Learning to Generate Highly Dynamic Videos using Synthetic Motion Data

Wonjoon Jin[1] Jiyun Won[1] Janghyeok Han[1] Qi Dai[2]
Chong Luo[2] Seung-Hwan Baek[1] Sunghyun Cho[1]

[1]POSTECH [2]Microsoft Research Asia

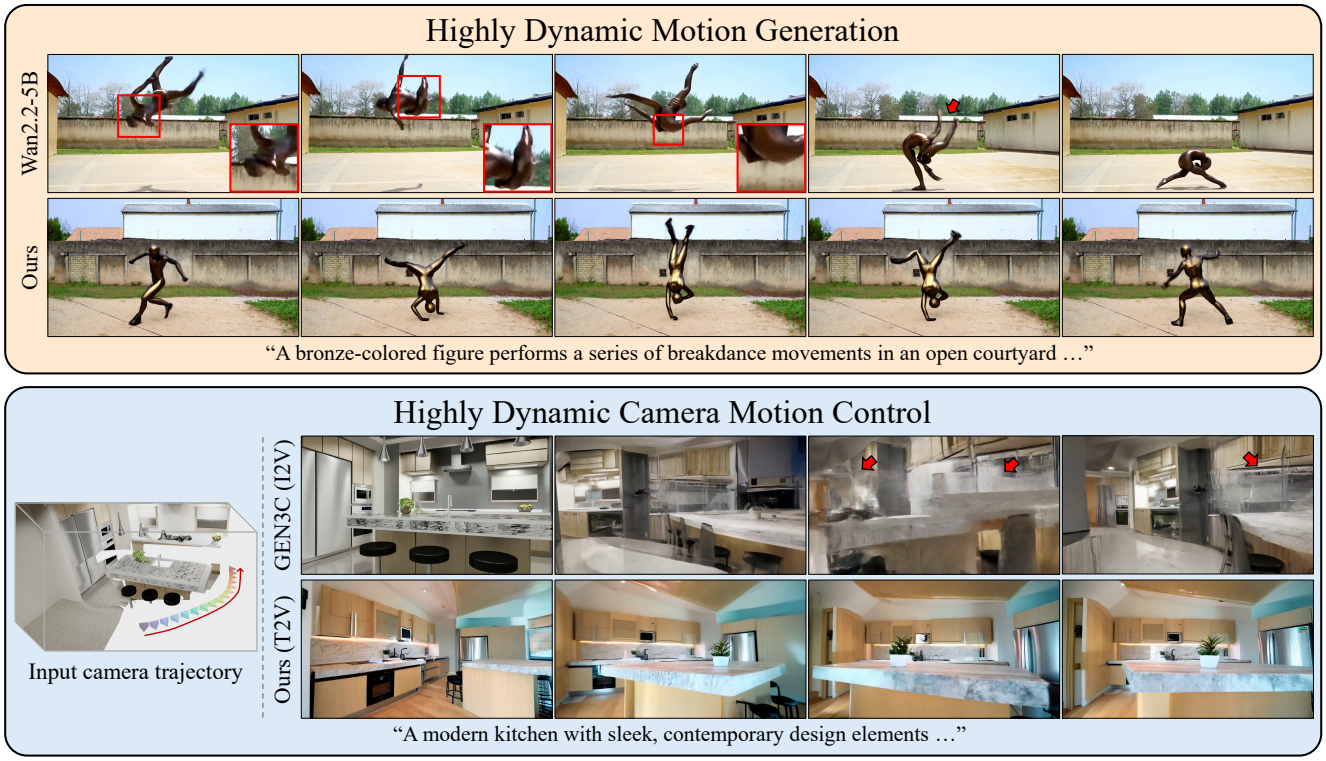


Figure 1. Examples of video synthesis results for highly dynamic object motion (top) and camera-controlled video generation with rapid viewpoint changes (bottom). Our method produces natural and highly dynamic motions, whereas Wan2.2-5B [48] generates unrealistic motion and GEN3C [40] exhibits noticeable visual artifacts. The synthesis results are best viewed in the supplementary video.

