2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

DRAW2ACT: Turning Depth-Encoded Trajectories into Robotic Demonstration Videos

Yang Bai[1,2,5], Liudi Yang[3,5], George Eskandar[5†], Fengyi Shen[4,5],
Mohammad Altillawi[5], Ziyuan Liu[5], Gitta Kutyniok[1,2]
[1]Ludwig-Maximilians-Universität München (LMU Munich), [2]Munich Center for Machine Learning (MCML)
[3]University of Freiburg, [4]Technical University of Munich, [5]Huawei Heisenberg Research Center (Munich)

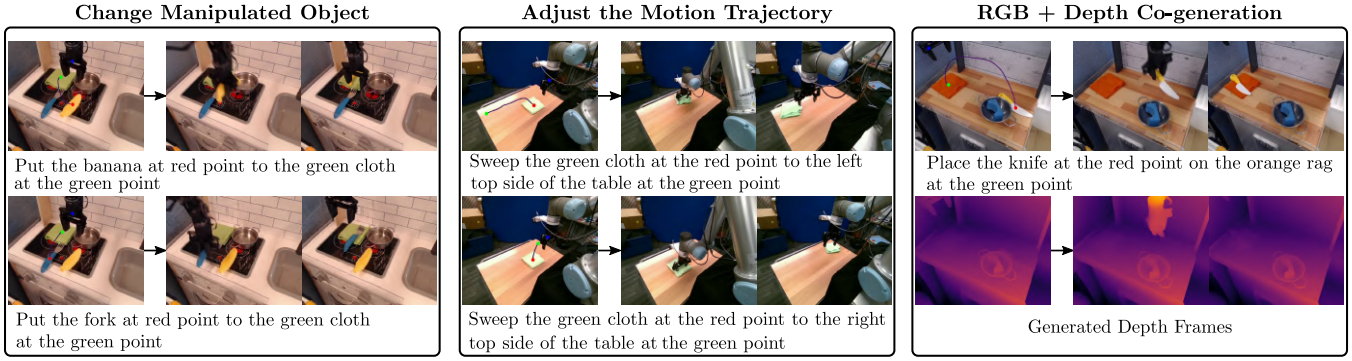

Fig. 1: Capabilities of our proposed model, DRAW2ACT, which supports a variety of features, including changing the manipulated object, adjusting motion trajectories, and co-generating RGB and depth videos.

***Abstract*— Video diffusion models provide powerful real-world simulators for embodied AI but remain limited in controllability for robotic manipulation. Recent works on trajectory-conditioned video generation address this gap but often rely on 2D trajectories or single modality conditioning, which restricts their ability to produce controllable and consistent robotic demonstrations. We present DRAW2ACT, a depth-aware trajectory-conditioned video generation framework that extracts multiple orthogonal representations from the input trajectory, capturing depth, semantics, shape and motion, and injects them into the diffusion model. Moreover, we propose to jointly generate spatially aligned RGB and depth videos, leveraging cross-modality attention mechanisms and depth supervision to enhance the spatio-temporal consistency. Finally, we introduce a multimodal policy model conditioned on the generated RGB and depth sequences to regress the robot's joint angles. Experiments on Bridge V2, Berkeley Autolab, and simulation benchmarks show that DRAW2ACT achieves superior visual fidelity and consistency while yielding higher manipulation success rates compared to existing baselines.**

I. INTRODUCTION

The growing interest in embodied intelligence has catalyzed extensive research on video-based learning methods for robotic arm training [1]–[7]. Recent frameworks such as Vid2Robot [8] have shown that policies can be learned directly from videos. However, collecting large-scale datasets with real robots is costly and often requires teleoperation or human supervision to ensure safe manipulation. Recently, video diffusion models [9] have shown an excellent ability to synthesize robotic demonstrations, thereby enriching training data and enabling the development of more capable video-based policy models [10], [11] and/or Vision-Language-Action models (VLAs) [12], [13]. Conventional video generation models typically rely on natural language prompts as conditions [14], [15]. While effective for general-purpose video synthesis, language alone often lacks the precision needed to describe detailed robotic motions or manipulation trajectories. In robotic video generation, fine-grained controllability is crucial to ensure alignment with physical constraints and task requirements. Therefore, multimodal inputs such as sketches, object start/end points, or explicit motion trajectories are needed to guide the generation process and achieve more accurate and controllable video synthesis.

Recent works such as Tora [16], DragAnything [17], and FreeTraj [18] explore 2D trajectory-conditioned video generation by either introducing motion encoders, conditioning on the VAE latents of displaced objects, or modifying the initial noise sampling. However, robotic arm motion is inherently defined in 3D space, making purely 2D guidance insufficient. Incorporating depth information can improve controllability and better handle occlusions and disocclusions in image space. Methods like LeviTor [19] take a step toward integrating 3D trajectory information, but when applied to robotic demonstrations, they often lack explicit action control, resulting in inconsistent interactions between the arm and objects. Achieving precise 3D trajectory control in the generated video while preserving object and interaction consistency thus remains an open challenge for current video generation models. It requires a more holistic approach that captures all relevant aspects of the trajectory.

Contributions: To address these limitations, we propose DRAW2ACT, a novel framework that encodes multiple complementary trajectory representations inside a video diffusion model, enhancing its trajectory-following capabilities. Instead of relying one one trajectory representation like previous works, we fuse depth-encoded 3D trajectories, high-

level semantic DINOv2 object features and pixel-coordinates augmented instruction prompts. Moreover, our model co-generates RGB and depth videos. Our insight is that adding multimodal generation capabilities to the model improves the video consistency through depth supervision. Finally, we propose a multimodal policy model conditioned on the generated video and depth sequence, enabling the regression of joint control commands of robotic arms. In summary, our contributions are:

1) We propose a trajectory-conditioned video diffusion model that integrates complementary trajectory representations into the diffusion transformer (DiT), namely: depth-aware 3D trajectories, high-level semantic features, and coordinate augmented text captions.
2) We introduce an object-centric feature control method, which extracts DINOv2 features of the manipulated object from the first frame and propagates them along the corresponding trajectory in subsequent frames. This encoding helps capture shape, semantics, and motion information, and is further injected inside the DiT.
3) We co-generate depth sequences alongside RGB videos, using depth supervision to enforce constraints and improve the quality of the generated videos. Moreover, we propose a multimodal policy model (RGB + Depth videos) for robotic manipulation, enabling more consistent learning.
4) Our experiments show that our framework produces better video quality and higher task success rates than previous trajectory-conditioned video methods, demonstrating its effectiveness for real-world robotic learning.

