith point in $P^3 = P^3_{init}$ as $\mathbf{p}_i$, the objectives are formulated as: **Collision Loss.** A signed distance field (SDF) is constructed to represent the field of distance toward the static background of the scene, and a safe distance hyperparameter is adapted. **Path Length Loss.** This target aims to minimize the path length, ensuring the effectiveness of robot manipulation. **Smoothness Loss.** Smooth loss includes two parts, an acceleration loss and a curvature loss. This loss avoids sharp acceleration or direction changes of robot end effector.

$$\mathcal{L}_{col} = \sum_{i=1}^{N} (SDF(\mathbf{p}_i)) \tag{1}$$

$$\mathcal{L}_{len} = \sum_{i=1}^{N-1} ||\mathbf{p}_i - \mathbf{p}_{i+1}||^2 \tag{2}$$

$$\mathcal{L}_{acc} = \frac{1}{2} \sum_{i=1}^{N-2} ||\mathbf{p}_{i+2} - 2\mathbf{p}_{i+1} + \mathbf{p}_i||^2 \tag{3}$$

$$\mathcal{L}_{cur} = \frac{1}{2} \sum_{i=1}^{N-2} \left(\frac{||\mathbf{v}_i \times \mathbf{a}_i||^2}{||\mathbf{v}_i^3||^2 + \epsilon} \right) \tag{4}$$

In the above formulation, $\mathbf{v}_i = \mathbf{p}_{i+1} - \mathbf{p}_i$ is velocity of each point, $\mathbf{a}_i = \mathbf{p}_{i+2} - 2\mathbf{p}_{i+1} + \mathbf{p}_i$ is the acceleration term, $\epsilon = 1e^{-6}$ is a small constant and $\times$ denotes the cross product. Note that the start and end are specially treated and are not optimized, therefore keeping their original position. We optimize the point group P^3 to minimize the following objective using Adam optimizer and a learning rate of 0.1 for a fixed number of iterations and finally obtain P^3_{opt}:

$$\min_{P^3} (\omega_{len}\mathcal{L}_{len} + \omega_{curv}\mathcal{L}_{cur} + \omega_{acc}\mathcal{L}_{acc} + \omega_{col}\mathcal{L}_{col}) \tag{5}$$

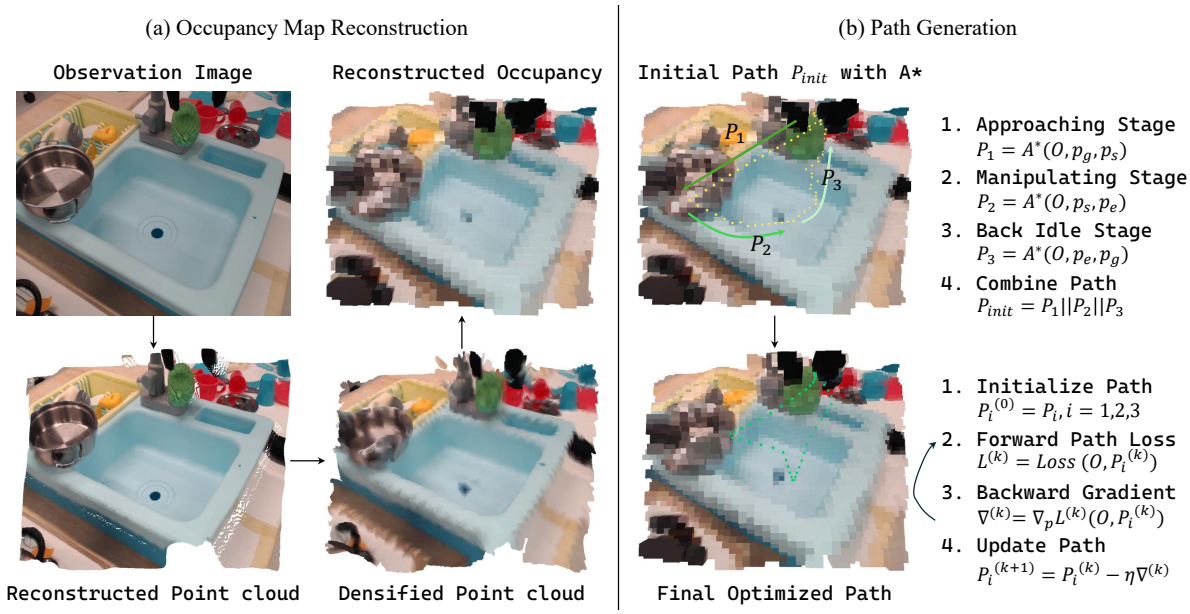

Figure 2: **The occupancy construction and trajectory pipeline.** (a) In the occupancy reconstruction process, we first estimate and densify the point cloud, then extract an occupancy out of the densified point cloud. (b) The procedure of path generation in occupancy. We first generate 3 initial sub-trajectories with A^*, and then each sub-trajectory is then optimized with gradient descent. We formulate the initialization of sub-trajectories and optimization process of a given path P on the right side.

