Dream.exe: Can Video Generation Models Dream Executable Robot Manipulation?

Rui Zhao [*1] Kaiming Yang [*1] Jifeng Zhu [†1] Siyang Chen [†1] Ziqi Wang [1] Weijia Wu [1]
Kevin Qinghong Lin [2] Heng Wang [3] Mike Zheng Shou [1]

Abstract

Video generation models have made impressive strides in synthesizing visually compelling content, yet their outputs remain confined to the virtual domain. A natural question follows: how well do these models reflect the physical world when their generated videos leave the screen and enter reality? We propose robotic manipulation as a concrete, measurable window onto this question: if a model has truly internalized physical laws, the motion it depicts should translate into executable robot behavior. We introduce Dream.exe, an evaluation framework that operationalizes this criterion through a video-to-execution pipeline. Given a scene image and a task description, Dream.exe synthesizes a manipulation video, converts the generated motion into robot trajectories, and executes them in a physics simulator, yielding a grounding signal that purely visual metrics cannot offer. Using this pipeline, we evaluate 8 models spanning frontier closed-source generators, open-source generators, and robot-specific models. Our benchmark covers 101 manually curated manipulation tasks at three levels of physical complexity, measured across visual quality, trajectory fidelity, and execution success. Encouragingly, several models achieve measurable execution success, suggesting that generative priors learned from internet-scale data already encode meaningful physical knowledge. Yet visual quality proves a poor predictor of executability, exposing a dimension of model capability that standard visual evaluations do not capture.

[*]Equal contribution. [†]Equal contribution (second authors). [1]Show Lab, National University of Singapore [2]University of Oxford [3]Tencent. Correspondence to: Mike Zheng Shou <mike.zheng.shou@gmail.com>.

Published as a paper at the 1st FoGen workshop, ICML 2026, Seoul, South Korea, 2026. Copyright 2026 by the author(s).

1. Introduction

Recent years have seen video generation models cross a qualitative threshold. Models such as Wan (Wan et al., 2025), Kling (Team et al., 2025), Imagen video (Ho et al., 2022a), and Veo (Google DeepMind, 2025) can synthesize photorealistic videos of fluid dynamics, human motion, and complex object interactions with a fidelity that was out of reach just two years ago. The community has begun to interpret this visual fluency as evidence of something deeper: that large-scale video generation models are learning implicit world models (Brooks et al., 2024; Kang et al., 2024; Ha & Schmidhuber, 2018), acquiring internal representations of physical causality from the statistical regularities of internet-scale data. This interpretation has become a foundation for an active line of research in robot learning, where generated videos are proposed as scalable behavioral priors that could reduce dependence on costly physical demonstrations (Du et al., 2023; Jang et al., 2025; Ye et al., 2026a; Liang et al., 2025).

The world model hypothesis, however, has never been directly tested. Standard video generation benchmarks evaluate models on visual quality, temporal consistency, and human aesthetic ratings, all of which measure how natural a video looks without asking whether its implied motions could actually accomplish the depicted task in the physical world. Under these metrics, a model that generates a robot arm gracefully passing through a table is indistinguishable from one whose motions are physically valid. As models grow larger and more visually convincing, the field has no principled way to know whether their physical knowledge and learning are keeping pace.

We argue that robotic manipulation offers the right test. The criterion is simple and unambiguous: if a model has internalized the physical laws governing a manipulation task, the trajectory implied by its generated video should produce task success when executed by a robot. We build Dream.exe on this intuition, treating task success in simulation as the grounding signal rather than only relying on perceptual quality scores.

As illustrated in Figure 2, Dream.exe operationalizes this

criterion at scale. Each model is given an initial scene image and a natural-language task description and asked to generate a manipulation video. The video is then assessed along three tracks: visual evaluation of robot stability, physical plausibility, and task adherence; a five-step video-to-trajectory extraction pipeline and the corresponding trajectory evaluation; and closed-loop execution in a physics simulator that yields fine-grained success scores and an overall task success rate.

Bridging video and physical execution is non-trivial. A generated video encodes motion only implicitly, in the form of pixel-level appearance changes, without any explicit representation of 3D geometry, contact forces, or gripper state. To recover executable trajectories, we develop a video-to-execution pipeline that lifts 2D end-effector motion into world-frame 3D trajectories using monocular depth estimation and known camera parameters, infers gripper timing from the interaction context, and converts the result into a structured action stream that a robot controller can follow. The task suite is built on 101 manually curated episodes from RoboCasa365 (Nasiriany et al., 2026), stratified into three difficulty levels ranging from single-object atomic manipulation to multi-stage composite tasks. Together, these three axes of assessment provide a capability profile that no prior benchmark offers.

Using Dream.exe, we evaluate 8 models spanning frontier closed-source generators, open-source generators, and a robot-specific policy model. Our experiments surface three findings. First, several models achieve measurable execution success rates, suggesting that generative priors trained on internet-scale data do encode meaningful physical knowledge. Second, visual quality is a poor predictor of executability: models that lead on standard visual metrics frequently fail in execution, while models with modest visual scores can produce physically valid trajectories. Third, the robot-specific policy model does not consistently outperform general generators, as the latter generalize better across diverse tasks and camera viewpoints.

Dream.exe will be open-sourced to support future work at the intersection of video generation and robot learning. Our contributions are summarized as follows:

- We introduce Dream.exe, the first benchmark to evaluate video generation models on physical executability, using task success in simulation as the primary criterion rather than perceptual quality scores.
- We propose a three-track evaluation protocol: visual assessment of generated videos; video-to-trajectory extraction pipeline and trajectory evaluation; and closed-loop execution in a physics simulator and evaluation.
- We provide a comprehensive empirical analysis of 8 video generation models spanning closed-source, open-source, and robot-specific models, characterizing systematic failure modes and revealing that visual quality is a poor predictor of physical executability.

2. Related Work

Video Generation Models. Video generation has evolved rapidly from early diffusion-based approaches into a diverse ecosystem of powerful models. Ho et al. (Ho et al., 2022b) established the core paradigm of applying diffusion models to video; Make-A-Video (Singer et al., 2022) demonstrated text-to-video generation without paired supervision; and Stable Video Diffusion (Blattmann et al., 2023) showed that large-scale image-to-video pretraining yields strong motion priors. More recent open-weight models such as CogVideoX (Yang et al., 2025) and HunyuanVideo (Kong et al., 2024) match or surpass earlier proprietary systems in quality and efficiency. On the frontier, Sora (Brooks et al., 2024) reframed video generation as world simulation, followed by Movie Gen (Polyak et al., 2024), and the current generation of image-to-video models evaluated in this work: Kling (Team et al., 2025), Wan (Wan et al., 2025), SeedDance (Seedance et al., 2026), Veo (Google DeepMind, 2025), and LTX-Video (HaCohen et al., 2025). Despite their visual fluency, these models are evaluated exclusively on perceptual quality metrics; whether their generated motions are physically executable has not been tested.

Video Generation Benchmarks. Standard benchmarks evaluate video models on visual and semantic quality. EvalCrafter (Liu et al., 2024b) proposes a holistic framework spanning visual quality, motion quality, and text-video alignment. VBench (Huang et al., 2024) decomposes evaluation into fine-grained dimensions including temporal consistency, subject identity, and aesthetics. T2V-CompBench (Sun et al., 2025) focuses on compositional reasoning over spatial relations, attributes, and actions. These benchmarks measure how natural a video looks; they do not probe whether its physics is correct. A growing body of work has begun to fill this gap. VideoPhy (Bansal et al., 2024) and PhyGenBench (Meng et al., 2024) test whether generated videos depict physically plausible phenomena, using VLM-based scorers and human raters as judges. WorldSimBench (Qin et al., 2024) adds an implicit manipulative evaluation that asks whether a video generation model could support downstream task execution via a learned policy. MIND (Ye et al., 2026b) evaluates memory consistency and action control in world models, testing whether generated scenes remain consistent under closed-loop revisiting. Kang et al. (Kang et al., 2024) probe model adherence to concrete physical laws and find systematic failures across all current generators.

Despite this progress, none of these works closes the loop

with a real robot controller: measuring physical plausibility through visual classifiers differs categorically from asking whether a generated trajectory succeeds when executed in a physics simulator. Dream.exe makes physical executability the primary metric, directly bridging this gap.

Robot Learning from Video. The idea of using video as a source of robot behavioral knowledge spans imitation from human demonstrations and pre-training on internet-scale video (Wu et al., 2023). An influential early direction treats video generation itself as the policy: UniPi (Du et al., 2023) frames planning as text-conditioned video generation; SuSIE (Black et al., 2023) synthesizes visual subgoals via image-editing diffusion models for hierarchical control; and Dreamitate (Liang et al., 2024) distills visuomotor policies directly from generated demonstrations. The most recent wave turns video world models into zero-shot and few-shot robot policies: Cosmos Policy (Kim et al., 2026) fine-tunes a video foundation model on robot demonstration data for visuomotor control; DreamGen (Jang et al., 2025) generates neural trajectories conditioned on novel environments to unlock out-of-distribution generalization; DreamZero (Ye et al., 2026a) embeds action generation into the video diffusion process, achieving zero-shot policy transfer across embodiments; VideoVLA (Shen et al., 2025) jointly models video, language, and action to turn video generators into generalizable robot manipulators; and Video Generators are Robot Policies (Liang et al., 2025) proposes a modular framework in which a single video generator serves as the policy backbone for a wide range of manipulation skills. Trajectory extraction methods recover executable actions from video without explicit labels: Video Prediction Policy (Hu et al., 2024) decodes implicit robot control signals from video diffusion representations; and Dream2Flow (Dharmarajan et al., 2025) lifts 3D object flow directly from generated videos for open-world manipulation. Our work differs fundamentally from all of the above: Dream.exe treats video generation as a *fixed test subject*, evaluating the physical content of generation as-is via execution in a physics simulator built on RoboCasa365 (Nasiriany et al., 2026) and robosuite (Zhu et al., 2020).

3. Dream.exe: Benchmark Design

3.1. Task Suite

A benchmark for physical executability must ensure that each task scenario is strictly reproducible: the same initial scene state must be recoverable on demand, so that different video generation models can be compared on an equal footing. We build our task suite on top of RoboCasa365 (Nasiriany et al., 2026), a large-scale simulation framework comprising 365 everyday manipulation tasks.

Data curation. Not all episodes are suitable for benchmarking video generation models. Cluttered viewpoints obscure end-effector motion; ambiguous object identities make trajectory evaluation ill-defined; and certain tasks require base navigation that the current extraction pipeline does not support. We therefore conducted a substantial manual curation effort: each candidate episode was reviewed for camera suitability, object visibility, trajectory clarity, and semantic unambiguity. Camera viewpoints were individually tuned to maximize both object and end-effector visibility in the rendered frame. After filtering, around 101 episodes were selected, as shown in Fig. 1, and are organized into a benchmark dataset with unified metadata, including the initial image and textual task prompt.

Three-level difficulty taxonomy. The tasks are stratified into three levels of increasing complexity, designed to probe different aspects of physical complexity in generated videos, as shown in Fig. 1.

Level 1: Atomic single-object manipulation. Each task involves a single object and a single continuous interaction primitive, such as pick-and-place, articulated joint actuation, button press, or knob rotation. These tasks require the model to generate geometrically consistent end-effector motion and correct grasp-release timing, but do not demand reasoning about object-to-object relationships.

Level 2: Multi-object interaction. Tasks at this level involve two or more objects whose states are coupled. Representative examples include placing one object into a container, stacking objects, or transferring contents between containers. Success requires that the generated video correctly represent the spatial relationships between objects and the sequential dependency between manipulation events.

Level 3: Multi-stage composite tasks. Each task at this level decomposes into two or more semantically distinct stages, such as opening a drawer before retrieving an object, or turning a stove knob before moving a cooking vessel. These tasks test whether a video generation model can maintain physical coherence across a long task horizon, correctly sequencing sub-goals and transitions between interactions.

3.2. Models Evaluated

A central goal of Dream.exe is to provide a broad and representative evaluation that spans the current landscape of video generation. We include three categories of models, with detailed generation settings reported in Appendix B.

