

Learning to Generate Rigid Body Interactions with Video Diffusion Models

David Romero¹, Ariana Bermudez¹, Viacheslav Iablochnikov¹
Hao Li^{1,2}, Fabio Pizzati¹, and Ivan Laptev¹

¹ MBZUAI, UAE

² Pinscreen, USA

{david.romero, fabio.pizzati, ivan.laptev}@mbzuai.ac.ae

Abstract. Recent video generation models have achieved remarkable progress and are now deployed in film, social media production, and advertising. Beyond their creative potential, such models also hold promise as world simulators for robotics and embodied decision making. Despite strong advances, current approaches still struggle to generate physically plausible object interactions and lack object-level control mechanisms. To address these limitations, we introduce *KineMask*, an approach for video generation that enables realistic rigid body control, interactions, and effects. Given a single image and a specified object velocity, our method generates videos with inferred motions and future object interactions. We propose a two-stage training strategy that gradually removes future motion supervision via object masks. Using this strategy we train video diffusion models (VDMs) on synthetic scenes of simple interactions and demonstrate significant improvements and generalization to rigid body and hand-object interactions in real scenes. Furthermore, *KineMask* integrates low-level motion control with high-level textual conditioning via predicted scene descriptions, leading to support for synthesis of complex dynamical phenomena. Our experiments show that *KineMask* generalizes to different VDMs and achieves strong improvements over recent models of comparable size. Ablation studies further highlight the complementary roles of low- and high-level conditioning in VDMs.

Project Page: <https://daromog.github.io/KineMask/>

Keywords: Object Interactions · Motion Conditioning Generation

1 Introduction

Recent years have seen substantial advances in video generation, with Video Diffusion Models (VDMs) emerging as a leading paradigm for high-resolution, temporally consistent synthesis [10, 26, 32, 63]. This progress has elevated visual quality and enabled early commercial use in creative performances [41], experimental filmmaking [12], and advertising [48]. Beyond content creation, VDMs have also been explored as world models [24], capable of anticipating real-world interactions and supporting robotics and embodied decision making [3, 5, 18]. Realizing this vision, however, requires strict physical plausibility and control,

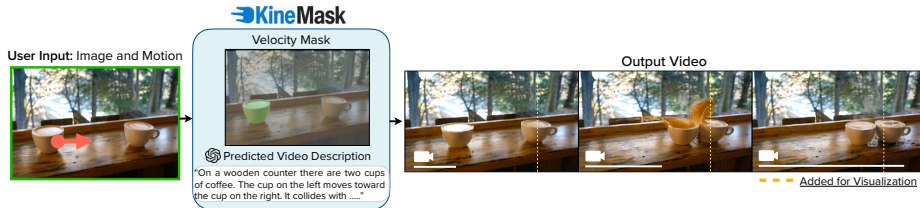


Fig. 1: We enable object-based control with a novel training strategy. Paired with synthetic data constructed for the task, **KineMask** enables pretrained VDMs to synthesize realistic rigid body interactions in real-world input scenes.

since small deviations from realistic physics can accumulate into large errors in predicted dynamics. Yet, current VDMs struggle to capture fundamental traits [31, 42] such as object permanence, collisions and causal interactions, even for large-scale models like Veo-3 [62]. Hence, methods for physically accurate video generation are a key step toward establishing VDMs as reliable world models. Recent frameworks such as WonderPlay [38], PhysGen [40] and PhysGen3D [13] integrate physics-based simulators into data-driven video generation, but rely on explicit scene reconstruction, a challenging task on its own. Differently, drag-based approaches [15, 19, 33, 36, 51, 68, 70] can move objects, but often requiring pre-defined target points, thus preventing the inference of causal effects of motion from starting conditions only. Around the time of this work, Force Prompting [20] introduced physics-guided controls into VDMs using simulated training data. While promising, it offers limited fine-grained control and object interactions realism, often producing unrealistic dynamics, and distorted shapes, showing a lack of understanding of causality. Motivated by these limitations, we propose an approach enabling video diffusion models to generate physically realistic object interactions, as a core requirement for robotics and video generation. Our approach aims to answer two central questions for world models: **(1)** Can a video diffusion model generate *realistic interactions* between rigid bodies given initial dynamic conditions?, and **(2)** how do data and textual conditioning influence the emergence of causal physical effects in generated videos?.

To address these questions, we introduce *KineMask*, a framework for generating accurate object interactions and effects in complex scenes (Figure 1). *KineMask* provides *low-level* kinematic control over parameters such as object direction and speed, while *inferring object interactions directly within the VDM*. *KineMask* is trained on simulator-rendered videos that capture not only physically valid dynamics but also explicit speed-dependent object interactions, paired with textual descriptions of the underlying events. We leverage object masks as guidance, enabling single-frame motion control and improving the model’s understanding of shapes. Beyond low-level control, *KineMask* integrates *high-level* prompt conditioning through textual descriptions of future scene dynamics. *KineMask* provides low-level Kinematic control through velocity masks, combined with high-level text conditioning, and learns physically-inspired causality and motion from simulated data, improving controllability and physical plausi-

bility. At inference, it predicts dynamics from a single input image and enables the generation of complex effects, such as liquid spilling as a result of a collision (Figure 1). Extensive experiments show that *KineMask* not only introduces new control capabilities but also outperforms state-of-the-art models of comparable size, while ablation studies highlight the importance of both the proposed training strategy and the integration of low- and high-level controls. In addition, we show that improvements of *KineMask* generalize to different VDMs. To this end, we first demonstrate advantages of *KineMask* for *CogVideoX-5B* [63] and then verify improvements by applying *KineMask* to *Wan2.2-5B* [57] and *Cosmos2.5-2B* [3] video models. In summary, we propose the following contributions:

1. We introduce KineMask, a mechanism for object motion conditioning in VDMs, based on a novel two-stage training and conditioning encoding.
2. We train KineMask on a synthetic video dataset composed of simple object interactions and demonstrate resulting models to enable generation of complex object interactions in realistic scenes.
3. We evaluate KineMask by combining low-level motion control with high-level textual conditioning, and demonstrate significant improvements of multiple video models in the task of generating realistic rigid body interactions.

2 Related Works

2.1 Video Diffusion Models

Early VDMs directly extended image generators by inserting temporal layers into denoising U-Nets [2, 7, 8, 14, 23, 60, 61]. Later, video synthesis improved through the usage of Diffusion Transformers (DiTs) [45], inheriting scaling properties from native transformer-based architectures. The use of DiTs allowed for higher-resolution videos and stronger visual quality [25, 32, 57, 63]. We build *KineMask* on CogVideoX [63], benefiting from the advantages of DiTs. Beyond creative purposes, diffusion-based video generation is now used for *world modeling*, by synthesizing the possible outcomes of actions in an environment. [5, 56] first proposed diffusion-based world models on restricted scenarios. Some like GAIA [27, 49] train at scale on specific domains such as autonomous driving, while Cosmos [3] uses large-scale data to handle heterogeneous domains. These models still suffer from limited realism in generating object interactions, motivating our study.