II. RELATED WORK

A. Video Diffusion for Robotics

Video generation [20]–[24] has recently emerged as a promising tool for robotic manipulation, serving as a world model, simulator, and data augmentation source. Early studies used generated videos to predict future robot actions and states, enabling policy learning [1], [2]. Other works incorporate gestures, text instructions, or robot end-effector's 3D trajectory-based control [25], which relies on precise motion data that is often difficult to obtain in practice. Large-scale synthetic datasets such as UniPi [9] and UniSim [26] have been proposed to support scalable robot learning, while model-based approaches like DreamerV3 [27] and DayDreamer [28] further combine latent world models with inverse dynamics to predict rewards and generate optimal actions. Despite these advances, many existing methods still face challenges in achieving fine-grained controllability.

In contrast, our method uses object-level trajectories, which are easier to acquire and allow explicit control over both robot arms and objects, providing more precise and interpretable manipulation videos.

B. Trajectory Control in Video Generation

Recent advances in controllable video generation have sought to improve motion precision while retaining the generative power of diffusion models. Early approaches employed optical flow or recurrent networks to encode object trajectories [16], [17], but achieved only coarse control. Subsequent methods introduced bounding-box representations (e.g., TrailBlazer [29]) and trajectory vector maps (e.g., MotionCtrl [30], DragNUWA [31]) for instance-level guidance, while entity-aware methods such as DragAnything exploited segmentation masks for higher-fidelity motion. To address the limitations of 2D control, approaches like 3DTrajMaster [32] and LeviTor [19] incorporated depth cues and 6-DoF pose sequences to enable more realistic object manipulation. Multi-object trajectory control has also been explored for better coordination in complex scenes.

Both training-based and training-free paradigms have been investigated. Methods such as Tune-A-Video [33] and MotionDirector [34] leverage reference videos to generalize motion patterns, whereas training-free approaches like Peekaboo [35], TrailBlazer [29], and DragAnything [17] use attention modulation or explicit trajectory guidance during inference, reducing annotation costs. Yet both methods face challenges when applied to robot demonstrations, particularly in maintaining consistency during robot-object interactions.

Building on these insights, our work trains a robotic arm video generation model with several explicit and implicit trajectories representations (depth, high-level semantics, corrdiantes) to achieve fine-grained controllable motion.

III. METHOD

This section introduces the key components and functionality of DRAW2ACT. As shown in Fig. 2, DRAW2ACT supports two output modalities, RGB and depth videos and incorporates three forms of control signals: depth-awared trajectories, object-level DINOv2 features and text prompt augmented with pixel coordinates. We first review our Task Formulation (Sec. III-A), followed by the details of our Representation Encoding pipeline (Sec. III-B) our Control Injection Module (Sec. III-C), and Multimodal Generation (Sec. III-D) and Multimodal Policy Model (Sec. III-E).

A. Task Formulation and Model Overview

Task Formulation: Our goal is to synthesize a realistic manipulation video with N frames, $V \in \mathbb{R}^{N\times 3\times H\times W}$, given an initial frame $I \in \mathbb{R}^{3\times H\times W}$ containing task-relevant objects, a binary mask $M \in \mathbb{R}^{H\times W}$ specifying the manipulated object, a user-defined textual prompt c, and an object trajectory sequence $q = \{(x_i, y_i, d_i)\}_{i=0}^{N-1}$, where x and y are pixel coordinates relative to the image origin and d represents the relative depth value at each frame i. The trajectory sequence q explicitly defines the manipulated object's motion from a starting point (e.g. for picking) to a target location (e.g. for placing), providing direct control over the task.

Model Overview: We design our model as a conditional multimodal Latent Diffusion Model (LDM). We encode different representations of the input trajectory, capturing different but complementary aspects of the video to be generated: depth, motion, object semantics and shape. Namely, we extract three representations: $\mathcal{D} = \{z_0^{\text{ref}}, y_{dino}, y_c\}$, where:

- z_0^{ref} is the latent representation of the first video frame with the color-coded depth-aware trajectory.

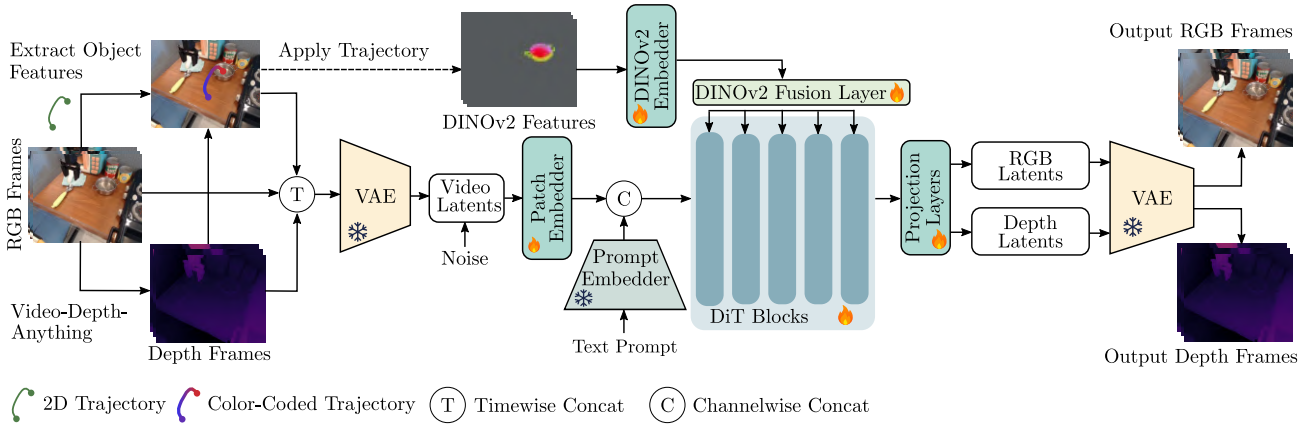

Fig. 2: Model Architecture. We extract several representations from the trajectory (DINOv2 object-level features, pixel coordinates augmented prompt and reference image with the depth-aware trajectory overlaid. DINOv2 features are injected into each block through DINOv2 fusion Layers. To enable multimodal output generation, we concatenate the reference frame, RGB frames, and depth frames in the time dimension, and process them with the same patch embedding module.