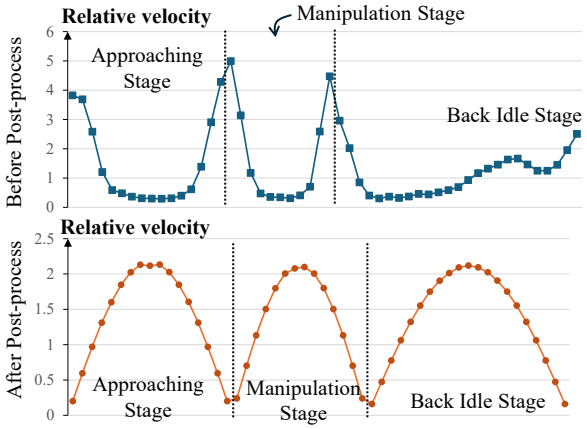

Figure 3: Distribution of velocity before and after path-aware time reallocation of one example.

After the optimization process, we obtain optimized sub-trajectories $P_i^3, i = 1, 2, 3$ that are plausible, as short and more safe in their path shape. However, due to the ignorance of the above objectives towards the mechanic nature of robot, the speed distribution of the above points follows poorly with real robots, which typically first accelerate and then decelerate within each sub-trajectory.

Path-aware Time Reallocation To adjust the robot's speed throughout the manipulation process according to a predefined velocity profile, we propose a post-processing method that redistributes trajectory points based on path length and desired speed characteristics. First, the number of points in each sub-trajectory is reassigned proportionally to its arc length. Then, each point is repositioned along the path via interpolation between the two nearest original points, ensuring the resulting spatial distribution matches the intended velocity profile. We use sine wave as our default velocity profile for each sub-trajectory.

As shown in Fig. 3, the velocity distribution of the original trajectory exhibits inconsistencies with physics rules, whereas the recalculated trajectory after post-processing adheres more closely to physically realistic motion constraints. This realignment results in a velocity profile that is both smoother and more applicable to real-world execution.

3D Trajectory Data Curation

3D Scene Reconstruction. We first generate temporally consistent point clouds of each frame of the third-view 2D video by processing the entire sequence with a VGGT model (Wang et al. 2025b). This produces a unified 3D point maps of the scene along with estimated camera parameters.

End-Effector Localization. Accurately localizing the 3D position of the robotic end-effector is challenging due to its dynamic motion and lack of fixed reference points. To address this, we employ a fine-tuned YOLO model (Yaseen 2024) specifically for detecting the gripper fingers. The 2D centers of the detected bounding boxes are then mapped to corresponding 3D coordinates, and the midpoint between these two positions is set as the 3D location of the gripper.

Object Detection and Segmentation. We first identify the grasping initiation moment—when the gripper is about to contact the target object. Following This&That (Wang et al. 2025a), we use the gripper's 2D center point at this mo-

