Robotic Manipulation by Imitating Generated Videos Without Physical Demonstrations

Anonymous CVPR submission

Paper ID *****

Abstract

This work introduces Robots Imitating Generated Videos (RIGVid), a system that enables robots to perform complex manipulation tasks—such as pouring, wiping, and mixing—purely by imitating AI-generated videos, without requiring any physical demonstrations or robot-specific training. Given a language command and an initial scene image, a video diffusion model generates potential demonstration videos, and a vision-language model (VLM) automatically filters out results that do not follow the command. A 6D pose tracker then extracts object trajectories from the video, which are retargeted to the robot in an embodiment-agnostic fashion. Through extensive real-world evaluations, we show that filtered generated videos can be as effective as real demonstrations, and that performance improves with generation quality. We also show that relying on generated videos outperforms more compact alternatives such as keypoint prediction using VLMs, and that strong 6D pose tracking outperforms other ways to extract trajectories, such as dense feature point tracking. These findings suggest that videos produced by a state-of-the-art off-the-shelf model can offer a scalable and effective source of supervision for robotic manipulation. Project page: rigvid25.github.io.

1. Introduction

Videos offer a rich and expressive source of training data for robotic manipulation, and numerous methods have successfully leveraged them for supervision. Such methods typically fall into two categories: (1) Learning from publicly available large-scale datasets of real-world videos [9, 13, 22, 36, 106, 125], and (2) Imitation of specific demonstrations captured under controlled conditions that closely match the execution setting [8, 21, 55, 65, 69, 114]. Unfortunately, both of these strategies come with challenges that limit scalability and broad deployment. Large-scale video datasets often introduce domain gaps [36, 119, 134] and require adaptation to specific robot embodiments and tasks [9, 87]. On the other hand, video-based imitation involves laborious data collection that must ensure close alignment in viewpoints, morphologies, and interaction modalities [7, 8, 26, 106].

Motivated by recent advances in video generation, we explore a potentially new paradigm: can a **single generated video**, synthesized to exactly match our input environment and task description, be used as the sole source of supervision for robotic manipulation? Recently released models like SORA [16] and Kling [1] have demonstrated impressive capabilities in producing realistic-seeming videos from language and image inputs. At the same time, it has been shown that such videos frequently suffer from distorted object geometries [73, 129], physically implausible interactions [83, 124], and unrealistic scene dynamics [11, 39]. Consequently, while the idea of synthesizing supervision is enticing, its usefulness in the robotics setting has not been convincingly established. Prior work incorporating video generation into robotics typically relies on additional supervision, such as task-specific training [30] or fine-tuning on offline robot trajectory datasets [14, 15]. By contrast, we ask whether a robot can perform real-world manipulation tasks solely by imitating generated videos—*without any additional supervision or task-specific training*.

To this end, we introduce **Robots Imitating Generated Videos (RIGVid)**, a framework that connects video generation models to real-world robotic execution. Fig. 1 shows an outline of the method. Given an input RGB-D image of the scene and a free-form language command (e.g., "pour water on the plant"), we use a state-of-the-art video diffusion model to generate a candidate video of the task being performed. The generated video may not accurately follow the language command, but we show that it is possible to use a VLM to automatically filter out unsuccessful generations. Next, we estimate per-frame depth on the video, segment the manipulated object, and track its 6D pose across the video using a pose tracker, FoundationPose [117], that requires a pre-computed object mesh. The resulting *6D object pose trajectory* serves as a high-level task representation that is retargeted to the robot for execution. Because

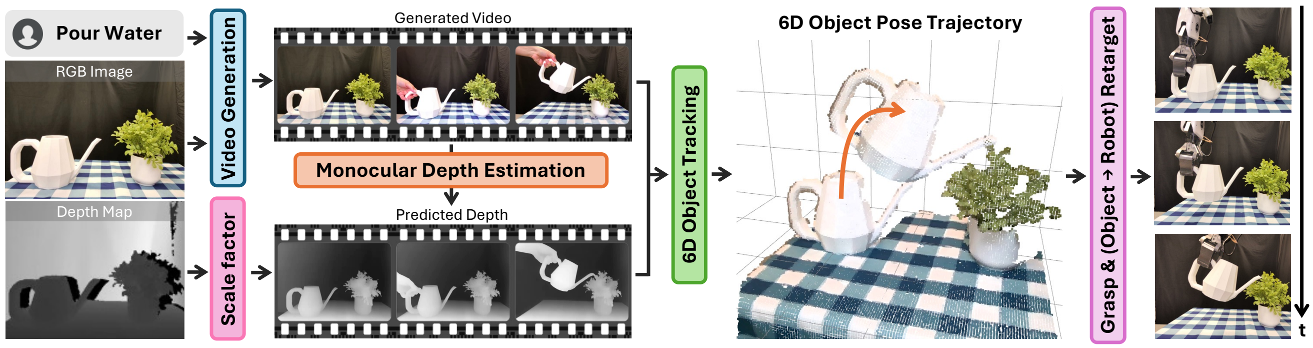


Figure 1. **Robots Imitating Generated Videos.** Given an initial scene image and depth, we generate a video conditioned on a language command. A monocular depth estimator recovers depth for each frame of the generated video, and these depth maps are combined with the corresponding RGB frames to produce 6D Object Pose Trajectory. After grasping, the trajectory is retargeted to the robot for execution.

this pose trajectory describes only how the object should move, rather than specifying robot-specific actions, it can be directly adapted to other robot platforms. During deployment, RIGVid also performs real-time object tracking and dynamically adjusts the robot's actions to handle disturbances and execution-time variations, promoting robust and adaptive behavior.

We evaluate RIGVid on four real-world manipulation tasks—pouring water, lifting a lid, placing a spatula, and sweeping trash. These tasks represent a wide range of manipulation challenges, including minimal vs. significant depth variation (pouring vs. lifting), thin and partially occluded objects (placing, sweeping), and different object geometries and actions. Our results show that, when paired with our filtering mechanism, generated videos can be as effective as real human videos for visual imitation, enabling robots to act entirely from synthetic supervision. Moreover, the performance of RIGVid improves with video quality, suggesting a promising trajectory where advances in generative models directly translate to stronger manipulation capabilities. While recent advances in VLMs offer a more compact alternative by predicting high-level task abstractions, it has remained unclear whether these representations capture sufficient detail for robust execution, especially given the substantial computational cost of full video generation. Our results indicate that generating videos is, in fact, crucial, yielding higher performance compared to SOTA VLM-based trajectory prediction method ReKep [49]. To validate our tracking approach, we compare against a broad range of existing tracking approaches that reflect the diversity of current paradigms–sparse point tracking [15], dense optical flow [60], 3D feature-fields based pose reasoning [58], and generated goal supervision [14]. Despite our method's reliance on a pre-computed mesh, this requirement consistently yields superior accuracy, confirming its practical advantage.

In summary, this work introduces a generative vision-driven paradigm for robotic manipulation that scales supervision through synthetic videos. Our key contributions are: (1) We propose RIGVid, a framework that enables robots to perform open-world manipulation tasks using only generated videos, by extracting structured, embodiment-agnostic representations that can be directly retargeted for execution—without requiring any real-world demonstrations. (2) We show that RIGVid, when using filtered generated videos, performs on par with real human videos, demonstrating that high-quality synthetic videos can serve as effective substitutes for real-world demonstrations. (3) We achieve state-of-the-art performance, outperforming representative methods based on VLMs, point tracking, optical flow, feature fields, and generated-goal supervision.

2. Related Work

Imitation from Videos. Imitation from videos seeks to acquire robotic skills directly from raw observational data, without requiring expert action labels or robot state information. This paradigm has attracted significant attention [12, 22, 32, 60, 78, 81, 91, 96, 100–102, 107, 114, 128, 133] because it eliminates the need for precisely labeled robot data. A first line of work focuses on learning actionable affordance models from internet-scale video datasets [9, 10, 27, 52, 54, 67, 68, 82, 108, 127]. However, these methods suffer from domain gap between training videos and task-specific environments, and require additional mechanisms to obtain task-conditioned affordances. To address this, many methods adopt direct imitation from videos, matching visual states in demonstration videos to those of the learner [8, 26, 34, 46, 55, 58, 93, 103, 104, 106, 112, 114, 121, 126]. While effective, this approach demands paired demonstrations in the same setting. A common strategy is to leverage visual correspondences—tracks [15] or optical flow [5, 35, 122]—to infer object trajectories. For example, Bharadhwaj *et al.* [15] predicts object tracks and uses PnP to recover poses for closed-loop task execution. Dense descriptor learning [33, 113, 135] has also proven powerful for handling variations in object geometry and appearance. Kerr *et al.* [58] recover object part trajectories from monocular videos using feature fields. Crucially, all these methods rely on demonstrations collected under conditions closely matching the target task. In contrast, our method removes this strict requirement by generating task and scene-conditioned videos. Our approach can be viewed as a hybrid between the affordance-based and direct imitation paradigms. Like the affordance-based approaches, our generated videos implicitly encode actionable affordances without requiring closely matched demonstrations. At the same time, similar to direct imitation approaches, we leverage visual imitation on these generated videos, providing the robot with task-specific guidance.

Video Generation for Robotics. Video generation has appeared as a promising avenue for robotics [3, 4, 14, 29, 30, 71, 71, 123, 132]. A common limitation of these approaches is their reliance on robot data, either to train the video generation model [71, 110], or to train policies [14], or both [3, 29, 30]. Bharadhwaj *et al*. [14] leverages tracks on generated videos to condition policy learning. Albaba *et al*. [4] uses generated videos to compute rewards for RL training. The closest related work is of Liang *et al*. [71], which executes robotic tasks by tracking tools attached to the robot's end effector. While effective, their method relies on 1,822 human-collected robot demonstrations for just four tasks, and is confined to tasks executable only by tools. In contrast, our approach requires no such data collection. Instead of tools, our method tracks objects—allowing for a significantly broader range of manipulation tasks without using any robot data.

6D Pose Estimation and Tracking. Instance-level object pose tracking methods fall into two main categories: model-based and model-free. Model-based approaches [19, 43, 44, 62, 63, 85, 89, 105] require a 3D CAD model and typically estimate pose by constructing 2D-3D correspondences and solving the PnP problem [89, 111]. In contrast, model-free methods [17, 42, 45, 66, 79, 90, 109] rely on multiple reference images instead of an explicit 3D mesh. Recent advances in vision foundation models and large datasets have enabled zero-shot methods [6, 19, 63, 77, 88], which extend to unseen objects and categories but still lag behind instance-level methods in performance. We employ FoundationPose [117], a versatile instance-level tracking method that supports model-based pose tracking. Notably, it does not require any instance-specific fine-tuning. Our choice is guided by its state-of-the-art performance and real-time execution speed, both of which are crucial for ensuring robustness against disturbances during execution.

Motion Retargeting for Object Manipulation. Early work in learning from demonstration established the foundation for object-centric motion retargeting [18, 38, 51, 80, 86, 95]. More recently, deep learning-based retargeting methods have emerged [24, 25, 41], with some incorporating object-centric representations to bridge the gap between the demonstrator and the robot [58, 69, 118]. Many approaches have applied retargeting to humanoid robots [47, 61, 72, 84, 94]. Other works have extended these techniques to dexterous manipulation [64, 97]. Like most prior work, we assume a fixed transformation between the end-effector and the object. While motion retargeting has traditionally relied on human demonstrations, RIGVid eliminates this dependency by leveraging generated videos.

3. Robots Imitating Generated Videos

We begin with an overview of RIGVid, introducing the key notations and constituent modules. We then describe each module and necessary implementation details for reproducing RIGVid.

3.1. Overview

Our method is shown in Fig. 1. Inputs are the initial scene RGB image, its corresponding depth map, and a free-form human command. Our goal is to predict the robot's 6DoF end-effector trajectory. To this end, we predict 6D pose rollout that can be easily retargeted to any robot for real-world execution. Concretely, RIGVid entails the following key steps: (1) Generate a scene- and task-conditioned video and predict its corresponding depth using a monocular depth estimator (Sec. 3.2); (2) Identify the active object mask (Sec. 3.3); (3) Compute 6D pose rollout via an object pose tracker (Sec. 3.4); (4) Grasp the object and retarget the pose trajectory to robot, and execute the resulting trajectory (Sec. 3.5).

3.2. Generating Videos and Corresponding Depth

We generate a video conditioned on the scene (using initial real RGBD observation) and the task (using human command) by employing a pre-trained video diffusion model. The recent surge in video generation research has enabled models to generate videos that are both photorealistic and also semantically aligned with open-ended instructions, an achievement that was unattainable just a year ago. We evaluate three generators–Sora [16], Kling v1.5, and Kling v1.6 [1].

Sora, introduced by OpenAI in early 2024, is notable for its ability to create highly cinematic videos with striking realism. In contrast, Kling v1.5 and its successor Kling v1.6 were both released by Kling AI in mid and late 2024 and are trained with a specific emphasis on command following. Our experiments reveal that Kling v1.6 produces the most reliable and physically plausible generated videos, resulting in higher downstream robotic performance compared to earlier models. As a result, we use Kling v1.6 as our default generator in all reported results. App. A details the practices that yielded the most reliable results in generated videos for us, although we expect these practices to become less important as the models improve.