- y_{dino} represents the sequence of DINOv2 features of the manipulated object, propagated along its trajectory.
- y_c is the text prompt augmented with detailed pixel-level positional information derived from the trajectory.

To encode the videos into a latent space, we employ a pre-trained open-source 3D causal variational autoencoder (VAE) $\mathcal{E}$. For a video $V = \{I_i\}_{i=0}^{N-1}$ of dimension $\mathbb{R}^{3\times N\times H\times W}$, where I_i denotes i^{th} image frame, N represents the number of frames, and H and W are the height and width of each frames, the VAE outputs a latent feature z of dimension $\mathbb{R}^{16\times n\times h\times w}$, where h and w refer to the height and width in the latent space and n represents the number of frames after the VAE's temporal compression.

We thus train the denoising network $\epsilon_\theta(\cdot, t)$, implemented as a DiT, to learn to progressively remove the added noise and reconstruct the video sequence given the set of input conditions. The training objective is:

$$\mathcal{L}_{\text{diffusion}} = \mathbb{E}_{z,\mathcal{D},\epsilon\sim\mathcal{N}(0,1),t} \left\|\epsilon - \epsilon_\theta(z_t, \mathcal{D}, t)\right\|_2^2$$

B. Encoding Trajectory Representations

To obtain the trajectory of the manipulated object, we first identify the object name from the caption and extract its binary mask in each frame using Grounded-SAM [36] and TrackAnything [37] (see Fig. 3). Given the sequence of object masks $M = \{M_i\}_{i=0}^{N-1}$, we compute the center of each mask to derive pixel-relative 2D coordinates $\{(x_i, y_i)\}_{i=0}^{N-1}$ forming the 2D trajectory sequence. Next, we employ the depth estimation network Video Depth Anything [38] to predict relative depth video $V_{\text{depth}} = \{I_i^{depth}\}_{i=0}^{N-1}$, $I_i^{depth} \in \mathbb{R}^{3\times H\times W}$, which matches the resolution and frame count of the original RGB video. Our approach does not require highly accurate metric depth measurements, simplifying user interaction during inference. Finally, we associate each 2D trajectory point with its corresponding depth value d_i extracted from the corresponding depth video:

$$d_i = I_i^{depth}(x_i, y_i).$$

By combining the 2D coordinates with estimated depth values, we construct depth-aware control trajectory sequences.

$$q = \{(x_i, y_i, d_i)\}_{i=0}^{N-1}.$$

Text Prompt. We provide a detailed description of the robotic arm task in the form of a textual prompt (e.g., "The robotic arm at blue point $(x_{\text{robot}}, y_{\text{robot}})$ moves to the object at red point $(x_{\text{obj}}, y_{\text{obj}})$, picks it up, and then moves to green point $(x_{\text{des}}, y_{\text{des}})$"), where all point coordinates are pixel-relative. The text prompt is encoded using the T5 model [39], producing a feature vector y_c that is incorporated into the denoising network ϵ_θ through cross-attention. The caption provides a representation for the trajectory on a coarse level.
Reference Image. We generate a reference image I_0^{ref} in the following way: we draw the object's start and end points as well as the trajectory line color-coded with the depth value on the first frame I_0. I_0^{ref} is then encoded by a VAE to obtain latent feature representations $z_0^{\text{ref}} \in \mathbb{R}^{16\times 1\times h\times w}$. The latent feature z_0^{ref} is then concatenated time-wise with z, and then fed into the denoising network ϵ_θ. This reference-frame latent feature encodes depth and motion providing a conditioning signal that propagates to other frames through self-attention.
DINOv2 Object Features. To provide object-centric semantic and motion information, we use a sequence of DINOv2 features as a key condition. First, we extract the manipulated object region from the initial frame I_0 using its mask M_0. To maximize the retention of object features and minimize background influence, we crop this region using the bounding box size and extract DINOv2 features from it. The DINOv2 model outputs features from its last layer that are rich in high-dimensional semantic information, which is essential for accurate object modeling.

We process the extracted object DINOv2 features on a blank background and resize them to match the latent space dimensions. This preserves object information while suppressing background noise. We then propagate these features along the object's motion trajectory q, placing them at their corresponding locations in each frame to create a feature

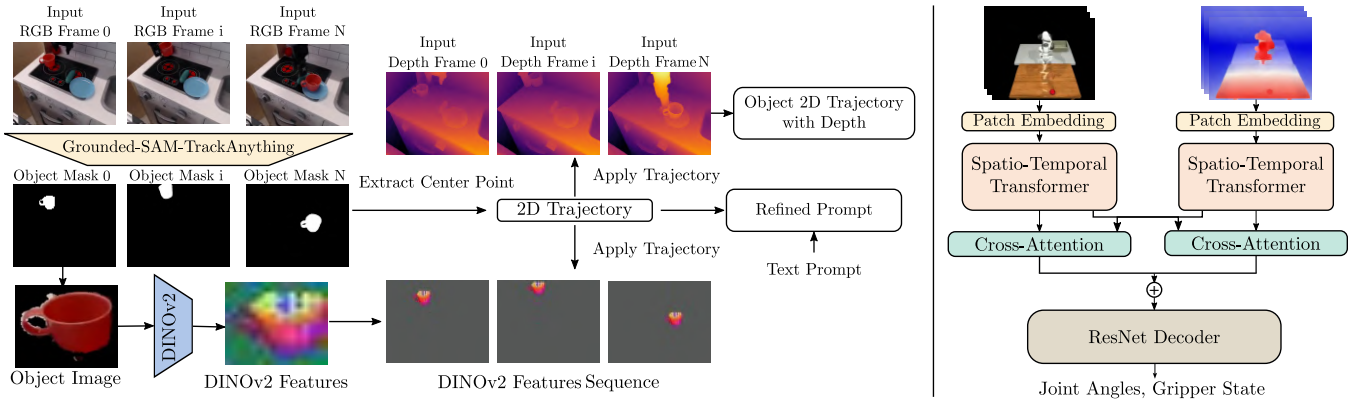

Fig. 3: Left: Representation Encoding. We first extract object masks M using Grounded-SAM-TrackAnything to obtain the 2D trajectory of the manipulated object and augment it with relative depth estimates. In parallel, we extract DINOv2 features from the object and align them to the trajectory coordinates. We augment the text prompt with the start and end positions. Right: utilize modal Policy Model to predict robot joint angles and gripper state

map of the same size as the video latent. This representation encodes the object's semantics, shape, and motion.