Frontier closed-source generators. We evaluate five state-of-the-art commercial image-to-video models: Hailuo 2.3 (MiniMax, 2025) from MiniMax, Kling 3.0 (Team et al., 2025; Kuaishou, 2026) from Kuaishou, Wan 2.7 (Wan et al., 2025; Lab, 2026) from

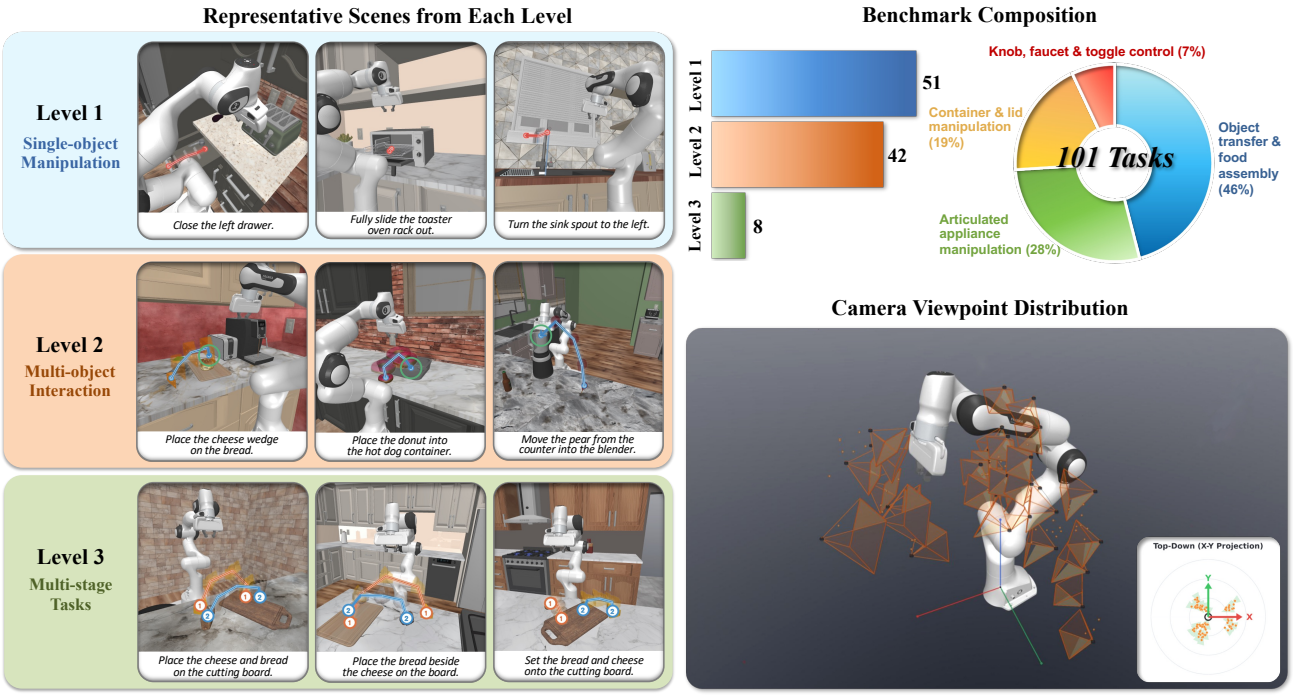

Figure 1. **Overview of the Dream.exe task suite.** Left: representative scenes and task prompts from each difficulty level. Top right: distribution of 101 tasks across the three levels. Bottom right: camera viewpoints are deliberately diversified across scenes to improve generalization coverage.

Alibaba, SeedDance 2.0 (Seedance et al., 2026) from ByteDance, and Veo 3.1 (Google DeepMind, 2025) from Google DeepMind. These models represent the current ceiling of general-purpose video generation quality and are the systems most commonly cited in the community. Including them is essential for answering whether the best available generators already encode sufficient physical understanding for robot execution.

Open-source generators. We include two open-weight models: Wan 2.2 (Wan et al., 2025) and LTX-Video (HaCohen et al., 2025; Lightricks, 2026). These models are fully reproducible and serve two purposes: they establish a baseline that the research community can build on, and they allow a controlled comparison between the open and closed variants of the same model family to isolate the effect of scale and proprietary training data. Additionally, we fine-tune Wan 2.2 on the RoboCasa365 episodes outside the test set to examine whether in-domain video data can close the domain gap between the general and robotic domains.

Robot-specific policy model. We include Cosmos Policy (Kim et al., 2026) from NVIDIA, a video generation model trained specifically on robot manipulation data. Its inclusion directly addresses the question of whether task-specific training confers an advantage in physical executability over models trained purely on general internet video.

3.3. Evaluation Pipeline

Stage 1: Video Generation. Each model receives the initial scene image and the task prompt and generates a manipulation video. For Level 1 and Level 2 tasks, a short clip is generated; for Level 3 multi-stage tasks, a longer video is generated to accommodate the extended task horizon. Full generation settings are provided in Appendix 6.

Stage 2: Visual Quality Evaluation. Generated videos are scored before trajectory extraction to characterize visual stability, physical plausibility, and task adherence. The additional human-evaluation protocol and results are described at the end of Section 4.

Stage 3: Video-to-Trajectory Extraction and Evaluation. The proposed video-to-trajectory extraction pipeline converts a manipulation video into a step-level robot action stream through a five-step chain.

Region Mask Initialization. On the first video frame, the module identifies the spatial region of the end-effector and the manipulated object. When a matching simulation scene is available, initialization-time instance segmentation provides pixel masks directly. Otherwise, open-vocabulary detection via Grounding DINO (Liu et al., 2024a) followed by SAM2 (Ravi et al., 2024) segmentation is used to obtain the corresponding masks.

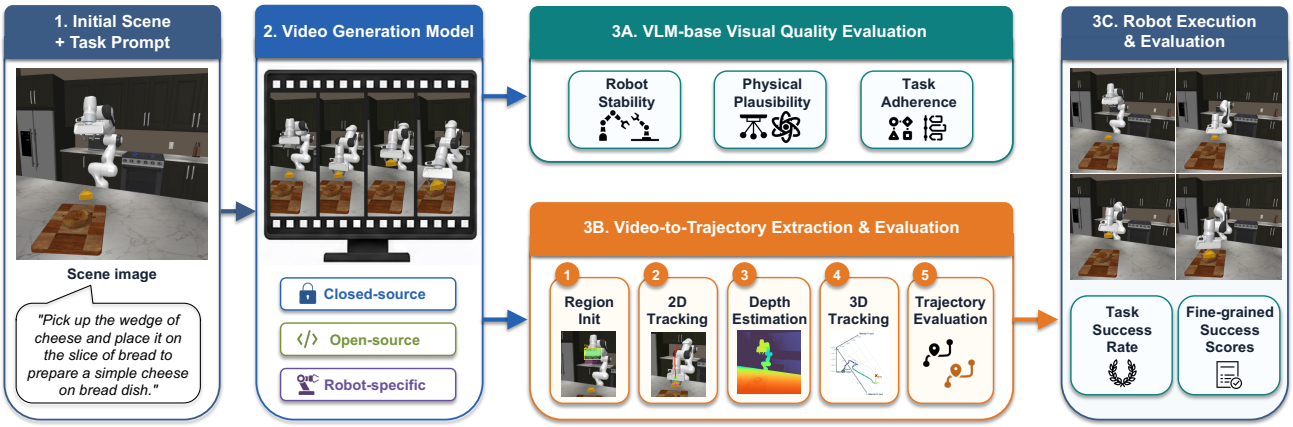

Figure 2. **The Dream.exe evaluation pipeline.** Given an initial scene image and a task prompt, a video generation model produces a manipulation video. The video is assessed for visual quality and physical plausibility, and its implied motion is extracted as a robot trajectory. The trajectory is then executed in a physics simulator, where task success is the final arbiter.

2D point tracking. A set of mask-based query points is sampled within each identified region and tracked across all video frames using CoTracker (Karaev et al., 2024), yielding per-frame pixel coordinates and visibility flags for both the end-effector and the object.

Depth estimation and 3D lifting. For generated videos, video depth is estimated using the DVD (Zhang et al., 2026) model with LoRA adaptation on robot rollout videos. The model predicts affine depth, which is calibrated to metric scale using depth from the initial scene. Each valid tracked pixel is transferred to a 3D point in the world frame using the camera intrinsic and extrinsic parameters associated with the scene. Lifted trajectories are maintained in the world frame, with action deltas later emitted in the configured controller reference frame.

End-Effector Trajectory Extraction. A per-frame visual center is estimated from the lifted point set. Since the visual center of the end-effector does not directly correspond to the robotic control site, we develop a module that applies a calibration derived from the initial state to convert the visual center trajectory into the trajectory of the controller reference point. This step is critical for physically valid execution and enables the same extraction pipeline to operate across different robot morphologies. End-effector orientation is estimated by applying Kabsch alignment to lifted end-effector points across frames.

Gripper-Aware Action Assembly. The gripper open-and-close schedule is inferred from the relative motion between the end-effector trajectory and the manipulated-object trajectory. When task annotations are available, the stage-level priors constrain the expected close/open events for each interaction type, while for multi-stage tasks, each stage is processed with its own target object and then merged into a single video-level gripper schedule. Combining this schedule with the calibrated end-effector motion yields the executable action stream.

Stage 4: Robot Execution and Evaluation. The extracted action stream is executed in MuJoCo via the robosuite (Zhu et al., 2020) control framework on a Franka Panda robot. The scene is restored to its exact initial state before each trial. Execution proceeds in closed loop: at each checkpoint boundary, the current end-effector pose is compared to the target pose, and a correction sequence is applied if the deviation exceeds a threshold. This prevents open-loop error accumulation and provides a controlled test of whether the trajectory extracted from the generated video can be reliably followed by the robot controller.

4. Experiments

Experimental setup. All models are evaluated on the full task suite under a unified protocol. For each task, the model receives the same initial scene image and natural-language prompt, while no additional context or few-shot demonstrations are provided. We consider two instruction variants: (1) *standard instructions* taken verbatim from the original dataset annotations, and (2) *enhanced instructions* rephrased by a VLM, Gemini 3 Pro, into a more descriptive natural-language style that better matches the input distribution of generative models. Each model generates a separate set of videos for each instruction variant, so every variant is evaluated end-to-end through the full video-to-execution pipeline. Unless otherwise noted, all results reported in the main paper are the average over the two instruction variants, while the individual results under standard and enhanced instructions are provided in Appendix D.

Several models deviate from the basic testing mode and are described below. Wan 2.2-LoRA$_{2K}$ and Wan 2.2-LoRA$_{7K}$

Table 1. **Visual quality evaluation results.** Results are grouped by difficulty level. Stab., Phys., and Task Adh. denote robot stability, physical plausibility, and task adherence. Higher is better (↑). Top-1, Top-2, and Top-3 results are highlighted in green, blue, and orange, respectively.

Model	Level 1			Level 2			Level 3			Overall		
	Stab.	Phys.	Task Adh.	Stab.	Phys.	Task Adh.	Stab.	Phys.	Task Adh.	Stab.	Phys.	Task Adh.
Hailuo 2.3	5.708	2.050	2.425	5.726	2.032	2.889	5.278	2.028	2.306	5.690	2.041	2.611
Kling 3.0	6.491	2.179	**2.780**	6.159	2.067	2.972	6.889	**2.361**	2.389	6.376	2.144	2.837
SeedDance 2.0	6.135	*2.123*	*2.739*	6.937	2.143	*3.036*	*7.917*	2.111	**3.111**	6.574	*2.130*	2.884
Veo 3.1	6.079	1.821	2.745	5.274	1.948	**3.385**	4.972	2.028	3.083	5.678	1.886	**3.031**
Wan 2.2	5.833	2.104	2.192	5.968	2.067	2.337	6.611	1.944	1.806	5.936	2.079	2.229
Wan 2.7	*6.428*	2.116	2.670	*6.996*	*2.135*	3.083	6.111	*2.111*	*2.833*	*6.645*	2.124	*2.851*
LTX 2.3	5.484	**2.557**	2.635	5.710	**2.246**	2.187	4.806	1.917	1.778	5.538	**2.389**	2.398
Wan 2.2-LoRA$_{2K}$	6.091	2.057	2.057	6.857	2.024	2.337	6.194	2.000	1.778	6.416	2.040	2.157
Wan 2.2-LoRA$_{7K}$	6.160	2.094	2.264	6.730	1.968	2.254	6.917	1.972	1.861	6.445	2.035	2.236
CosmosPolicy-DefaultCam	6.417	2.063	2.143	7.518	2.035	1.930	8.111	1.944	1.750	6.881	2.045	2.047
CosmosPolicy-BenchCam	**7.262**	2.020	2.024	**7.702**	1.921	1.921	**8.889**	1.944	1.722	**7.532**	1.985	1.968

Table 2. **Trajectory evaluation results.** EEF vis, EEF tcp, and OBJ are the end-effector visual center, end-effector tool center point, and manipulated object. HSD, DYN, and NDTW measure trajectory shape, dynamics, and temporal-alignment similarity. Higher is better (↑).

Model	EEF vis			EEF tcp			OBJ		
	HSD	DYN	NDTW	HSD	DYN	NDTW	HSD	DYN	NDTW
Hailuo 2.3	0.623	0.638	0.704	0.716	0.715	0.822	0.555	0.821	0.720
Kling 3.0	*0.733*	0.740	0.831	*0.734*	0.740	0.836	0.517	0.766	0.689
SeedDance 2.0	0.692	0.698	0.796	0.700	0.704	0.816	0.485	0.751	0.659
Veo 3.1	0.537	0.564	0.601	0.716	0.754	0.812	0.526	0.789	0.675
Wan 2.2	0.639	0.708	0.727	0.663	0.738	0.764	0.505	0.717	0.588
Wan 2.7	0.753	0.778	**0.838**	0.762	0.789	**0.862**	*0.599*	0.852	0.750
LTX 2.3	0.569	0.627	0.623	0.710	*0.782*	0.797	0.602	*0.838*	*0.742*
Wan 2.2-LoRA$_{2K}$	0.645	0.717	0.736	0.669	0.740	0.769	0.498	0.696	0.581
Wan 2.2-LoRA$_{7K}$	0.671	*0.754*	0.766	0.677	0.765	0.780	0.466	0.700	0.526
CosmosPolicy-BenchCam	**0.770**	**0.833**	*0.823*	**0.770**	**0.839**	*0.835*	**0.629**	**0.873**	**0.798**

are fine-tuned versions of Wan 2.2 trained on RoboCasa episodes that do not overlap with our test suite, using 2K and 7K optimization steps, respectively. CosmosPolicy requires multi-view input by design, so we evaluate two variants: CosmosPolicy-DefaultCam follows the standard inference protocol with three camera views, while all other models receive a single task-specific view curated per scene. To make a fairer comparison, CosmosPolicy-BenchCam replaces the primary view with a curated main camera view of Dream.exe while keeping the two remaining views at their default positions.