Since the generated videos may not necessarily follow the language command, we introduce a filtering mechanism to discard inaccurate generations. We prompt GPT-4o to assess whether the generated video depicts a successful execution of the human command. We sample four evenly spaced frames in the video and concatenate them vertically into a single image to create a video summary. The VLM determines whether this summary depicts a successful execution of the task. If the response is negative, we regenerate the video and repeat the process for up to five attempts. If all attempts fail, we default to the final attempt. App. B provides the full prompt used for filtering, examples of video summaries with their corresponding VLM responses, and filtering statistics.

The resulting filtered video is a plausible execution of the task, but it is in raw pixel space. To extract actionable information from this video, we predict its corresponding depth by employing the depth predictor from Ke *et al*. [56]. Although directly using the predicted depth is the intuitive choice here, we are faced with the challenge of scale and shift ambiguity [40], where the estimated depth values are not grounded in real-world units. Consistent with prior works adopting depth estimators in vision-based robotics [23, 37], we compute a linear scale-and-shift transformation that aligns the predicted depth in the first frame with the initial real depth map. This transformation is then applied to the entire predicted video to resolve scale ambiguity.

3.3. Identifying Active Object Mask

Our next step is to identify the active object–the one being manipulated in the generated video. We require a binary mask in the initial RGB image, which is essential both for object tracking (Sec. 3.4) and for determining which object to grasp (Sec. 3.5). Given the initial image and the task command, we prompt GPT-4o [2] to identify which object in the scene is likely to be manipulated. We then ground the predicted object category into a bounding box using Grounding DINO [76], and further refine this bounding box into a segmentation mask using SAM-2 [99].

3.4. 6D Object Pose Trajectory

We then track the active object, localized by the binary mask, across the generated video using the scaled predicted depth. This yields the 6D pose rollout. Tracking objects in videos is a rich area of research, and we experimented with several video trackers in 6D pose space [63, 116, 117]. With the goal of real-world deployment, we found the tracker from Wen *et al*. [117] to perform the best. It requires an object mesh, which we pre-compute using BundleSDF [116]. For this, we record a short RGBD video in which the object is held in front of the camera and rotated, so that it is observed from all sides. While this process is straightforward, it does constrain our method to objects for which a mesh can be precomputed. However, as shown in App. C, our method is compatible with mesh-free approaches, though their inference speed is currently infeasible for real-time deployment. To ensure real-world feasibility, we apply an averaging filter to smooth abrupt pose changes, particularly in the rotational component, to prevent jerky movements. This refinement stabilizes the object pose trajectory and enables more realistic executions. App. D provides more details on pose smoothing.

3.5. Object to Robot Motion Retargeting

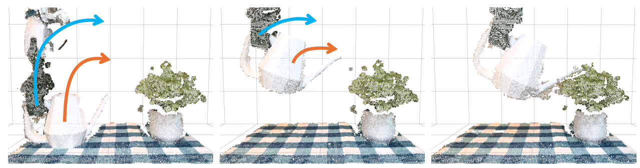


Figure 2. **Re-targeting RIGVid to a robot trajectory.** Assuming a fixed transformation between the end-effector and the object after grasping, the 6D Object Pose Trajectory (*orange arrow*) is re-targeted to the robot (*blue arrow*). This formulation is embodiment agnostic and can be transferred to a different robot.

Once the object trajectory is obtained, the first step is to grasp the object. We use an off-the-shelf grasper, Any-Grasp [31], to identify and execute the highest-scoring grasp within a defined boundary around the active object mask. After grasping, we retarget its trajectory to the robot's end-effector. Since the object remains firmly grasped, we assume a fixed transformation between the robot's end-effector and the object. This transformation is composed of two components: (1) the pose of the object relative to the gripper at the moment it is grasped and (2) the offset between the gripper and the robot's end-effector. By combining these two components, we obtain a single transformation from the end-effector to the object.

The corresponding end-effector trajectory is obtained by applying the fixed end-effector-to-object transformation to the object's pose along the entire trajectory. This formulation ensures that the retargeted 6D pose rollout follows the object's motion while maintaining a stable grasp. These are executed on the physical robot, enabling it to reproduce the object's movement as observed in the generated video. A key strength of this approach is that it is robot-agnostic. Specifically, to accommodate a different robot or gripper, only the end-effector to the object transformation needs to be updated to reflect the new end-effector configuration.

4. Experiments

We describe our experimental setup, evaluation metrics, and overview of the baselines. We provide empirical evidence to answer key research questions about RIGVid: **(1)** How does the choice of video generation model impact performance, and how does the robot perform with real videos of humans demonstrating these tasks? **(2)** How does RIGVid compare with VLM-based trajectory prediction methods that allow zero-shot robot executions? **(3)** How does RIGVid compare to the most relevant visual imitation methods?

4.1. Robot Setup, Tasks, and Evaluation

We conduct experiments on an xArm7 robot arm with a stationary Orbbec Femto Bolt camera, positioned next to the robot to capture RGBD observations. We evaluate our method on four everyday manipulation tasks that together span a diverse range of robotic challenges:

1. **Pouring water** requires the robot to position and tilt a watering can above a plant. While the depth of the can from the camera remains largely constant, successful execution demands a smooth trajectory spanning the pick-up, transport, and pouring phases. A trial is considered successful if the spout of the watering can is positioned above the plant at the end of the execution.
2. **Lifting a lid** requires the robot to lift a pot lid. Unlike pouring, this task involves significant variation in object depth, as the lid moves closer to the camera during execution. This task assesses the method's adaptability to changing object-to-camera distances and the robustness of trajectory extraction under substantial depth shifts. Success is achieved if the lid is no longer in contact with the pot at the end of the trial.
3. **Placing a spatula on a pan** requires the robot to place the head of a spatula into a pan. The spatula presents a thin, elongated geometry and is often partially occluded during manipulation, which presents a challenge for object tracking. This task evaluates the method's ability to handle objects with small surface area and persistent occlusion, both of which are particularly difficult for non-mesh-based approaches. The task is considered successful if the spatula's head is in the pan at the end of execution.
4. **Sweeping trash** requires the robot to sweep trash into a dustpan. This task is especially challenging as it combines the need for precise positioning to align the head of the sweeping brush with the trash, along with all the challenges encountered from the previous task. A trial is successful if the trash is touching the base of the dustpan at the end of the execution.

Task success is determined via human judgment based on these criteria, though the procedure could be readily automated with a VLM. The initial setup is fixed across trials of the same task and each trial has a different generated video. All baselines use the same videos for consistent comparison and reporting. We run all experiments on a single Nvidia TitanX GPU machine with 32 GB RAM.

4.2. Quality and Filtering of Generated Videos

As mentioned in Sec. 3.2, we experimented with Sora, Kling v1.5, and Kling v1.6 and compared different video filtering strategies. Next, we summarize our key empirical findings.

How do different video generation models compare for robotic imitation? Sora is known for creating visually impressive and cinematic videos. However, these videos often prioritize aesthetics over following the human command. For example, as seen in the top row of Fig. 3, Sora frequently alters the camera viewpoint, changes object positions, or even swaps out objects mid-sequence. This lack of scene and object consistency makes Sora poorly suited for imitation. Kling v1.5 places more emphasis on following language instructions. In our evaluations, videos from Kling v1.5 generally preserved the original scene and correctly depicted the target object. Nonetheless, it still exhibited physically implausible behaviors, such as objects moving in unnatural ways or actions that defy basic physical constraints. These issues, although less frequent than in Sora, still prevent successful downstream robot execution. In addition, we frequently observed that Kling v1.5 would fail to follow the commanded instruction at all, nothing happens in the video, and the intended manipulation is simply not attempted. Kling v1.6 further improved command following and physical plausibility. Videos generated by it were the most consistent with the initial scene and the intended task. As shown in the bottom row of Fig. 3, Kling v1.6 avoids altering the scene layout, maintains the positions of all objects, and depicts motions that are physically reasonable and closely aligned with the human command. Hence, Kling v1.6 proved to be the most reliable video generator for us.

Does higher video quality lead to better robot performance? To quantify this, Fig. 4 plots RIGVid 's task success across five video sources: unfiltered Sora, unfiltered Kling v1.5, unfiltered Kling v1.6, filtered Kling v1.6, and real human demonstration videos. For each source, we used 10 videos per task. We observe a clear trend: as video quality improves, so does success rate. Sora's videos led to the lowest success, Kling v1.5 performed better, and Kling v1.6 gave the highest results among all generated videos. Filtering further improved reliability: by discarding failed generations using our automatic approach, performance with filtered Kling v1.6 videos matched that obtained with real demonstration videos.

How effective is automatic video filtering? Filtering success rates varied by task: 83% for pouring, 66% for lifting, 55% for placing, and 45% for sweeping. Most errors were false negatives—i.e., the filter occasionally discarded some usable videos, but almost never passed an incorrect one. Compared to standard metrics like video-text consistency and subject consistency from VBench++ [50, 115], our VLM-based filter correlated much more strongly with true task completion (Tab. 1).

Can generated videos replace real videos for imitation? The results in Fig. 4 indicate that, when using filtered Kling v1.6 videos, RIGVid 's performance is similar to that achieved with real human demonstration videos. This find-

Figure 3. **Qualitative comparison of video generation.** Sora (top) drastically alters the scene layout and objects. Kling v1.5 (middle) is better but exhibits physically-implausible interations. Kling v1.6 (bottom) produces the most consistent and realistic videos.

ing suggests that, at current model quality, generated videos are already sufficient for visual imitation, substantially reducing the need for manual data collection.

What causes failure? Aside from one case where the object slipped out of the gripper, all failures are attributed to errors in monocular depth estimation. These errors result in inaccurate 6D trajectories and lead to tracking failures. Notably, similar depth estimation issues are also observed in real videos, suggesting that the limitation lies in the depth model itself. App. I provides a detailed analysis with qualitative examples.

4.3. RIGVid vs. VLM-based Trajectory Prediction

Video generation is computationally expensive, prompting the question of whether more efficient alternatives can enable robot manipulation without any demonstrations. VLMs offer one by predicting simplified task abstractions—goal states [48], constraints [49], or reward functions [92]—without generating full visual sequences, making them cheaper in computation and inference time. We compare against the state-of-the-art VLM-based method ReKep [49] in Fig. 5, where RIGVid achieves 85% vs. 50% success over four tasks. App. F illustrates ReKep's failures, which we attribute to inaccurate keypoint predictions. This comparison suggests that while VLM-generated abstractions are compact, they may lack the rich, necessary details. Thus, despite its higher cost, video generation provides crucial supervision rather than being a wasteful expense.

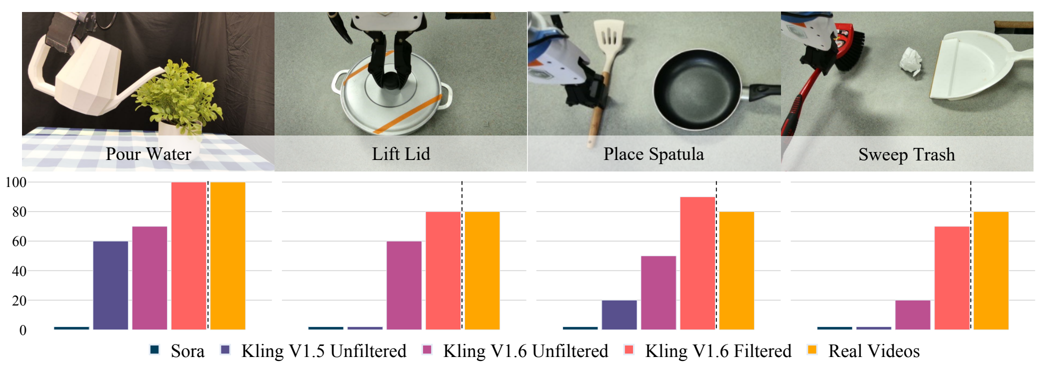


Figure 4. **RIGVid performance vs. video quality.** The dashed lines separate performance on generated videos from real videos. Kling V1.6 produces most reliable videos and leads to highest RIGVid success. Filtered videos yields performance on par with real videos.

Filtering Metrics	Description	Pour Water	Lift Lid	Place Spatula	Sweep Trash	Average
Video-text Consistency	Text-video match	0.06	0.47	0.70	0.11	0.34
I2V Subject Consistency	Image-video subject match	0.35	0.58	-0.09	0.63	0.37
Querying GPT o1	VLM-based filtering	**0.91**	**0.91**	**0.91**	**0.66**	**0.84**

Table 1. **Comparison of video filtering metrics.** Pearson correlation coefficients measure each metric's effectiveness in assessing whether a generated video follows the language command. Each task has 10 success and failure cases. Querying GPT o1 proves to be most effective.

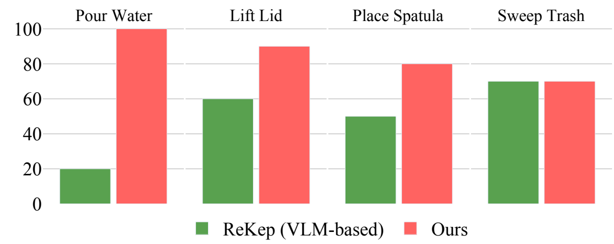


Figure 5. **RIGVid vs. VLM-Based Trajectory Prediction.** By leveraging rich spatial and temporal structure, RIGVid provides accurate execution for challenging manipulation tasks.

4.4. Baseline Tracking Methods

We adapted several well-studied and competitive baselines that use different types of tracking for visual imitation to be able to work *without any demonstrations*. For each baseline, we describe its inputs and outputs, core approach, our modifications, and the motivation for its inclusion. We describe these baselines below and defer further details to App. E.