Finally, we apply temporal interpolation to the DINOv2 features, compressing the time dimension to match that of the latent representation. The final feature has the same temporal and spatial dimensions as the latent space after the VAE process $y_{dino} \in \mathbb{R}^{1024\times n\times h\times w}$.

C. Control Injection Module

We design a DINOv2 patch embedding module that first performs spatio-temporal downsampling on the input y_{dino} to match the dimensions of the latents features after patch embedding $y_{dino} \in \mathbb{R}^{C\times(n/2\times h/2\times w/2)}$.

Then, we inject the patchified DINOv2 features into the DiT backbone using a specialized Fusion Block. This module employs a gating mechanism to selectively modulate the DINOv2 features, followed by LayerNorm to stabilize the output. The modulated features are then residually integrated with the Transformer's hidden states. This allows for the progressive fusion of the DINOv2 features into each DiT block while preserving the integrity of the original hidden state. The fusion process is defined by:

$$h' = h + \text{LayerNorm}(y_{dino} \odot G) \odot (y_{dino} \odot G)$$

$$G = \sigma(W_g y_{dino} + b_g)$$

where h denotes the Transformer hidden states, σ is the Sigmoid activation function, $\odot$ denotes element-wise multiplication, W_g and b_g are the learnable weights and biases of the gate, G is the gating mask, and h' is the fused output.

D. Multimodal Generation

We incorporate the depth video, generated by Video Depth Anything, as an additional modality to leverage its rich spatial information (as illustrated in Fig. 2). By simultaneously generating both depth and RGB frames, the model is able to produce robust spatial information while preserving the scene's contextual structure. During training, the model applies self-attention across both modalities, enabling it to jointly capture and leverage the complementary information from the RGB and depth video streams.

To encode the depth information, we use the same VAE we use for the RGB frames. The two modalities are concatenated along the temporal dimension at the input, rather than along the channel dimension. In this setting, the input is effectively a single long sequence formed by concatenating the RGB and depth videos in time. This approach has the advantage of avoiding the need for additional embedding layers and separate convolution projection layers to predict independent depth noise, which reduces training complexity. Consequently, the final denoiser is formulated as:

$$D_\theta\big((z\| z_{\text{depth}}); \sigma, \{z_0^{ref}, y_{dino}, y_c\}\big).$$

E. Multimodal Policy Model

We propose an architecture see right part of Fig. 3 to estimate robot joint states from RGB and depth videos. RGB and depth videos are first encoded using VAE to obtain latent representations, which are then processed separately through their respective patch embedding. Each latent feature is first processed through a spatial transformer and then a temporal transformer to capture spatial and temporal dependencies. The two features then interact through a cross-attention block, exchanging information, and are finally summed before being decoded through a ResNet-based decoder to predict the robot's gripper state and joint angles.

IV. EXPERIMENTS AND RESULTS

To thoroughly evaluate the effectiveness of fine-grained control through trajectories and DINOv2 features, we conducted extensive experiments on multiple datasets and diffusion base model. Sec. IV-A details the experimental setup, and Sec. IV-B presents the results and comparative analysis, then we do the ablation study in Sec. IV-C

A. Experimental setup

Datasets: We used four robotic arm datasets, including two publicly available datasets: BridgeDataV2(WidowX Robot) [40] and Berkeley Autolab(UR5) [41]. The third dataset was

Dataset	Method	Vbench Evaluation				Trajectory Deviation	Task Evaluation
		Mot.Cons.↑	Bg.Cons.↑	Subj.Cons.↑	Tem.Fli.↑	Object Traj. Error ↓	Success Rate(%)↑
Bridge V2	LeviTor	0.9712	0.9289	0.9272	0.9817	46.52	N/A
	Tora	0.9875	0.9507	0.9346	0.9821	35.67	N/A
	MotionCtrl	0.9792	0.9471	0.9317	0.9811	38.24	N/A
	DragAnything	0.9810	0.9442	0.9289	0.9832	37.11	N/A
	Ours	**0.9891**	**0.9512**	**0.9383**	**0.9849**	**25.30**	N/A
Berkeley UR5	LeviTor	0.9803	0.9436	0.9325	0.9761	47.29	N/A
	Tora	0.9818	0.9502	0.9410	**0.9833**	35.73	N/A
	MotionCtrl	0.9844	0.9472	0.9391	0.9761	37.91	N/A
	DragAnything	0.9827	0.9488	0.9402	0.9761	39.76	N/A
	Ours	**0.9845**	**0.9509**	**0.9417**	**0.9833**	**22.37**	N/A
Simulator	LeviTor	0.9711	0.9394	0.9381	0.9819	33.52	0.0
	Tora	0.9844	0.9441	0.9452	0.9803	35.44	36.8
	MotionCtrl	0.9832	0.9387	0.9392	0.9813	29.83	29.6
	DragAnything	0.9811	0.9424	0.9433	0.9811	30.27	31.2
	Ours	**0.9865**	**0.9473**	**0.9495**	**0.9821**	**19.88**	**65.2**

TABLE I: Main comparison of trajectory-conditioned video generation models. Bold numbers indicate the best performance.