4.1. Visual Evaluation

We score each generated video with two VLM judges, Gemini 3 Pro and Qwen3-VL-Plus, along three dimensions: *robot-subject stability*, *physical plausibility*, and *task adherence*. For each dimension the VLM is shown sampled frames from the video together with the task prompt and produces a numeric score. Full prompt templates and the scoring rubric are provided in Appendix A. Table 1 reports the results, which are the average over the two VLM judges, while the per-judge scores are provided in Appendix D.

CosmosPolicy-BenchCam scores highest on robot-subject stability, consistent with its domain-specific training on robotic footage. Veo 3.1 leads on task adherence and LTX 2.3 on physical plausibility. To complement these automatic scores and mitigate the uncertainty inherent in black-box VLM evaluation, we also conduct a human study with the same dimensions, reported in Section 4.4.

4.2. Video-to-Trajectory Evaluation

As reported in Table 2, we compare extracted 3D trajectories against ground-truth rollout trajectories with three metrics. HSD is the symmetric Hausdorff distance computed on the most spatially extended sub-trajectory of the ground truth, capturing worst-case shape deviation. DYN measures the Wasserstein-1 distance between the per-frame speed distributions of the generated and reference trajectories, reflecting how closely the motion dynamics are reproduced. NDTW is the DTW alignment cost divided by the alignment path length, penalising local temporal mismatches. All three raw distances are divided by a per-task normalization factor derived from the spatial extent and speed scale of the ground-truth trajectory, then mapped to a $[0, 1]$ similarity score where higher is better. Metrics are computed separately for the end-effector visual center, the end-effector tool center point, and the manipulated object.

Wan 2.7 leads on or is competitive on end-effector trajectory similarity, while CosmosPolicy-BenchCam leads on object trajectory similarity. Notably, general-purpose models such as Wan 2.7 and Kling 3.0 match or exceed CosmosPolicy on several end-effector metrics, suggesting that large-scale pretraining on general video can rival robot-specific training in terms of generating suitable robot trajectories.

4.3. Robot Execution Evaluation

The extracted trajectories are executed in the corresponding robosuite simulation environments and evaluated at two levels. Table 3 reports trajectory executability metrics that measure how easily the video-implied trajectory can be realized by the robot controller: E-SR is the fraction of intermediate checkpoints reached, nDTW measures dense TCP tracking disagreement between the commanded and executed trajectories, Pos95 and Rot95 are 95th-percentile position and rotation errors, and Smth is path-normalized executed smoothness. Table 4 reports task-level execution evaluation results, measuring whether the robot actually completes the manipulation task. SR-B is the binary success rate and SR-P is a continuous 0–1 progress score that remains informative even when SR-B is zero. The sub-goal columns Rel, Place, Art, and Core measure end-effector release quality, target placement proximity, articulation completion degree, and core sub-goal fraction respectively, while their availability depends on task category and difficulty level.

Trajectory executability metrics show consistent trends across models, where overall E-SR ranges from 0.40 to 0.75, with the robot-specific CosmosPolicy variants reaching the highest values, while nDTW, positional, and rotational errors quantify how faithfully the extracted trajecto-

Table 3. **Trajectory executability evaluation results.** Results are broken down by difficulty level and overall. E-SR is strict checkpoint executability, where higher is better (↑). nDTW is commanded-vs-executed TCP tracking disagreement, Pos95 and Rot95 are 95th-percentile position and rotation tracking errors in cm and degrees, and Smth is $10^3\times$ path-normalized executed smoothness, where lower is better (↓).

Model	Level 1					Level 2					Level 3					Overall				
	E-SR↑	nDTW↓	Pos95↓	Rot95↓	Smth↓	E-SR↑	nDTW↓	Pos95↓	Rot95↓	Smth↓	E-SR↑	nDTW↓	Pos95↓	Rot95↓	Smth↓	E-SR↑	nDTW↓	Pos95↓	Rot95↓	Smth↓
Hailuo 2.3	0.508	26.247	53.096	25.658	16.865	0.510	251.421	833.866	22.175	17.753	0.689	6.552	2.364	13.272	18.354	0.519	118.714	374.759	23.469	17.323
Kling 3.0	0.421	8.964	9.304	28.324	18.216	0.514	23.765	300.346	27.538	17.445	0.607	*3.665*	*2.180*	10.638	19.883	0.470	14.804	129.908	26.900	17.995
SeedDance 2.0	0.437	20.302	135.486	28.916	19.579	0.558	11.193	65.576	15.356	20.273	0.604	5.241	5.431	12.006	*18.121*	0.497	15.619	98.689	22.351	19.781
Veo 3.1	*0.522*	28.826	96.989	24.300	16.142	0.513	9.098	8.728	*12.301*	19.906	0.631	7.466	4.138	10.658	17.648	0.527	19.248	54.632	*17.823*	17.821
Wan 2.2	0.448	112.789	918.201	21.013	*14.066*	0.472	*8.235*	*7.661*	23.045	15.594	0.534	4.078	**2.119**	12.099	51.818	0.463	62.853	485.140	21.282	16.944
Wan 2.7	0.513	8.965	9.056	24.398	19.206	0.617	39.153	141.321	18.069	17.519	0.616	5.518	6.553	15.151	18.718	*0.562*	21.314	63.909	21.249	18.476
LTX 2.3	0.422	9.789	9.817	20.798	15.255	0.392	11.315	9.501	21.148	20.762	0.252	16.515	3.205	23.584	62.001	0.401	10.813	9.391	21.225	19.931
Wan 2.2-LoRA$_{2K}$	0.465	*7.707*	*7.704*	*17.991*	14.247	0.464	8.810	17.107	19.332	**15.069**	0.612	3.158	2.156	*7.927*	52.913	0.474	*7.895*	11.285	17.907	*16.886*
Wan 2.2-LoRA$_{7K}$	0.471	8.889	8.497	18.952	13.668	0.445	8.958	8.607	27.265	*15.717*	0.553	3.815	2.561	9.811	53.702	0.465	8.613	*8.187*	21.788	16.930
CosmosPolicy-DefaultCam	**0.662**	**4.376**	**3.928**	**4.949**	**12.538**	**0.841**	**2.905**	**3.372**	**4.127**	17.354	**0.891**	**2.794**	3.170	**2.319**	18.464	**0.750**	**3.670**	**3.652**	**4.451**	**14.893**
CosmosPolicy-BenchCam	0.627	4.639	4.355	6.261	14.306	*0.563*	4.827	5.124	5.474	18.025	*0.662*	4.098	4.685	4.038	**17.518**	0.603	4.685	4.695	5.802	16.044

Table 4. **Task-level execution evaluation results.** SR-B is the binary task success rate and SR-P is a continuous partial-completion score. Rel, Place, Art, and Core report sub-goal completion for end-effector release, target placement, articulation progress, and core sub-goal fraction, whose availability depends on the task category and difficulty. Higher is better (↑).

Model	Level 1			Level 2						Level 3					Overall					
	SR-B↑	SR-P↑	Art↑	SR-B↑	SR-P↑	Rel↑	Place↑	Art↑	Core↑	SR-B↑	SR-P↑	Rel↑	Place↑	Core↑	SR-B↑	SR-P↑	Rel↑	Place↑	Art↑	Core↑
Hailuo 2.3	0.104	0.230	*0.197*	*0.143*	0.592	0.778	0.305	0.751	0.188	0.000	0.359	**0.688**	*0.031*	*0.031*	0.112	0.387	0.763	*0.251*	*0.304*	*0.156*
Kling 3.0	0.123	*0.270*	**0.230**	0.190	0.607	0.547	**0.463**	0.754	**0.352**	**0.062**	0.297	0.438	**0.156**	**0.156**	0.146	*0.409*	0.529	**0.402**	**0.331**	**0.312**
SeedDance 2.0	*0.151*	**0.283**	0.216	**0.214**	0.656	0.815	0.298	0.759	0.188	0.000	0.328	0.625	*0.031*	*0.031*	**0.165**	**0.439**	0.785	0.244	0.320	*0.156*
Veo 3.1	0.033	0.105	0.087	0.120	*0.611*	0.882	0.278	*0.764*	0.143	0.000	0.266	*0.500*	*0.031*	*0.031*	0.069	0.345	*0.820*	0.228	0.228	0.120
Wan 2.2	0.038	0.132	0.076	0.060	0.509	0.587	0.290	0.773	0.156	0.000	0.188	0.375	0.000	0.000	0.044	0.290	0.553	0.232	0.210	0.125
Wan 2.7	0.094	0.215	0.168	**0.214**	**0.667**	**0.884**	0.325	0.760	0.188	0.000	**0.375**	**0.688**	0.062	0.062	*0.136*	0.412	**0.853**	0.272	0.282	0.163
LTX 2.3	0.047	0.140	0.100	0.037	0.503	0.712	0.293	0.722	0.154	0.000	0.250	*0.500*	0.000	0.000	0.039	0.294	0.678	0.233	0.220	0.122
Wan 2.2-LoRA$_{2K}$	0.038	0.122	0.079	0.071	0.500	0.545	0.302	0.763	*0.172*	0.000	0.219	0.438	0.000	0.000	0.049	0.284	0.528	0.241	0.210	0.138
Wan 2.2-LoRA$_{7K}$	0.029	0.144	0.090	0.071	0.517	0.595	0.286	**0.799**	0.156	0.000	0.219	0.438	0.000	0.000	0.044	0.303	0.570	0.229	0.226	0.125
CosmosPolicy-DefaultCam	0.179	0.241	0.186	0.024	0.534	0.597	*0.307*	0.749	*0.172*	0.000	0.250	*0.500*	0.000	0.000	0.102	0.361	0.581	0.246	0.294	0.138
CosmosPolicy-BenchCam	**0.208**	0.271	0.188	0.000	0.594	*0.849*	0.292	0.708	0.156	0.000	*0.344*	**0.688**	0.000	0.000	0.107	0.408	0.823	0.234	0.288	0.125
Rollout Video†	0.765	0.851	0.818	0.381	0.742	0.811	0.562	0.755	0.516	0.750	0.938	1.000	0.875	0.875	0.604	0.812	0.842	0.625	0.805	0.588
Rollout Video w/ GT Depth‡	1.000	1.000	0.950	0.952	0.979	0.905	0.866	1.000	0.953	1.000	1.000	1.000	1.000	1.000	0.981	0.991	0.920	0.893	0.960	0.963

Note. † Rollout Video uses the same depth estimation pipeline as generated videos, while ‡ Rollout Video (w/ GT Depth) replaces estimated depth with simulator depth. These rows serve as reference bounds for the video-to-execution pipeline. Rankings are computed among generation models only; oracle/reference rows are shaded in gray.

ries can be followed by the robot controller. Task-level execution results reveal a clear difficulty gradient. At Level 1, CosmosPolicy-BenchCam achieves the highest SR-B of 20.8%, and articulation sub-goal scores vary noticeably across models. At Level 2, SeedDance 2.0 and Wan 2.7 lead with 21.4% SR-B. Rel scores are generally high across several models, indicating that end-effector release is reliably achieved, while Place and Core scores are more discriminative. At Level 3, only Kling 3.0 achieves non-zero task success with 6.2% SR-B, while most generation models remain at zero. Nevertheless, non-zero sub-goal scores indicate partial progress on multi-step tasks after execution.

It is worth emphasizing that CosmosPolicy outputs robot actions, whereas the general-purpose video generators obtain their actions through our proposed video-to-trajectory pipeline. Even under this indirect route, the general generators reach task-level SR-B that is comparable to or even exceeds CosmosPolicy, reflecting their stronger generalization across tasks and camera viewpoints. The rollout-video reference rows further contextualize these results: replacing estimated depth with simulator ground-truth depth produces a further improvement, indicating that depth estimation remains a bottleneck in the pipeline. Crucially, this bottleneck applies uniformly to all general-purpose generators, ensuring a fair comparison across them.

4.4. Human Evaluation

Four independent human annotators rated each generated video on a 1–5 scale across four dimensions: robot stability, physical plausibility, task adherence, and expected execution result. The rating results are shown in Table 5.