Abstract

Despite recent progress, video diffusion models still struggle to synthesize realistic videos involving highly dynamic motions or requiring fine-grained motion controllability. A central limitation lies in the scarcity of such examples in commonly used training datasets. To address this, we introduce DynaVid, a video synthesis framework that leverages synthetic motion data in training, which is represented as optical flow and rendered using computer graphics pipelines. This approach offers two key advantages. First, synthetic motion offers diverse motion patterns and precise control signals that are difficult to obtain from real data. Second, unlike rendered videos with artificial appearances, rendered optical flow encodes only motion and is decoupled from appearance, thereby preventing models from reproducing the unnatural look of synthetic videos. Building on this idea, DynaVid adopts a two-stage generation framework: a motion generator first synthesizes motion, and then a motion-guided video generator produces video frames conditioned on that motion. This decoupled formulation enables the model to learn dynamic motion patterns from synthetic data while preserving visual realism from real-world videos. We validate our framework on two challenging scenarios, vigorous human motion generation and extreme camera motion control, where existing datasets are particularly limited. Extensive experiments demonstrate that DynaVid improves the realism and controllability in dynamic motion generation and camera motion control.

1. Introduction

Highly dynamic motions, such as breakdancing and rapid camera movements, are prominent elements that enhance the visual impact in modern video content, including films, animations, and social media. However, despite recent progress in video generation [9, 38, 48, 56], synthesizing realistic videos with such motions remains challenging (Fig. 1). A primary bottleneck lies in the training data: although existing video diffusion models are trained on extremely large-scale datasets [2, 7, 15], video clips containing highly dynamic motions are relatively underrepresented. Moreover, manually collecting such videos is labor-intensive and difficult to scale, making it challenging to construct balanced datasets that adequately capture dynamic motion.

Another challenge is controllability, i.e., the ability to guide a model to generate videos with specific target motions. For example, controlling rapidly changing viewpoints during video synthesis remains particularly difficult. Existing camera-controllable video diffusion models [5, 22, 50] typically require accurate ground-truth 3D camera poses during training. However, estimating accurate camera poses for videos with extreme camera motion is highly unreliable due to minimal frame overlap. As a result, these models are usually trained on relatively simple datasets with limited camera motions, leading to degraded performance when synthesizing complex camera motions (Fig. 1).

A straightforward way to address these dataset limitations is to use synthetic data. Synthetic data generated from computer graphics pipelines provides dynamic motion scenes and precise control signals, such as camera parameters, which are difficult to capture in practice. Recent studies have explored training video diffusion models on rendered synthetic videos to improve motion fidelity and controllability [44, 60]. However, building synthetic video datasets with both highly dynamic motions and natural appearances is itself a formidable task. Rendered videos often contain artificial textures, lighting, and shadows, creating a large domain gap from real footage. As a result, models trained on such datasets tend to reproduce the artificial look of synthetic videos rather than realistic visuals.

To more effectively harness the advantages of synthetic data while mitigating its drawbacks, we propose DynaVid, a video synthesis framework that leverages *synthetic motion data*, represented as optical flow and rendered from computer graphics engines, instead of synthetic videos. Unlike videos, optical flow encodes only motion information and is naturally decoupled from appearance, thereby substantially reducing the domain gap from real motion. We render synthetic motion data along with precise control signals and use them to train video diffusion models. This approach enables the model to learn highly dynamic motion patterns and fine-grained controllability without sacrificing visual realism.

Based on this idea, DynaVid is designed as a two-stage generation framework: a motion generator synthesizes optical flow maps and a motion-guided video generator produces RGB video frames from the generated flow. We train this framework using both synthetic and real-world datasets. The synthetic motion data provides supervision for the motion generator, allowing it to learn diverse and highly dynamic motion patterns. In parallel, optical flows are estimated from real-world videos to construct motion-video pairs, which are then used to train the motion-guided video generator. Overall, our decoupled framework, equipped with the complementary strengths of synthetic motion data and real-world videos, results in natural-looking video synthesis that faithfully captures highly dynamic motions.