Dataset	Method	Vbench Evaluation				Depth Deviation				Trajectory Deviation
		Mot.Smth..↑	Bg.Cons.↑	Subj.Cons.↑	Tem.Fli.↑	LPIPS ↓	SSIM ↑	PSNR ↑	FVD↓	Object Traj. Error ↓
Bridge V2	first frame (RGB)	0.9777	0.9183	0.9155	0.9817	N/A	N/A	N/A	N/A	39.88
	first frame (RGB and Depth)	0.9813	0.9202	0.9164	0.9821	0.3480	0.7022	15.33	222.7	45.27
	point frame	0.9892	0.9221	0.9268	**0.9849**	0.3449	0.6822	15.12	196.6	36.41
	2D traj. frame	0.9864	0.9369	0.9373	0.9804	0.3485	0.6643	14.98	181.8	29.67
	3D(depth) traj. frame	0.9887	0.9509	0.9377	0.9845	0.3199	0.7305	15.86	**169.4**	26.81
	3D(depth) traj.&Dino	**0.9891**	**0.9512**	**0.9383**	**0.9849**	**0.3152**	**0.7326**	**15.91**	177.1	**25.30**
Simulator	first frame (RGB)	0.9832	0.9018	0.9082	0.9818	N/A	N/A	N/A	N/A	61.32
	first frame (RGB and Depth)	0.9854	0.9125	0.9091	0.9820	0.3972	0.4888	7.230	174.6	69.17
	point frame	0.9855	0.9226	0.9130	0.9816	0.3959	0.4906	7.331	163.05	26.83
	2D traj. frame	0.9852	0.9253	0.9317	0.9814	0.3889	0.4920	7.329	164.39	22.17
	3D(depth) traj. frame	0.9851	0.9357	0.9396	0.9811	0.3889	**0.4981**	**7.350**	159.45	21.39
	3D(depth) traj.&Dino	**0.9865**	**0.9473**	**0.9495**	**0.9821**	**0.3886**	**0.4981**	7.347	**158.44**	**19.88**

TABLE II: Ablation study on different different types of input condition.

generated in the MuJoCo simulator with a Franka Panda robotic arm. In total, we curated approximately 50.6K videos. All datasets were preprocessed following the procedure outlined in Method Sect. III.

Baseline Models: To evaluate the effectiveness of our method, we compare it with several state-of-the-art trajectory-controlled video generation approaches: LeviTor [19], Tora [16], MotionCtrl [30], DragAnything [17]. For fair comparison, all baselines are retrained on the same datasets with their respective optimal training configurations.

Evaluation: We evaluate our approach on a total of 300 test samples, consisting of 100 samples from each dataset. The evaluation set covers diverse environments, robotic arms, manipulation instructions, and object types. Our evaluation is based on three criteria:

1) **Video Quality**: We use the following quantitative metrics for automatic evaluation: Motion Smoothness(Mot. Smth.), I2V Background consistency (Bg. Cons.), I2V Subject Consistency(Subj. Cons), Temporal flickering (Tem. Fli.). The metrics are calculated using VBench-2.0 [42] to assess the overall quality of the videos.
2) **Trajectory Accuracy**: We measure object trajectory by computing the Trajectory Error (Object Traj. Error), defined as the mean L1 distance between the input trajectories and those extracted from generated videos.
3) **Task Success Rate**: While not the focus of our work, we measure the efficiency of the multimodal policy model compared to the RGB only policy and the existing baselines. The RGB only policy model is used to evaluate task success rates for baselines on simulated datasets, as they do not generate depth videos. Furthermore, we compare task success rates when regressing trajectories from our generated videos using both RGB and depth versus only RGB.
4) **Depth Video Quality**:We report LPIPS, SSIM, PSNR and FVD by comparing our generated depth videos with depth maps inferred from the generated RGB videos using Video Depth Anything.

Implementation details: Our conditional video diffusion model is implemented based on the pre-trained CogVideoX-Fun-5B architecture [43]. We use a $69 \times 320 \times 320$ resolution video for both training and inference. The model is trained with the AdamW optimizer on 8 NVIDIA A800 GPUs for approximately 3 days, over a total of 50K training steps.

B. Qualitative and Quantitative Evaluation

Quantitative evaluation: Our model outperforms all baselines across all evaluation metrics, including video quality, trajectory accuracy, and task success rate on the simulation dataset(see Tab. I). LeviTor, due to the lack of prompt-based control, fails to drive interactions between the robot arm and objects. Other methods can complete the tasks, but deformations occurring during object–robot interactions degrade video quality, resulting in low success rates.

Qualitative evaluation: As shown in Fig. 4, our model produces the best visual quality among all methods. Competing approaches often struggle to move objects correctly along the intended trajectories and exhibit issues such as

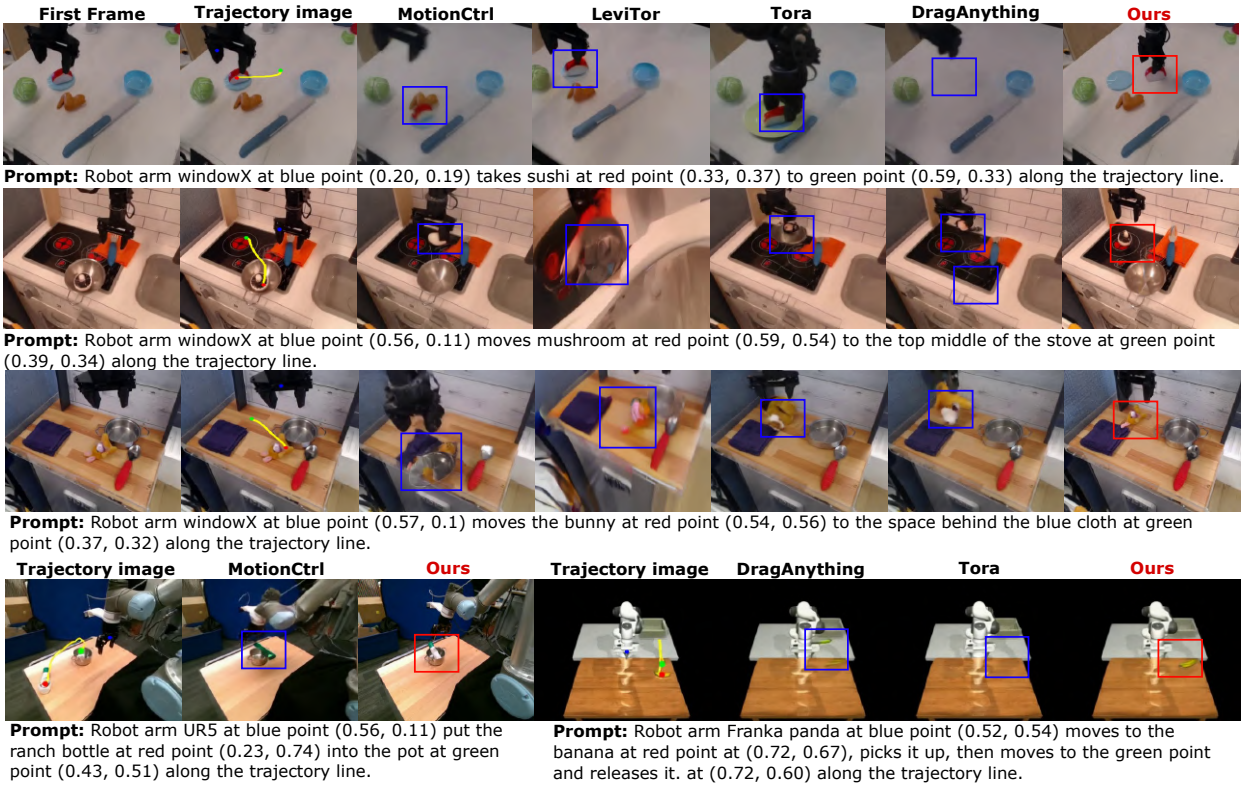

Fig. 4: Qualitative comparison of trajectory-conditioned video generation approaches.