Among general-purpose video generators, Wan 2.7 receives the highest stability rating and SeedDance 2.0 the highest physical-plausibility rating, while Kling 3.0 leads on task adherence and expected execution result. CosmosPolicy variants score high on stability and physical plausibility but low on task adherence and expected execution result, consistent with their tendency to produce visually robotic motion without completing the specified task.

4.5. More Findings

Visual quality does not equal executability. Visual quality is an unreliable predictor of executability. Physical plausibility, the dimension most tied to physical correctness, is essentially uncorrelated with task success across Tables 1 and 4, with a Pearson correlation of $r = -0.03$ against SR-B. The mismatch is stark per model: LTX 2.3 ranks first on

Table 5. **Human evaluation results.** Stab., Phys., Task Adh., and Exec are annotator ratings of robot stability, physical plausibility, task adherence, and expected execution result on a 1–5 scale. Higher is better (↑). Top-1, Top-2, and Top-3 results are highlighted in green, blue, and orange, respectively.

Model	Level 1				Level 2				Level 3				Overall			
	Stab.	Phys.	Task Adh.	Exec	Stab.	Phys.	Task Adh.	Exec	Stab.	Phys.	Task Adh.	Exec	Stab.	Phys.	Task Adh.	Exec
Hailuo 2.3	3.192	2.346	3.500	*2.423*	3.045	2.500	3.864	1.864	2.000	1.500	3.000	1.250	2.900	2.233	3.533	1.983
Kling 3.0	*3.885*	*3.269*	**4.769**	2.462	3.773	3.909	4.591	**2.955**	4.083	2.833	*4.167*	**3.083**	*3.883*	3.417	**4.583**	**2.767**
SeedDance 2.0	3.538	3.462	4.423	2.346	4.000	4.045	**4.682**	*2.273*	*3.833*	3.917	4.583	2.750	3.767	3.767	4.550	*2.400*
Veo 3.1	2.692	2.423	3.346	1.654	2.773	3.182	4.227	1.591	2.400	2.300	4.000	2.300	2.672	2.690	3.793	1.741
Wan 2.2	3.115	2.615	2.308	1.577	3.273	2.773	2.682	1.773	2.500	1.900	1.800	1.300	3.069	2.552	2.362	1.603
Wan 2.7	4.000	3.192	*4.154*	2.269	4.000	*3.955*	*4.545*	2.773	3.500	*3.900*	**4.800**	*2.600*	3.914	*3.603*	*4.414*	2.517
LTX 2.3	2.042	1.625	2.250	1.667	1.722	1.500	1.889	1.111	1.300	1.000	1.300	1.000	1.788	1.462	1.942	1.346
Wan 2.2-LoRA$_{2K}$	3.318	2.682	2.545	1.636	3.889	3.333	2.778	1.889	3.000	2.600	1.500	1.300	3.460	2.900	2.420	1.660
Wan 2.2-LoRA$_{7K}$	3.400	2.850	2.750	1.800	*4.000*	3.278	2.833	2.056	3.100	2.300	2.000	1.200	3.562	2.896	2.625	1.771
CosmosPolicy-DefaultCam	**4.343**	**4.100**	2.286	**2.476**	**4.426**	**4.370**	1.870	1.906	**4.900**	**4.700**	1.700	1.500	**4.418**	**4.254**	2.075	2.171
CosmosPolicy-BenchCam	3.343	3.108	2.186	2.098	3.688	3.422	1.531	1.344	3.333	2.333	1.083	1.083	3.466	3.169	1.876	1.715

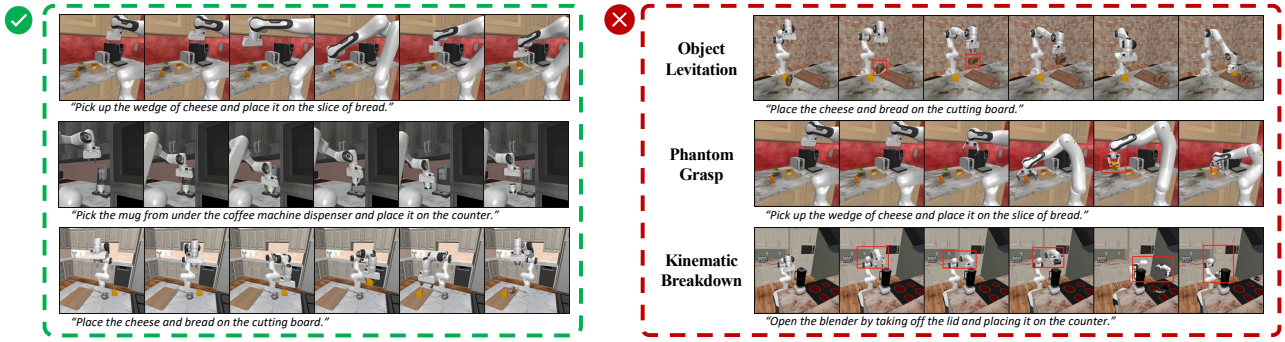

Figure 3. **Success and failure mode taxonomy.** We provide representative examples for each failure category.

physical plausibility yet last on SR-B, while Veo 3.1 leads on task adherence yet reaches only 3.3% Level-1 success. Conversely, visually weaker models such as SeedDance 2.0 and Kling 3.0 achieve the strongest task-level outcomes. Human evaluation confirms the same pattern.

Generative priors help, but struggle at long horizons. Several general-purpose models achieve non-trivial task success without any robot-specific supervision: SeedDance 2.0 reaches 15.1% SR at Level 1, while SeedDance 2.0 and Wan 2.7 reach 21.4% SR at Level 2. However, Level 3 remains difficult, where only Kling 3.0 achieves non-zero task success, and most models fail to complete.

Robot-specific training sharpens geometry more than task success. CosmosPolicy leads on checkpoint executability at Levels 1–2, yet falls substantially behind general generators on task SR at Level 2, i.e. 2.4% vs. SeedDance 2.0 and Wan 2.7 21.4%. Robot-specific models are sensitive to camera viewpoint and task domain, which limit generalization despite strong geometric precision.

In-domain fine-tuning improves appearance, not physics. Fine-tuning Wan 2.2 on in-domain episodes shifts the generated video appearance toward robotic motion and improves trajectory similarity, but does not improve task success rates significantly. This suggests that injecting physical knowledge through robot video fine-tuning alone is insufficient, where the model learns the visual style of robot manipulation without acquiring the underlying physical constraints that drive task success.

Failure modes. Figure 3 illustrates three recurring failure categories: object levitation, phantom grasp, and kinematic breakdown. Phantom grasps and kinematic breakdowns account for the majority of failed trials across all models.

5. Conclusion

The rapid progress of video generation has fueled excitement about using these models as world models and behavioral priors for robotics. Dream.exe puts this idea to a direct test: can the manipulation videos these models dream be grounded back into the physical world through robotic execution? Evaluating 8 models across 101 tasks, we find the answer is a qualified yes. Generative priors trained on internet-scale data encode physically meaningful motion, and several models achieve measurable execution success without any robot-specific supervision. Yet visual quality remains a poor predictor of executability, and long-horizon tasks expose the limits of current models. We hope Dream.exe offers the community both the diagnostic tools and the motivation to close this gap.

References

Bansal, H., Lin, Z., Xie, T., Zong, Z., Yarom, M., Bitton, Y., Jiang, C., Sun, Y., Chang, K.-W., and Grover, A. Videophy: Evaluating physical commonsense for video generation. *arXiv preprint arXiv:2406.03520*, 2024.

Black, K., Nakamoto, M., Atreya, P., Walke, H., Finn, C., Kumar, A., and Levine, S. Zero-shot robotic manipulation with pretrained image-editing diffusion models. *arXiv preprint arXiv:2310.10639*, 2023.

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

Brooks, T., Peebles, B., Holmes, C., DePue, W., Guo, Y., Jing, L., Schnurr, D., Taylor, J., Luhman, T., Luhman, E., et al. Video generation models as world simulators. *OpenAI Blog*, 1(8):1, 2024.

Deng, Y., Pan, Z., Zhang, H., Li, X., Hu, R., Ding, Y., Zou, Y., Zeng, Y., and Zhou, D. Rethinking video generation model for the embodied world. *arXiv preprint arXiv:2601.15282*, 2026.

Dharmarajan, K., Huang, W., Wu, J., Fei-Fei, L., and Zhang, R. Dream2flow: Bridging video generation and open-world manipulation with 3d object flow. *arXiv preprint arXiv:2512.24766*, 2025.

Du, Y., Yang, S., Dai, B., Dai, H., Nachum, O., Tenenbaum, J., Schuurmans, D., and Abbeel, P. Learning universal policies via text-guided video generation. *Advances in neural information processing systems*, 36:9156–9172, 2023.

Google DeepMind. Veo 3 technical report. Technical report, Google DeepMind, 2025. URL https://storage.googleapis.com/deepmind-media/veo/Veo-3-Tech-Report.pdf.

Ha, D. and Schmidhuber, J. World models. *arXiv preprint arXiv:1803.10122*, 2(3):440, 2018.

HaCohen, Y., Chiprut, N., Brazowski, B., Shalem, D., Moshe, D., Richardson, E., Levin, E., et al. Ltx-video: Realtime video latent diffusion. *arXiv preprint arXiv:2501.00103*, 2025.

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

Ho, J., Salimans, T., Gritsenko, A., Chan, W., Norouzi, M., and Fleet, D. J. Video diffusion models. *Advances in neural information processing systems*, 35:8633–8646, 2022b.

Hu, Y., Guo, Y., Wang, P., Chen, X., Wang, Y.-J., Zhang, J., Sreenath, K., Lu, C., and Chen, J. Video prediction policy: A generalist robot policy with predictive visual representations. *arXiv preprint arXiv:2412.14803*, 2024.

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

Jang, J., Ye, S., Lin, Z., Xiang, J., Bjorck, J., Fang, Y., Hu, F., Huang, S., Kundalia, K., Lin, Y.-C., et al. Dreamgen: Unlocking generalization in robot learning through video world models. *arXiv preprint arXiv:2505.12705*, 2025.

Kang, B., Yue, Y., Lu, R., Lin, Z., Zhao, Y., Wang, K., Huang, G., and Feng, J. How far is video generation from world model: A physical law perspective. *arXiv preprint arXiv:2411.02385*, 2024.

Karaev, N., Rocco, I., Graham, B., Neverova, N., Vedaldi, A., and Rupprecht, C. Cotracker: It is better to track together. In *European conference on computer vision*, pp. 18–35. Springer, 2024.

Kim, M. J., Gao, Y., Lin, T.-Y., Lin, Y.-C., Ge, Y., Lam, G., Liang, P., Song, S., Liu, M.-Y., Finn, C., et al. Cosmos policy: Fine-tuning video models for visuomotor control and planning. *arXiv preprint arXiv:2601.16163*, 2026.

Kong, W., Tian, Q., Zhang, Z., Min, R., Dai, Z., Zhou, J., Xiong, J., Li, X., Wu, B., Zhang, J., et al. Hunyuanvideo: A systematic framework for large video generative models. *arXiv preprint arXiv:2412.03603*, 2024.

Kuaishou. Kling ai 3.0, 2026. URL https://kling.ai.

Lab, A. T. Wan 2.7, 2026. URL https://wan.video.

Liang, J., Liu, R., Ozguroglu, E., Sudhakar, S., Dave, A., Tokmakov, P., Song, S., and Vondrick, C. Dreamitate: Real-world visuomotor policy learning via video generation. *arXiv preprint arXiv:2406.16862*, 2024.

Liang, J., Tokmakov, P., Liu, R., Sudhakar, S., Shah, P., Ambrus, R., and Vondrick, C. Video generators are robot policies. *arXiv preprint arXiv:2508.00795*, 2025.

Lightricks. Ltx-2.3 video engine, 2026. URL https://ltx.io/model/ltx-2-3.

Liu, S., Zeng, Z., Ren, T., Li, F., Zhang, H., Yang, J., Jiang, Q., Li, C., Yang, J., Su, H., et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *European conference on computer vision*, pp. 38–55. Springer, 2024a.

Liu, Y., Cun, X., Liu, X., Wang, X., Zhang, Y., Chen, H., Liu, Y., Zeng, T., Chan, R., and Shan, Y. Evalcrafter: Benchmarking and evaluating large video generation models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 22139–22149, 2024b.

Meng, F., Liao, J., Tan, X., Shao, W., Lu, Q., Zhang, K., Cheng, Y., Li, D., Qiao, Y., and Luo, P. Towards world simulator: Crafting physical commonsense-based benchmark for video generation. *arXiv preprint arXiv:2410.05363*, 2024.

MiniMax. Hailuo AI video: Technical report. https://hailuoai.com/video, 2025.

MiniMax. Minimax hailuo 2.3: A new level of complex video expression, 2025. URL https://www.minimax.io/news/minimax-hailuo-23.

Nasiriany, S., Nasiriany, S., Maddukuri, A., and Zhu, Y. Robocasa365: A large-scale simulation framework for training and benchmarking generalist robots. *arXiv preprint arXiv:2603.04356*, 2026.