2.2 Control for video generation

Significant efforts have been made to extend control on generated videos beyond textual conditioning. On images, ControlNet [65] and similar approaches [43, 69] proposed plug-and-play control trainable components for dense conditioning such as edges, depth, or human poses. In videos, VideoControlNet [28] propagated conditions frame by frame using optical flow, while others focused on pose-driven human video generation [67] or keyframe-to-video propagation [35]. More recent works such as SparseCtrl [22] uses just a few keyframes for sketch, depth, or

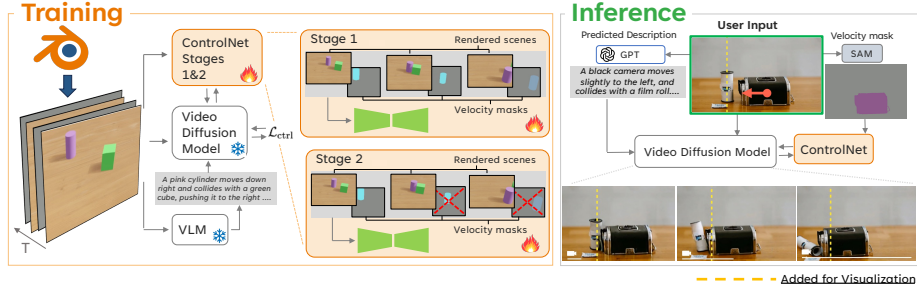


Fig. 2: **KineMask pipeline.** We encode our *low-level* control signal as a mask encoding the velocity of the moving objects, to train a ControlNet (left) in two stages using Blender-generated videos of objects in motion. In the first stage, we train with all mask frames as control. In the second stage, we randomly drop parts of the final mask frames. Additionally, we provide a *high-level* textual control extracted by a VLM. At inference (right), we construct the low-level conditioning with SAM and use GPT to infer high-level outcomes of object motion from a single frame.

image conditioning, ignoring motion conditions. MotionI2V [50] first infers object motion with flow generation, then it renders a video using flow as control. A significant line of works allows motion control with dragging trajectories [17, 19, 33, 36, 51, 58, 64, 68, 70]. While effective, these require pre-defined trajectories, and are unable to infer motion causal effects from initial conditions. Some drive motion by directly guiding attentions [46] or noise [11] using reference videos. Beyond single modalities, Cosmos-Transfer [1] demonstrates adaptive multimodal conditioning. To our knowledge, *KineMask* is the first approach training VDMs to generate rigid body interactions by controlling *initial object velocity only*.

2.3 Physics-aware video generation

The interactions between physical understanding and video generation is a growing research field. A first line of work integrates physical simulations with learning-based techniques. In particular, PhysDreamer [66] generates oscillatory motions on 3D Gaussians, while DreamPhysics learns physical properties of dynamic 3D Gaussians with video diffusion priors [29]. On video generation, WonderPlay [38] bridges physics solvers and generative video models to synthesize dynamic 3D scenes across diverse physical phenomena. PhysGen3D [13] and PhysGen [40] apply a similar approach using 3D and 2D information respectively for rigid body interactions. However, the use of a simulator requires significant engineering efforts and limits the flexibility of models. Differently, others [9, 16] map input kinematics to future frames, but requiring dataset-specific training. Instead, C-Drag [37] infers causal motion on output videos with LLMs, and combines it with tracking control to allow video generation. Using purely diffusion models, InterDyn [4] explores the capability of video models to render realistic object dynamics. However, it exploits frame-wise masks of controlling elements, which

are typically unavailable at test time. Similarly to us, [34] explores physics post-training but focuses on gravity effects. Goal Force [21] focuses on goal forces and Force Prompting [20] explores similar ideas to ours but does not consider object interactions and deploys simpler motion control. We compare to Force Prompting and demonstrate improved performance synthesizing object interactions.

3 KineMask

KineMask is designed to synthesize realistic interactions among rigid bodies given an input image and the object velocity encoded by an object mask, as in Figure 2. Below, we introduce preliminaries in Section 3.1. We then describe our conditioning mechanism in Section 3.2 and outline the training and inference procedure in Section 3.3.

3.1 Preliminaries

Video diffusion Models. VDMs generate data by reversing the noising process. In training, a clean video $\mathbf{x}_0 \in \mathbb{R}^{F \times H \times W \times C}$ is perturbed into a noisy version $\mathbf{x}_t$ by adding Gaussian noise at a randomly sampled timestep t , and the model is optimized to approximate the corresponding reverse transition. At inference, the process is inverted: starting from pure Gaussian noise $\mathbf{x}_T$, the model denoises through intermediate states $\{\mathbf{x}_t\}_{t=1}^T$ until it recovers a clean output video $\mathbf{x}_0$ after T steps. To be grounded to real scenes, we use image-to-video (I2V) models, where the video synthesis is conditioned on a reference image $\mathbf{y}$. Formally, we denote by p_θ the VDM with parameters θ . Conditioned on a high-level textual description c and a reference image $\mathbf{y}$, the denoising step is:

$$\mathbf{x}_{t-1} \sim p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t, c, \mathbf{y}), \quad (1)$$

where $\mathbf{x}_{t-1}$ is a tensor defined over the frame dimensions. The training loss minimizes the KL divergence between the true reverse conditional p and the model distribution:

$$\mathcal{L}_{\text{diff}}(\theta; \mathbf{x}_0, \mathbf{y}, c, t) = D_{\text{KL}}(p(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0) \parallel p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t, c, \mathbf{y})). \quad (2)$$

In practice, this objective is implemented by a noise prediction task in which the network learns to estimate the Gaussian noise added to $\mathbf{x}_0$.

ControlNet. To allow additional guidance, a ControlNet [65] branch ψ_ϕ , parameterized by ϕ , can encode an arbitrary dense control signal $\mathbf{u} \in \mathbb{R}^{F \times H \times W \times D}$ driving the output generation. For more details, please refer to [65]. The denoising step then becomes:

$$\mathbf{x}_{t-1} \sim p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t, c, \mathbf{y}, \psi_\phi(\mathbf{u})). \quad (3)$$

When training the ControlNet, the parameters of the backbone model θ are kept frozen, while only the control branch ϕ is optimized. The corresponding loss is:

$$\mathcal{L}_{\text{ctrl}}(\phi; \mathbf{x}_0, \mathbf{y}, \mathbf{u}, c, t) = D_{\text{KL}}(q(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0) \parallel p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t, c, \mathbf{y}, \psi_\phi(\mathbf{u}))). \quad (4)$$