Our framework can be further extended to incorporate controllability. We integrate a control branch into the motion generator to condition the generation process on additional control signals. For example, by training on paired data of synthetic motion and corresponding camera parameters, the motion generator learns to produce flow maps reflecting rapidly changing camera motions. These generated motion then provides control signals to the motion-guided video generator, enabling camera-controlled video synthesis with complex viewpoint change.

We demonstrate the effectiveness of our framework with two particularly challenging scenarios: (1) vigorous human motions such as breakdancing, and (2) camera-controlled video synthesis with rapid camera motions. Extensive experiments show that our approach can generate natural-looking videos with highly dynamic motions and precise motion controllability for extreme scenarios. Our main contributions are summarized as follows:

- We construct synthetic datasets that capture dynamic motion scenes and provide precise control signals that are difficult to obtain from real-world videos.
- Instead of rendered videos with artificial appearances, we leverage rendered optical flow for training, which aligns closely with real motion and avoids appearance gaps. This enables the model to learn dynamic motion patterns from synthetic data while preserving visual realism from real videos through a decoupled, two-stage framework.
- Extensive experiments demonstrate that our method outperforms existing approaches in terms of dynamic motion generation and fine-grained camera motion control.

2. Related Work

Video diffusion models. Recent video diffusion models [9, 10, 29, 37, 38, 45, 48, 56] have achieved remarkable progress in text-to-video synthesis. Leveraging large-scale datasets and advanced diffusion frameworks [23, 31], they successfully learn spatio-temporal structures of real videos. However, open-source models such as CogVideoX and Wan [48, 56] still struggle to generate highly dynamic

scenes, primarily due to the limited diversity of motion in available training data.

Object motion synthesis. Synthesizing and controlling object motion is crucial in content creation and visual storytelling. Recent video diffusion models leverage large-scale video corpora paired with detailed text descriptions, enabling realistic motion generation [10, 45, 48]. Another line of work introduces explicit motion control using additional conditioning signals such as 2D pose sequences or motion trajectories [11, 14, 19, 24, 26, 34, 43, 50–53, 55, 57–59]. For instance, Animate Anyone [24] guides human motion generation using pose sequences, while Go-with-the-Flow [11] warps latent noise volumes according to optical flow. However, these methods often struggle to synthesize highly dynamic motions due to the scarcity of such examples in their training datasets.

Camera-controlled video synthesis. Controlling camera motion during video synthesis has recently gained significant attention [4, 5, 8, 21, 22, 28, 40, 50]. These models are typically trained on paired datasets of video frames and corresponding camera parameters so that the generated videos can follow specified camera trajectories. CameraCtrl [22] represents camera rays using Plücker embeddings computed from camera parameters, while VD3D [4] and AC3D [5] extend this idea with Diffusion Transformers. FloVD [28] and GEN3C [40] leverage depth estimation to enable 3D-aware video generation with accurate camera control. However, since these models are trained mostly on videos with limited viewpoint variation, they still struggle to synthesize videos under extreme camera motions, such as rapid rotations or large translations.

Training with synthetic data. Synthetic data created from rendering engines [16, 18] have long been employed for dense prediction tasks [12, 17, 20, 33, 39, 46, 49] and is increasingly explored in generative modeling [6, 32, 42, 44, 60]. In generative settings, Sharma et al. [42] train models on paired rendered images and physical parameters for controllable material synthesis. Zhao et al. [60] leverage rendered videos to learn natural object motion generation, while Shuai et al. [44] employ rendered videos with full 6D annotations of object and camera poses for joint object-camera control. However, models trained directly on rendered videos often suffer from unrealistic appearances due to the substantial visual discrepancy between rendered and real footage. In contrast, we leverage synthetic motion data represented as optical flow instead of rendered videos, allowing the model to learn complex motion dynamics while effectively mitigating the appearance domain gap between synthetic and real videos.

Multi-Modal generation. Diffusion models have shown strong capability in modeling cross-modal relationships across diverse domains. In video generation, a few prior studies exploit dual modalities (e.g., optical flow and RGB video) through two-stage pipelines [28, 30, 35, 36, 43]. For instance, MoVideo [30] and LFDM [35] first synthesize optical flow maps and subsequently use them to guide conditional video generation. However, these methods primarily rely on real-world data and offer limited motion controllability. In contrast, our approach introduces a control branch to enhance motion controllability and employs synthetic motion data to learn highly dynamic motion patterns beyond what real datasets provide.