Polyak, A., Zohar, A., Brown, A., Tjandra, A., Sinha, A., Lee, A., Vyas, A., Shi, B., Ma, C.-Y., Chuang, C.-Y., et al. Movie gen: A cast of media foundation models. *arXiv preprint arXiv:2410.13720*, 2024.

Qin, Y., Shi, Z., Yu, J., Wang, X., Zhou, E., Li, L., Yin, Z., Liu, X., Sheng, L., Shao, J., et al. Worldsimbench: Towards video generation models as world simulators. *arXiv preprint arXiv:2410.18072*, 2024.

Ravi, N., Gabeur, V., Hu, Y.-T., Hu, R., Ryali, C., Ma, T., Khedr, H., Rädle, R., Rolland, C., Gustafson, L., et al. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024.

Seedance, T., Chen, D., Chen, L., Chen, X., Chen, Y., et al. Seedance 2.0: Advancing video generation for world complexity. *arXiv preprint arXiv:2604.14148*, 2026.

Shen, Y., Wei, F., Du, Z., Liang, Y., Lu, Y., Yang, J., Zheng, N., and Guo, B. Videovla: Video generators can be generalizable robot manipulators. *arXiv preprint arXiv:2512.06963*, 2025.

Singer, U., Polyak, A., Hayes, T., Yin, X., An, J., Zhang, S., Hu, Q., Yang, H., Ashual, O., Gafni, O., et al. Make-a-video: Text-to-video generation without text-video data. *arXiv preprint arXiv:2209.14792*, 2022.

Sun, K., Huang, K., Liu, X., Wu, Y., Xu, Z., Li, Z., and Liu, X. T2v-compbench: A comprehensive benchmark for compositional text-to-video generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 8406–8416, 2025.

Team, K., Chen, J., Ci, Y., Du, X., Feng, Z., Gai, K., Guo, S., Han, F., He, J., He, K., et al. Kling-omni technical report. *arXiv preprint arXiv:2512.16776*, 2025.

Wan, T., Wang, A., Ai, B., Wen, B., Mao, C., Xie, C.-W., Chen, D., Yu, F., Zhao, H., Yang, J., et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

Wu, H., Jing, Y., Cheang, C., Chen, G., Xu, J., Li, X., Liu, M., Li, H., and Kong, T. Unleashing large-scale video generative pre-training for visual robot manipulation. *arXiv preprint arXiv:2312.13139*, 2023.

Yang, Z., Teng, J., Zheng, W., Ding, M., Huang, S., Xu, J., Yang, Y., Hong, W., Zhang, X., Feng, G., et al. Cogvideox: Text-to-video diffusion models with an expert transformer. In *The Thirteenth International Conference on Learning Representations*, 2025.

Ye, S., Ge, Y., Zheng, K., Gao, S., Yu, S., Kurian, G., Indupuru, S., Tan, Y. L., Zhu, C., Xiang, J., et al. World action models are zero-shot policies. *arXiv preprint arXiv:2602.15922*, 2026a.

Ye, Y., Lu, X., Jiang, Y., Gu, Y., Zhao, R., Liang, Q., Pan, J., Zhang, F., Wu, W., and Wang, A. J. Mind: Benchmarking memory consistency and action control in world models. *arXiv preprint arXiv:2602.08025*, 2026b.

Zhang, H., Chen, H. H., Liao, C., He, J., Zhang, Z., Li, H., Liang, Y., Chen, K., Ren, B., Zheng, X., et al. Dvd: Deterministic video depth estimation with generative priors. *arXiv preprint arXiv:2603.12250*, 2026.

Zhu, Y., Wong, J., Mandlekar, A., Martín-Martín, R., Joshi, A., Lin, K., Maddukuri, A., Nasiriany, S., and Zhu, Y. robosuite: A modular simulation framework and benchmark for robot learning. *arXiv preprint arXiv:2009.12293*, 2020.

A. VLM Visual Evaluation Details

Visual quality is assessed along three dimensions, each implemented as a separate VLM query run independently with Gemini 3 Pro and Qwen3-VL-Plus. We describe the protocol and give a concrete prompt example for each dimension below.

Robot-Subject Stability

Following the previous VLM evaluation work (Deng et al., 2026), two frames are sampled from each video: the first frame and the frame at 75% of the total duration. They are concatenated horizontally to form a side-by-side image, with the first frame on the left serving as a reference and the later frame on the right representing the generated output. The VLM receives this composite image and is asked two questions in sequence.

Question 1 evaluates the robot subject (e.g., the robotic gripper or arm):

> The provided image shows two sequential frames from an AI-generated video about a robot doing a task. The left frame is the correct reference image, while the right frame is the AI-generated video frame. Focus on how '*[robot subject]*' appears in both frames, and evaluate its consistency between the reference and the generated frame.
>
> Note:
> 1) Pay special attention to distinguishing between robotic gripper and robotic hand. A robotic gripper usually has a small number of rigid gripping jaws or prongs, while a robotic hand has multiple articulated fingers.
> 2) Changes in orientation or position are acceptable and should not affect the consistency rating.
> 3) Do NOT assign option A or B lightly.
>
> Question:
> A: '*[robot subject]*' in the right frame is clear and consistent with the left image.
> B: mostly consistent, with minor visual issues.
> C: noticeable inconsistencies in shape or structure.
> D: highly inconsistent; transforms into another type of '*[robot subject]*'.
> E: *[type-specific disappearance or substitution option]*.
> Select the most suitable option and respond in JSON: `{"option": "A", "explanation": "...", "adjust": "A"}`.

Question 2 repeats the same protocol for the manipulated object, with option E defined as the object being absent in the right frame.

The adjusted option from each question is combined into a pair, which is mapped to a score from 1 to 15 via a fixed lookup table. The mapping is symmetric: (A, A) $\mapsto$ 15 and (E, E) $\mapsto$ 1, with intermediate combinations interpolated monotonically.

Physical Plausibility

Six frames are sampled uniformly from the video and arranged into a 3×2 grid image. The VLM receives this grid together with the camera viewpoint description and the task description, and is asked:

> The provided image presents sequential frames, arranged in a grid, from a *[view]* perspective AI-generated task video about *[task description]*. Does this video comply with common-sense expectations for human-level interactions?
>
> **A. Anomaly Checks:**
> 1) Physical grounding violation: any part of the robot appears floating, or intersecting/penetrating other geometry.
> 2) Spontaneous object appearance: any object that suddenly appears between frames without a plausible cause.
> 3) Non-contact attachment / false grasp: if the video involves grasping, check whether the object moves with the gripper without clear physical contact or closure.
> If any anomaly is present, assign a low score (1–2).
>
> **B. Common-Sense Consistency:**
> Rate the video on a scale from 1 to 5, where 5 means fully consistent with human common sense and 1 means major violations. Be cautious when assigning 4 or 5; do not give high scores lightly. Respond in JSON: `{"reason": "...", "score": 3}`.

Task Adherence

The same 3×2 frame grid is used. The VLM is asked:

The provided image presents sequential frames, arranged in a grid, from a *[view]* perspective AI-generated task video. In this AI-generated video, does the robot successfully perform the task: "*[task description]*"? Please rate the video on a scale from 1 to 5, where 5 indicates a perfect match and 1 indicates no relevance. Be cautious when assigning scores of 4 or 5. Respond in JSON: `{"reason": "...", "score": 3}`.

B. Model Details

Table 6 summarizes the evaluated models and their generation settings. All models are conditioned on the initial scene image paired with a task instruction. Resolution and video duration were set to the suitable option that does not exceed the task horizon: a short clip for Level 1 and Level 2 tasks, and a longer video for Level 3 multi-stage tasks.

Table 6. Models evaluated in Dream.exe, organized by category. Open-weight models are marked with ✓; closed-source API models are marked with ✗.

Model	Category	Open	Resolution	Frames	FPS	Duration (s)
Hailuo 2.3 (MiniMax, 2025)	Closed-source	✗	960×960	121/193	24	5/8
Kling 3.0 (Team et al., 2025)	Closed-source	✗	960×960	121/193	24	5/8
Wan 2.7 (Wan et al., 2025)	Closed-source	✗	1440×1440	150/240	30	5/8
SeedDance 2.0 (Seedance et al., 2026)	Closed-source	✗	960×960	121/193	24	5/8
Veo 3.1 (Google DeepMind, 2025)	Closed-source	✗	1280×720	144/192	24	6/8
Wan 2.2 (Wan et al., 2025)	Open-source	✓	480×480	81/129	16	5/8
LTX 2.3 (HaCohen et al., 2025)	Open-source	✓	512×512	121	24	5
Cosmos Policy (rollout) (Kim et al., 2026)	Robot-specific	✓	512×512	400	15	27
Cosmos Policy (generated) (Kim et al., 2026)	Robot-specific	✓	224×224	50	1	50

C. Video2Traj Implementation Details

Figure 4 provides an expanded view of the video-to-execution pipeline described in Section 3.3.

C.1. Trajectory Extraction and Execution Details

2D point tracking. We initialize tracking regions on the first frame using simulator-provided segmentation masks when available, and fall back to visual region proposals such as manual boxes, open-vocabulary detection (Liu et al., 2024a), and SAM2 (Ravi et al., 2024) segmentation. Within each region, query points are sampled from the mask using farthest-point sampling, using 3D sampling when initial depth and camera calibration are available and 2D sampling otherwise. The selected points are tracked through the video using CoTracker (Karaev et al., 2024), which returns per-frame point locations and visibility estimates. During 3D lifting, points with low visibility, out-of-frame locations, or invalid depth are ignored on a per-frame basis.

Depth estimation. For generated videos, we use a robot-adapted DVD (Zhang et al., 2026) depth model to estimate temporally consistent video depth. Specifically, we fine-tune low-rank adapters on top of the DVD DiT using robot rollout videos rendered from simulation, while keeping the remaining model weights frozen. Each training sample consists of an RGB rollout clip paired with simulator-rendered metric depth. Rather than supervising metric depth directly, we convert valid metric depth values into disparity, normalize them within each clip using robust percentiles, and train the LoRA adapters in this normalized-disparity space. The training objective combines a masked reconstruction loss over valid depth pixels with spatial and temporal gradient matching losses, encouraging both sharp local depth boundaries and temporal consistency. When end-effector weighting is used, simulator instance segmentation up-weights end-effector pixels in the reconstruction loss. This end-effector emphasis improves the reliability of depth estimates around the robot hand, where small depth errors can lead to large errors in the recovered 3D end-effector trajectory. At inference time, the LoRA weights are merged into the DVD DiT and used as the depth backend. The predicted depth remains affine rather than metrically scaled; therefore, before 3D lifting, we align it to the first-frame simulator depth using a robust affine calibration over valid task regions.

In our implementation, the LoRA adapters are trained on fixed-length robot rollout clips at 512×512 resolution, with rank-512 adapters inserted into the DVD DiT attention and feed-forward projections.

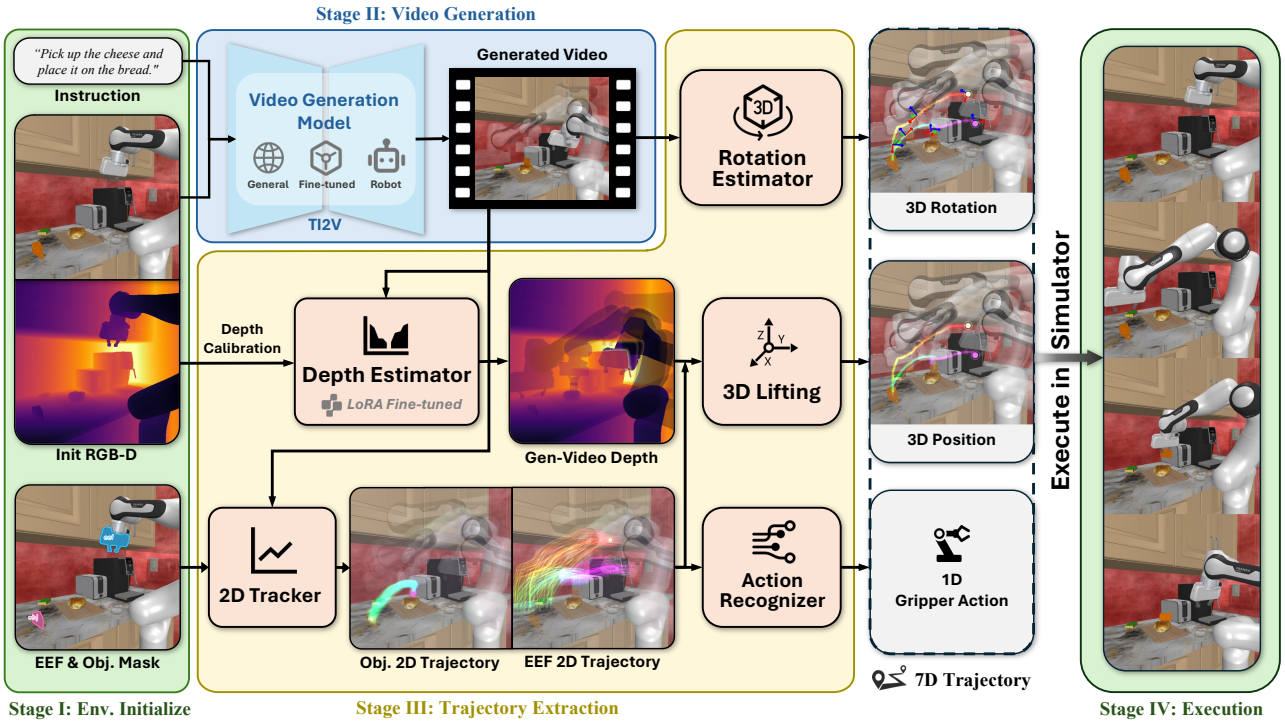

Figure 4. **Detailed video-to-execution pipeline.** The diagram expands the trajectory extraction and execution components of Figure 2, showing how generated video is converted into 2D tracklets, calibrated depth, lifted 3D point trajectories, end-effector rotation, and gripper actions. These signals are fused into a 7D executable trajectory and replayed in the simulator for evaluation.