3.2 Enabling Motion Control

First-Stage training. We now want to enable object-wise motion control for KineMask. Specifically, our goal is to move an object in an input scene $\mathbf{y}$ with controlled direction and velocity, allowing us to study the effects of object interactions in the videos generated with diffusion models. To do so, we assume access to a dataset $\mathcal{D}$ of captioned videos depicting objects in motion. Let $f \in \{1, \dots, F\}$ denote the frame index. For each frame f and object in the scene, we are given a mask $\mathbf{m}_f \in \mathbb{R}^{H \times W \times 3}$ aligned with the image resolution. The three channels encode the instantaneous velocity vector, with the red, green, and blue channels corresponding to motion along the x -, y -, and z -axes, respectively, in the pixels defined by a segmentation mask of the object. In this way, $\mathcal{D}$ provides not only spatial information about object locations, but also explicit ground-truth dynamics. The velocity masks are then aggregated into a tensor $\mathbf{m} \in \mathbb{R}^{F \times H \times W \times 3}$ and used to condition ψ_ϕ . Similarly to [4], *we annotate in this way only the velocity of the objects moving in the first frame of the rendered video*, leaving blank the mask for objects potentially moved by interactions. We visualize our strategy in Figure 2 (top). This enforces the model to synthesize interactions without explicitly relying on pixel control. We can then train a *first-stage* KineMask ControlNet ϕ' by solving:

$$\phi' = \arg \min_{\phi} \mathbb{E}_{(\mathbf{x}_0, \mathbf{y}, \mathbf{m}, c) \sim \mathcal{D}, t} \left[\mathcal{L}_{\text{ctrl}}(\phi; \mathbf{x}_0, \mathbf{y}, \mathbf{m}, c, t) \right], \quad (5)$$

The ϕ' network learns to map dense pixel-wise supervision into structured guidance for object motion in the generated videos. We show the training masks in Figure 2 (top), more examples can be seen in Appendix.

Second-Stage Training KineMask ϕ' enables motion control given motion masks $\mathbf{m}$ provided for all the frames of a video. While such a setup simplifies training, it does not correspond to our desired scenario of video generation conditioned *only* by the object motion at the first video frame. Towards this goal, we use a mask dropout strategy, randomly erasing part of the velocity masks in $\mathbf{m}$ at training time, as shown in Figure 2 (bottom). Formally, we define a truncated mask tensor:

$$\mathbf{m}_\odot = \{\mathbf{m}_{\odot, f} = \mathbf{m}_f \text{ if } f \in f^*, \mathbf{0} \text{ otherwise}\}_{f=0}^F. \quad (6)$$

where f^* denotes the masked frame indices, and depends on the dropout ratio. Thus, only some frames contain velocity supervision, while the remainder of the sequence is set to zero. We then train a *second-stage* KineMask ϕ'' by finetuning ϕ' with this strategy, solving:

$$\phi'' = \arg \min_{\phi''} \mathbb{E}_{(\mathbf{x}_0, \mathbf{y}, \mathbf{m}, c) \sim \mathcal{D}, t} \left[\mathcal{L}_{\text{ctrl}}(\phi'; \mathbf{x}_0, \mathbf{y}, \mathbf{m}_\odot, c, t) \right]. \quad (7)$$

As a result of the dropout during training, the VDM equipped with ϕ'' is able to move objects by taking as input only the *initial* velocity, with $\mathbf{m}_\odot = \{\mathbf{m}_0, \mathbf{0}, \dots, \mathbf{0}\}$. Ultimately, to render realistic videos, the VDM *must synthesize motion dynamics* starting from initial conditions only.

3.3 Data

Training. We show our training pipeline in Figure 2 (left). For training ϕ'' , we assumed the availability of a dataset $\mathcal{D} = \{(\mathbf{x}_0, \mathbf{y}, \mathbf{m}, c)\}$. Besides the target video $\mathbf{x}_0$ and the reference conditioning image $\mathbf{y}$, we require both *low-level* and *high-level* conditioning for physical dynamics. At the low level, we require the aggregated velocity masks $\mathbf{m}$ defined in Section 3.2. At the high level, instead, we associate each video with a textual description c summarizing the effects of physical interactions. Since collecting real-world videos with such annotations is impractical, we generate synthetic data in Blender. Importantly, such simulated data still allows to generalize to real scenes, as we empirically verify in our experiments. We render scenes with boxes and cylinders placed on textured surfaces, and assign to each controlled object an initial velocity with random direction and magnitude. This procedure yields $\mathbf{x}_0$ as the rendered video, $\mathbf{y}$ as the first frame of the sequence (used for image-to-video conditioning), and $\mathbf{m}$ as the stack of per-frame velocity masks, which provide supervision of motion. To obtain the high-level descriptions c , we instead process each rendered video with a vision-language model (VLM), prompted to provide detailed video captions with particular focus on object interactions. The full prompt used for caption generation is reported in Appendix.

Inference. We assume as input an unseen image $\mathbf{y}$. An object mask is obtained for the target object using SAM2 [47], while the desired object velocity at the first frame is assumed to be provided by the user. We use this information to construct $\mathbf{m}_\odot$. We also prompt GPT-5 [44] for a description c_{infer} of the effects on the scene if the object starts moving in the direction indicated by the user. The full prompt is in Appendix. Combining those with random noise $\mathbf{x}_T \sim \mathcal{N}(0, 1)$, we construct the input tuple $\{\mathbf{x}_T, \mathbf{y}_{\text{input}}, \mathbf{m}_\odot, c_{\text{infer}}\}$ compatible with our VDM equipped with ϕ'' . Our inference pipeline is illustrated in Figure 2 (right). Please note that at inference the user simply draws an arrow, mapped to RGB masks automatically.

4 Experiments

We describe the experimental setup in Section 4.1. Section 4.2 presents our main results and compares KineMask to the state of the art. Finally, Section 4.3 shows a comprehensive analysis with ablations for different components of our method. Further ablations are in Appendix.

4.1 Setup

Datasets. We generate two synthetic datasets for training and evaluation. Following Section 3.3, we render cubes and cylinders with random colors, moving on textured backgrounds from AmbientCG [6]. The first *Interactions* dataset contains objects moving in random directions and interacting with each other. We also construct a *Simple Motion* dataset, where isolated objects are moving in random directions without collisions. For both, we generate 10,000 training and 100 test samples. Test videos use disjoint colors and textures compared to



Fig. 3: **Comparison with CogVideoX.** While CogVideoX often suffers from several failure modes, such as hallucinations and incorrect motions, *KineMask* follows target motion and generates realistic object interactions. We improve object interactions in collisions and show causal effects of object motion.

training to ensure diversity. We visualize samples of the generated data in Appendix. We also include a *Real World* set of 50 images, collected from the web [54] or generated with ChatGPT’s image generator [30], used to assess generalization to real scenes with complex objects. Note that unlike *Simple Motion* and *Interactions*, *Real World* does not include ground-truth motion trajectories. We use Tarsier [59] to extract captions c for synthetic data.

Implementation. We adopt CogVideoX-I2V-5B (CogVideoX) as backbone for KineMask. We use ControlNet on the first 8 layers of the model, with weight at 0.5. For ϕ'' , we also apply a non-uniform sampling strategy for f^* , where frame selection is biased toward earlier frames, as rigid body interactions occur most frequently at the beginning of the simulated sequences. Further details are provided in Appendix. We generate 49 frames at inference.