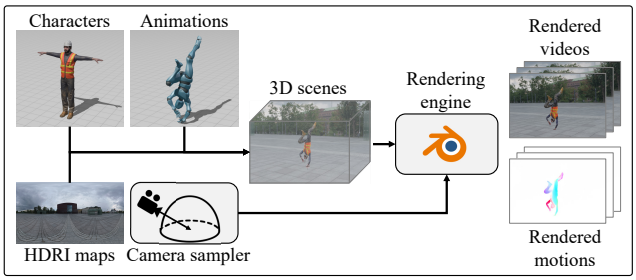


Figure 2. Overview of our dataset generation pipeline.

3. Method

We propose a novel approach that leverages synthetic motion data to enable highly dynamic video generation. Our approach particularly targets two challenging scenarios: vigorous human motions like breakdancing and dynamic camera control involving rapid viewpoint changes. To achieve this, we first construct a dataset generation pipeline that produces synthetic motion data using computer graphics pipelines. We then introduce DynaVid, our two-stage video generation framework equipped with a training strategy using both synthetic and real data.

3.1. DynaVid Datasets

We construct a dataset generation pipeline based on the Cycles renderer in Blender [16] and use it to synthesize two datasets: *DynaVid-Human* and *DynaVid-Camera*. The pipeline consists of three stages: 3D scene construction, camera trajectory definition, and rendering (Fig. 2).

DynaVid-Human. We first construct 3D scenes using publicly available assets, where vigorous human actions are represented by integrating animatable human characters with motion sequences from the Mixamo dataset [3]. We then define camera trajectories $\mathcal{C}^{syn} = \{C_n^{syn}\}_{n=1}^{N}$, where n denotes the frame index. For DynaVid-Human, a single camera position is used across all frames, randomly sampled on a hemisphere and fixed while oriented toward its center. Finally, we render synthetic RGB videos and their corresponding optical flow maps, $\mathcal{I}^{syn} = \{I_n^{syn}\}_{n=1}^{N}$ and $\mathcal{F}^{syn} = \{f_n^{syn}\}_{n=1}^{N}$, using Blender's Cycles renderer. Optical flow is computed by measuring the 3D displacement of each visible surface point between consecutive frames and

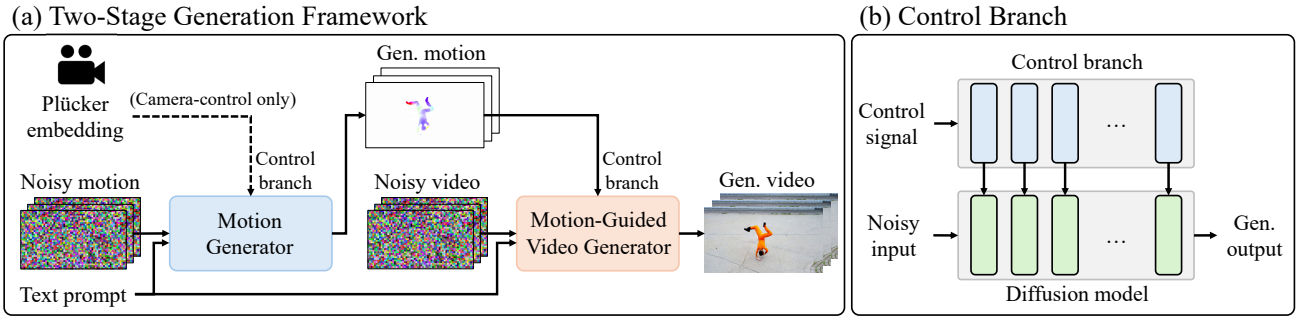


Figure 3. Overview of DynaVid. (a) The motion generator first synthesizes motion and then produces video frames conditioned on the generated motion. For camera-controlled video synthesis, Plücker embeddings are provided as additional input. (b) Our framework adopts VACE [27] to incorporate control signals such as Plücker embeddings or optical flow maps.

projecting it onto the image plane, yielding pixel-wise motion vectors. These rendered flows serve as synthetic motion data used to train the motion generator in our framework. For simplicity, we omit the frame index n in the following.