3D Point Lifting Each valid tracked pixel is back-projected into the world frame using the calibrated depth and the simulator camera intrinsics and extrinsics. The lifted point tracks are then summarized into per-frame visual-center trajectories for the end-effector and the manipulated object. For each frame, the visual center is estimated robustly from valid lifted points, with carry-forward and interpolation used when too few reliable observations are available. These visual-center trajectories provide the positional inputs for controller reference calibration and gripper schedule inference.

TCP Trajectory Extraction. The end-effector visual-center trajectory does not directly correspond to the robot tool-center point (TCP) or controller reference site. We convert the recovered visual-center trajectory to a TCP trajectory through translation-only alignment to the simulator initial TCP pose: the offset is estimated from the median of the first valid visual-center observations and then applied to all frames. When the controller reference site differs from the TCP, we further apply the fixed TCP-to-controller offset specified by the scene metadata. This calibration uses the initial simulation state and does not require matching a full ground-truth rollout.

3D Rotation Estimation. In addition to the positional trajectory, we estimate the end-effector orientation for 6-DoF action generation. Frame-wise rotations are estimated by rigidly aligning lifted 3D end-effector points to their first-frame anchors with Kabsch alignment. The simulator-provided controller reference pose defines the initial geometric anchor for this alignment. A lightweight temporal guard is applied to suppress implausible frame-to-frame rotation jumps while preserving the current translational trajectory. The resulting orientation sequence is fused with the calibrated end-effector position trajectory before action generation.

Gripper Action Recognition. The gripper open-and-close schedule is inferred from the relative motion between the end-effector and the manipulated object. We detect grasp and release events using geometric motion cues, including distance, relative velocity, co-motion, visibility, and invalid-track gating. When task annotations are available, we identify the interaction mode of each manipulation stage and use its expected close/open pattern as a task prior, as summarized in Table 7. This prior constrains the number and ordering of gripper events, while their timing is still determined from motion evidence. For multi-stage tasks, each stage is processed with its own target object and interaction prior, and the resulting

Table 7. **Gripper action priors by interaction mode.** Each interaction mode specifies the expected number of close and open events used by the task-prior gripper recognizer. The occurrence count reports how often the interaction mode appears in the benchmark annotations; a single task may contain multiple interaction instances.

Interaction mode	Description	Level	Close	Open	# Occ.
Pick-and-place	Grasp an object and place it at a target location or into another container.	level 2 & 3	1	1	37
Object pushing	Move an object by pushing without requiring a grasp.	level 1	0	0	16
Drawer closing	Push a prismatic drawer from an open state toward a closed state.	level 1	0	0	14
Stacking	Grasp an object and place it on top of another object.	level 2	1	1	11
Object relocation	Grasp a removable object, lift it from its support or attachment, and place it at a target location.	level 2	1	1	10
Lever turning	Rotate an articulated lever or handle without a grasp-release cycle.	level 1	0	0	10
Door closing	Close a revolute articulated door.	level 1	0	0	7
Drawer opening	Pull or slide a prismatic drawer from a closed state toward an open state.	level 1	0	0	3
Switch activation	Activate an appliance or switch through direct contact.	level 1	0	0	1

stage-local schedules are merged into a single frame-aligned gripper schedule. The resulting schedule is combined with the extracted end-effector trajectory during action generation.

Closed-loop execution. The extracted trajectory is converted into a sequence of delta 6-DoF actions and gripper commands, and replayed in the robosuite (Zhu et al., 2020) operational-space controller. Before each trial, the simulator scene is restored to the recorded initial state. During action-mode execution, each retained trajectory checkpoint is reached through one or more controller steps. At checkpoint boundaries, the executor compares the current configured controller-reference pose with the target checkpoint pose and applies a bounded number of corrective actions until the position and orientation errors fall below the configured tolerances. In the current execution configuration, these tolerances are 5 mm and 0.03 rad. This checkpoint-level correction reduces accumulated open-loop tracking error during simulation execution.

C.2. Rollout-Video Reference Bounds

To isolate the limitations of the video-to-execution pipeline from the limitations of generated videos, we also evaluate two rollout-video reference settings. In both settings, the input video is the ground-truth rollout rendered from the simulator, rather than a generated video. These rows therefore provide reference bounds on how well our video-to-execution pipeline can extract executable robot behavior when the visual motion is correct.

The *Rollout Video* setting runs the same pipeline used for generated videos, including the learned depth estimator. In contrast, *Rollout Video w/ GT Depth* replaces the estimated depth with simulator-rendered metric depth while keeping the rest of the pipeline unchanged. The gap between these two rows diagnoses the effect of depth estimation on downstream 3D lifting and execution. As shown in Table 4, using ground-truth depth substantially improves task-level success, indicating that depth remains a major bottleneck in the current video-to-execution pipeline. This is expected that small temporally inconsistent depth errors around the end-effector or manipulated object can be amplified after 3D lifting, leading to inaccurate TCP positions, contact timing errors, and ultimately lower execution success.

D. Additional Quantitative Results

D.1. Additional Visual Quality Scores under Different Instruction Settings

Table 8 provides a by-level breakdown of visual-quality scores across instruction settings, together with per-judge and judge-averaged scores.

D.2. Trajectory Similarity Metrics under Different Instruction Settings

Table 9 reports trajectory similarity scores under standard and enhanced instructions, complementing Table 2 in the main paper.

Table 8. **By-level visual quality evaluation under different instruction settings.** Results are grouped by VLM judge, difficulty level, and instruction setting. Stab., Phys., and Task Adh. denote robot stability, physical plausibility, and task adherence. Judge Avg. reports the average of Gemini 3 Pro and Qwen3-VL-Plus scores. Higher is better (↑). Rankings are computed separately within each instruction setting. Top-1, Top-2, and Top-3 results are highlighted in green, blue, and orange, respectively.

Model	Gemini 3 Pro									Qwen3-VL-Plus									Judge Avg.								
	Level 1			Level 2			Level 3			Level 1			Level 2			Level 3			Level 1			Level 2			Level 3		
	Stab.	Phys.	Task	Stab.	Phys.	Task	Stab.	Phys.	Task	Stab.	Phys.	Task	Stab.	Phys.	Task	Stab.	Phys.	Task	Stab.	Phys.	Task	Stab.	Phys.	Task	Stab.	Phys.	Task
Videos generated from standard instructions; evaluated with standard instructions																											
Hailuo 2.3	5.717	1.000	2.019	5.690	1.143	3.429	5.500	1.000	3.000	**6.302**	2.189	3.189	6.024	2.119	*3.071*	6.000	2.500	2.333	6.009	1.594	2.604	5.857	1.631	3.250	5.750	1.750	2.667
Kling 3.0	7.136	**1.321**	**3.057**	5.929	1.095	3.381	9.333	1.167	*3.167*	6.039	2.283	**3.189**	6.098	2.143	2.714	*6.000*	*2.333*	2.500	6.519	1.802	**3.123**	6.083	1.619	3.048	7.667	1.750	2.833
SeedDance 2.0	5.755	*1.125*	2.660	7.690	1.439	4.056	8.500	**1.833**	**5.000**	5.981	2.302	*3.132*	6.000	*2.214*	2.929	5.833	2.000	2.500	5.868	*1.755*	2.896	6.845	1.833	*3.369*	7.167	**1.917**	**3.500**
Veo 3.1	5.925	1.094	2.811	4.286	**1.548**	**4.071**	2.500	1.667	4.167	5.962	2.302	2.849	6.143	**2.220**	2.690	6.000	2.000	*2.667*	5.943	1.698	2.830	5.214	**1.869**	3.381	4.250	1.833	3.417
Wan 2.2	6.170	1.000	1.078	7.095	1.114	1.219	*11.667*	1.000	1.000	5.906	*2.415*	2.849	5.951	2.214	2.714	6.000	2.000	2.500	6.038	1.717	2.009	6.476	*1.786*	2.095	*8.833*	1.500	1.750
Wan 2.7	6.792	1.321	*2.811*	*7.895*	*1.214*	*4.048*	5.000	*1.667*	2.833	*6.094*	2.189	2.906	6.000	2.000	3.262	6.000	2.000	**3.333**	6.406	1.755	*2.858*	6.905	1.607	**3.655**	5.417	*1.833*	*3.083*
LTX 2.3	4.800	1.000	1.769	5.889	1.000	1.061	2.750	1.000	1.000	6.000	**2.453**	2.788	6.000	2.214	2.929	6.000	2.000	2.250	5.786	**2.123**	2.547	5.964	1.774	2.179	4.375	1.500	1.625
Wan 2.2-LoRA$_{2K}$	6.132	1.075	1.226	8.381	1.024	1.190	8.000	1.000	1.167	6.208	2.208	2.895	6.171	2.119	**3.270**	**7.000**	**2.667**	2.333	6.170	1.642	1.792	7.238	1.571	2.119	7.500	1.833	1.750
Wan 2.2-LoRA$_{7K}$	6.667	1.020	1.180	7.821	1.000	1.167	12.167	1.000	1.000	6.000	2.231	3.000	**6.524**	2.167	2.878	6.000	2.167	2.500	6.144	1.651	2.132	*7.107*	1.583	2.000	9.083	1.583	1.750
CosmosPolicy-DefaultCam	*6.800*	1.000	1.595	7.750	1.158	1.000	11.167	1.000	1.000	6.073	2.439	3.119	*6.316*	2.053	3.053	6.000	2.000	2.833	*6.427*	1.702	2.357	7.026	1.605	2.026	**9.083**	1.500	1.917
CosmosPolicy-BenchCam	**8.000**	1.000	1.357	**9.706**	1.000	1.000	**12.500**	1.000	1.000	5.810	2.341	2.927	6.316	2.211	2.789	4.833	2.000	2.333	**6.881**	1.655	2.119	**7.816**	1.605	1.895	8.667	1.500	1.667
Videos generated from enhanced instructions; evaluated with standard instructions																											
Hailuo 2.3	4.981	1.094	2.302	5.381	1.167	3.310	3.333	1.000	1.833	5.941	2.340	2.849	5.881	**2.310**	*2.976*	5.500	2.000	**3.000**	5.415	1.717	2.575	5.631	1.738	3.143	4.417	1.500	2.417
Kling 3.0	7.333	**1.300**	*2.938*	6.316	1.286	**4.048**	7.000	**1.667**	3.000	5.887	2.377	3.075	*6.220*	2.190	2.833	6.000	1.833	1.667	*6.585*	**1.877**	*3.019*	6.321	*1.738*	3.440	6.500	**1.750**	2.333
SeedDance 2.0	6.245	1.189	3.321	7.786	1.310	3.976	11.000	1.167	**4.000**	6.038	2.321	2.925	5.976	*2.214*	2.738	6.000	2.000	2.667	6.142	1.755	3.123	6.881	1.762	*3.357*	8.500	*1.583*	**3.333**
Veo 3.1	6.389	1.056	**3.444**	4.909	*1.364*	4.000	5.167	*1.167*	*3.667*	6.000	2.333	**3.500**	5.909	1.818	3.000	6.000	2.000	2.833	6.194	1.694	**3.472**	5.409	1.591	**3.500**	5.583	1.583	3.250
Wan 2.2	5.434	1.075	2.094	5.357	1.119	2.833	6.000	1.000	1.167	6.057	*2.358*	2.774	6.095	2.095	2.714	6.000	2.000	2.500	5.745	1.717	2.434	5.726	1.607	2.774	6.000	1.500	1.833
Wan 2.7	*7.321*	1.226	2.925	*8.143*	**1.762**	*4.000*	6.500	1.167	3.833	6.000	2.340	*3.000*	5.976	2.119	2.643	6.000	**2.167**	2.500	6.660	1.783	2.962	*7.060*	**1.940**	3.321	6.250	1.667	*3.167*
LTX 2.3	4.887	1.000	2.600	3.900	1.000	1.852	3.833	1.000	1.333	6.321	2.151	2.962	6.000	2.100	2.641	*6.000*	2.000	2.167	5.321	1.821	2.868	4.950	1.575	2.282	4.917	1.500	1.750
Wan 2.2-LoRA$_{2K}$	6.038	1.132	2.057	7.071	1.405	2.929	4.667	1.000	1.167	5.952	2.216	3.000	6.000	2.065	2.825	6.000	2.000	2.500	5.962	1.651	2.396	6.905	1.702	2.845	5.333	1.500	1.833
Wan 2.2-LoRA$_{7K}$	6.269	*1.208*	2.216	7.128	1.071	2.476	4.500	1.000	1.000	*6.192*	**2.442**	2.980	6.073	2.095	2.833	6.000	2.000	*2.833*	6.231	*1.811*	2.594	6.595	1.583	2.655	5.250	1.500	1.917
CosmosPolicy-DefaultCam	6.610	1.000	1.381	**9.312**	1.053	1.000	*9.500*	1.000	1.000	6.073	2.333	2.929	**6.526**	2.263	2.737	5.500	*2.000*	2.500	6.321	1.667	2.155	**7.816**	1.658	1.868	*7.500*	1.417	1.750
CosmosPolicy-BenchCam	**8.590**	1.000	1.119	8.765	1.000	1.000	**11.833**	1.000	1.000	**6.439**	2.250	2.976	6.316	1.947	**3.222**	**6.000**	2.000	2.167	**7.405**	1.595	2.048	7.447	1.474	2.053	**8.917**	1.500	1.583
Videos generated from enhanced instructions; evaluated with enhanced instructions																											
Hailuo 2.3	5.094	1.019	1.415	5.405	1.000	1.643	5.333	1.000	1.000	6.294	4.731	2.808	5.976	4.452	2.905	6.000	*4.667*	2.667	5.698	2.840	2.094	5.690	2.726	2.274	5.667	2.833	1.833
Kling 3.0	6.604	1.019	**1.755**	6.049	1.049	**2.024**	12.000	*1.000*	1.333	6.058	**4.769**	2.642	6.095	4.548	2.833	6.000	4.500	2.667	6.387	2.858	2.198	6.071	2.845	2.429	6.500	**3.583**	2.000
SeedDance 2.0	6.660	1.000	*1.566*	*8.143*	*1.119*	*1.857*	*10.167*	1.000	**2.000**	6.096	*4.717*	2.830	6.024	4.548	*2.905*	6.000	4.667	3.000	6.396	2.858	*2.198*	*7.083*	2.833	*2.381*	8.083	2.833	2.500
Veo 3.1	6.889	*1.056*	1.444	5.273	**1.364**	2.000	4.667	1.000	2.000	**6.588**	4.389	2.778	5.455	**4.909**	**3.091**	5.500	4.333	**3.167**	6.667	2.722	2.111	5.364	3.136	**2.545**	5.083	2.667	**2.583**
Wan 2.2	5.358	**1.094**	1.396	4.929	1.000	1.452	4.000	1.000	1.000	6.075	4.660	*2.885*	6.476	4.619	2.833	6.000	4.667	2.667	5.717	2.877	2.132	5.702	2.810	2.143	5.000	2.833	1.833
Wan 2.7	6.434	1.038	1.604	8.048	1.143	1.738	6.333	1.000	*1.833*	5.962	4.585	2.774	6.000	4.571	2.810	**7.000**	4.667	2.667	6.217	2.811	2.189	7.024	*2.857*	2.274	6.667	2.833	*2.250*
LTX 2.3	4.571	1.000	1.208	4.500	1.000	1.115	3.667	1.000	1.000	6.571	4.642	**2.942**	6.025	*4.718*	2.650	6.000	4.167	2.667	6.107	**3.726**	**2.491**	5.875	**3.487**	2.175	4.833	2.583	1.833
Wan 2.2-LoRA$_{2K}$	5.852	1.077	1.321	6.659	1.000	1.476	5.333	1.000	1.000	6.170	4.623	2.933	6.071	4.595	2.853	6.167	4.333	2.500	6.142	*2.877*	1.981	6.429	2.798	2.048	5.750	2.667	1.750
Wan 2.2-LoRA$_{7K}$	5.846	1.038	1.314	7.000	1.000	1.476	6.833	1.000	1.000	6.135	4.596	2.808	5.927	4.405	2.738	6.000	4.667	2.833	6.106	2.821	2.066	6.488	2.738	2.107	6.417	2.833	1.917
CosmosPolicy-DefaultCam	*6.690*	1.048	1.024	8.812	1.000	1.105	9.500	1.000	1.000	*6.357*	4.595	2.810	*6.471*	4.889	2.684	6.000	**4.833**	2.167	*6.524*	2.821	1.917	7.711	2.842	1.895	*7.750*	2.917	1.583
CosmosPolicy-BenchCam	**8.850**	1.000	1.000	**8.947**	1.000	1.000	**12.167**	**1.000**	1.000	6.098	4.619	2.854	**6.556**	4.556	2.632	*6.000*	4.667	*2.833*	**7.500**	2.810	1.905	**7.842**	2.684	1.816	**9.083**	*2.833*	1.917