Baselines. We consider two baselines using pretrained image-to-video models: CogVideoX [63] and Wan2.2-I2V-5B [57] (Wan). We prompt both baselines with the same c_{infer} as used for KineMask. Since we use CogVideoX as a backbone for KineMask, these two methods only differ by our training procedure. We choose TORA [68], also based on CogVideoX, as a drag-based baseline. Since TORA requires the full drag trajectory, we draw it starting from the target object depending on the input velocity (see Appendix). We also include MotionI2V [50] as mask-constrained drag method, that we prompt using SAMv2 masks and a drag trajectory similar to TORA. Finally, we evaluate Force Prompting (FP) [20], by mapping our input velocity to input force prompts [20]. Force Prompting is also built on CogVideoX. More details are in Appendix.

Metrics. For visual quality, we report the Fréchet Video Distance (FVD) [55] and the mean squared error (MSE) between generated and ground-truth videos

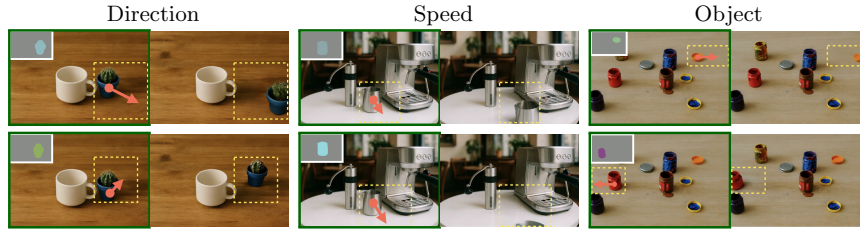


Fig. 4: **Degrees of freedom.** We show control of different aspects of KineMask outputs. We can choose different directions (left), speed (middle), and objects to move (right), opening potential for world modeling.

in our synthetic test sets. For motion, we compute the Fréchet Video Motion Distance (FVMD) [39], which isolates motion quality from appearance. Finally, we use SAM2 [47] to extract semantic masks of objects in generated videos, and compute Intersection over Union (IoU) with ground-truth masks and assess geometric consistency.

4.2 Comparison with baselines

Qualitative Comparison. We demonstrate qualitative improvements on *Real World* scenes. While for space reasons we compare only with the backbone CogVideoX in Figure 3, we include comparisons against all baselines in the Appendix. We noticed that CogVideoX suffers from unrealistic interactions, making objects fly or disappear. Instead, KineMask generates realistic interactions with other objects if they are present in the path of motion of the initial moving object, showing a correct understanding of rigid body dynamics. We also show (Figure 3, bottom left) complex interactions that require implicit 3D understanding, making the glass of juice fall and crash as a result of motion. Our realistic real world outputs show a strong generalization of the knowledge acquired from simulated videos.

User Study. In Figure 5, we show results of a user study with 30 participants, who perform pairwise comparisons of videos generated by KineMask and baselines on *Real World*. Participants are presented with initial images and input object velocities indicated by arrows. We ask them to

evaluate outputs on three different axes: motion fidelity to the control signal (Motion), realism of the object interactions (Interactions), overall physical consistency (Physical). Each answer is collected with a three-answer forced choice format, where we compare our outputs with baselines and we ask for user preferences, allowing also to reply “both have the same quality”. The full details are

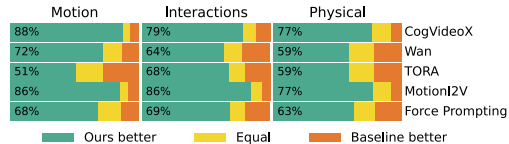


Fig. 5: We widely outperform baselines on motion fidelity, interaction quality, and overall physical consistency.

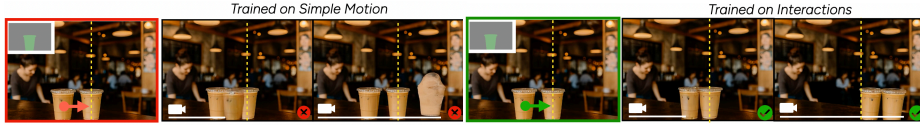


Fig. 6: **Impact of training data.** While *KineMask* trained on *Simple Motion* is able to generalize to *Real World* images, the lack of object interactions in *Simple Motion* results in hallucinations (top). Training on *Interactions* results in collisions and plausible motion of pushed objects (bottom).

presented in Appendix. The collected user preferences indicate that *our method significantly outperforms all baselines in all questions*. This demonstrates that existing methods are unsuitable for rendering interactions with starting conditions only. Kinemask, instead, renders realistic interactions.

4.3 Analysis and ablation studies

Low-level motion control. In our first set of experiments, we evaluate KineMask on low-level motion control. To do so, we train KineMask on *Simple Motion* with a prompt $c_\emptyset = \text{“An object moving on a surface”}$, hence discarding the rich description c extracted by Tarsier. At inference, we assume $c_{\text{infer}} = c_\emptyset$. Doing so, we isolate the effects of low-level conditioning from high-level textual control, allowing for a fair assessment of KineMask as a motion conditioning method.

Fine-grained control. In Figure 4 we present results on different degrees of freedom of KineMask. Despite being trained on basic synthetic data, KineMask generalizes motion control to complex real-world scenes, coherently with the findings in 20. In particular, we show that KineMask achieves disentangled control over different directions, speed, and objects. This allows for a fine-grained evaluation of motion dynamics, precious for world modeling.

Two-stage training. Table 1 shows the importance of our two-stage training introduced in Section 3.2. We test KineMask on *Simple Motion* only, to study the effects of motion control in isolation. We report results of CogVideoX prompted with c as a lower bound. As baseline, we train directly ϕ'' (2nd stage only), without first-stage pretraining. As an upper bound, we also evaluate ϕ' using ground truth object masks provided for every frame at test time (1st stage w/ masks). Our two-stage strategy boosts considerably all metrics compared to training ϕ'' directly, approaching the upper bound. We noticed that the IoU is very sensitive to minor displacement errors of the moving objects (0.367 vs 0.684), that do not impact the quality of motion as captured by the other metrics.

Simple Motion				
Method	MSE↓	FVD↓	FMVD↓	IoU↑
CogVideoX	158.3	601.1	1504.6	0.051
2nd stage only	86.3	288.8	201.0	0.237
Ours (1st + 2nd stage)	47.2	160.3	199.8	0.367
1st stage w/ masks	24.9	89.7	165.6	0.684

Table 1: **Ablation on training.** We show that our two-stage training strategy considerably boosts results and approaches an upper bound exploiting privileged information (1st stage w/ masks).