DynaVid-Camera. We adapt the same pipeline with minor modifications to simulate dynamic camera motion scenarios. Specifically, we construct complex 3D environments (e.g., urban and natural scenes) using diverse mesh assets from BlendSwap [1]. Camera trajectories are defined by preselecting several key camera positions in each scene and interpolating them using Non-Uniform Rational B-Spline (NURBS) curves to produce highly dynamic trajectories with rapid viewpoint changes. Additional details on both datasets are provided in the supplementary document.

3.2. Video and Motion Representations

Following latent diffusion models [41], we represent both videos and motions in the latent space of a pre-trained Variational Auto-Encoder (VAE). To reuse the VAE trained on RGB videos [48] for optical flow, we convert the flow sequences into the RGB domain via HSV mapping [13].

Specifically, each pixel's flow vector $f(\mathrm{p})$ is first normalized so that the flow magnitudes are more evenly distributed within the $[0, 1]$ range while preserving direction:

$$f'(\mathrm{p}) = s(f(\mathrm{p}))\frac{f(\mathrm{p})}{\|f(\mathrm{p})\|_2},\ s(f(\mathrm{p})) = \min\left(1, \sqrt{\frac{\|f(\mathrm{p})\|_2}{s_f}}\right), \tag{1}$$

where $\|\cdot\|_2$ denotes the L2 norm and s_f is a dataset-level scale factor defined as the 99th percentile of flow magnitudes. The magnitude $m = \|f'(\mathrm{p})\|_2$ and the angle $\alpha = \arctan2(f'(\mathrm{p})_y, f'(\mathrm{p})_x)$ of the normalized flow are mapped to the value and hue channels in HSV space, respectively, with saturation fixed to 1. The resulting HSV image is then converted into RGB format and encoded to the latent space by the VAE. Further details and discussion of this conversion process are provided in the supplementary document. For simplicity, we omit the notation of the latent representations for video and motion in the following sections.

3.3. DynaVid Framework

Fig. 3 (a) illustrates the overall two-stage video generation framework. This framework decouples motion synthesis from video synthesis: the motion generator synthesizes motion represented as optical flow, and the motion-guided video generator produces RGB videos conditioned on this motion. This decoupled design enables the use of synthetic motion data while avoiding the visual domain gaps that arise from synthetic videos. We build both stages on top of a recent video diffusion model [48], so the framework is largely compatible with existing diffusion-based generators.

Motion generator. Given a noisy motion $\mathcal{F}_t$ and a text condition c_{txt}, the motion generator iteratively denoises $\mathcal{F}_t$ to obtain a clean motion $\mathcal{F}$, represented as optical flow maps. This stage is responsible only for deciding "how things move," not for rendering appearance. To enable camera-controlled motion generation, we extend the motion generator with a control branch following a conditional video diffusion architecture, e.g., VACE [27], as shown in Fig. 3 (b). Specifically, the control branch receives Plücker-embeddings computed from camera parameters $\mathcal{C}$ as input and produces context feature maps. These features are injected into the transformer blocks of the motion generator through pixel-wise addition, so that the denoising process can produce flow maps that are consistent with the given camera motion. In this way, camera trajectories become an explicit control signal for the motion stage.

Motion-guided video generator. The motion-guided video generator takes a noisy video $\mathcal{I}_t$ along with the text condition c_{txt} and generated motion $\mathcal{F}$, and denoises it to produce the final video $\mathcal{I}$. We again adopt the conditional video diffusion architecture [27] to inject control signals. The overall structure is similar to that of the motion generator, except that the control signal is the optical flow, rather than the camera parameters. Because optical flow captures both object and camera motions, and the motion-guided video generator is explicitly trained to follow the provided flow, the same architecture applies to both dynamic object motion and camera-controlled video synthesis.