Track2Act [15] (Tracks-Based). It takes as input RGBD image of the initial scene, together with a goal image that specifies the desired final configuration. Since we have no way to get the goal configuration, we generate its goal image by using the last frame of the generated video. The method predicts a dense grid of 2D point tracks between the initial and goal image to estimate pixel-wise correspondences. These tracks are then lifted to 3D using the depth map from the initial frame and converted into a sequence of 3D object poses via the Perspective-n-Point (PnP) algorithm. All other components of the method are kept unchanged. Track2Act is notable for its use of a dedicated track prediction network, which is conceptually similar to approaches that predict affordances, but here enables object motion to be inferred directly from observation pairs.

AVDC [60] (Flow-Based). Given an initial image and a task description, AVDC predicts object motion by first generating a task-conditioned video and then computing optical flow between frames. This optical flow is used in an optimization process to reconstruct the object trajectory. In our adaptation, we substitute AVDC's original video generator with our improved model, while preserving all downstream processing. This method is compelling because it leverages dense correspondences across all points on the object, providing more correspondences for tracking.

4D-DPM [58] (Feature Field-Based). We modify this method from tracking object part poses to tracking single objects. It takes as input a 3D Gaussian splatting field of the object and a video of the task, and outputs estimated object trajectories over time. To build the field, this requires a separate video where object is captured from all sides. In our adaptation, since 4D-DPM typically expects a real human demonstration video, we instead use a generated video as the task input video. This approach is compelling because it applies geometric, feature-based reasoning to track objects, capturing entire object structure from video, without relying on explicit correspondences.

Gen2Act [14] (Generated Goal-Based). Gen2Act takes as input an RGBD image of the scene and a task description, and outputs robot actions predicted by a learned policy. The method first generates a video of the task as an intermediate step, then extracts object tracks from this video. These tracks, together with offline robot demonstrations, are used to train a policy for robot execution. In our adaptation, we remove the fine-tuning on real robot demonstrations and instead directly estimate object poses from the video tracks, eliminating any dependence on expert demonstration data. Gen2Act is notable for leveraging sparse correspondences extracted from the generated video, enabling task-relevant object motion to be tracked and retargeted without requiring explicit actions.

4.5. RIGVid vs Other Robotics Tracking Methods

In the following, we include takeaways based on the results in Fig. 6 and Fig. 7.

How does RIGVid compare to prior works that use other tracking methods, and what accounts for its advantage? Fig. 6 reveal that RIGVid achieves a success rate of 85.0%, compared to 67.5% for Gen2Act and considerably lower

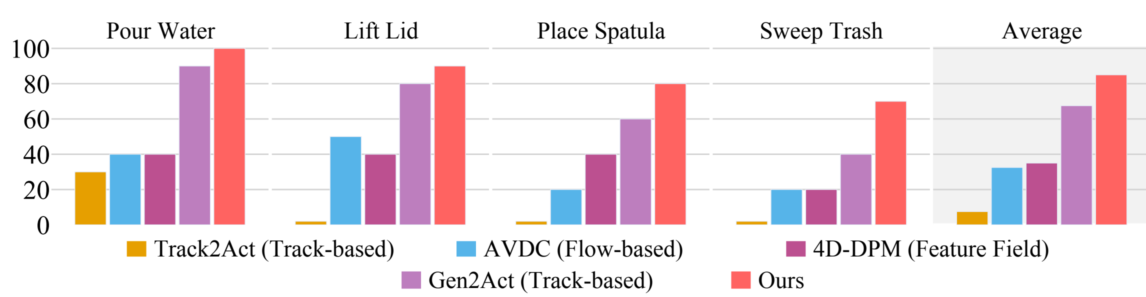


Figure 6. **Performance on everyday manipulation tasks.** Shown in the table, RIGVid, which uses 6D Object Pose Trajectory, consistently achieves higher success rates (*bottom*) across all four tasks (*top*). Relative improvements are higher as tasks become harder (*i.e.*, from left to right in the bar plot). All results are on valid video generations (*i.e.*, human filtered; detailed study in Sec. 4.2) and ten episodes for each reported metric.

rates for all other baselines. This margin grows on the more complex tasks. Crucially, all approaches are evaluated using the same set of generated videos, isolating the effect of the trajectory representation itself. Methods such as Track2Act (7.5%), AVDC (32.5%), and 4D-DPM (35.0%) rely on point tracks or optical flow, but their performance remains limited—especially as object rotations or occlusions become severe. Gen2Act, which combines video generation with point-based tracking, closes part of the gap but consistently struggles when large portions of the object become untrackable. In contrast, RIGVid's use of a structured 6D object pose trajectory enables robust execution across all tasks, accounting for the observed 17.5% absolute improvement over Gen2Act. This advantage persists even when more powerful tracking models like CoTracker3 [53] are used, as shown in App. G. These results indicate that it is the accuracy and stability of the 6D object pose trajectory that is key to RIGVid's stronger performance across a range of manipulation tasks.

How does RIGVid perform on tasks involving depth variation, small objects, and occlusion? Looking at the task-wise breakdown in Fig. 6, we find that RIGVid consistently maintains high success rates even as object depth varies significantly (such as in the lifting task) or when the manipulated objects are thin, small, or partially occluded (such as in placing a spatula or sweeping trash). Other methods frequently struggle in these settings, often failing to recover accurate object trajectories when objects become partially hidden or change distance rapidly. The advantage of RIGVid is most pronounced on the most challenging tasks: for both spatula placement and sweeping, RIGVid achieves success rates 20–25% higher than the next best baseline. These results suggest that the structured 6D pose trajectory not only enables robust tracking under depth changes and occlusion, but also scales to manipulation scenarios where traditional correspondence-based methods break down.

What do the intermediate predictions reveal about these methods? Visualizing the outputs in Fig. 7 for the same generated video, we observe the intermediate predictions and resulting robot executions produced by each method. For Track2Act, the predicted tracks diverge from the true object path, leading to failed execution. AVDC generates reasonable optical flow in individual frames, but when summed across the entire video, the resulting trajectory is often phys-

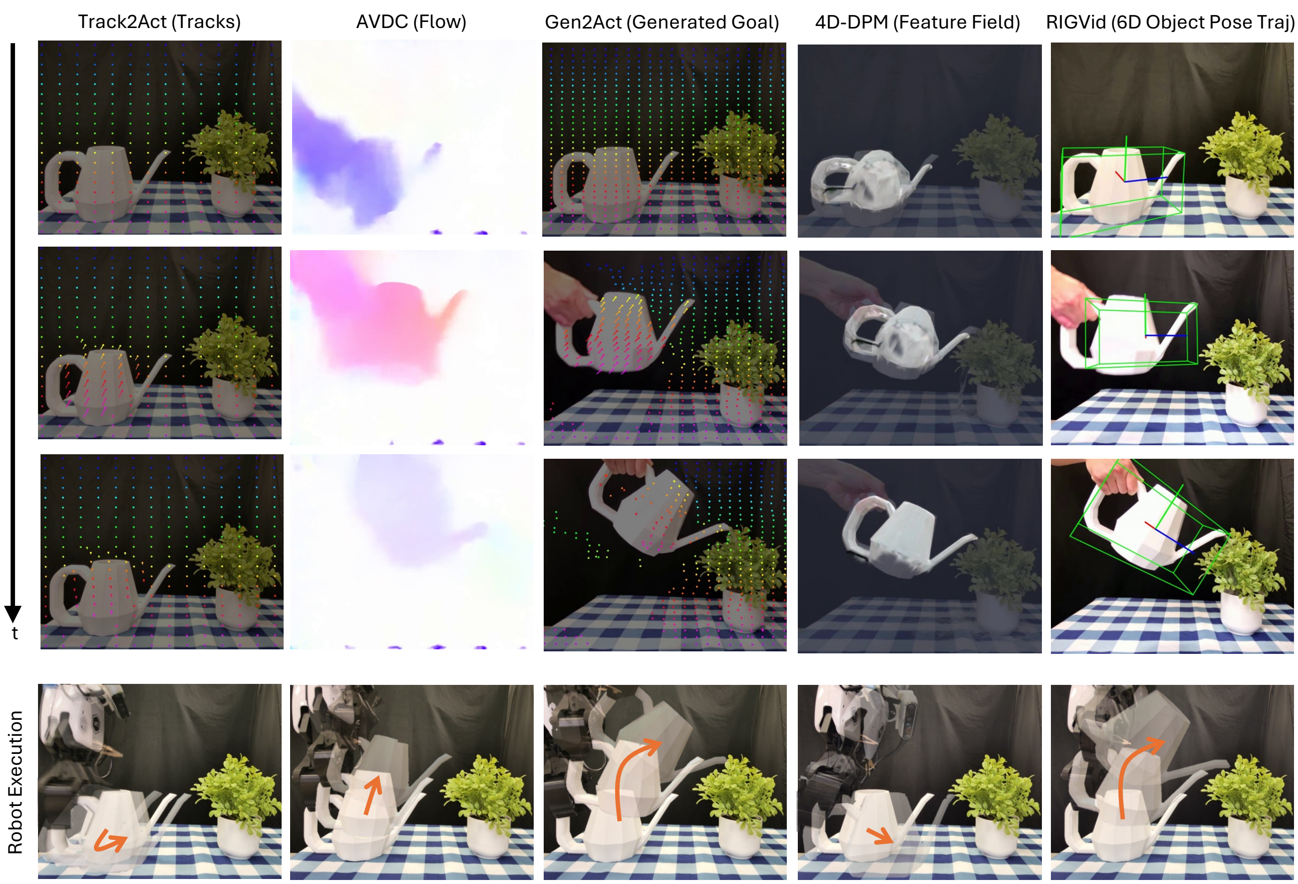


Figure 7. **Analyzing intermediate visual representations.** Our 6D Object Pose Trajectory can correctly track the position and rotation of the watering can (rightmost column), leading to a successful execution.

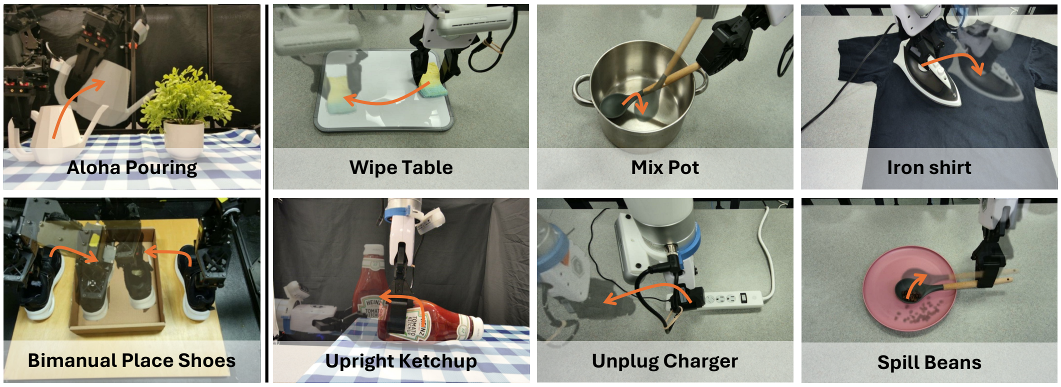


Figure 8. **RIGVid's embodiment-agnostic capabilities and examples on solving complex, open-world tasks.** RIGVid can readily work on ALOHA setup [131] as shown on top left. On the bottom left, RIGVid is retargeted to the ALOHA setup. On the right, it generates trajectories for diverse manipulation tasks—including wiping, mixing, and ironing—without using any physical demonstrations.

ically implausible and the execution fails. Gen2Act yields plausible tracks and leads to successful manipulation. 4D-DPM exhibits inconsistent tracking performance. While it accurately follows the object in certain segments, the example shown reveals incorrect tracking during the first half of the episode, which ultimately causes the rollout to fail. In contrast, the 6D object pose trajectories produced by RIGVid remain stable throughout the episode and closely match the actual object motion, resulting in successful execution.

4.6. Testing Generalization and Robustness

Embodiment-Agnostic Transfer. We test RIGVid's generalizability to another embodiment by deploying it on the ALOHA robot for the pouring task (Fig. 8, top left). On this setup, it achieves 80% success, compared to 100% on our default xArm setup.[1] RIGVid also generalizes to a bimanual setup, simultaneously placing a pair of shoes into a box using both arms (Fig. 8, bottom left).

Extensions to New Tasks. RIGVid enables zero-shot execution of diverse and challenging manipulation tasks that involve complex and unconstrained trajectories. As shown in Fig. 8 (right), it completes tasks such as wiping, mixing, and ironing. It also handles physically intricate scenarios, including uprighting a ketchup bottle, rotating a spoon to spill beans, and unplugging a charger, despite these tasks involving extreme rotations.

Recovery from Perturbations. A key strength of RIGVid is its robustness to external disturbances during execution, as shown in Fig. 9. The system continuously tracks the object's position using f_{track} and updates the robot's end-effector trajectory in real time. To detect a deviation (image 1), the current object pose is compared to the precomputed motion plan. If it strays beyond 3 cm in position or 20 degrees in orientation (image 2), the system classifies it as a disturbance. The robot then backtracks to the last successfully executed trajectory point (image 3) and resumes the planned motion (image 4). This realignment allows RIGVid to maintain stable task execution even under physical perturbations. Additional robustness examples are discussed in App. H.