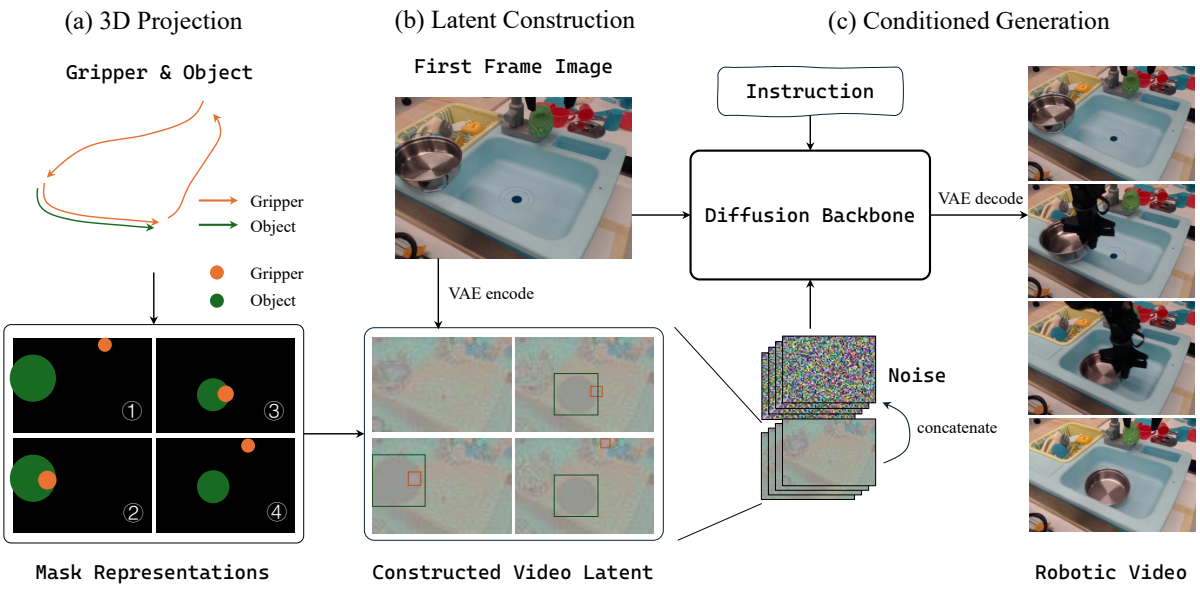

Figure 4: The conditioned Video generation pipeline. (a) We use a **3D-to-2D projection** to create masks that represent the position and distance of object or gripper in each frame, we draw the mask of object and gripper in the same mask for clarity. (b) We apply a **latent editing method** using the first frame latent and the corresponding masks to create a video latent. (c) We concatenate the constructed video latent with noise to guide the denoising process of video in a **parameter-free** manner.

Method		Video Quality			Trajectory Accuracy	
		FVD ↓	PSNR ↑	SSIM ↑	$\text{TrajError}_{\text{robot}}$ ↓	$\text{TrajError}_{\text{obj}}$ ↓
SVD-based	DragAnything	158.42	21.13	0.792	18.97	27.41
	This&That	148.69	20.93	0.758	62.07	37.12
	ManipDreamer3D(SVD)	**143.33**	**22.75**	**0.807**	**17.40**	**18.77**
DiT-based	RoboMaster	147.31	21.55	0.803	16.47	24.16
	ManipDreamer3D(DiT)	**93.98**	**23.64**	**0.847**	**15.38**	**16.59**

Table 2: Common video metric results of our ManipDreamer3D SVD version, DiT version, and baselines.

ment as 2D indication of the object's position. We then leverage Qwen-VL (Bai et al. 2023) for visual grounding, using a prompt that combines both the estimated position and the object name (parsed from the instruction via an LLM). Finally, we apply SAM (Kirillov et al. 2023) to generate precise object masks based on the 2D position and bbox.

This multi-stage approach ensures high-fidelity 3D trajectory extraction for both the robotic arm and the manipulated object from original videos, forming a critical foundation for the training process of our video generation model.

Trajectory-Guided Video Synthesis

3D-to-2D Trajectory Representation Our framework transforms 3D manipulation trajectories into a compact 2D latent representation that effectively guides video synthesis. As shown in Fig. 4, the representation consists of three key components: (1) first-frame latent encoding with VAE, (2) projecting dynamic object and end-effector trajectory and construct masks, and (3) temporal latent construction.

Latent Editing Preparation. We initialize our representation by extracting the latent code of the first video frame using the video diffusion model's variational autoencoder. Following RoboMaster (Fu et al. 2025), we employ pooled latent vectors to represent key scene elements: For the manipulated object, we interpolate its mask to match the latent spatial resolution and compute the mean-pooled latent vector within the masked region. For the end-effector, we generate a predefined latent vectors through similar operations and add biases to represent open/closed states of the gripper.

2D Mask Projection from 3D trajectories. The projection process involves three computational steps: *Distance Estimation*: We establish the object's depth by assuming constant camera-object distance during grasping, using the end-effector's known 3D trajectory as a reference to estimate the change of object distance. *Geometric Abstraction*: Object and gripper are modeled as spheres, the radius is defined as the maximum edge length of the box of object, while the end-effector uses a predefined fixed small radius. *Perspective Projection*: We projection 2D circles from their 3D spheres for both object and end-effector using perspective

Method		Aesthetic Quality ↑	Imaging Quality ↑	Temporal Flickering ↑	Motion Smoothness ↑	Subject Consistency ↑	background Consistency ↑
SVD-based	DragAnything	49.53	67.15	97.83	98.25	93.01	95.14
	This&That	**57.27**	**70.09**	97.20	97.91	94.43	95.45
	ManipDreamer3D(SVD)	52.46	69.24	**97.98**	**98.47**	**95.39**	**96.57**
DiT-based	RoboMaster	50.32	67.49	98.27	98.81	93.55	95.40
	ManipDreamer3D(DiT)	**51.44**	**68.65**	98.18	98.70	**94.38**	**95.87**

Table 3: The VBench metric results of our ManipDreamer3D SVD version, DiT version, and baselines.

projection, thus the scale reflects distance.