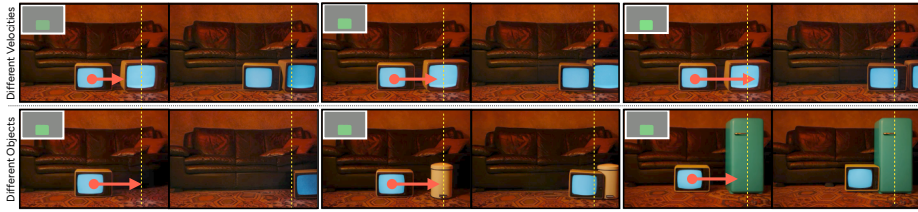


Fig. 7: On top, we show how different velocities impact motion effects, suggesting causal understanding. At the bottom, we show that the environment impacts results, for the same velocity but different objects, suggesting mass awareness.

Impact of data. Next, we explore the effects of training data on KineMask for generating realistic rigid body interactions. We compare models trained on synthetic samples with and without interactions, and draw conclusions on the capability of VDMs to learn motion. We still use $c_\emptyset$ at both training and inference, as in Section 4.3 (Low level motion control).

Data influence. We evaluate KineMask’s capability to synthesize object interactions depending on the training data. We first test the model trained on *Simple Motion* and apply it to *Real World* data. We deliberately set input velocities to directions that should generate collisions among objects in the scene. The VDM fails to render realistic collisions (Figure 6, left). However, *KineMask* trained on the *Interactions* set produces accurate interactions for the same input velocities (Figure 6, right). Hence, *KineMask trained on appropriate data allows to render complex object interactions*. Moreover, this experiment yields important insights: while VDMs are robust to visual distribution shift, as even synthetic training data do not influence the realism of *Real World* generated scenes, their preservation of motion understanding seems to be more impacted, since training on *Simple Motion* prevents to generate interactions.

Quantitative evaluation. We validate the importance of specific data with a quantitative evaluation on *Interactions*, following the same protocol as in Section 4.3 (Low-level motion control). As shown in Table 2, KineMask achieves the best results when trained on the *Interactions* dataset. Notably, even when trained only on *Simple Motion*, it greatly outperforms CogVideoX Zero-Shot and CogVideoX finetuned on *Interactions*, highlighting the need of our research to improve object interactions.

Method	Training	Interactions			
		MSE↓	FVD↓	FMVD↓	IoU↑
CogVideoX	-	344.6	807.3	3514.9	0.192
CogVideoX	Interactions	201.9	368.2	162.9	0.243
KineMask	<i>Simple Motion</i>	166.2	301.1	160.5	0.334
KineMask	<i>Interactions</i>	158.7	250.7	143.8	0.355

Table 2: **Effects of data.** Compared with models trained on *Simple Motion*, training on *Interactions* data considerably boosts performance in all metrics.

Emergence of causality. Figure 7 shows results of KineMask trained on *Interactions* and tested with different inputs on a *Real World* scene with interacting objects. On top, as velocity increases, the interactions also change, indicating

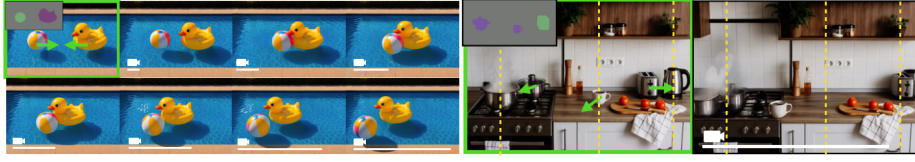


Fig. 8: **Multi-Object Control.** Generation examples of *KineMask* for multiple object control and interactions, using velocity masks for different objects as input generates objects colliding with each other or with other objects in the scene.

that the model captures causality of collisions. At the bottom, *with the same velocity*, we inpaint the input image, removing the interacting object, or replacing it with objects of different mass. The different final position of the moving object demonstrates understanding of mass in collisions. Note that this result is challenging for drag-based methods since they imply prior knowledge of the object final position. *KineMask* instead maps different target positions to the same input, depending on the environment.

Multi-Object Control In Figure 8 we show examples of multi-object control, and interactions. We set initial velocity masks for different objects, and show that *KineMask* can render motion and interactions for multiple objects at the same time. Left, shows two objects moving towards each other, while right shows multiple interactions with 3 objects in a kitchen. As visible, we preserve realism of interactions if both objects collide with each other or with other objects.

Generalization. By training on interactions *KineMask* is able to generalize its knowledge not only to object-object interaction but also to hand-object and hand-tool-object interaction as it is shown in Figure 9. By exploiting the knowledge gained in our two stage training with synthetic data to learn plausible collisions, and its powerful prior knowledge from the real world, *KineMask* is able to generate plausible interactions. Notice that CogVideoX generates unrealistic motion, where the objects start to move in a certain direction without any touch from the hand-tool. While *KineMask* follows the input given by the user, respects causality and generates plausible interactions. Furthermore, our work focuses on understanding causality and rigid body interactions to study potential outcomes of actions in the world, which can be applicable to fields like robotics. Figure 10 show a sequential generation using an image of hands or a robotic gripper manipulating objects, in which we apply *KineMask* sequentially. We also propose actions using text, to showcase the preservation of textual control.

High-level text conditioning. We now enable the usage of c at training and c_{infer} at inference. This allows us to assess the effects of text introduced in *KineMask*, and to draw insights on generalization.

Qualitative evaluation. We compare *KineMask* trained with c_0 against training with c , while using c_{infer} for inference in both cases. Doing so, we aim to evaluate the effects of textual prompts describing interactions during training. As shown

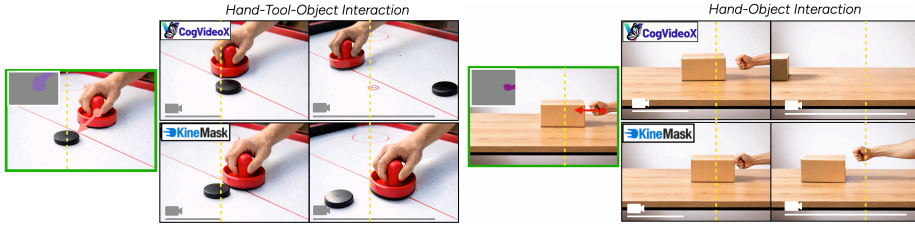


Fig. 9: **Generalization.** *KineMask* trained on *Interactions* is also able to generalize its knowledge to *Hand-Tool-Object* (Left) and *Hand-Object* (Right) interactions, showing plausible generations that respect causality.