[1]The slight performance drop stems primarily from camera calibration challenges, as ALOHA's arms yield less accurate pose estimates.

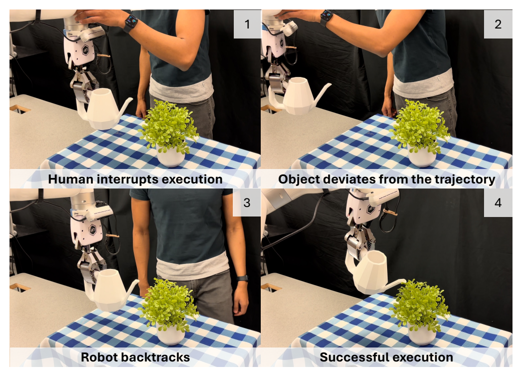


Figure 9. **RIGVid is robust to perturbations.** The robot detects and recovers from external disturbances by backtracking to the last achieved pose before resuming execution.

5. Conclusions and Limitations

We introduced Robots Imitating Generated Videos (RIGVid), the first method for robotic manipulation that works without demonstrations—no teleoperation, no human demonstration, or expert policy rollouts. By leveraging recent advances in generative vision models and 6D pose estimation, RIGVid enables robots to execute complex tasks entirely from generated video. We extract 6D Object Pose Trajectory from the generated videos and retarget it to the robot, demonstrating a scalable, data-efficient approach to robotic skill acquisition. Our analysis shows a clear correlation between video quality and task success: as generation improves, RIGVid approaches real demo performance. Additionally, our comparisons with SOTA VLM-based zero-shot manipulation methods confirm that leveraging detailed visual and temporal cues from generated videos yields substantially superior performance. We also show that RIGVid significantly outperforms baselines across a diverse set of visual imitation tasks, and demonstrate the robustness of our approach to environmental disturbances. Our work represents a step toward enabling robots to learn from the vast visual knowledge in generative models, reducing reliance on costly and time-consuming real-world data collection.

Despite the advancements, our method has certain limitations. We need a precomputed mesh of the objects. In the future, it can be simplified by using single-image to mesh reconstruction methods [74, 75]. Additionally, our approach depends on the video generation quality, which may struggle with complex prompts or scenes. As video generation improves, we anticipate this limitation will become less significant. Our work aims to democratize robotics by removing the need for demonstrations, which could enable broader accessibility of robotic capabilities. However, we also acknowledge potential risks. Highly generalizable manipulation methods could be misused in applications such as automated weapons or unsafe industrial automation.

References

[1] Kling ai. https://www.klingai.com/. Accessed: 2024-02-10. 1, 3

[2] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 4

[3] Anurag Ajay, Seungwook Han, Yilun Du, Shuang Li, Abhi Gupta, Tommi Jaakkola, Josh Tenenbaum, Leslie Kaelbling, Akash Srivastava, and Pulkit Agrawal. Compositional foundation models for hierarchical planning. *Advances in Neural Information Processing Systems*, 36: 22304–22325, 2023. 3

[4] Mert Albaba, Chenhao Li, Markos Diomataris, Omid Taheri, Andreas Krause, and Michael Black. Nil: No-data imitation learning by leveraging pre-trained video diffusion models. *arXiv preprint arXiv:2503.10626*, 2025. 3

[5] Max Argus, Lukas Hermann, Jon Long, and Thomas Brox. Flowcontrol: Optical flow based visual servoing. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7534–7541. IEEE, 2020. 2

[6] Philipp Ausserlechner, David Haberger, Stefan Thalhammer, Jean-Baptiste Weibel, and Markus Vincze. Zs6d: Zero-shot 6d object pose estimation using vision transformers. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 463–469. IEEE, 2024. 3

[7] Arpit Bahety, Priyanka Mandikal, Ben Abbatematteo, and Roberto Martín-Martín. Screwmimic: Bimanual imitation from human videos with screw space projection. *arXiv preprint arXiv:2405.03666*, 2024. 1

[8] Shikhar Bahl, Abhinav Gupta, and Deepak Pathak. Human-to-robot imitation in the wild. *arXiv preprint arXiv:2207.09450*, 2022. 1, 2

[9] Shikhar Bahl, Russell Mendonca, Lili Chen, Unnat Jain, and Deepak Pathak. Affordances from human videos as a versatile representation for robotics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13778–13790, 2023. 1, 2

[10] Bowen Baker, Ilge Akkaya, Peter Zhokov, Joost Huizinga, Jie Tang, Adrien Ecoffet, Brandon Houghton, Raul Sampedro, and Jeff Clune. Video pretraining (vpt): Learning to act by watching unlabeled online videos. *Advances in Neural Information Processing Systems*, 35:24639–24654, 2022. 2

[11] Hritik Bansal, Zongyu Lin, Tianyi Xie, Zeshun Zong, Michal Yarom, Yonatan Bitton, Chenfanfu Jiang, Yizhou Sun, Kai-Wei Chang, and Aditya Grover. Videophy: Evaluating physical commonsense for video generation. *arXiv preprint arXiv:2406.03520*, 2024. 1

[12] Leonardo Barcellona, Andrii Zadaianchuk, Davide Allegro, Samuele Papa, Stefano Ghidoni, and Efstratios Gavves. Dream to manipulate: Compositional world models empowering robot imitation learning with imagination. *arXiv preprint arXiv:2412.14957*, 2024. 2

[13] Homanga Bharadhwaj, Abhinav Gupta, Shubham Tulsiani, and Vikash Kumar. Zero-shot robot manipulation from passive human videos. *arXiv preprint arXiv:2302.02011*, 2023. 1

[14] 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. 1, 2, 3, 7, 17

[15] Homanga Bharadhwaj, Roozbeh Mottaghi, Abhinav Gupta, and Shubham Tulsiani. Track2act: Predicting point tracks from internet videos enables diverse zero-shot robot manipulation, 2024. 1, 2, 7, 16

[16] Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, Clarence Ng, Ricky Wang, and Aditya Ramesh. Video generation models as world simulators. 2024. 1, 3

[17] Ming Cai and Ian Reid. Reconstruct locally, localize globally: A model free method for object pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3153–3163, 2020. 3

[18] Sylvain Calinon. A tutorial on task-parameterized movement learning and retrieval. *Intelligent service robotics*, 9: 1–29, 2016. 3

[19] Andrea Caraffa, Davide Boscaini, Amir Hamza, and Fabio Poiesi. Freeze: Training-free zero-shot 6d pose estimation with geometric and vision foundation models. *European Conference on Computer Vision (ECCV)*, 2024. 3

[20] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021. 20

[21] Elliot Chane-Sane, Cordelia Schmid, and Ivan Laptev. Learning video-conditioned policies for unseen manipulation tasks. In *2023 IEEE International Conference on*

Robotics and Automation (ICRA), pages 909–916. IEEE, 2023. 1

[22] Matthew Chang, Arjun Gupta, and Saurabh Gupta. Semantic visual navigation by watching youtube videos. In *NeurIPS*, 2020. 1, 2

[23] Matthew Chang, Theophile Gervet, Mukul Khanna, Sriram Yenamandra, Dhruv Shah, So Yeon Min, Kavit Shah, Chris Paxton, Saurabh Gupta, Dhruv Batra, Roozbeh Mottaghi, Jitendra Malik, and Devendra Singh Chaplot. Goat: Go to any thing. *arXiv preprint arXiv:2311.06430*, 2023. 4

[24] Xuxin Cheng, Yandong Ji, Junming Chen, Ruihan Yang, Ge Yang, and Xiaolong Wang. Expressive whole-body control for humanoid robots. *arXiv preprint arXiv:2402.16796*, 2024. 3

[25] Sungjoon Choi, Matthew KXJ Pan, and Joohyung Kim. Nonparametric motion retargeting for humanoid robots on shared latent space. In *Robotics: science and systems*, 2020. 3

[26] Sudeep Dasari and Abhinav Gupta. Transformers for one-shot visual imitation. In *Conference on Robot Learning*, pages 2071–2084. PMLR, 2021. 1, 2

[27] Sudeep Dasari, Mohan Kumar Srirama, Unnat Jain, and Abhinav Gupta. An unbiased look at datasets for visuo-motor pre-training. In *Conference on Robot Learning*, pages 1183–1198. PMLR, 2023. 2

[28] Carl Doersch, Yi Yang, Dilara Gokay, Pauline Luc, Skanda Koppula, Ankush Gupta, Joseph Heyward, Ross Goroshin, João Carreira, and Andrew Zisserman. Bootstap: Bootstrapped training for tracking-any-point. *arXiv preprint arXiv:2402.00847*, 2024. 17

[29] Yilun Du, Mengjiao Yang, Pete Florence, Fei Xia, Ayzaan Wahid, Brian Ichter, Pierre Sermanet, Tianhe Yu, Pieter Abbeel, Joshua B Tenenbaum, et al. Video language planning. *arXiv preprint arXiv:2310.10625*, 2023. 3

[30] Yilun Du, Sherry Yang, Bo Dai, Hanjun Dai, Ofir Nachum, Josh Tenenbaum, Dale Schuurmans, and Pieter Abbeel. Learning universal policies via text-guided video generation. *Advances in Neural Information Processing Systems*, 36, 2024. 1, 3

[31] Hao-Shu Fang, Chenxi Wang, Hongjie Fang, Minghao Gou, Jirong Liu, Hengxu Yan, Wenhai Liu, Yichen Xie, and Cewu Lu. Anygrasp: Robust and efficient grasp perception in spatial and temporal domains. *IEEE Transactions on Robotics*, 2023. 4

[32] Chelsea Finn, Tianhe Yu, T. Zhang, P. Abbeel, and Sergey Levine. One-shot visual imitation learning via meta-learning. In *CoRL*, 2017. 2

[33] Peter R Florence, Lucas Manuelli, and Russ Tedrake. Dense object nets: Learning dense visual object descriptors by and for robotic manipulation. *arXiv preprint arXiv:1806.08756*, 2018. 2

[34] Zipeng Fu, Qingqing Zhao, Qi Wu, Gordon Wetzstein, and Chelsea Finn. Humanplus: Humanoid shadowing and imitation from humans. *arXiv preprint arXiv:2406.10454*, 2024. 2

[35] Chongkai Gao, Haozhuo Zhang, Zhixuan Xu, Zhehao Cai, and Lin Shao. Flip: Flow-centric generative planning for general-purpose manipulation tasks. *arXiv preprint arXiv:2412.08261*, 2024. 2

[36] Shenyuan Gao, Siyuan Zhou, Yilun Du, Jun Zhang, and Chuang Gan. Adaworld: Learning adaptable world models with latent actions. *arXiv preprint arXiv:2503.18938*, 2025. 1

[37] Theophile Gervet, Soumith Chintala, Dhruv Batra, Jitendra Malik, and Devendra Singh Chaplot. Navigating to objects in the real world. *Science Robotics*, 2023. 4

[38] Michael Gleicher. Retargetting motion to new characters. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 33–42, 1998. 3

[39] Xuyang Guo, Jiayan Huo, Zhenmei Shi, Zhao Song, Jiahao Zhang, and Jiale Zhao. T2vphysbench: A first-principles benchmark for physical consistency in text-to-video generation. *arXiv preprint arXiv:2505.00337*, 2025. 1

[40] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003. 4

[41] Tairan He, Zhengyi Luo, Wenli Xiao, Chong Zhang, Kris Kitani, Changliu Liu, and Guanya Shi. Learning human-to-humanoid real-time whole-body teleoperation. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8944–8951. IEEE, 2024. 3

[42] Xingyi He, Jiaming Sun, Yuang Wang, Di Huang, Hujun Bao, and Xiaowei Zhou. Onepose++: Keypoint-free one-shot object pose estimation without cad models. *Advances in Neural Information Processing Systems*, 35: 35103–35115, 2022. 3

[43] Yisheng He, Wei Sun, Haibin Huang, Jianran Liu, Haoqiang Fan, and Jian Sun. Pvn3d: A deep point-wise 3d keypoints voting network for 6dof pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11632–11641, 2020. 3

[44] Yisheng He, Haibin Huang, Haoqiang Fan, Qifeng Chen, and Jian Sun. Ffb6d: A full flow bidirectional fusion network for 6d pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3003–3013, 2021. 3

[45] Yisheng He, Yao Wang, Haoqiang Fan, Jian Sun, and Qifeng Chen. Fs6d: Few-shot 6d pose estimation of novel objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6814–6824, 2022. 3

[46] Cheng-Chun Hsu, Bowen Wen, Jie Xu, Yashraj Narang, Xiaolong Wang, Yuke Zhu, Joydeep Biswas, and Stan Birchfield. Spot: Se (3) pose trajectory diffusion for object-centric manipulation. *arXiv preprint arXiv:2411.00965*, 2024. 2

[47] Kai Hu, Christian Ott, and Dongheui Lee. Online human walking imitation in task and joint space based on quadratic programming. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3458–3464. IEEE, 2014. 3

[48] Wenlong Huang, Chen Wang, Ruohan Zhang, Yunzhu Li, Jiajun Wu, and Li Fei-Fei. Voxposer: Composable 3d value maps for robotic manipulation with language models. *arXiv preprint arXiv:2307.05973*, 2023. 6

[49] Wenlong Huang, Chen Wang, Yunzhu Li, Ruohan Zhang, and Li Fei-Fei. Rekep: Spatio-temporal reasoning of relational keypoint constraints for robotic manipulation. *arXiv preprint arXiv:2409.01652*, 2024. 2, 6