Temporal Latent Construction. The dynamic representation is constructed in the following way: For the first frame, we simply maintain the original latent to keep the initial spatial and semantic representation. For the following frames, we overlay the object and gripper latent vectors within their masked regions sequentially onto the first latent.

This representation efficiently encodes pixel-level position and distance of the manipulation trajectory while still being compatible with the diffusion model's architecture. The experimental results in the Experiments section demonstrate its effectiveness in guiding realistic video synthesis.

Conditioned Video Generation Unlike previous trajectory conditioned methods that rely on additional ControlNet (Wang et al. 2025a) or injection modules (Fu et al. 2025) to bridge trajectory conditions with video features, our approach directly concatenates the constructed latent with the noisy video latent along the channel dimension, replacing the traditional repeated static first-frame condition with a dynamic video condition, leading to precise control.

For UNet-based models such as SVD, where the VAE performs only spatial compression, the constructed latent can be directly fed into the denoising model. However, for DiT-based architectures like CogVideoX-5B (Yang et al. 2024), which uses a VAE that employs a $4\times$ temporal compression aside from spatial compression, we simply apply a two-layer $2\times$ average pooling on temporal dimension for alignment.

This latent-editing strategy offers two key advantages: it maintains the original diffusion model framework by replacing the repeated first-frame latent with our dynamically edited version, introducing no additional parameters. Furthermore, by using the first-frame latent as the editing base, we minimize distribution shift while preserving the inherent condition mechanism of model and achieve precise control.

Experiments

Experiment Setups

Model and data details. We train our models on a combination of the bridge V1 and bridge V2 datasets. After our curation pipeline, we finally acquired 8.7k valid episodes. These examples are randomly split by 9:1 for training and test. We implement our condition video generation method based on both pretrained SVD model (Blattmann et al. 2023) and CogVideoX-5B (Yang et al. 2024), we denote these two versions of our models as ManipDreamer3D(SVD) and ManipDreamer3D(DiT) respectively.

Metrics. To evaluate the quality of generated videos, we use common video qualities including FVD (Unterthiner et al. 2018), SSIM (Wang et al. 2004), and PSNR (Hore and Ziou 2010), as well as VBench (Huang et al. 2024b) metrics. Aside from that, we also evaluate our model with trajectory error, which reveals how well the model follows the trajectory. Following (Fu et al. 2025), we evaluated this metric for the robot end-effector and the target object, respectively.

Video Quality Results

The quantitative comparisons of video generation quality are presented in Tab. 2 and Tab. 3. Our ManipDreamer3D demonstrates superior performance across both evaluation frameworks. In common video metrics (Tab. 2), our SVD variant achieves best-in-class scores with FVD, PSNR, and SSIM, while significantly reducing trajectory errors compared to SVD-based competitors. The DiT version further elevates performance in these metrics.

The trajectory accuracy advantage stems from our gripper-object collaborative representation design. While This&That relies on ambiguous two-key-gesture control and DragAnything operates at the whole-entity movement level, lacking end-effector precision, our method explicitly models the spatial position of both gripper and object throughout the interaction. RoboMaster, which uses the same DiT backbone, implicitly models the position of the end-effector during interaction, underperforms in trajectory accuracy.

VBench evaluation results in Tab. 3 reveals our method's balanced strengths: the SVD model leads in temporal stability (flickering score) and consistency metrics, while the DiT variant maintains competitive Imaging quality with robust motion handling. These results collectively validate our approach's effectiveness in both visual quality and trajectory fidelity. We also visualize some of the same examples generated by our ManipDreamer3D(SVD) and This&That. As shown in Fig. 5, our model keeps the original shape of the original objects, while This&That suffers from object deformation. This aligns with our quantitative metrics, proving our superior visual quality.

Key Components Analysis

Precision Manipulation Control To evaluate the model's capability in precision manipulation control, we conduct experiments using different grasp parts (affordances) of objects to generate manipulation videos. The experiments employ our SVD-based model architecture to demonstrate fine-grained control over manipulation actions.