Policy Model	Condition	Success Rate(%)↑
RGB	First Frame RGB	15.4
RGB & Depth	First Frame Multimodal	21.0
	Point Frame	35.6
	2D Trajectory Image	52.4
	3D Trajectory Image	61.8
	3D Trajectory w/ DINOv2	**65.2**

TABLE III: Ablation study on different types of input condition's task success rate.

object disappearance, geometric distortions, or inconsistent appearances across frames. For example, in the first task of moving sushi, none of the baselines are able to follow the trajectory and move the object correctly. In the second task, while some models succeed in following the trajectory, the toy suffer from noticeable deformations. In the third task, MotionCtrl generates videos in which the bottle undergoes severe distortions. In the simulator task, TORA results in object disappearance, whereas DragAnything places the object in an incorrect location. In contrast, our method reliably follows the given trajectories while preserving object geometry, resulting in more coherent and realistic manipulation videos.

C. Ablation Study

To demonstrate the novel contributions and design choices of our method, we conduct ablation studies using different types of input conditions to the DiT: First Frame RGB (without trajectory or point information, vanilla I2V), First Frame Multimodal (I2V with both RGB and depth videos), Point Image (object start and end points), 2D Trajectory Image (adding a 2D path on the point image), 3D Trajectory Image (adding depth information), and 3D Trajectory with DINOv2 features (combining depth trajectory and DINOv2 features). From Tab. II, we can see that co-generating RGB and depth videos can help improve video quality. Moreover, using trajectory reference images further enhances video quality. Finally, depth-aware trajectories yield the best visual results and produce depth videos that are closer to those inferred by Video Depth Anything, highlighting the importance of depth information for fidelity.

We can see from Fig. 5, inputs based on the first frame or points provide coarse control, making task completion difficult. For example, when executing the first task, conditioning on the first frame causes the robot arm to mistakenly manipulate the yellow cloth instead of the can, and conditioning on the point image also results in the object failing to move as expected. Similarly, As shown in Fig. 6, we observe that using the first frame does not move the bowl to the designated position, while using the point frame leads to severe deformation of both the robot arm and the object. As shown in Fig. 5, while 2D and depth trajectories improve control precision, they still suffer from object inconsistency, such as appearance changes in the can and varying the shape of the bowl. In contrast, our final method—combining depth trajectories with DINOv2 features—not only follows the intended trajectories but also preserves object consistency, thereby achieving best video quality and trajectory accuracy. Task success rates are reported in Tab. III, showing that using RGB and depth videos for trajectory regression outperforms using RGB only. Furthermore, the 3D trajectory with DINOv2 features achieves the highest video quality, and the highest task success rate.

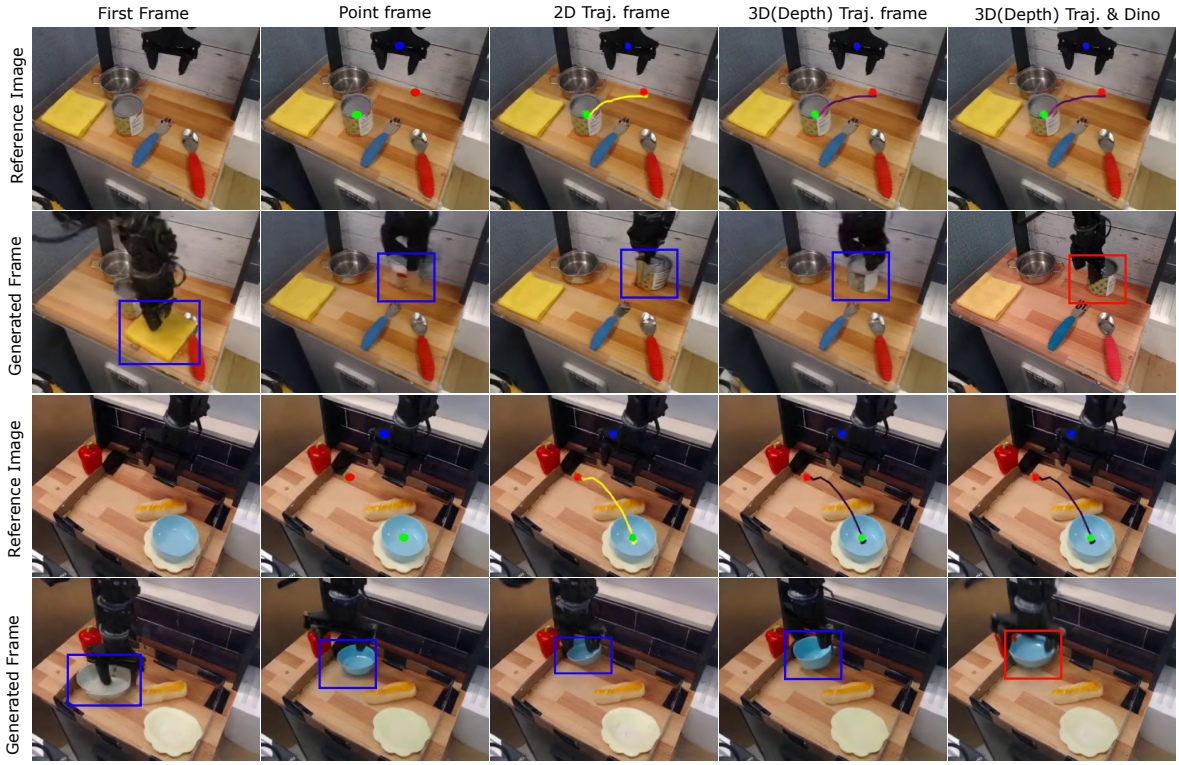

Fig. 5: Qualitative Ablation study against different control approaches on the Bridge dataset