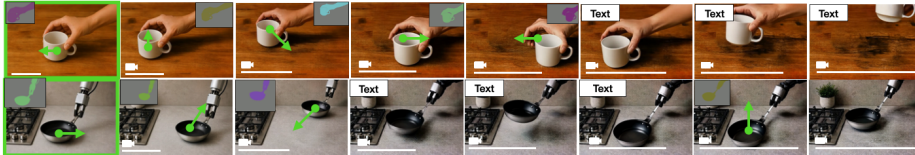


Fig. 10: **Sequential Generation.** Generation examples applying *KineMask* sequentially, by combining our velocity masks and text only.

in Figure 11, training with rich captions c allows the method to render interactions beyond those used for training, successfully exploiting the VDM prior knowledge in the rendered videos. Indeed, the model trained with c_0 fails to follow some effects in c_{infer} , such as vase breaking, or small waves forming. Instead, the model trained with c is able to benefit from additional information described in c_{infer} , correctly breaking the vase and perturbing the water. Please also note that the captions used in training are always tied to synthetic elements in *Interactions*, and therefore have limited diversity. Nevertheless, this still enables successful transfer to c_{infer} prompts describing complex effects that go beyond the training distribution of *KineMask*.

Quantitative evaluation. We evaluate metrics with different textual conditioning in Table 3. We consider trainings with c_0 , using c_0 or c_{infer} at inference. Results demonstrate that using c at training time boosts object consistency (IoU 0.376 vs 0.356 of the second best configuration), overall realism (FVD 231.3 vs 238.8, MSE on par), while we lose some motion consistency (FMVD 174.4 vs 143.8 of the best model). However, note that Table 3 includes *only evaluation on synthetic data*, where we have access to a motion ground truth. Considering the improved qualitatives on *Real World* and the benefit on complex interactions (Figure 11) we believe the use of captions c during training is still beneficial for the final model to improve robustness and preserve its capabilities.

		<i>Interactions</i>			
Train	Infer	MSE↓	FVD↓	FMVD↓	IoU↑
c_0	c_0	158.7	250.7	143.8	0.355
c_0	c_{infer}	174.4	238.8	161.3	0.356
c	c_{infer}	160.9	231.3	174.4	0.376

Table 3: **Ablation on prompts.** Training with c improve object consistency and quality on synthetic data.

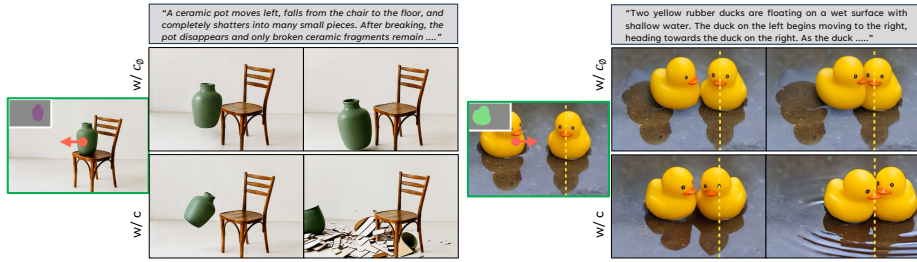


Fig. 11: **Impact of text.** Training with rich captions c makes KineMask generate effects that go beyond the synthetic data used for training, exploiting the prior knowledge of the VDM. While we prompt both models with c_{infer} , KineMask trained with c (bottom row) is able to generate complex effects of the interactions, such as crashing objects, and water effects. Full prompts in Appendix.

Generalization to different models.

Beyond CogVideoX, the *KineMask* approach generalizes to other VDM’s. To verify the generalization, we test *KineMask* in combination with Wan2.2-5B [57] and Cosmos2.5-2B [3]. Following the same protocol as in 4.2, we perform an additional human study (Figure 12) to evaluate the generations of *KineMask-Wan*, *KineMask-Cosmos* and compare with the respective backbone models for each case. The collected responses indicate that all versions of *KineMask* outperform the original models, showing better motion, interactions, and physical realism. Additionally, Figure 13 top row shows the comparison of Wan vs KineMask-Wan, at the bottom we show Cosmos vs KineMask-Cosmos. In both cases the implementation of *KineMask* improves the original generation of each model respectively, synthesizing better object interactions, showing plausible collisions with liquid or flame movement, and glass breaking as a result of a fall, contrary to the original backbones that show hallucinations in motion and interactions. As expected, we saw that the implementation of KineMask in models that have better physical prior knowledge about the real world leads to more realistic generations with *KineMask*, showing better physical effects created from interactions.

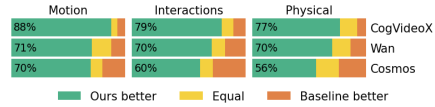


Fig. 12: All versions of *KineMask* outperform the original models, on motion fidelity, interaction quality and overall physical consistency. Showing its applicability, and generalization, bringing improvements to different video diffusion models.

5 Conclusion

We introduced *KineMask*, a framework to enable video generation of object interactions and effects in complex scenes. Our experiments show that *KineMask* achieves significant improvements over state-of-the-art models of comparable size, synthesizing realistic multi-object interactions while allowing control over

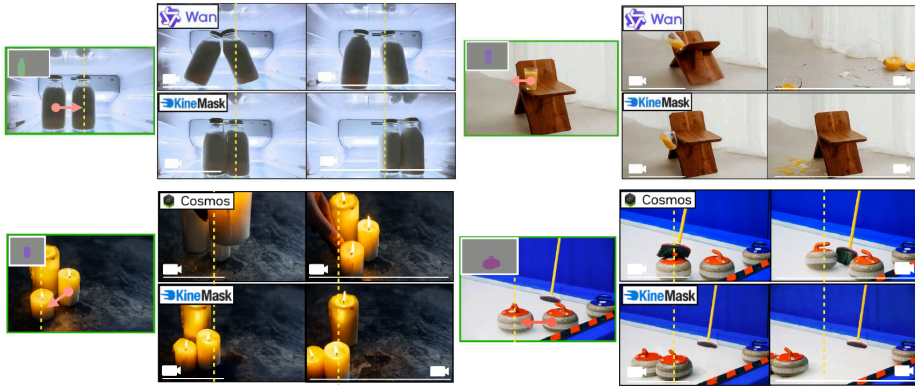


Fig. 13: **Qualitative comparison with Wan and Cosmos.** *KineMask-Wan* and *KineMask-Cosmos* follows target motion and generates realistic object interactions, while the original backbones show implausible motions. This demonstrates that apart from our original implementation on top of CogVideoX, KineMask generalize to different VDM’s bringing improvements to all models.

variable object velocities, and demonstrating its applicability and generalization to different VDMs. We believe this methodology can inform future work on world models, with potential implications for robotic manipulation, planning, and embodied decision making.

Limitations and future work. While effective, KineMask’s low-level conditioning is limited to velocity, whereas real-world motion depends on factors such as friction, mass, shape and air resistance. Moreover, we believe that extending to soft-body interactions will further improve world modeling capabilities. As shown in Section 4.3, combining text-based conditioning with KineMask enhances realism, highlighting the complementary roles of high-level textual guidance and low-level video control. These results support the joint use of both modalities for physically grounded world modeling, and we speculate that advances in multimodal language models [52, 53] may enable text-based physical reasoning to complement video generation. KineMask is an initial study of object-centric rigid-body control, focusing on center-of-mass motion rather than part-level actuation, this is not represented in our training data, but they could in principle be simulated and used to build richer control, making them a direction for future work. On the other hand, KineMask is trained with static cameras, and does not explicitly render or control camera motion. However, camera motion could also be simulated, making camera-motion also a direction for future work.