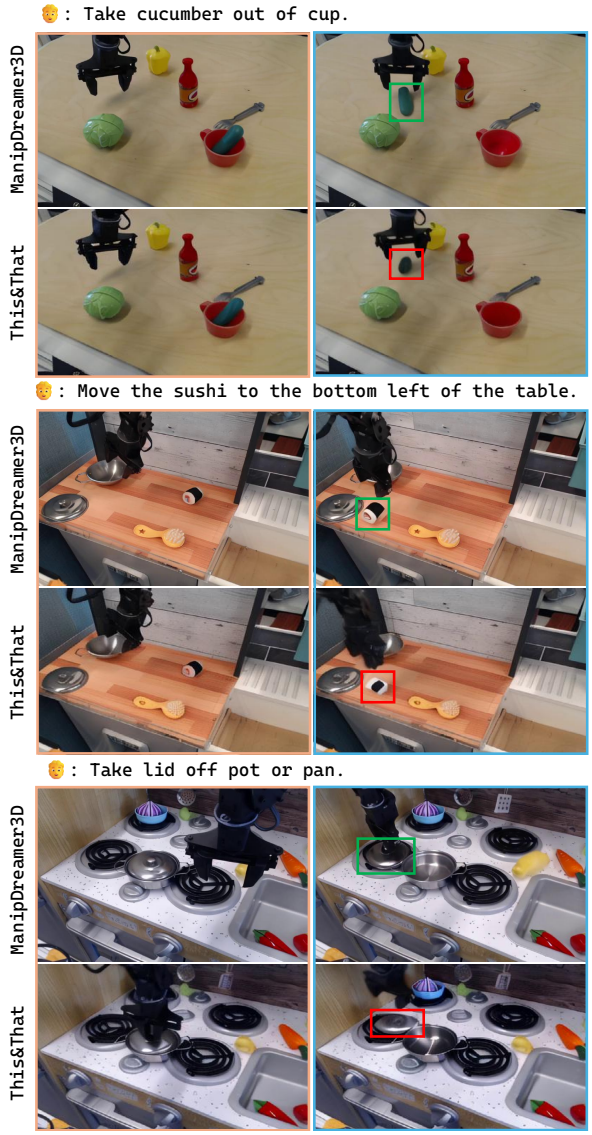

Figure 5: Qualitative comparison between our ManipDreamer3D and This&That (both SVD-based). Our method better preserves object appearance compared to baseline This&That, which exhibits noticeable shape distortions in manipulation results.

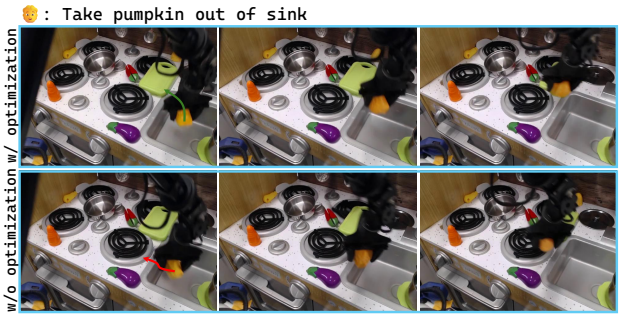

Figure 6: Video synthesis comparison between using initial trajectories and optimized trajectories.

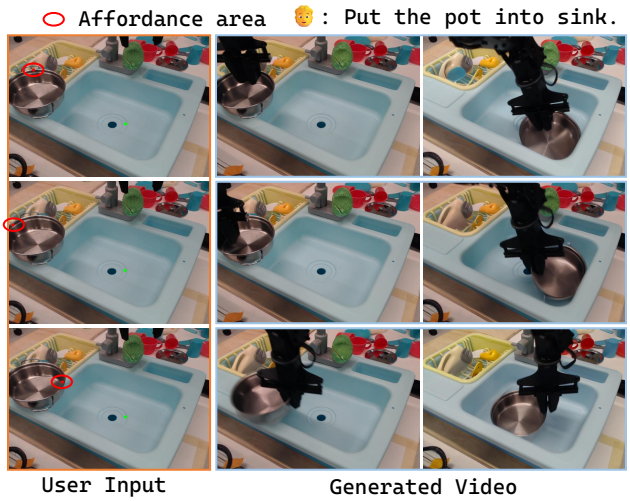

Figure 7: Precision manipulation control demonstrations. ManipDreamer3D generates manipulation videos with affordance level condition, achieving fine-grained controlling.

As illustrated in Fig. 7, we systematically vary the contact points on a pot object to test the model's responsiveness to different manipulation positions. The results show that our model successfully generates manipulation videos that accurately follow the specified conditions, demonstrating its ability to handle intricate manipulation tasks.

The impact of trajectory optimization To evaluate the impact of trajectory optimization, we compare videos generated using initial path P^3_{init} and optimized path P^3_{opt} by our optimization method. As shown in Fig. 6, with the P^3_{opt} trajectory, the generated video safely avoids colliding by moving upward, while P^3_{init} produces an possibly unsafe path close to the sink edge. Direct usage of videos from initial trajectory may cause unsafe behaviors in downstream VLA models, confirming the importance to optimize trajectories.