[50] Ziqi Huang, Fan Zhang, Xiaojie Xu, Yinan He, Jiashuo Yu, Ziyue Dong, Qianli Ma, Nattapol Chanpaisit, Chenyang Si, Yuming Jiang, Yaohui Wang, Xinyuan Chen, Ying-Cong Chen, Limin Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. Vbench++: Comprehensive and versatile benchmark suite for video generative models. *arXiv preprint arXiv:2411.13503*, 2024. 5, 20, 21

[51] Biao Jiang, Xin Chen, Wen Liu, Jingyi Yu, Gang Yu, and Tao Chen. Motiongpt: Human motion as a foreign language. *Advances in Neural Information Processing Systems*, 36:20067–20079, 2023. 3

[52] Yuanchen Ju, Kaizhe Hu, Guowei Zhang, Gu Zhang, Mingrun Jiang, and Huazhe Xu. Robo-abc: Affordance generalization beyond categories via semantic correspondence for robot manipulation. In *European Conference on Computer Vision*, pages 222–239. Springer, 2024. 2

[53] Nikita Karaev, Iurii Makarov, Jianyuan Wang, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Cotracker3: Simpler and better point tracking by pseudo-labelling real videos. *arXiv preprint arXiv:2410.11831*, 2024. 8

[54] Siddharth Karamcheti, Suraj Nair, Annie S Chen, Thomas Kollar, Chelsea Finn, Dorsa Sadigh, and Percy Liang. Language-driven representation learning for robotics. *arXiv preprint arXiv:2302.12766*, 2023. 2

[55] Simar Kareer, Dhruv Patel, Ryan Punamiya, Pranay Mathur, Shuo Cheng, Chen Wang, Judy Hoffman, and Danfei Xu. Egomimic: Scaling imitation learning via egocentric video. *arXiv preprint arXiv:2410.24221*, 2024. 1, 2

[56] Bingxin Ke, Dominik Narnhofer, Shengyu Huang, Lei Ke, Torben Peters, Katerina Fragkiadaki, Anton Obukhov, and Konrad Schindler. Video depth without video models, 2024. 4

[57] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023. 17

[58] Justin Kerr, Chung Min Kim, Mingxuan Wu, Brent Yi, Qianqian Wang, Ken Goldberg, and Angjoo Kanazawa. Robot see robot do: Imitating articulated object manipulation with monocular 4d reconstruction. *arXiv preprint arXiv:2409.18121*, 2024. 2, 3, 7, 17

[59] Chung Min Kim, Mingxuan Wu, Justin Kerr, Ken Goldberg, Matthew Tancik, and Angjoo Kanazawa. Garfield: Group anything with radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21530–21539, 2024. 17

[60] Po-Chen Ko, Jiayuan Mao, Yilun Du, Shao-Hua Sun, and Joshua B Tenenbaum. Learning to act from actionless videos through dense correspondences. *arXiv preprint arXiv:2310.08576*, 2023. 2, 7, 17

[61] Scott Kuindersma, Robin Deits, Maurice Fallon, Andrés Valenzuela, Hongkai Dai, Frank Permenter, Twan Koolen, Pat Marion, and Russ Tedrake. Optimization-based locomotion planning, estimation, and control design for the atlas humanoid robot. *Autonomous robots*, 40:429–455, 2016. 3

[62] Yann Labbé, Justin Carpentier, Mathieu Aubry, and Josef Sivic. Cosypose: Consistent multi-view multi-object 6d pose estimation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVII 16*, pages 574–591. Springer, 2020. 3

[63] Yann Labbé, Lucas Manuelli, Arsalan Mousavian, Stephen Tyree, Stan Birchfield, Jonathan Tremblay, Justin Carpentier, Mathieu Aubry, Dieter Fox, and Josef Sivic. Megapose: 6d pose estimation of novel objects via render & compare. In *Proceedings of the 6th Conference on Robot Learning (CoRL)*, 2022. 3, 4, 19

[64] Arjun S Lakshmipathy, Jessica K Hodgins, and Nancy S Pollard. Kinematic motion retargeting for contact-rich anthropomorphic manipulations. *arXiv preprint arXiv:2402.04820*, 2024. 3

[65] Marion Lepert, Jiaying Fang, and Jeannette Bohg. Phantom: Training robots without robots using only human videos. *arXiv preprint arXiv:2503.00779*, 2025. 1

[66] Fu Li, Shishir Reddy Vutukur, Hao Yu, Ivan Shugurov, Benjamin Busam, Shaowu Yang, and Slobodan Ilic. Nerf-pose: A first-reconstruct-then-regress approach for weakly-supervised 6d object pose estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2123–2133, 2023. 3

[67] Gen Li, Deqing Sun, Laura Sevilla-Lara, and Varun Jampani. One-shot open affordance learning with foundation models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3086–3096, 2024. 2

[68] Gen Li, Nikolaos Tsagkas, Jifei Song, Ruaridh Mon-Williams, Sethu Vijayakumar, Kun Shao, and Laura Sevilla-Lara. Learning precise affordances from egocentric videos for robotic manipulation. *arXiv preprint arXiv:2408.10123*, 2024. 2

[69] Jinhan Li, Yifeng Zhu, Yuqi Xie, Zhenyu Jiang, Mingyo Seo, Georgios Pavlakos, and Yuke Zhu. Okami: Teaching humanoid robots manipulation skills through single video imitation. In *8th Annual Conference on Robot Learning*, 2024. 1, 3

[70] Zhen Li, Zuo-Liang Zhu, Ling-Hao Han, Qibin Hou, Chun-Le Guo, and Ming-Ming Cheng. Amt: All-pairs multi-field transforms for efficient frame interpolation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 20

[71] Junbang Liang, Ruoshi Liu, Ege Ozguroglu, Sruthi Sudhakar, Achal Dave, Pavel Tokmakov, Shuran Song, and Carl Vondrick. Dreamitate: Real-world visuomotor policy learning via video generation. *arXiv preprint arXiv:2406.16862*, 2024. 3

[72] Yuwei Liang, Weijie Li, Yue Wang, Rong Xiong, Yichao Mao, and Jiafan Zhang. Dynamic movement primitive based motion retargeting for dual-arm sign language mo-

tions. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8195–8201. IEEE, 2021. 3

[73] Fangfu Liu, Wenqiang Sun, Hanyang Wang, Yikai Wang, Haowen Sun, Junliang Ye, Jun Zhang, and Yueqi Duan. Reconx: Reconstruct any scene from sparse views with video diffusion model. *arXiv preprint arXiv:2408.16767*, 2024. 1

[74] Minghua Liu, Chao Xu, Haian Jin, Linghao Chen, Mukund Varma T, Zexiang Xu, and Hao Su. One-2-3-45: Any single image to 3d mesh in 45 seconds without per-shape optimization. *Advances in Neural Information Processing Systems*, 36:22226–22246, 2023. 10

[75] Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3d object. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9298–9309, 2023. 10

[76] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2023. 4

[77] Xingyu Liu, Gu Wang, Ruida Zhang, Chenyangguang Zhang, Federico Tombari, and Xiangyang Ji. Unopose: Unseen object pose estimation with an unposed rgb-d reference image. *arXiv preprint arXiv:2411.16106*, 2024. 3

[78] YuXuan Liu, Abhishek Gupta, Pieter Abbeel, and Sergey Levine. Imitation from observation: Learning to imitate behaviors from raw video via context translation. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1118–1125. IEEE, 2018. 2

[79] Yuan Liu, Yilin Wen, Sida Peng, Cheng Lin, Xiaoxiao Long, Taku Komura, and Wenping Wang. Gen6d: Generalizable model-free 6-dof object pose estimation from rgb images. In *European Conference on Computer Vision*, pages 298–315. Springer, 2022. 3

[80] Zhengyi Luo, Jinkun Cao, Kris Kitani, Weipeng Xu, et al. Perpetual humanoid control for real-time simulated avatars. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10895–10904, 2023. 3

[81] Ajay Mandlekar, Yuke Zhu, Animesh Garg, Jonathan Booher, Max Spero, Albert Tung, Julian Gao, John Emmons, Anchit Gupta, Emre Orbay, et al. Roboturk: A crowdsourcing platform for robotic skill learning through imitation. In *Conference on Robot Learning*, pages 879–893. PMLR, 2018. 2

[82] Russell Mendonca, Shikhar Bahl, and Deepak Pathak. Structured world models from human videos. *arXiv preprint arXiv:2308.10901*, 2023. 2

[83] Saman Motamed, Laura Culp, Kevin Swersky, Priyank Jaini, and Robert Geirhos. Do generative video models learn physical principles from watching videos? *arXiv preprint arXiv:2501.09038*, 2025. 1

[84] Shinichiro Nakaoka, Atsushi Nakazawa, Fumio Kanehiro, Kenji Kaneko, Mitsuharu Morisawa, and Katsushi Ikeuchi. Task model of lower body motion for a biped humanoid robot to imitate human dances. In *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3157–3162. IEEE, 2005. 3

[85] Van Nguyen Nguyen, Thibault Groueix, Mathieu Salzmann, and Vincent Lepetit. Gigapose: Fast and robust novel object pose estimation via one correspondence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9903–9913, 2024. 3

[86] Scott Niekum, Sarah Osentoski, George Konidaris, and Andrew G Barto. Learning and generalization of complex tasks from unstructured demonstrations. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5239–5246. IEEE, 2012. 3

[87] Abby O'Neill, Abdul Rehman, Abhinav Gupta, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, et al. Open x-embodiment: Robotic learning datasets and rt-x models. *arXiv preprint arXiv:2310.08864*, 2023. 1

[88] Evin Pınar Örnek, Yann Labbé, Bugra Tekin, Lingni Ma, Cem Keskin, Christian Forster, and Tomáš Hodaň. Foundpose: Unseen object pose estimation with foundation features. *European Conference on Computer Vision (ECCV)*, 2024. 3

[89] Kiru Park, Timothy Patten, and Markus Vincze. Pix2pose: Pixel-wise coordinate regression of objects for 6d pose estimation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 7668–7677, 2019. 3

[90] Keunhong Park, Arsalan Mousavian, Yu Xiang, and Dieter Fox. Latentfusion: End-to-end differentiable reconstruction and rendering for unseen object pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10710–10719, 2020. 3

[91] Austin Patel, Andrew Wang, Ilija Radosavovic, and Jitendra Malik. Learning to imitate object interactions from internet videos. *arXiv preprint arXiv:2211.13225*, 2022. 2

[92] Shivansh Patel, Xinchen Yin, Wenlong Huang, Shubham Garg, Hooshang Nayyeri, Li Fei-Fei, Svetlana Lazebnik, and Yunzhu Li. A real-to-sim-to-real approach to robotic manipulation with vlm-generated iterative keypoint rewards. *arXiv preprint arXiv:2502.08643*, 2025. 6

[93] Deepak Pathak, Parsa Mahmoudieh, Guanghao Luo, Pulkit Agrawal, Dian Chen, Yide Shentu, Evan Shelhamer, Jitendra Malik, Alexei A Efros, and Trevor Darrell. Zero-shot visual imitation. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 2050–2053, 2018. 2

[94] Luigi Penco, Nicola Scianca, Valerio Modugno, Leonardo Lanari, Giuseppe Oriolo, and Serena Ivaldi. A multimode teleoperation framework for humanoid loco-manipulation: An application for the icub robot. *IEEE Robotics & Automation Magazine*, 26(4):73–82, 2019. 3

[95] Xue Bin Peng, Ze Ma, Pieter Abbeel, Sergey Levine, and Angjoo Kanazawa. Amp: Adversarial motion priors for stylized physics-based character control. *ACM Transactions on Graphics (ToG)*, 40(4):1–20, 2021. 3

[96] Georgy Ponimatkin, Martin Cífka, Tomáš Souček, Médéric Fourmy, Yann Labbé, Vladimir Petrik, and Josef Sivic. 6d object pose tracking in internet videos for robotic manipulation. *arXiv preprint arXiv:2503.10307*, 2025. 2

[97] Yuzhe Qin, Yueh-Hua Wu, Shaowei Liu, Hanwen Jiang, Ruihan Yang, Yang Fu, and Xiaolong Wang. Dexmv: Im-

itation learning for dexterous manipulation from human videos. In *European Conference on Computer Vision*, pages 570–587. Springer, 2022. 3

[98] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021. 20

[99] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. Sam 2: Segment anything in images and videos, 2024. 4

[100] Juntao Ren, Priya Sundaresan, Dorsa Sadigh, Sanjiban Choudhury, and Jeannette Bohg. Motion tracks: A unified representation for human-robot transfer in few-shot imitation learning. *arXiv preprint arXiv:2501.06994*, 2025. 2

[101] Pierre Sermanet, Corey Lynch, Yevgen Chebotar, Jasmine Hsu, Eric Jang, Stefan Schaal, Sergey Levine, and Google Brain. Time-contrastive networks: Self-supervised learning from video. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 1134–1141. IEEE, 2018.