E. More Results

Figure 5 provides additional qualitative examples for our evaluation protocol. For each example, we compare the generated manipulation video with the rollout video obtained after executing the corresponding action trajectory.

The successful cases show that coherent object motion and stable robot-object interactions can be recovered as executable trajectories. The failure cases illustrate the opposite behavior, where artifacts such as spurious objects or inconsistent contacts introduce visual evidence that cannot be mapped to a valid robot action sequence.

Table 9. **Trajectory similarity results under different instruction settings.** Results are reported for videos generated from enhanced and standard instructions. EEF vis, EEF tcp, and OBJ denote the end-effector visual center, end-effector tool center point, and manipulated object. HSD, DYN, and NDTW measure trajectory shape, dynamics, and temporal-alignment similarity. Higher is better (↑). Top-1, Top-2, and Top-3 results are highlighted in green, blue, and orange, respectively.

Model	EEF vis			EEF tcp			OBJ		
	HSD	DYN	NDTW	HSD	DYN	NDTW	HSD	DYN	NDTW
Videos generated from enhanced instructions									
Hailuo 2.3	0.636	0.654	0.721	0.725	0.722	*0.838*	0.536	*0.814*	0.699
Kling 3.0	*0.758*	*0.761*	0.844	*0.747*	0.755	0.834	0.514	0.764	0.678
SeedDance 2.0	0.679	0.671	0.785	0.709	0.706	0.827	0.490	0.758	0.646
Veo 3.1	0.601	0.642	0.671	0.714	0.768	0.808	0.511	0.774	0.662
Wan 2.2	0.625	0.665	0.736	0.657	0.706	0.785	0.427	0.645	0.536
Wan 2.7	0.777	0.807	**0.861**	0.780	0.814	**0.870**	0.582	0.847	*0.736*
LTX 2.3	0.551	0.595	0.606	0.706	*0.778*	0.797	*0.573*	0.811	0.737
Wan 2.2-LoRA$_{2K}$	0.634	0.669	0.745	0.663	0.698	0.783	0.427	0.623	0.535
Wan 2.2-LoRA$_{7K}$	0.666	0.716	0.781	0.669	0.723	0.791	0.397	0.637	0.478
CosmosPolicy-BenchCam	**0.785**	**0.845**	*0.831*	**0.791**	**0.866**	0.848	**0.637**	**0.898**	**0.806**
Videos generated from standard instructions									
Hailuo 2.3	0.609	0.622	0.688	0.707	0.708	0.805	0.574	0.827	0.740
Kling 3.0	*0.708*	0.718	**0.819**	*0.720*	0.724	0.838	0.520	0.769	0.700
SeedDance 2.0	0.704	0.724	0.808	0.691	0.703	0.804	0.480	0.744	0.671
Veo 3.1	0.473	0.485	0.530	0.718	0.741	0.817	0.542	0.804	0.687
Wan 2.2	0.652	0.751	0.718	0.668	0.770	0.744	0.583	0.789	0.639
Wan 2.7	0.729	0.749	*0.815*	0.745	0.764	**0.854**	*0.615*	0.857	0.765
LTX 2.3	0.586	0.658	0.640	0.714	*0.785*	0.797	**0.631**	**0.864**	*0.748*
Wan 2.2-LoRA$_{2K}$	0.656	*0.765*	0.727	0.675	0.782	0.755	0.569	0.769	0.627
Wan 2.2-LoRA$_{7K}$	0.676	0.791	0.752	0.686	0.807	0.769	0.534	0.762	0.574
CosmosPolicy-BenchCam	**0.756**	**0.822**	0.816	**0.750**	**0.811**	*0.823*	0.622	*0.849*	**0.789**

Table 10. **Trajectory execution feasibility under different instruction settings.** Results are reported for trajectories extracted from videos generated with standard and enhanced instructions, broken down by task difficulty and overall. E-SR measures checkpoint reachability (↑), while nDTW, Pos95/Rot95, and Smth measure TCP tracking disagreement, 95th-percentile position/rotation error, and executed-trajectory smoothness (↓). The Overall block aggregates over all active tasks without stratifying by difficulty. Top-1, Top-2, and Top-3 results are highlighted in green, blue, and orange, respectively.

Model	Level 1					Level 2					Level 3					Overall				
	E-SR↑	nDTW↓	Pos95↓	Rot95↓	Smth↓	E-SR↑	nDTW↓	Pos95↓	Rot95↓	Smth↓	E-SR↑	nDTW↓	Pos95↓	Rot95↓	Smth↓	E-SR↑	nDTW↓	Pos95↓	Rot95↓	Smth↓
Videos generated from standard instructions																				
Hailuo 2.3	0.506	37.227	81.936	24.791	17.171	0.523	21.510	91.097	24.127	19.415	0.625	5.238	2.585	13.067	19.575	0.520	28.791	81.031	23.788	18.247
Kling 3.0	0.426	*7.367*	8.625	21.881	16.864	0.503	21.960	260.739	25.128	17.416	0.595	4.307	1.932	12.289	19.976	0.468	13.253	113.067	22.601	17.279
SeedDance 2.0	0.447	9.973	11.458	25.139	16.516	0.552	10.896	63.382	15.558	20.508	0.650	3.706	4.487	10.136	18.970	0.503	9.985	32.636	20.309	18.322
Veo 3.1	0.537	46.636	181.877	21.522	16.294	0.470	9.510	8.082	*12.709*	23.351	0.694	6.118	**1.074**	12.964	18.478	0.519	28.790	98.865	*17.407*	19.359
Wan 2.2	0.438	9.324	*8.078*	22.117	14.511	0.513	*7.882*	6.318	22.284	16.336	0.604	3.296	*1.697*	11.199	16.647	0.479	8.366	6.967	21.502	15.397
Wan 2.7	0.496	9.046	9.063	23.789	17.561	0.614	71.339	266.288	18.331	16.997	0.647	6.018	6.664	16.571	*16.886*	*0.554*	34.770	115.885	21.120	17.287
LTX 2.3	0.407	11.584	12.131	22.198	15.167	0.399	13.181	10.059	25.779	19.875	0.427	3.077	1.301	*3.628*	**14.120**	0.404	11.918	10.814	22.901	17.122
Wan 2.2-LoRA$_{2K}$	0.430	8.196	8.283	20.251	14.223	0.513	7.982	*6.007*	19.361	**14.935**	*0.710*	**2.449**	1.849	5.533	18.514	0.481	*7.766*	*6.954*	18.969	14.774
Wan 2.2-LoRA$_{7K}$	0.448	9.192	9.472	*17.756*	*13.814*	0.468	9.072	7.162	28.689	*16.424*	0.656	2.540	2.212	6.083	17.035	0.469	8.743	8.066	21.507	*15.103*
CosmosPolicy-DefaultCam	**0.549**	**5.198**	**4.268**	**5.737**	**10.936**	**0.877**	**2.651**	**3.120**	**4.247**	16.939	**0.896**	*2.767*	3.102	**1.983**	19.007	**0.706**	**3.994**	**3.721**	**4.895**	**13.911**
CosmosPolicy-BenchCam	*0.522*	5.387	4.706	6.661	13.644	*0.590*	4.644	4.777	5.781	17.706	0.734	3.575	4.120	3.269	17.770	0.563	4.970	4.701	6.094	15.578
Videos generated from enhanced instructions																				
Hailuo 2.3	0.510	15.268	24.257	26.524	16.560	0.496	481.331	1576.635	20.223	16.091	0.754	7.866	**2.143**	13.478	17.132	0.519	208.637	668.487	23.149	*16.399*
Kling 3.0	0.416	10.561	9.983	34.767	19.569	0.524	25.571	339.954	29.949	17.474	*0.619*	3.023	2.428	8.987	19.790	0.473	16.355	146.750	31.198	18.711
SeedDance 2.0	0.427	30.630	259.515	32.694	22.642	*0.564*	11.490	67.771	15.155	20.039	0.557	6.775	6.375	13.876	17.271	0.492	21.254	164.742	24.393	21.240
Veo 3.1	0.507	11.017	12.101	27.078	15.990	0.556	8.687	9.374	*11.892*	16.462	0.569	8.814	7.202	*8.352*	**16.819**	0.535	9.706	10.400	18.239	16.283
Wan 2.2	0.459	216.255	1828.324	19.910	13.620	0.431	8.588	9.005	23.806	**14.853**	0.464	4.859	2.540	13.000	86.990	0.448	117.340	963.313	21.061	18.491
Wan 2.7	*0.530*	8.885	9.050	25.006	20.850	0.620	*6.968*	16.354	17.807	18.041	0.585	5.018	6.442	13.730	20.551	*0.571*	*7.858*	11.933	21.377	19.664
LTX 2.3	0.437	7.994	7.503	19.398	15.343	0.385	9.448	*8.943*	16.516	21.649	0.077	29.953	5.108	43.539	109.881	0.397	9.708	*7.968*	19.550	22.740
Wan 2.2-LoRA$_{2K}$	0.500	*7.218*	*7.126*	*15.732*	14.271	0.416	9.637	28.208	19.303	*15.203*	0.514	*3.867*	*2.463*	10.322	87.312	0.466	8.025	15.616	*16.845*	18.998
Wan 2.2-LoRA$_{7K}$	0.494	8.585	7.522	20.149	**13.521**	0.422	8.845	10.052	25.842	15.010	0.450	5.091	2.909	13.539	90.370	0.461	8.484	8.308	22.069	18.757
CosmosPolicy-DefaultCam	**0.774**	**3.554**	**3.587**	**4.161**	*14.140*	**0.805**	**3.159**	**3.624**	**4.007**	17.770	**0.887**	**2.821**	3.239	**2.655**	17.922	**0.794**	**3.346**	**3.582**	**4.008**	**15.874**
CosmosPolicy-BenchCam	0.732	3.891	4.004	5.861	14.969	0.536	5.011	5.471	5.166	18.344	0.590	4.620	5.250	4.807	*17.266*	0.642	4.400	4.688	5.510	16.509

Note. Scores use the same trajectory-validity adjustment as Table 3 to penalize unrealistically short motions. All reported metrics are means over active UIDs within each difficulty level or over all active UIDs for Overall. Rankings are computed separately within each instruction setting.