Conclusion

We propose ManipDreamer3D, a framework for generating robotic manipulation videos guided by 3D occupancy-aware trajectories. First, we reconstruct the scene in 3D, then plan efficient gripper and object trajectories, and finally synthesize a coherent video from the first-frame latent. This design enables comprehensive control at keypoint, full-trajectory, and affordance levels while requiring minimal manual annotation. Extensive experiments on diverse scenes demonstrate that ManipDreamer3D improves visual quality and spatial consistency, achieving stronger trajectory adherence compared with prior motion-controlled video generators.

Limitations and future work. Our current planning primarily targets rigid-body interactions and quasi-static grasps. Future work will explore more contact- and compliance-aware objectives, together with generative priors that better capture articulation and deformation.

Acknowledgments

This work was supported by the Beijing Natural Science Foundation (L252060).

References

An, S.; Meng, Z.; Tang, C.; Zhou, Y.; Liu, T.; Ding, F.; Zhang, S.; Mu, Y.; Song, R.; Zhang, W.; et al. 2025. Dexterous manipulation through imitation learning: A survey. *arXiv preprint arXiv:2504.03515*.

Bai, J.; Bai, S.; Yang, S.; Wang, S.; Tan, S.; Wang, P.; Lin, J.; Zhou, C.; and Zhou, J. 2023. Qwen-VL: A Versatile Vision-Language Model for Understanding, Localization, Text Reading, and Beyond. arXiv:2308.12966.

Black, K.; Brown, N.; Driess, D.; Esmail, A.; Equi, M.; Finn, C.; Fusai, N.; Groom, L.; Hausman, K.; Ichter, B.; et al. 2024. $\pi 0$: A vision-language-action flow model for general robot control. CoRR, abs/2410.24164, 2024. doi: 10.48550. *arXiv preprint ARXIV.2410.24164*.

Blattmann, A.; Dockhorn, T.; Kulal, S.; Mendelevitch, D.; Kilian, M.; Lorenz, D.; Levi, Y.; English, Z.; Voleti, V.; Letts, A.; et al. 2023. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*.

Cao, J.; Zhang, Q.; Jia, P.; Zhao, X.; Lan, B.; Zhang, X.; Li, Z.; Wei, X.; Chen, S.; Li, L.; et al. 2025. Fastdrivevla: Efficient end-to-end driving via plug-and-play reconstruction-based token pruning. *arXiv preprint arXiv:2507.23318*.

Chen, J.; Lin, B.; Xu, R.; Chai, Z.; Liang, X.; and Wong, K.-Y. 2024. MapGPT: Map-Guided Prompting with Adaptive Path Planning for Vision-and-Language Navigation. In Ku, L.-W.; Martins, A.; and Srikumar, V., eds., *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 9796–9810. Bangkok, Thailand: Association for Computational Linguistics.

Chen, Q.; Kiami, S. C.; Gupta, A.; and Kumar, V. 2023. GenAug: Retargeting behaviors to unseen situations via Generative Augmentation. In *Proceedings of Robotics: Science and Systems*. Daegu, Republic of Korea.

Chen, T.; Shorinwa, O.; Bruno, J.; Swann, A.; Yu, J.; Zeng, W.; Nagami, K.; Dames, P.; and Schwager, M. 2025a. Splat-nav: Safe real-time robot navigation in gaussian splatting maps. *IEEE Transactions on Robotics*.

Chen, Y.; Bai, X.; Wang, Z.; Bai, C.; Dai, Y.; Lu, M.; and Zhang, S. 2025b. StreamKV: Streaming Video Question-Answering with Segment-based KV Cache Retrieval and Compression. arXiv:2511.07278.

Ding, W.; Chen, B.; Xu, M.; and Zhao, D. 2020. Learning to collide: An adaptive safety-critical scenarios generating method. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2243–2250. IEEE.

Fang, Y.; Yang, Y.; Zhu, X.; Zheng, K.; Bertasius, G.; Szafir, D.; and Ding, M. 2025. Rebot: Scaling robot learning with real-to-sim-to-real robotic video synthesis. *arXiv preprint arXiv:2503.14526*.

Fu, X.; Wang, X.; Liu, X.; Bai, J.; Xu, R.; Wan, P.; Zhang, D.; and Lin, D. 2025. Learning Video Generation for Robotic Manipulation with Collaborative Trajectory Control. *arXiv preprint arXiv:2506.01943*.

Gu, Z.; Yan, R.; Lu, J.; Li, P.; Dou, Z.; Si, C.; Dong, Z.; Liu, Q.; Lin, C.; Liu, Z.; et al. 2025. Diffusion as shader: 3d-aware video diffusion for versatile video generation control. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers*, 1–12.