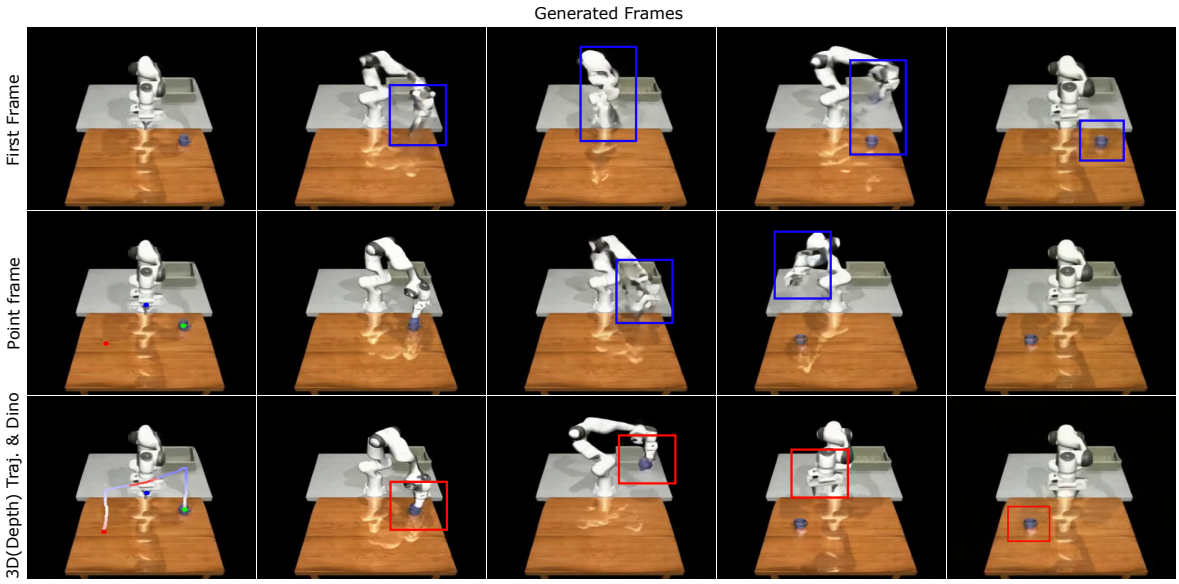

Fig. 6: Qualitative Ablation study against different control approaches on the Simulation dataset

V. CONCLUSIONS

We propose DRAW2ACT, a depth-aware trajectory-conditioned video generation framework for robotic manipulation. Our approach advances three fronts: (1) depth-aware 3D trajectories enable precise control of both robot arms and manipulated objects; (2) object-centric DINOv2 features enhance geometric and motion consistency, avoiding trajectory drift, object disappearance, and distortion observed in baselines; (3) RGB and depth videos co-generation improves visual quality and provides multimodal data that boosts downstream policy performance, as confirmed by ablation studies. Together, these contributions yield more accurate, coherent, and controllable robotic manipulation videos, validated across open-source benchmarks and simulator datasets.

While promising, challenges remain that model does not support long-horizon controllability. Moreover, we currently support the controllability of only one object. Future works can address generation conditioned on multiple trajectories in parallel or sequentially to enable multi-object manipulation.

REFERENCES

[1] Y. Liu, A. Gupta, P. Abbeel, and S. Levine, "Imitation from observation: Learning to imitate behaviors from raw video via context translation," in *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2018, pp. 1118–1125.

[2] H. Xiong, Q. Li, Y.-C. Chen, H. Bharadhwaj, S. Sinha, and A. Garg, "Learning by watching: Physical imitation of manipulation skills from human videos," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 7827–7834.

[3] L. Smith, N. Dhawan, M. Zhang, P. Abbeel, and S. Levine, "Avid: Learning multi-stage tasks via pixel-level translation of human videos," *arXiv preprint arXiv:1912.04443*, 2019.

[4] P. Sharma, D. Pathak, and A. Gupta, "Third-person visual imitation learning via decoupled hierarchical controller," in *NeurIPS*, 2019.

[5] K. Schmeckpeper, O. Rybkin, K. Daniilidis, S. Levine, and C. Finn, "Reinforcement learning with videos: Combining offline observations with interaction," *Conference on Robot Learning*, 2020.

[6] M. Xu, Z. Xu, C. Chi, M. Veloso, and S. Song, "Xskill: Cross embodiment skill discovery," in *Conference on Robot Learning*. PMLR, 2023, pp. 3536–3555.

[7] L. Yang, Y. Bai, G. Eskandar, F. Shen, M. Altillawi, D. Chen, S. Majumder, Z. Liu, G. Kutyniok, and A. Valada, "Roboenvision: A long-horizon video generation model for multi-task robot manipulation," *arXiv preprint arXiv:2506.22007*, 2025.

[8] V. Jain, M. Attarian, N. J. Joshi, Wahid, and Others, "Vid2Robot: End-to-end Video-conditioned Policy Learning with Cross-Attention Transformers," in *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, July 2024.

[9] Y. Du, M. Yang, B. Dai, H. Dai, O. Nachum, J. B. Tenenbaum, D. Schuurmans, and P. Abbeel, "Learning universal policies via text-guided video generation," *arXiv e-prints*, pp. arXiv–2302, 2023.

[10] C. Chi, S. Feng, Y. Du, Z. Xu, E. Cousineau, B. Burchfiel, and S. Song, "Diffusion policy: Visuomotor policy learning via action diffusion," in *Proceedings of Robotics: Science and Systems (RSS)*, 2023.

[11] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn, "Learning fine-grained bimanual manipulation with low-cost hardware," *arXiv preprint arXiv:2304.13705*, 2023.

[12] M. Kim, K. Pertsch, S. Karamcheti, T. Xiao *et al.*, "Openvla: An open-source vision-language-action model," *arXiv preprint arXiv:2406.09246*, 2024.