Table 11. **Task-level execution results under different instruction settings.** Results are reported for videos generated from standard and enhanced instructions, broken down by task difficulty and overall. SR-B is the binary task success rate and SR-P is a continuous partial-completion score. Rel, Place, Art, and Core report sub-goal completion for end-effector release, target placement, articulation progress, and core sub-goal fraction, whose availability depends on the task category and difficulty. Higher is better for all metrics (↑). Top-1, Top-2, and Top-3 results are highlighted in green, blue, and orange, respectively.

Model	Level 1			Level 2						Level 3					Overall					
	SR-B↑	SR-P↑	Art↑	SR-B↑	SR-P↑	Rel↑	Place↑	Art↑	Core↑	SR-B↑	SR-P↑	Rel↑	Place↑	Core↑	SR-B↑	SR-P↑	Rel↑	Place↑	Art↑	Core↑
Videos generated from standard instructions																				
Hailuo 2.3	0.075	0.179	0.140	0.167	*0.610*	0.778	0.297	*0.774*	*0.172*	0.000	*0.312*	0.625	0.000	0.000	0.107	0.365	0.754	0.238	0.262	*0.138*
Kling 3.0	0.132	*0.256*	**0.245**	0.167	0.583	0.507	**0.466**	0.718	**0.359**	**0.125**	0.344	*0.500*	**0.188**	**0.188**	**0.146**	0.396	0.506	**0.410**	**0.336**	**0.325**
SeedDance 2.0	*0.151*	0.283	0.221	0.167	0.656	*0.834*	0.293	0.756	0.156	0.000	**0.406**	**0.750**	0.062	0.062	**0.146**	**0.445**	0.821	*0.247*	0.324	*0.138*
Veo 3.1	0.038	0.126	0.088	*0.119*	0.586	0.844	0.276	0.701	0.141	0.000	0.344	0.625	0.062	0.062	0.068	0.331	*0.809*	0.233	0.206	0.125
Wan 2.2	0.038	0.110	0.057	0.024	0.472	0.541	0.276	**0.809**	0.141	0.000	0.188	0.375	0.000	0.000	0.029	0.264	0.514	0.221	0.202	0.113
Wan 2.7	0.094	0.205	0.183	**0.214**	**0.687**	**0.916**	*0.307*	0.787	*0.172*	0.000	*0.312*	0.625	0.000	0.000	0.136	*0.410*	**0.869**	0.246	0.299	*0.138*
LTX 2.3	0.038	0.119	0.089	0.000	0.445	0.603	0.264	0.715	0.125	0.000	0.188	0.375	0.000	0.000	0.019	0.257	0.567	0.211	0.209	0.100
Wan 2.2-LoRA$_{2K}$	0.019	0.095	0.066	0.024	0.451	0.486	0.274	0.740	0.141	0.000	0.250	*0.500*	0.000	0.000	0.019	0.252	0.488	0.219	0.196	0.113
Wan 2.2-LoRA$_{7K}$	0.038	0.148	0.101	0.024	0.487	0.535	0.276	0.734	0.141	0.000	0.250	*0.500*	0.000	0.000	0.029	0.296	0.529	0.221	0.223	0.113
CosmosPolicy-DefaultCam	0.189	0.248	0.195	0.048	0.507	0.530	0.338	0.728	0.203	0.000	*0.312*	0.625	0.000	0.000	*0.117*	0.359	0.545	0.270	0.297	0.163
CosmosPolicy-BenchCam	**0.264**	**0.307**	*0.212*	0.000	0.583	0.810	0.307	0.700	*0.172*	0.000	*0.312*	0.625	0.000	0.000	0.136	0.420	0.781	0.245	*0.306*	*0.138*
Videos generated from enhanced instructions																				
Hailuo 2.3	*0.132*	*0.281*	**0.255**	0.119	0.574	0.777	0.314	0.728	*0.203*	0.000	0.406	**0.750**	0.062	0.062	0.117	0.410	0.773	*0.263*	**0.346**	*0.175*
Kling 3.0	0.113	**0.284**	0.214	0.214	0.631	0.586	**0.460**	*0.790*	**0.344**	0.000	0.250	0.375	**0.125**	**0.125**	0.146	0.423	0.552	**0.393**	0.325	**0.300**
SeedDance 2.0	0.151	0.283	*0.211*	**0.262**	**0.656**	0.797	0.302	0.762	0.219	0.000	0.250	*0.500*	0.000	0.000	**0.184**	**0.433**	0.749	0.242	*0.317*	*0.175*
Veo 3.1	0.028	0.084	0.086	*0.122*	*0.635*	**0.919**	0.281	0.826	0.145	0.000	0.188	0.375	0.000	0.000	0.071	0.359	*0.830*	0.223	0.250	0.115
Wan 2.2	0.038	0.154	0.094	0.095	0.545	0.633	0.304	0.738	0.172	0.000	0.188	0.375	0.000	0.000	0.058	0.316	0.592	0.243	0.218	0.138
Wan 2.7	0.094	0.225	0.154	0.214	0.647	*0.852*	0.343	0.733	*0.203*	0.000	**0.438**	**0.750**	**0.125**	**0.125**	*0.136*	*0.414*	0.836	0.299	0.266	0.188
LTX 2.3	0.057	0.161	0.112	0.075	0.561	0.821	0.323	0.729	0.183	0.000	0.312	0.625	0.000	0.000	0.059	0.331	0.789	0.255	0.230	0.145
Wan 2.2-LoRA$_{2K}$	0.057	0.149	0.091	0.119	0.550	0.604	*0.329*	0.786	*0.203*	0.000	0.188	0.375	0.000	0.000	0.078	0.315	0.568	0.263	0.225	0.163
Wan 2.2-LoRA$_{7K}$	0.019	0.140	0.078	0.119	0.546	0.656	0.296	**0.865**	0.172	0.000	0.188	0.375	0.000	0.000	0.059	0.311	0.611	0.237	0.229	0.138
CosmosPolicy-DefaultCam	**0.170**	0.233	0.178	0.000	0.561	0.663	0.276	0.769	0.141	0.000	0.188	0.375	0.000	0.000	0.087	0.363	0.617	0.221	0.292	0.113
CosmosPolicy-BenchCam	0.151	0.235	0.165	0.000	0.605	0.887	0.277	0.716	0.141	0.000	*0.375*	**0.750**	0.000	0.000	0.078	0.397	**0.865**	0.222	0.271	0.113
Rollout Video†	0.765	0.851	0.818	0.381	0.742	0.811	0.562	0.755	0.516	0.750	0.938	1.000	0.875	0.875	0.604	0.812	0.842	0.625	0.805	0.588
Rollout Video w/ GT Depth‡	1.000	1.000	0.950	0.952	0.979	0.905	0.866	1.000	0.953	1.000	1.000	1.000	1.000	1.000	0.981	0.991	0.920	0.893	0.960	0.963

Note. † Rollout Video and ‡ Rollout Video (w/ GT Depth) serve as oracle/reference bounds. Rankings are computed separately within each instruction setting among generation models only; oracle/reference rows are shaded in gray. Zero-valued entries are not highlighted, even when tied.

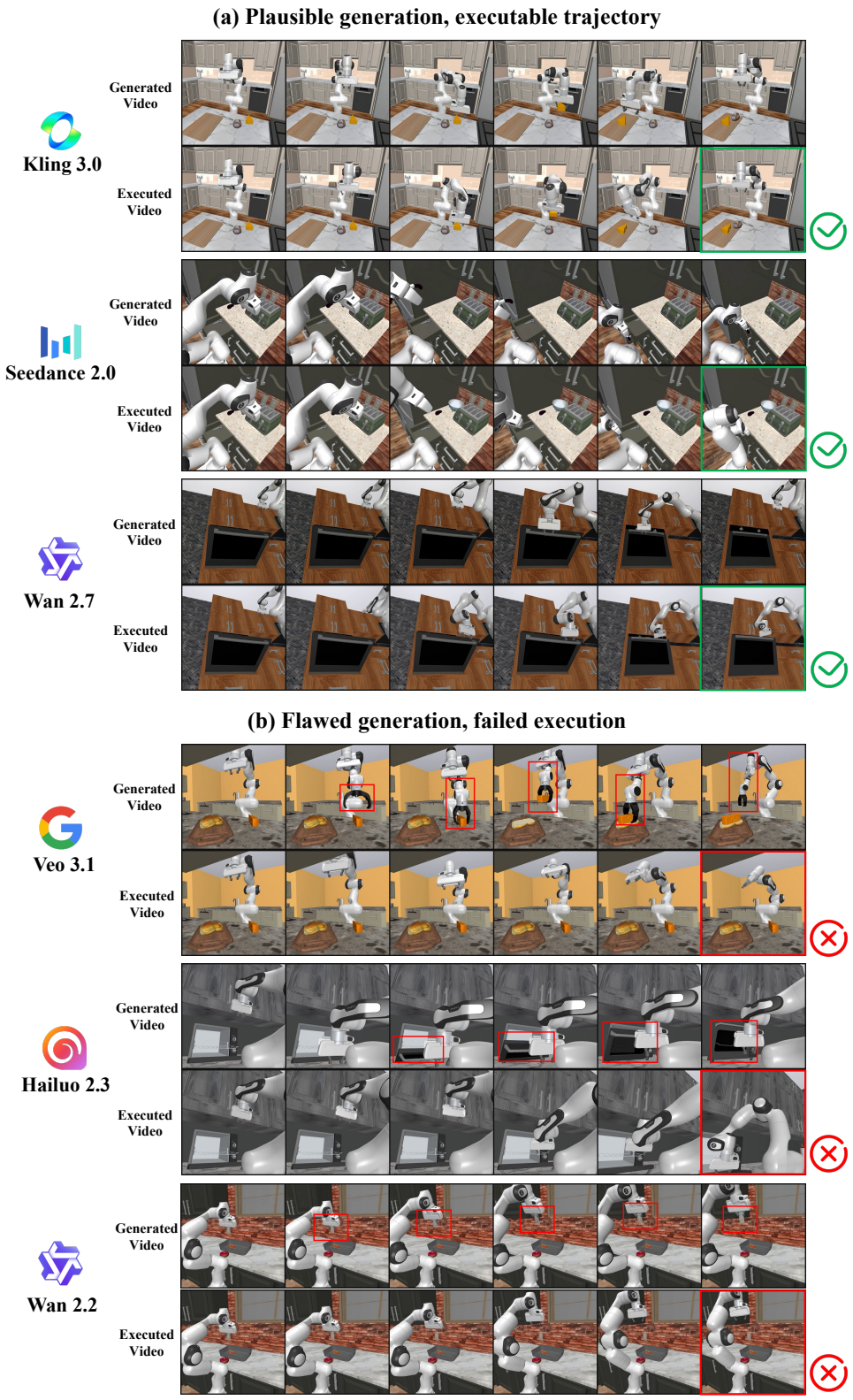

Figure 5. **Qualitative examples of video-to-execution outcomes.** Each example shows six temporally aligned frames from the generated video and the recovered execution rollout. **(a)** Successful cases show that visually plausible robot-object motion can be converted into executable trajectories and completed rollouts. **(b)** Failure cases illustrate how generation artifacts, such as inconsistent robot geometry, object-state hallucinations, and unreliable contact cues, propagate through trajectory extraction and lead to failed execution.