Han, X.; Liu, M.; Chen, Y.; Yu, J.; Lyu, X.; Tian, Y.; Wang, B.; Zhang, W.; and Pang, J. 2025. Re^{3}Sim: Generating High-Fidelity Simulation Data via 3D-Photorealistic Real-to-Sim for Robotic Manipulation. *arXiv preprint arXiv:2502.08645*.

Hanselmann, N.; Renz, K.; Chitta, K.; Bhattacharyya, A.; and Geiger, A. 2022. King: Generating safety-critical driving scenarios for robust imitation via kinematics gradients. In *European Conference on Computer Vision*, 335–352. Springer.

Hore, A.; and Ziou, D. 2010. Image quality metrics: PSNR vs. SSIM. In *2010 20th international conference on pattern recognition*, 2366–2369. IEEE.

Huang, J.; Gojcic, Z.; Atzmon, M.; Litany, O.; Fidler, S.; and Williams, F. 2023. Neural kernel surface reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4369–4379.

Huang, N.; Wei, X.; Zheng, W.; An, P.; Lu, M.; Zhan, W.; Tomizuka, M.; Keutzer, K.; and Zhang, S. 2024a. S3Gaussian: Self-Supervised Street Gaussians for Autonomous Driving. *arXiv preprint arXiv:2405.20323*.

Huang, Z.; He, Y.; Yu, J.; Zhang, F.; Si, C.; Jiang, Y.; Zhang, Y.; Wu, T.; Jin, Q.; Chanpaisit, N.; et al. 2024b. Vbench: Comprehensive benchmark suite for video generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 21807–21818.

Kim, M. J.; Pertsch, K.; Karamcheti, S.; Xiao, T.; Balakrishna, A.; Nair, S.; Rafailov, R.; Foster, E.; Lam, G.; Sanketi, P.; et al. 2024. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*.

Kirillov, A.; Mintun, E.; Ravi, N.; Mao, H.; Rolland, C.; Gustafson, L.; Xiao, T.; Whitehead, S.; Berg, A. C.; Lo, W.-Y.; et al. 2023. Segment anything. In *Proceedings of the IEEE/CVF international conference on computer vision*, 4015–4026.

LaValle, S. 1998. Rapidly-exploring random trees: A new tool for path planning. *Research Report 9811*.

Li, Y.; Wei, X.; Chi, X.; Li, Y.; Zhao, Z.; Wang, H.; Ma, N.; Lu, M.; and Zhang, S. 2025. ManipDreamer: Boosting Robotic Manipulation World Model with Action Tree and Visual Guidance. *arXiv preprint arXiv:2504.16464*.

Ma, W.-D. K.; Lewis, J. P.; and Kleijn, W. B. 2024. Trailblazer: Trajectory control for diffusion-based video generation. In *SIGGRAPH Asia 2024 Conference Papers*, 1–11.

Mandi, Z.; Bharadhwaj, H.; Moens, V.; Song, S.; Rajeswaran, A.; and Kumar, V. 2022. Cacti: A framework

for scalable multi-task multi-scene visual imitation learning. *arXiv preprint arXiv:2212.05711*.

Park, Y.; Bhatia, J. S.; Ankile, L.; and Agrawal, P. 2024. Dexhub and dart: Towards internet scale robot data collection. *arXiv preprint arXiv:2411.02214*.

Ratliff, N.; Zucker, M.; Bagnell, J. A.; and Srinivasa, S. 2009. CHOMP: Gradient optimization techniques for efficient motion planning. In *2009 IEEE international conference on robotics and automation*, 489–494. IEEE.

Schmidt, L.; Santurkar, S.; Tsipras, D.; Talwar, K.; and Madry, A. 2018. Adversarially robust generalization requires more data. *Advances in neural information processing systems*, 31.

Unterthiner, T.; Van Steenkiste, S.; Kurach, K.; Marinier, R.; Michalski, M.; and Gelly, S. 2018. Towards accurate generative models of video: A new metric & challenges. *arXiv preprint arXiv:1812.01717*.

Wang, B.; Sridhar, N.; Feng, C.; Van der Merwe, M.; Fishman, A.; Fazeli, N.; and Park, J. J. 2025a. This&that: Language-gesture controlled video generation for robot planning. In *IEEE International Conference on Robotics & Automation (ICRA)*.

Wang, J.; Chen, M.; Karaev, N.; Vedaldi, A.; Rupprecht, C.; and Novotny, D. 2025b. Vggt: Visual geometry grounded transformer. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 5294–5306.

Wang, Z.; Bovik, A. C.; Sheikh, H. R.; and Simoncelli, E. P. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4): 600–612.