[102] Pratyusha Sharma, Lekha Mohan, Lerrel Pinto, and Abhinav Gupta. Multiple interactions made easy (mime): Large scale demonstrations data for imitation. *arXiv:1810.07121*, 2018. 2

[103] Pratyusha Sharma, Deepak Pathak, and Abhinav Gupta. Third-person visual imitation learning via decoupled hierarchical controller. *Advances in Neural Information Processing Systems*, 32, 2019. 2

[104] Junyao Shi, Zhuolun Zhao, Tianyou Wang, Ian Pedroza, Amy Luo, Jie Wang, Jason Ma, and Dinesh Jayaraman. Zeromimic: Distilling robotic manipulation skills from web videos. *arXiv preprint arXiv:2503.23877*, 2025. 2

[105] Ivan Shugurov, Fu Li, Benjamin Busam, and Slobodan Ilic. Osop: A multi-stage one shot object pose estimation framework. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6835–6844, 2022. 3

[106] Aravind Sivakumar, Kenneth Shaw, and Deepak Pathak. Robotic telekinesis: Learning a robotic hand imitator by watching humans on youtube. *arXiv preprint arXiv:2202.10448*, 2022. 1, 2

[107] Laura Smith, Nikita Dhawan, Marvin Zhang, Pieter Abbeel, and Sergey Levine. Avid: Learning multi-stage tasks via pixel-level translation of human videos. *arXiv preprint arXiv:1912.04443*, 2019. 2

[108] Mohan Kumar Srirama, Sudeep Dasari, Shikhar Bahl, and Abhinav Gupta. Hrp: Human affordances for robotic pre-training. *arXiv preprint arXiv:2407.18911*, 2024. 2

[109] Jiaming Sun, Zihao Wang, Siyu Zhang, Xingyi He, Hongcheng Zhao, Guofeng Zhang, and Xiaowei Zhou. Onepose: One-shot object pose estimation without cad models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6825–6834, 2022. 3

[110] Yihong Sun, Hao Zhou, Liangzhe Yuan, Jennifer J Sun, Yandong Li, Xuhui Jia, Hartwig Adam, Bharath Hariharan, Long Zhao, and Ting Liu. Video creation by demonstration. *arXiv preprint arXiv:2412.09551*, 2024. 3

[111] Jonathan Tremblay, Thang To, Balakumar Sundaralingam, Yu Xiang, Dieter Fox, and Stan Birchfield. Deep object pose estimation for semantic robotic grasping of household objects. *arXiv preprint arXiv:1809.10790*, 2018. 3

[112] Eugene Valassakis, Georgios Papagiannis, Norman Di Palo, and Edward Johns. Demonstrate once, imitate immediately (dome): Learning visual servoing for one-shot imitation learning. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8614–8621. IEEE, 2022. 2

[113] Mel Vecerik, Carl Doersch, Yi Yang, Todor Davchev, Yusuf Aytar, Guangyao Zhou, Raia Hadsell, Lourdes Agapito, and Jon Scholz. Robotap: Tracking arbitrary points for few-shot visual imitation. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5397–5403. IEEE, 2024. 2

[114] Chen Wang, Linxi Fan, Jiankai Sun, Ruohan Zhang, Li Fei-Fei, Danfei Xu, Yuke Zhu, and Anima Anandkumar. Mimicplay: Long-horizon imitation learning by watching human play. *arXiv preprint arXiv:2302.12422*, 2023. 1, 2

[115] Yi Wang, Yinan He, Yizhuo Li, Kunchang Li, Jiashuo Yu, Xin Ma, Xinhao Li, Guo Chen, Xinyuan Chen, Yaohui Wang, et al. Internvid: A large-scale video-text dataset for multimodal understanding and generation. In *The Twelfth International Conference on Learning Representations*, 2023. 5, 20

[116] Bowen Wen, Jonathan Tremblay, Valts Blukis, Stephen Tyree, Thomas Muller, Alex Evans, Dieter Fox, Jan Kautz, and Stan Birchfield. Bundlesdf: Neural 6-dof tracking and 3d reconstruction of unknown objects. *CVPR*, 2023. 4, 16

[117] Bowen Wen, Wei Yang, Jan Kautz, and Stan Birchfield. Foundationpose: Unified 6d pose estimation and tracking of novel objects. *arXiv preprint arXiv:2312.08344*, 2023. 1, 3, 4, 19

[118] Albert Wu, Ruocheng Wang, Sirui Chen, Clemens Eppner, and C Karen Liu. One-shot transfer of long-horizon extrinsic manipulation through contact retargeting. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 13891–13898. IEEE, 2024. 3

[119] 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. 1

[120] Haofei Xu, Jing Zhang, Jianfei Cai, Hamid Rezatofighi, and Dacheng Tao. Gmflow: Learning optical flow via global matching. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8121–8130, 2022. 17

[121] Mengda Xu, Zhenjia Xu, Cheng Chi, Manuela Veloso, and Shuran Song. Xskill: Cross embodiment skill discovery. In

Conference on robot learning, pages 3536–3555. PMLR, 2023. 2

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

[123] Mengjiao Yang, Yilun Du, Kamyar Ghasemipour, Jonathan Tompson, Dale Schuurmans, and Pieter Abbeel. Learning interactive real-world simulators. *arXiv preprint arXiv:2310.06114*, 1(2):6, 2023. 3

[124] Xindi Yang, Baolu Li, Yiming Zhang, Zhenfei Yin, Lei Bai, Liqian Ma, Zhiyong Wang, Jianfei Cai, Tien-Tsin Wong, Huchuan Lu, et al. Vlipp: Towards physically plausible video generation with vision and language informed physical prior. *arXiv e-prints*, pages arXiv–2503, 2025. 1

[125] Seonghyeon Ye, Joel Jang, Byeongguk Jeon, Sejune Joo, Jianwei Yang, Baolin Peng, Ajay Mandlekar, Reuben Tan, Yu-Wei Chao, Bill Yuchen Lin, et al. Latent action pretraining from videos. *arXiv preprint arXiv:2410.11758*, 2024. 1

[126] Tianhe Yu, Chelsea Finn, Annie Xie, Sudeep Dasari, Tianhao Zhang, Pieter Abbeel, and Sergey Levine. One-shot imitation from observing humans via domain-adaptive meta-learning. *arXiv preprint arXiv:1802.01557*, 2018. 2

[127] Chengbo Yuan, Chuan Wen, Tong Zhang, and Yang Gao. General flow as foundation affordance for scalable robot learning. *arXiv preprint arXiv:2401.11439*, 2024. 2

[128] Kevin Zakka, Andy Zeng, Pete Florence, Jonathan Tompson, Jeannette Bohg, and Debidatta Dwibedi. Xirl: Cross-embodiment inverse reinforcement learning. In *Conference on Robot Learning*, pages 537–546. PMLR, 2022. 2

[129] Qihang Zhang, Shuangfei Zhai, Miguel Angel Bautista, Kevin Miao, Alexander Toshev, Joshua Susskind, and Jiatao Gu. World-consistent video diffusion with explicit 3d modeling. *arXiv preprint arXiv:2412.01821*, 2024. 1

[130] Zhengyou Zhang. A flexible new technique for camera calibration. *IEEE Transactions on pattern analysis and machine intelligence*, 22(11):1330–1334, 2000. 16, 17

[131] Tony Z Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. *arXiv preprint arXiv:2304.13705*, 2023. 9

[132] Haoyu Zhen, Qiao Sun, Hongxin Zhang, Junyan Li, Siyuan Zhou, Yilun Du, and Chuang Gan. Tesseract: Learning 4d embodied world models. 2025. 3

[133] Huayi Zhou, Ruixiang Wang, Yunxin Tai, Yueci Deng, Guiliang Liu, and Kui Jia. You only teach once: Learn one-shot bimanual robotic manipulation from video demonstrations. *arXiv preprint arXiv:2501.14208*, 2025. 2

[134] Jiaming Zhou, Teli Ma, Kun-Yu Lin, Zifan Wang, Ronghe Qiu, and Junwei Liang. Mitigating the human-robot domain discrepancy in visual pre-training for robotic manipulation. *arXiv preprint arXiv:2406.14235*, 2024. 1

[135] Junzhe Zhu, Yuanchen Ju, Junyi Zhang, Muhan Wang, Zhecheng Yuan, Kaizhe Hu, and Huazhe Xu. Densematcher: Learning 3d semantic correspondence for category-level manipulation from a single demo. *arXiv preprint arXiv:2412.05268*, 2024. 2

Appendix

A. Best Practices for Video Generation

We found that the following practices lead to reliable video generation: (1) having a clean background without visual distractions, (2) minimizing the number of distractor objects in the scene, (3) ensuring objects are reasonably large and viewed from a natural, human-like perspective, (4) ensuring there is one clearly identifiable task that can be performed, (5) using simple and concise text prompts, and (6) setting the relevance factor to 0.7 with the negative prompt "fast motion" led to the most reliable video generations.

B. Prompting for Video Filtering and Filtering Statistics

The prompt for GPT o1-based filtering is shown in Figure 10. We provide GPT o1 with the prompt, the video summary—constructed by vertically concatenating evenly sampled frames from the video—and the language command (e.g., "pour water"). GPT o1 then responds with "Yes" or "No" to indicate whether the specified task is successfully performed in the video. The filtering success rates are: 83% for pouring, 66% for lifting, 55% for placing, and 45% for sweeping.

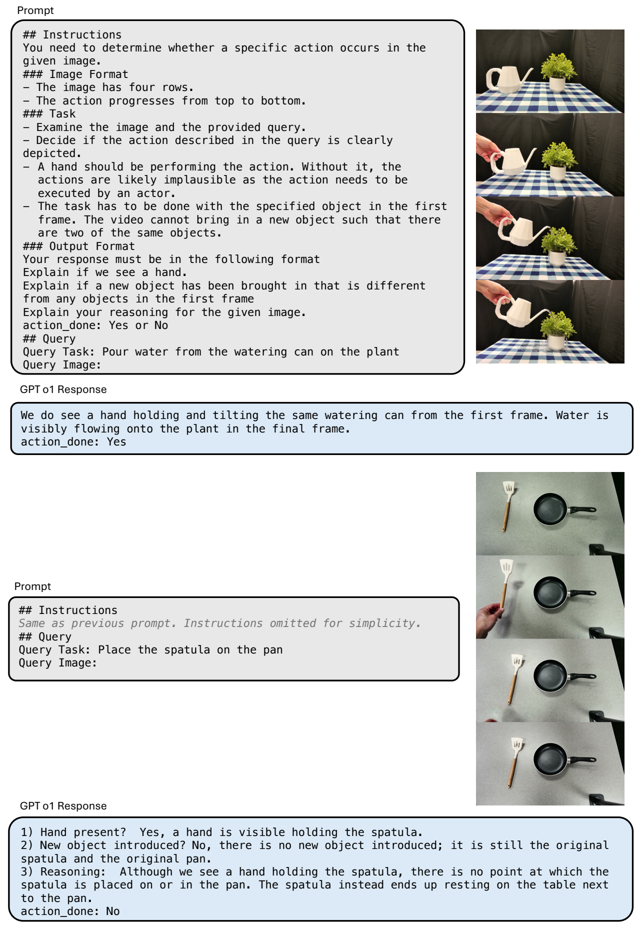


Figure 10. **Examples of prompting GPT o1 to filter generated videos.** We sample generated video frames and prompt GPT o1 to assess whether the specified task is performed successfully in the video.

C. Mesh-Free Object Tracking

We experiment with a mesh-free object tracking version of our method. Specifically, we use BundleSDF [116], which jointly performs 6-DoF object tracking and reconstruction from RGBD observations. For the *pouring* task, we evaluate our method using trajectories obtained via BundleSDF over 10 trials and observe a success rate of (90%), matching our default tracking setup. While the BundleSDF paper reports real-time capabilities, we found that its official implementation takes approximately 30 minutes to process each video in practice, which limits its applicability for real-time deployment. In contrast, our default tracker operates in real-time, enabling closed-loop execution and recovery from disturbances as discussed in Sec. 4.6. While the BundleSDF paper reports real-time capabilities, we observed significantly higher runtimes in practice with the official implementation. We expect that future advances in model-free tracking will address these efficiency bottlenecks, allowing for real-time mesh-free deployment.

D. Smoothing Object Trajectories

To reduce noise and jitter in the estimated object poses, we apply a moving average filter with a fixed sliding window (centered on each point), separately to the position and orientation components. Translations are smoothed independently along each axis, while orientation undergoes a similar process after conversion from quaternions to rotation vectors. This approach mitigates abrupt changes, resulting in a more stable and realistic object trajectory with smoother transitions.

E. Description of Baselines

Track2Act [15]: We adapt Track2Act's procedure to our setup preserving its core idea of object-centric trajectory estimation from point tracks. Track2Act generates a future interaction plan by predicting 2D point trajectories (using a DiT-based diffusion model) between an initial image and a goal image, then recovers a sequence of 3D object transforms via Perspective-and-Point (PnP) [130].

To integrate this into our pipeline, we use their published checkpoint but modify the input formulation–while the initial image remains identical to our real camera's view, the goal image is taken from the last frame of a generated video rather than being physically captured. We then use PnP on the predicted point tracks along with the initial depth image to estimate the object's rigid motion across frames, thereby defining the end-effector trajectory. We use interpolation between consecutive poses because Track2Act generates only a sparse set of frames, and denser sampling is needed for smooth trajectory estimation and execution. However, we do not include Track2Act's closed-loop residual policy correction, focusing solely on open-loop 6D object-pose estimation and execution. This adaptation allows us to directly evaluate how well a vision-based, open-loop approach generalizes to our setting without additional corrections.

Gen2Act [14]: Gen2Act introduces a video-conditioned policy learning framework that first generates a human video using a video generation model from a scene image and a task description. The system then extracts object tracks using BootsTAP [28], and trains a policy using behavior cloning with an auxiliary track prediction loss and offline robot demonstrations. At inference, Gen2Act only uses the generated video and the learned policy to predict robot actions.