[13] Black, Kevin, Brown, Noah, Driess, Danny, Esmail *et al.*, "pi_0: A vision-language-action flow model for general robot control," *arXiv preprint arXiv:2410.24164*, 2024.

[14] X. Peng, Z. Zheng, C. Shen *et al.*, "Open-sora 2.0: Training a commercial-level video generation model in 200k," *arXiv preprint arXiv:2503.09642*, 2025.

[15] T. Wan, A. Wang, B. Ai, B. Wen *et al.*, "Wan: Open and advanced large-scale video generative models," *arXiv preprint arXiv:2503.20314*, 2025.

[16] Z. Zhang, J. Liao, M. Li, Z. Dai, B. Qiu, S. Zhu, L. Qin, and W. Wang, "Tora: Trajectory-oriented diffusion transformer for video generation," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 2063–2073.

[17] W. Wu, Z. Li, Y. Gu, R. Zhao, Y. He, D. J. Zhang, M. Z. Shou, Y. Li, and D. Z. Tingting Gao, "Draganything: Motion control for anything using entity representation," 2024.

[18] H. Qiu, Z. Chen, Z. Wang, Y. He, M. Xia, and Z. Liu, "Freetraj: Tuning-free trajectory control in video diffusion models," 2024.

[19] H. Wang, H. Ouyang, Q. Wang, W. Wang, K. L. Cheng, Q. Chen, Y. Shen, and L. Wang, "Levitor: 3d trajectory oriented image-to-video synthesis," *arXiv preprint arXiv:2412.15214*, 2024.

[20] A. Blattmann, R. Rombach, H. Ling, T. Dockhorn, S. W. Kim, S. Fidler, and K. Kreis, "Align your latents: High-resolution video synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 22 563–22 575.

[21] J. Ho, W. Chan, C. Saharia, J. Whang, R. Gao, A. Gritsenko, D. P. Kingma, B. Poole, M. Norouzi, D. J. Fleet *et al.*, "Imagen video: High definition video generation with diffusion models," *arXiv preprint arXiv:2210.02303*, 2022.

[22] H. Chen, Y. Zhang, X. Cun, M. Xia, X. Wang, C. Weng, and Y. Shan, "Videocrafter2: Overcoming data limitations for high-quality video diffusion models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 7310–7320.

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

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

[25] F. Zhu, H. Wu, S. Guo, Y. Liu, C. Cheang, and T. Kong, "Irasim: Learning interactive real-robot action simulators," *arXiv preprint arXiv:2406.14540*, 2024.

[26] M. Yang, Y. Du, K. Ghasemipour, J. Tompson, D. Schuurmans, and P. Abbeel, "Learning interactive real-world simulators," *arXiv preprint arXiv:2310.06114*, 2023.

[27] D. Hafner, J. Pasukonis, J. Ba, and T. Lillicrap, "Mastering diverse domains through world models," *arXiv preprint arXiv:2301.04104*, 2023.

[28] P. Wu, A. Escontrela, D. Hafner, P. Abbeel, and K. Goldberg, "Daydreamer: World models for physical robot learning," in *Conference on robot learning*. PMLR, 2023, pp. 2226–2240.

[29] L. D. English, S. Humble, and V. E. Barnes, "Trailblazers," *Teaching children mathematics*, vol. 16, no. 7, pp. 402–409, 2010.

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

[31] S. Yin, C. Wu, J. Liang, J. Shi, H. Li, G. Ming, and N. Duan, "Dragnuwa: Fine-grained control in video generation by integrating text," *Image, and Trajectory*, 2023.

[32] X. Fu, X. Liu, X. Wang, S. Peng, M. Xia, X. Shi, Z. Yuan, P. Wan, D. Zhang, and D. Lin, "3dtrajmaster: Mastering 3d trajectory for multi-entity motion in video generation," *arXiv preprint arXiv:2412.07759*, 2024.

[33] J. Z. Wu, Y. Ge, X. Wang, S. W. Lei, Y. Gu, Y. Shi, W. Hsu, Y. Shan, X. Qie, and M. Z. Shou, "Tune-a-video: One-shot tuning of image diffusion models for text-to-video generation," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2023, pp. 7623–7633.

[34] R. Zhao, Y. Gu, J. Z. Wu, D. J. Zhang, J.-W. Liu, W. Wu, J. Keppo, and M. Z. Shou, "Motiondirector: Motion customization of text-to-video diffusion models," in *European Conference on Computer Vision*. Springer, 2024, pp. 273–290.

[35] H. Wu, Ö. Alay, A. Brunstrom, S. Ferlin, and G. Caso, "Peekaboo: Learning-based multipath scheduling for dynamic heterogeneous environments," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 10, pp. 2295–2310, 2020.

[36] T. Ren, S. Liu, A. Zeng, J. Lin, K. Li, H. Cao, J. Chen, X. Huang, Y. Chen, F. Yan *et al.*, "Grounded sam: Assembling open-world models for diverse visual tasks," *arXiv preprint arXiv:2401.14159*, 2024.

[37] J. Yang, M. Gao, Z. Li, S. Gao, F. Wang, and F. Zheng, "Track anything: Segment anything meets videos," 2023.

[38] S. Chen, H. Guo, S. Zhu, F. Zhang, Z. Huang, J. Feng, and B. Kang, "Video depth anything: Consistent depth estimation for super-long videos," *arXiv:2501.12375*, 2025.

[39] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *Journal of machine learning research*, vol. 21, no. 140, pp. 1–67, 2020.

[40] H. Walke, K. Black, A. Lee, M. J. Kim, M. Du, C. Zheng, T. Zhao, P. Hansen-Estruch, Q. Vuong, A. He, V. Myers, K. Fang, C. Finn, and S. Levine, "Bridgedata v2: A dataset for robot learning at scale," in *Conference on Robot Learning (CoRL)*, 2023.

[41] L. Y. Chen, S. Adebola, and K. Goldberg, "Berkeley UR5 demonstration dataset."

[42] Z. Huang, F. Zhang, X. Xu, Y. He, J. Yu, Z. Dong, Q. Ma, N. Chanpaisit, C. Si, Y. Jiang, Y. Wang, X. Chen, Y.-C. Chen, L. Wang, D. Lin, Y. Qiao, and Z. Liu, "Vbench++: Comprehensive and versatile benchmark suite for video generative models," *arXiv preprint arXiv:2411.13503*, 2024.

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