Wang, Z.; Li, M.; Wu, M.; Moens, M.-F.; and Tuytelaars, T. 2025c. Instruction-guided path planning with 3D semantic maps for vision-language navigation. *Neurocomputing*, 625: 129457.

Wang, Z.; Yuan, Z.; Wang, X.; Li, Y.; Chen, T.; Xia, M.; Luo, P.; and Shan, Y. 2024. Motionctrl: A unified and flexible motion controller for video generation. In *ACM SIGGRAPH 2024 Conference Papers*, 1–11.

Wei, X.; Wuwu, Q.; Zhao, Z.; Wu, Z.; Huang, N.; Lu, M.; Ma, N.; and Zhang, S. 2024. Emd: Explicit motion modeling for high-quality street gaussian splatting. *arXiv preprint arXiv:2411.15582*.

Wei, X.; Zhang, X.; Wang, H.; Wuwu, Q.; Lu, M.; Zheng, W.; and Zhang, S. 2025. OmniIndoor3D: Comprehensive Indoor 3D Reconstruction. *arXiv preprint arXiv:2505.20610*.

Wu, W.; Li, Z.; Gu, Y.; Zhao, R.; He, Y.; Zhang, D. J.; Shou, M. Z.; Li, Y.; Gao, T.; and Zhang, D. 2024. Draganything: Motion control for anything using entity representation. In *European Conference on Computer Vision*, 331–348. Springer.

Xu, H.; and Mannor, S. 2012. Robustness and generalization. *Machine learning*, 86(3): 391–423.

Yang, S.; Yu, W.; Zeng, J.; Lv, J.; Ren, K.; Lu, C.; Lin, D.; and Pang, J. 2025a. Novel demonstration generation with gaussian splatting enables robust one-shot manipulation. *arXiv preprint arXiv:2504.13175*.

Yang, X.; Li, B.; Xu, S.; Wang, N.; Ye, C.; Zhaoxi, C.; Qin, M.; Yikang, D.; Jin, X.; Zhao, H.; and Zhao, H. 2025b. ORV: 4D Occupancy-centric Robot Video Generation. *arXiv preprint arXiv:2506.03079*.

Yang, Z.; Teng, J.; Zheng, W.; Ding, M.; Huang, S.; Xu, J.; Yang, Y.; Hong, W.; Zhang, X.; Feng, G.; et al. 2024. Cogvideox: Text-to-video diffusion models with an expert transformer. *arXiv preprint arXiv:2408.06072*.

Yaseen, M. 2024. What is YOLOv8: An in-depth exploration of the internal features of the next-generation object detector. arXiv 2024. *arXiv preprint arXiv:2408.15857*.

Zhang, J.; Wang, K.; Wang, S.; Li, M.; Liu, H.; Wei, S.; Wang, Z.; Zhang, Z.; and Wang, H. 2024. Uni-navid: A video-based vision-language-action model for unifying embodied navigation tasks. *arXiv preprint arXiv:2412.06224*.

Zhou, G.; Hong, Y.; and Wu, Q. 2024. Navgpt: Explicit reasoning in vision-and-language navigation with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 7641–7649.

Zhou, S.; Du, Y.; Chen, J.; Li, Y.; Yeung, D.-Y.; and Gan, C. 2024. Robodreamer: Learning compositional world models for robot imagination. *arXiv preprint arXiv:2404.12377*.

Zitkovich, B.; Yu, T.; Xu, S.; Xu, P.; Xiao, T.; Xia, F.; Wu, J.; Wohlhart, P.; Welker, S.; Wahid, A.; Vuong, Q.; Vanhoucke, V.; Tran, H.; Soricut, R.; Singh, A.; Singh, J.; Sermanet, P.; Sanketi, P. R.; Salazar, G.; Ryoo, M. S.; Reymann, K.; Rao, K.; Pertsch, K.; Mordatch, I.; Michalewski, H.; Lu, Y.; Levine, S.; Lee, L.; Lee, T.-W. E.; Leal, I.; Kuang, Y.; Kalashnikov, D.; Julian, R.; Joshi, N. J.; Irpan, A.; Ichter, B.; Hsu, J.; Herzog, A.; Hausman, K.; Gopalakrishnan, K.; Fu, C.; Florence, P.; Finn, C.; Dubey, K. A.; Driess, D.; Ding, T.; Choromanski, K. M.; Chen, X.; Chebotar, Y.; Carbajal, J.; Brown, N.; Brohan, A.; Arenas, M. G.; and Han, K. 2023. RT-2: Vision-Language-Action Models Transfer Web Knowledge to Robotic Control. In Tan, J.; Toussaint, M.; and Darvish, K., eds., *Proceedings of The 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, 2165–2183. PMLR.