Our approach presents a simplified adaptation of this framework that removes the need for behavior cloning, and offline demonstrations. Instead of using the extracted tracks as an auxiliary loss, we directly process them for pose estimation. To recover 3D object positions, we leverage an initial depth image corresponding to the scene image, allowing us to obtain depth values for the extracted 2D tracks. We apply RANSAC filtering to remove outlier track points and then use the Perspective-n-Point (PnP) [130] to estimate the object's 6DoF pose. This adaptation preserves the core idea of leveraging video and track-based signals while eliminating the need for supervised policy learning.

AVDC [60]: The AVDC approach models action trajectories by synthesizing a task-driven video (using a trained text-conditioned video generation model) and using optical flow from GMFlow [120] to estimate dense pixel correspondences. It then reconstructs 3D object motion using an optimization step that refines pose estimates based on the tracked flow and depth information. To improve robustness, AVDC also includes a replanning mechanism that re-executes the pipeline when predicted motion stagnates.

Since the trained text-conditioned video generation model did not generalize well to our setup, we instead use the same generated video as in other experiments to ensure a fair comparison. While we do not employ AVDC's replanning strategy, we predict object poses using a similar optimization framework based on flow and depth information.

4D-DPM [58]: 4D-DPM is designed to track the 3D motion of articulated object parts from a single video. It first constructs a 3D Gaussian splatting [57] representation of the scene to capture object features, then applies GARField [59] to cluster the Gaussians into discrete object components. In our adaptation, we modify this approach to operate on entire objects rather than individual parts. Specifically, we set the clustering parameters to treat the object as a single entity, ensuring that motion estimation is performed at the object level rather than segmenting it into multiple parts. This allows us to track and execute trajectories for the whole object.

F. ReKep Predictions and Executions

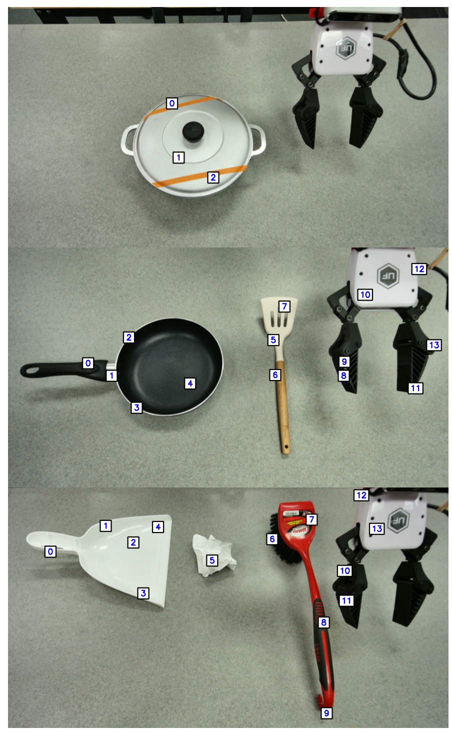
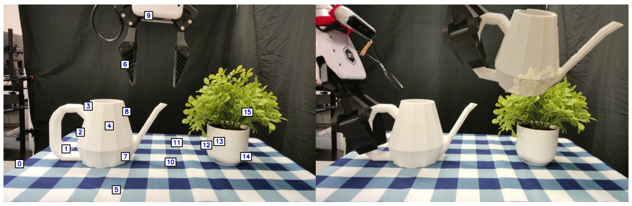


Figure 11. **Examples of ReKep's Keypoint Locations.** The keypoint placements are often suboptimal, except for sweeping task, where the keypoints are reasonable.

A detailed example of ReKep's keypoint and VLM predictions for pouring task is shown in Fig. 12. The VLM first predicts to grasp the watering can at keypoint 1. For the transport phase, it instructs moving keypoint 8 above keypoint 15, while keeping its height above keypoint 7. For the pouring action, keypoint 8 remains above 15 (to place the spout over the plant) and above keypoint 4 (to induce tilting). The resulting robot execution fails. We attribute most ReKep failures to inaccurate keypoint predictions, as shown in Fig. 11. In the lid image, there is no keypoint at the handle of the lid. In the placing task, keypoints cluster around pan corners. For the sweeping task, the keypoints are generally well-placed, and executions succeeded. Because the initial keypoints are suboptimal, downstream VLM predictions are also inaccurate.




```
def stage1_subgoal_constraint1(end_effector, keypoints):
    """Align the end-effector with the watering can handle (keypoint 1)."""
    handle_point = keypoints[1]
    cost = np.linalg.norm(end_effector - handle_point)
    return cost
```

```
def stage2_path_constraint1(end_effector, keypoints):
    """The robot must still be grasping the watering can handle (keypoint 1)."""
    return get_grasping_cost_by_keypoint_idx(1)

def stage2_path_constraint2(end_effector, keypoints):
    """The watering can must stay upright to avoid spilling (spout above base)."""
    spout = keypoints[8]
    base = keypoints[7]
    # Ensure spout is not lower than base
    cost = max(0, spout[2] - base[2])
    return cost
```

```
def stage2_subgoal_constraint1(end_effector, keypoints):
    """The watering can spout (keypoint 8) needs to be 10cm above the
    center of the plant (keypoint 15)."""
    spout = keypoints[8]
    plant = keypoints[15]
    offsetted_point = plant + np.array([0, 0, 0.10])  # 10cm above the plant
    cost = np.linalg.norm(spout - offsetted_point)
    return cost
```

```
def stage3_path_constraint1(end_effector, keypoints):
    """The robot must still be grasping the watering can handle (keypoint 1)."""
    return get_grasping_cost_by_keypoint_idx(1)

def stage3_path_constraint2(end_effector, keypoints):
    """The watering can spout (keypoint 8) is directly above the plant (keypoint 15)
    in x-y plane."""
    spout = keypoints[8]
    plant = keypoints[15]
    cost = np.linalg.norm(spout[:2] - plant[:2])
    return cost
```

```
def stage3_subgoal_constraint1(end_effector, keypoints):
    """The watering can spout (keypoint 8) needs to be 10cm above the
    center of the plant (keypoint 15)."""
    spout = keypoints[8]
    plant = keypoints[15]
    offsetted_point = plant + np.array([0, 0, 0.1])  # 10cm above the plant
    cost = np.linalg.norm(spout - offsetted_point)
    return cost

def stage3_subgoal_constraint2(end_effector, keypoints):
    """The watering can spout (keypoint 8) must be tilted below the base (keypoint 4) to pour."""
    spout = keypoints[8]
    base = keypoints[4]
    # Ensure spout is lower than base
    cost = max(0, spout[2] - base[2])
    return cost
```

Figure 12. **ReKep's output for the pouring task and the resulting robot execution (top-right).** The VLM predictions on the generated keypoints lead to failed execution.

G. Limitation of Tracking with Point Tracks

All point tracks fail under extreme rotations, as initially visible points often become occluded. This is a fundamental limitation of any correspondence-based tracking method that relies solely on visible surface features. We show this failure in Fig. 13. As the object rotates, most initial points are lost, resulting in insufficient 2D-3D correspondences to solve a stable PnP problem. This degrades pose estimation quality, leading to large drift or abrupt jumps in estimated object motion. Such instability cascades into robot execution errors, often causing the robot to fail at the task altogether. As a result, both variants of Gen2Act—despite stronger tracking backbones like CoTracker—still fail under large out-of-plane rotations. In contrast, RIGVid's model-based 6D tracking handles these situations more robustly, as it uses full-object geometry and SE(3) filtering to maintain stable trajectories.

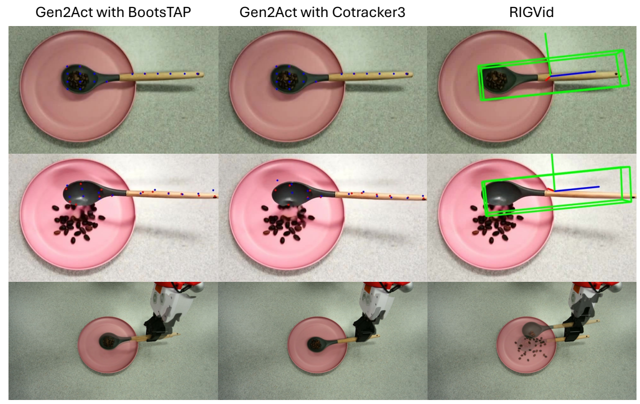


Figure 13. **Gen2Act with BootsTAP, CoTracker, and RIGVid.** Blue points denote the tracked points used for PnP; red points represent the reprojected 3D points. For a good PnP solution, these should align, as seen in the first frame. For Gen2Act, the blue points drift significantly from the red ones in later frames, indicating failure in pose estimation due to tracking loss, which leads to failed robot execution.

H. Additional Robustness Examples

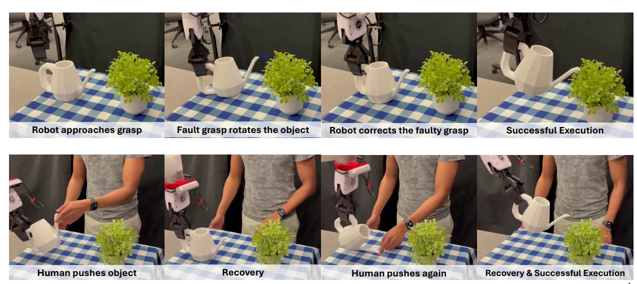


Figure 14. **Additional examples of RIGVid's robustness.** In the top row, RIGVid recovers from a faulty initial grasp by reorienting the object before continuing execution. In the bottom row, it corrects for external disturbances on the object when a human pushes the object mid-execution, realigning and successfully completing the task.

Examples of RIGVid's robustness are shown in Fig. 14. In the first row, the robot initially grasps the object, but due to a misaligned grasp, the object rotates unexpectedly. The robot compensates by rotating the object back to the correct orientation and then resumes the planned trajectory, ultimately completing the task successfully. In the bottom row, a human perturbs the object during execution while it is held by the robot. RIGVid detects the resulting change in the relative transformation and automatically re-aligns the object before continuing. When the human intervenes a second time, RIGVid again corrects the deviation, ultimately leading to successful task completion.

I. Errors from Depth Estimation

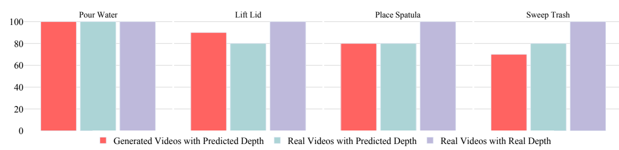


Figure 15. **Impact of Depth Estimation Errors on RIGVid performance.** Errors in monocular depth estimation result in worse performance of generated and real videos. RIGVid achieves perfect success across all tasks with real videos and real depth.

In Fig. 15, we isolate the impact of depth estimation errors. Robot executions on real videos with real depth (captured using an RGBD camera) achieve a 100% success rate, whereas executions from real videos with generated depth result in 85% average success. Similarly, executions from Kling V1.6-generated videos with generated depth also achieve 85% success, suggesting that the primary source of error lies in monocular depth estimation. Upon inspection, we observe two common undesirable behaviors in the predicted depth: inaccurate depth values and temporal flickering. An example of inaccurate depth is shown in Fig. 16. In the generated video, when the spatula is brought close to the camera, the depth changes by only 6.8 cm, which is visibly inconsistent with the video and likely much smaller than the real-world change. Inaccuracies also occur in real videos, as shown in the figure—the head of the spatula is estimated to be far from the camera, despite appearing close, revealing another failure mode in monocular depth estimation. An illustration of flickering is shown in Fig. 17. Although the position of the watering can relative to the camera remains nearly unchanged across three consecutive frames, the estimated depth varies significantly. In particular, the zoomed-in region on the right shows the can appearing much whiter than on the left, indicating a substantial change in predicted depth. The average depth of the can changes from 40.1 cm to 38.2 cm–a 1.9 cm difference over just 0.066 seconds–which is physically implausible for the generated video. We find similar flickering behavior in real videos as well, where the depth changes from 43.2 cm to 40.9 cm in the given example–a 2.3 cm difference.

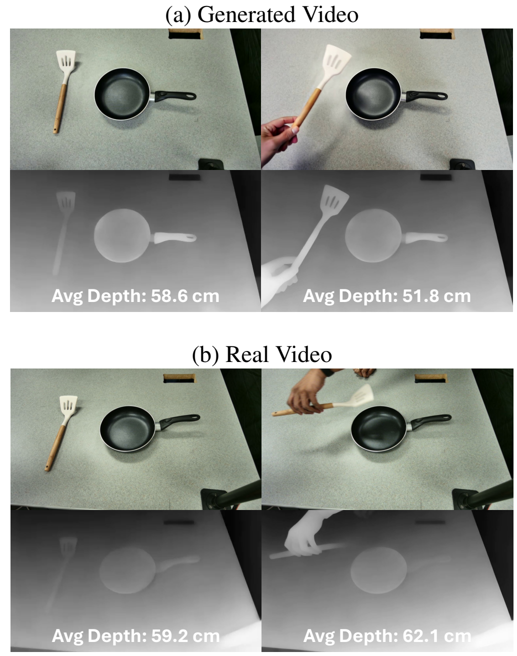


Figure 16. **Errors in Monocular Depth Estimation.** In the generated video (top), the depth of the spatula changes only slightly despite a large visual change. In the real video (bottom), the spatula's head is predicted to lie farther away, contradicting the visual appearance.