Acknowledgement. This work was supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korean Government (MSIT) (No. RS-2024-00457882, National AI Research Lab Project).

References

1. Abu Alhaija, H., Alvarez, J., Bala, M., et al.: Cosmos-transfer: Conditional world generation with adaptive multimodal control. *arXiv* (2025)
2. Agarwal, N., Ali, A., Bala, M., Balaji, Y., Barker, E., Cai, T., Chattopadhyay, P., Chen, Y., Cui, Y., Ding, Y., et al.: Video diffusion models. In: *NeurIPS* (2022)
3. Agarwal, N., Ali, A., Bala, M., Balaji, Y., Barker, E., Cai, T., Chattopadhyay, P., Chen, Y., Cui, Y., Ding, Y., et al.: Cosmos world foundation model platform for physical ai. *arXiv* (2025)
4. Akkerman, R., Feng, H., Black, M.J., Tzionas, D., Abrevaya, V.F.: Interdyn: Controllable interactive dynamics with video diffusion models. In: *CVPR* (2025)
5. Alonso, E., Jelley, A., Micheli, V., Kanervisto, A., Storkey, A.J., Pearce, T., Fleuret, F.: Diffusion for world modeling: Visual details matter in atari. *NeurIPS* (2024)
6. AmbientCG: AmbientCG. [Link](#) (2018-2025)
7. Bar-Tal, O., Chefer, H., Tov, O., Herrmann, C., Paiss, R., et al.: Lumiere: A space-time diffusion model for video generation. In: *SIGGRAPH Asia* (2024)
8. Blattmann, A., Dockhorn, T., Kulal, S., Tang, S., Wang, Z., Esser, P., Rombach, R.: Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127* (2023)
9. Blattmann, A., Milbich, T., Dorkenwald, M., Ommer, B.: ipoke: Poking a still image for controlled stochastic video synthesis. In: *ICCV* (2021)
10. Blattmann, A., Rombach, R., Ling, H., Dockhorn, T., Kim, S.W., Fidler, S., Kreis, K.: Align your latents: High-resolution video synthesis with latent diffusion models. In: *CVPR* (2023)
11. Burgert, R., Xu, Y., Xian, W., Pilarski, O., Clausen, P., He, M., Ma, L., Deng, Y., Li, L., Mousavi, M., et al.: Go-with-the-flow: Motion-controllable video diffusion models using real-time warped noise. In: *CVPR* (2025)
12. Chayka, K.: An a.i.-generated film depicts human loneliness, in ‘thank you for not answering’. [Link](#) (2023)
13. Chen, B., Jiang, H., Liu, S., Gupta, S., Li, Y., Zhao, H., Wang, S.: Physgen3d: Crafting a miniature interactive world from a single image. In: *CVPR* (2025)
14. Chen, H., Xia, M., He, Y., Zhang, Y., Cun, X., Wang, X., Shan, Y., Li, X.: Videocrafter: Open diffusion models for high-quality video generation. *arXiv* (2023)
15. Chen, T.S., Lin, C.H., Tseng, H.Y., Lin, T.Y., Yang, M.H.: Motion-conditioned diffusion model for controllable video synthesis. *arXiv preprint arXiv:2304.14404* (2023)
16. Davtyan, A., Favaro, P.: Learn the force we can: Enabling sparse motion control in multi-object video generation. In: *AAAI* (2024)
17. Deng, Y., Wang, R., Zhang, Y., Tai, Y.W., Tang, C.K.: Dragvideo: Interactive drag-style video editing. In: *ECCV*. Springer (2024)
18. Ding, Z., Zhang, A., Tian, Y., Zheng, Q.: Diffusion world model: Future modeling beyond step-by-step rollout for offline reinforcement learning. *arXiv* (2024)
19. Geng, D., Herrmann, C., Hur, J., Cole, F., Zhang, S., Pfaff, T., Lopez-Guevara, T., Doersch, C., Aytar, Y., Rubinstein, M., Sun, C., Wang, O., Owens, A., Sun, D.: Motion prompting: Controlling video generation with motion trajectories. In: *CVPR* (2025)
20. Gillman, N., Herrmann, C., Freeman, M., Aggarwal, D., Luo, E., Sun, D., Sun, C.: Force prompting: Video generation models can learn and generalize physics-based control signals. In: *NeurIPS* (2025)

21. Gillman, N., Zhou, Y., Tang, Z., Luo, E., Chakravarthy, A., Aggarwal, D., Freeman, M., Herrmann, C., Sun, C.: Goal force: Teaching video models to accomplish physics-conditioned goals (2026), <https://arxiv.org/abs/2601.05848>
22. Guo, Y., Yang, C., Rao, A., Agrawala, M., Lin, D., Dai, B.: Sparsectrl: Adding sparse controls to text-to-video diffusion models. In: ECCV (2024)
23. Guo, Y., Yang, C., Rao, A., Liang, Z., Wang, Y., Qiao, Y., Agrawala, M., Lin, D., Dai, B.: Animatediff: Animate your personalized text-to-image diffusion models without specific tuning. ICLR (2024)
24. Ha, D., Schmidhuber, J.: World models. In: NeurIPS (2018)
25. HaCohen, Y., Chiprut, N., Brazowski, B., Shalem, D., Moshe, D., et al.: Ltx-video: Realtime video latent diffusion. arXiv (2024)
26. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. NeurIPS (2020)
27. Hu, A., Russell, L., Yeo, H., Murez, Z., Fedoseev, G., Kendall, A., Shotton, J., Corrado, G.: Gaia-1: A generative world model for autonomous driving. arXiv (2023)
28. Hu, Z., Xu, D.: Videocontrolnet: Motion-guided video-to-video translation via diffusion models. In: ICLR (2024)
29. Huang, T., Zhang, H., Zeng, Y., Zhang, Z., Li, H., Zuo, W., Lau, R.W.: Dream-physics: Learning physics-based 3d dynamics with video diffusion priors. In: AAAI (2025)
30. Hurst, A., Lerer, A., Goucher, A.P., Perelman, A., Ramesh, A., Clark, A., Ostrow, A., Welihinda, A., Hayes, A., Radford, A., et al.: Gpt-4o system card. arXiv (2024)
31. Kang, B., Yue, Y., Lu, R., Lin, Z., Zhao, Y., Wang, K., Huang, G., Feng, J.: How far is video generation from world model: A physical law perspective. In: ICML (2025)
32. Kong, W., Dong, W., Wang, D., Li, Y., et al.: Hunyuanvideo: A systematic framework for large video generative models. arXiv (2024)
33. Lei, G., Wang, C., Zhang, R., Wang, Y., Li, H., Xu, W.: Animateanything: Consistent and controllable animation for video generation. In: CVPR (2025)
34. Li, C., Michel, O., Pan, X., Liu, S., Roberts, M., Xie, S.: Pisa experiments: Exploring physics post-training for video diffusion models by watching stuff drop. In: ICML (2025)
35. Li, H., Zhang, X., Xu, W., Liu, J., Wang, Y.: Keyvid: Video generation with keyframe propagation and temporal control. arXiv:2403.12006 (2024)
36. Li, X., Lin, D., Dai, B.: Motionctrl: A unified model for controllable text-to-motion generation. In: SIGGRAPH (2024)
37. Li, Y., Angel, M.C., Khan, S., Zhu, Y., Sun, J., Zhang, Y., Khan, F.S.: C-drag: Chain-of-thought driven motion controller for video generation. arXiv (2025)
38. Li, Z., Yu, H.X., Liu, W., Yang, Y., Herrmann, C., Wetzstein, G., Wu, J.: Wonderplay: Dynamic 3d scene generation from a single image and actions. In: ICCV (2025)
39. Liu, J., Qu, Y., Yan, Q., Zeng, X., Wang, L., Liao, R.: Frechet video motion distance: A metric for evaluating motion consistency in videos. In: ICML (2024)
40. Liu, S., Ren, Z., Gupta, S., Wang, S.: Physgen: Rigid-body physics-grounded image-to-video generation. In: ECCV (2024)
41. Miller, H.: Elvis is back in the building, thanks to ai—and u2. [Link](#) (2023)
42. Motamed, S., Culp, L., Swersky, K., Jaini, P., Geirhos, R.: Do generative video models understand physical principles? arXiv (2025)

43. Mou, C., Zhou, Y., Song, X., Ge, Y., Liu, W., Zhang, Y., Wang, Q., Shan, Y.: T2i-adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models. In: AAAI (2024)
44. OpenAI: GPT-5 System Card. [Link](#) (2025)
45. Peebles, W., Xie, S.: Scalable diffusion models with transformers. In: ICCV (2023)
46. Pondaven, A., Siarohin, A., Tulyakov, S., Torr, P., Pizzati, F.: Video motion transfer with diffusion transformers. In: CVPR (2025)
47. 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. In: ICLR (2025)
48. Roth, E.: Kalshi’s surreal \$2,000 ai-generated ad aired during the nba finals. [Link](#) (2025)
49. Russell, L., Hu, A., Bertoni, L., Fedoseev, G., Shotton, J., Arani, E., Corrado, G.: Gaia-2: A controllable multi-view generative world model for autonomous driving. arXiv (2025)
50. Shi, X., Huang, Z., Wang, F.Y., Bian, W., Li, D., Zhang, Y., Zhang, M., Cheung, K.C., See, S., Qin, H., et al.: Motion-i2v: Consistent and controllable image-to-video generation with explicit motion modeling. In: SIGGRAPH 2024 (2024)
51. Shi, Y., Liu, S., Zhou, H., Yang, R., Qiao, Y.: Dragdiffusion: Harnessing diffusion models for interactive point-based image editing. In: CVPR (2024)
52. Shukor, M., Fini, E., da Costa, V.G.T., Cord, M., Susskind, J., El-Nouby, A.: Scaling laws for native multimodal models. In: ICCV (2025)
53. Team, G., Anil, R., Borgeaud, S., Alayrac, J.B., Yu, J., Soricut, R., Schalkwyk, J., Dai, A.M., Hauth, A., Millican, K., et al.: Gemini: a family of highly capable multimodal models. arXiv (2023)
54. Unsplash: Unsplash: The internet’s source for freely-usable images. <https://unsplash.com/> (2024), accessed: June 2025
55. Unterthiner, T., Van Steenkiste, S., Kurach, K., Marinier, R., Michalski, M., Gelly, S.: Fvd: A new metric for video generation. In: ICLR Workshops (2019)
56. Valevski, D., Leviathan, Y., Arar, M., Fruchter, S.: Diffusion models are real-time game engines. In: ICLR (2025)
57. 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 (2025)
58. Wan, Z., Tang, S., Wei, J., Zhang, R., Cao, J.: Dragentity: Trajectory guided video generation using entity and positional relationships (2024), <https://arxiv.org/abs/2410.10751>
59. Wang, J., Yuan, L., Zhang, Y., Sun, H.: Tarsier: Recipes for training and evaluating large video description models. In: ICLR (2025)
60. Wang, J., Yuan, H., Chen, D., Zhang, Y., Wang, X., Zhang, S.: Modelscope text-to-video technical report. arXiv (2023)
61. Wang, Y., Chen, X., Ma, X., Zhou, S., Huang, Z., Wang, Y., Yang, C., He, Y., Yu, J., Yang, P., Guo, Y., Wu, T., Si, C., Jiang, Y., Chen, C., Loy, C.C., Dai, B., Lin, D., Qiao, Y., Liu, Z.: Lavie: High-quality video generation with cascaded latent diffusion models. *International Journal of Computer Vision* **133**(5), 3059–3078 (2025)
62. Wiedemer, T., Li, Y., Vicol, P., Gu, S.S., Matarese, N., Swersky, K., Kim, B., Jaini, P., Geirhos, R.: Video models are zero-shot learners and reasoners. arXiv preprint arXiv:2509.20328 (2025)
63. Yang, Z., Teng, J., Zheng, W., Ding, M., Huang, S., et al.: Cogvideox: Text-to-video diffusion models with an expert transformer. In: ICLR (2025)

64. Yin, S., Wu, C., Liang, J., Shi, J., Li, H., Ming, G., Duan, N.: Dragnuwa: Fine-grained control in video generation by integrating text, image, and trajectory. arXiv preprint arXiv:2308.08089 (2023)
65. Zhang, L., Rao, A., Agrawala, M.: Adding conditional control to text-to-image diffusion models. In: ICCV (2023)
66. Zhang, T., Yu, H.X., Wu, R., Feng, B.Y., Zheng, C., Snavely, N., Wu, J., Freeman, W.T.: Physdreamer: Physics-based interaction with 3d objects via video generation. In: European Conference on Computer Vision. pp. 388–406. Springer (2024)
67. Zhang, Y., Gu, J., Wang, L.W., Wang, H., Cheng, J., Zheng, W., Fan, C.: Mimic-motion: High-quality human motion video generation with confidence-aware pose guidance. In: CVPR (2024)
68. Zhang, Z., Liao, J., Li, M., Dai, Z., Qiu, B., Zhu, S., Qin, L., Wang, W.: Tora: Trajectory-oriented diffusion transformer for video generation. In: CVPR (2025)
69. Zhao, S., Chen, D., Chen, Y.C., Bao, J., Yuan, L., Wang, D., Wong, K.Y.K.: Uni-controlnet: All-in-one control to text-to-image diffusion models. In: NeurIPS (2023)
70. Zhou, H., Wang, C., Nie, R., Liu, J., Yu, D., Yu, Q., Wang, C.: Trackgo: A flexible and efficient method for controllable video generation. In: AAAI (2025)