J. Choice between MegaPose and FoundationPose

We compare the stability of trajectories obtained from MegaPose [63] and FoundationPose [117] by computing the translational and rotational RMS jitter. For each method, we apply a Gaussian smoothing filter ($\sigma = 2$ frames) to the raw SE(3) pose sequences, compute the residual between the original and smoothed trajectories, and then calculate:

$$\text{jitter}_{\text{trans}} = \sqrt{\frac{1}{N}\sum_{t=1}^{N} \|\Delta \mathbf{t}_t\|^2}, \quad \text{jitter}_{\text{rot}} = \sqrt{\frac{1}{N}\sum_{t=1}^{N} \theta_t^2},$$

where $\Delta \mathbf{t}_t$ is the translational residual at frame t, and θ_t is the angular magnitude (in radians) of the relative rotation $R_{\text{smooth}}^{-1} R_{\text{raw}}$, converted to degrees. We average these metrics over ten pouring trajectories extracted from generated videos.

MegaPose yields an average translational RMS jitter of 0.0045m and rotational RMS jitter of 37.47°, whereas FoundationPose achieves 0.0029m translational and 14.31° rotational jitter. These results demonstrate that FoundationPose produces significantly smoother and more stable trajectories. Additionally, it allows for real-time tracking during the execution, allowing us to make RIGVid robust to external disturbances.

CVPR
#*****

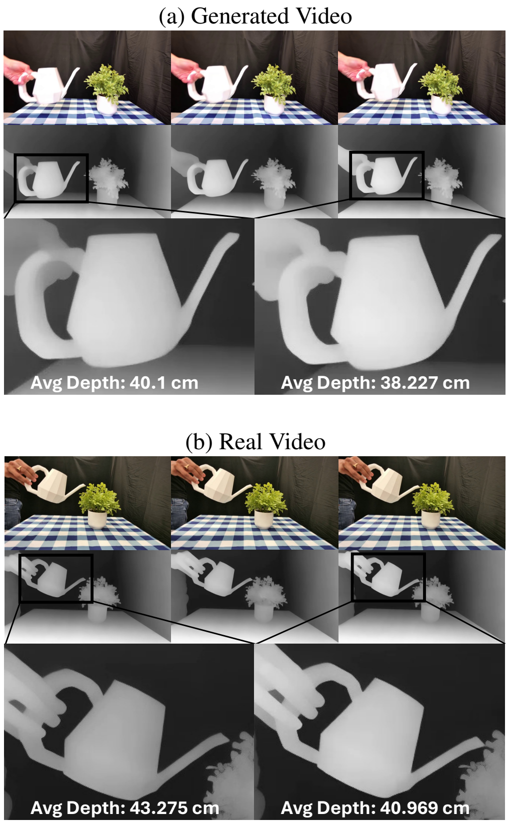


Figure 17. **Flickering in Depth Prediction.** We show three consecutive frames of the video and its corresponding predicted depth. The depth of the watering can change noticeably across frames—appearing significantly whiter in the third frame despite minimal actual motion. We observe this behavior in both generated and real videos.

K. Comparing Video Generative Models

To further assess video quality, we report VBench++ [50] metrics in Table 2 and explain them below. The numbers in the table are scaled 100× for easier interpretation. We collect these metrics on 40 randomly selected and unfiltered videos per model, 10 for each of the four tasks. Kling v1.6 outperformed the other models on most metrics but performed similarly or worse in video-text consistency and dynamic degree. Human evaluations discussed in Sec. 4.2 suggest that the video-text consistency and I2V subject consistency are not reliable indicators of whether a generated video correctly follows a given command. Sora scored high on dynamic degree, likely due to its tendency to drastically alter the scene, resulting in exceptionally large motions. Generated videos from these models and their corresponding metrics are shown in Fig. 18 and further details on these metrics can be found the next section.

VBench++ Metric Definitions:

- **Subject Consistency.** Subject consistency describes whether subjects' appearance remain consistent, which is computed by DINOv1 [20] similarities across video frames.
- **Background Consistency.** Background temporal consistency by CLIP [98] similarities across frames.
- **Motion Smoothness.** Evaluates smoothness of videos by utilizing video frame interpolation model AMT [70].
- **Dynamic Degree.** Describes whether the video contains large motions as a binary metric.
- **Aesthetic Quality.** Human perceived artistic and beauty value such as photo-realism, layout and color harmony.
- **Imaging Quality.** Assesses the presence of distortion in a video, such as noisiness, blurriness, and over-exposure.
- **Video-Text Consistency.** Text-to-video alignment score calculated by ViCLIP [115].
- **I2V Subject Consistency.** Similarity between subjects in input image and each video frame, as well as similarity between consecutive frames. Features are extracted from DINOv1 [20].

Metrics	**Video Generation Models**			**Human Demos**
	Kling V1.6	Kling V1.5	Sora	
Subject Consistency	**96.34**	91.66	83.09	94.91
Background Consistency	**96.64**	93.97	89.34	95.00
Motion Smoothness	**99.68**	99.57	99.06	99.51
Dynamic Degree	52.5	57.5	**70.0**	80.0
Aesthetic Quality	**51.75**	49.77	46.22	49.30
Imaging Quality	**72.80**	71.48	68.68	72.52
Video-Text Consistency	22.01	**22.61**	21.42	21.57
I2V Subject Consistency	**97.88**	95.96	89.09	97.89

Table 2. **Video generation quality metrics for real human demonstration videos and different models.** Higher values indicate better quality. Kling v1.6 performs comparably to or surpasses other models on most metrics.

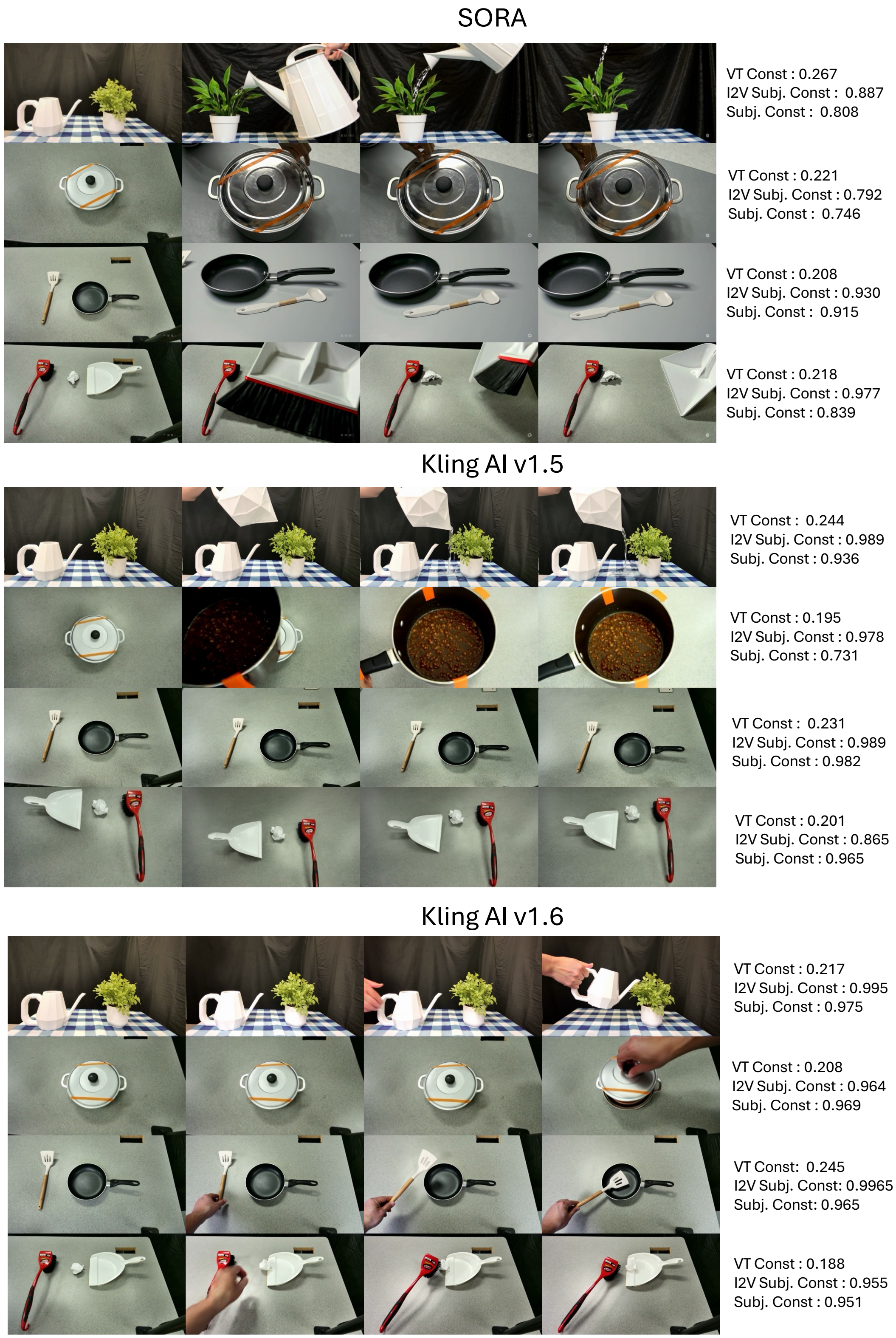


Figure 18. **Qualitative Comparison of Different Video Generative Models.** Videos generated by three models are shown in evenly sampled frames. We show VBench++ [50] metrics including video-text consistency, image-to-video subject consistency, and subject consistency.

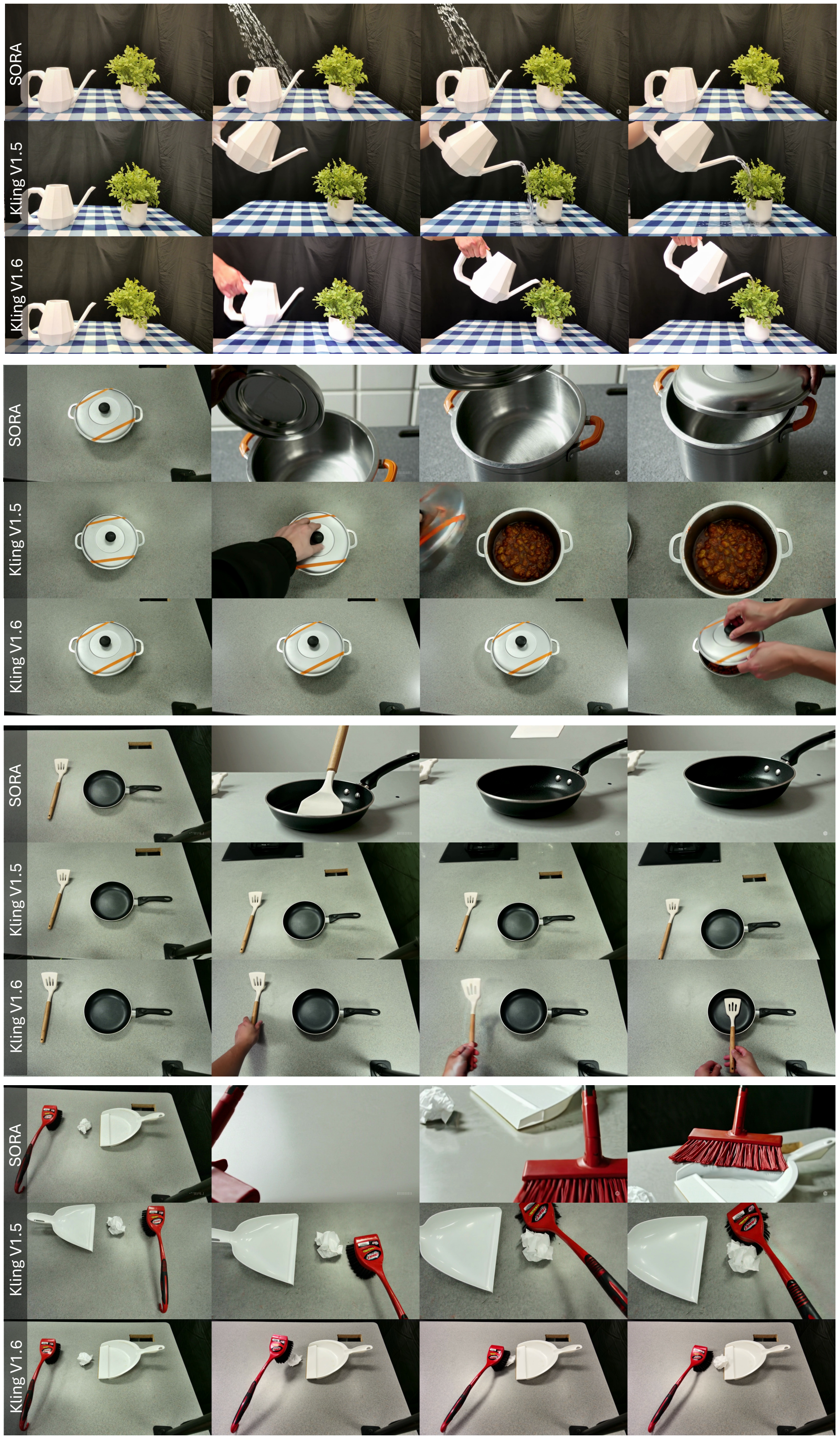


Figure 19. **Qualitative comparison of video generation.**