3.4. Training DynaVid

We train DynaVid using both synthetic and real-world datasets. During training, real-world data provide natural appearance and general motion statistics, whereas synthetic data supply precisely controlled and highly dynamic motion patterns. This combination enables the model to learn realistic video generation while capturing a wide spectrum of motion dynamics.

Motion generator. The motion generator is trained from two motion sources: synthetic motion data $\mathcal{F}^{\text{syn}}$ and real motion data $\mathcal{F}^{\text{real}}$. For $\mathcal{F}^{\text{real}}$, we extract optical flow from real videos using an off-the-shelf flow estimator [49]. We use an internal video dataset, containing scenes similar to those in the Pexels dataset [2], for dynamic-object scenarios, and the RealEstate10K dataset (RE10K) [61] for camera-controlled scenarios. For $\mathcal{F}^{\text{syn}}$, we use rendered motion data from DynaVid-Human and DynaVid-Camera, where ground-truth optical flow is directly obtained from the renderer (Sec. 3.1).

Training proceeds in two stages. We first pretrain the motion generator on $\mathcal{F}^{\text{real}}$ to learn general, in-the-wild motion statistics. We then fine-tune it with $\mathcal{F}^{\text{syn}}$ to expand its capability toward highly dynamic motions. During fine-tuning, each batch contains a mixture of real and synthetic flows so that the model does not forget natural motions while acquiring extreme ones. We use the Flow Matching objective [31] in both training stages:

$$\mathbb{E}_{\mathcal{F},c_{\text{txt}},C,\epsilon,t_f}\left[\|\hat{u}^{\mathcal{F}}(\mathcal{F}_{t_f};c_{\text{txt}},C,t_f)-v^{\mathcal{F}}\|_2^2\right], \tag{2}$$

where $\mathcal{F}_{t_f}=(1-t_f)\mathcal{F}_0+t_f\mathcal{F}_1$, $v^{\mathcal{F}}=\epsilon-\mathcal{F}$ is the target velocity and $\epsilon\sim\mathcal{N}(0,I)$. The camera parameter C is used only for the camera-controlled setting.

Motion-guided video generator. The motion-guided video generator is trained on real-world paired datasets of video frames $\mathcal{I}^{real}$ and their corresponding optical flow maps $\mathcal{F}^{real}$, where $\mathcal{F}^{real}$ are extracted from $\mathcal{I}^{real}$ as described above. We use our internal dataset for both dynamic-object and camera-controlled scenarios. This supervision guides the model to synthesize realistic visual appearances while faithfully following the input motion.

To further improve motion fidelity, we introduce a simple yet effective dataset filtering technique. Since real-world motion-video pairs are constructed using estimated optical flow, they inevitably contain estimation errors. We quantify these errors via flow cycle consistency computed from forward and backward flows, where the maximum observed error reaches 1080.05 pixels. To mitigate this issue, we discard motion-video pairs with large consistency errors by applying a threshold of 1.19 pixels, corresponding to the 90th percentile of the consistency error distribution. This filtering reduces the influence of inaccurate flow supervision and ensures that the model faithfully follows the input motion.

The motion-guided video generator is trained with the Flow Matching objective [31]:

$$\mathbb{E}_{\mathcal{I},\mathcal{F},c_{\text{txt}},\epsilon,t_I}\left[\|\hat{u}^{\mathcal{I}}(\mathcal{I}_{t_I};c_{\text{txt}},\mathcal{F},t_I)-v^{\mathcal{I}}\|_2^2\right], \tag{3}$$

where $\mathcal{I}_{t_I}=(1-t_I)\mathcal{I}_0+t_I\mathcal{I}_1$ and $v^{\mathcal{I}}=\epsilon-\mathcal{I}$ denotes the target velocity for the video representation. Note that camera parameters are not used in this stage, as camera control is already encoded in the motion generated by the motion generator. This training scheme enables the model to synthesize natural-looking videos while capturing highly dynamic motions that are rare or absent in real-world data.

4. Experiments

Implementation details. We use Wan2.2-5B [48] as the backbone for both the motion generator and the motion-guided video generator. Following Wan2.2-5B, we synthesize 121 frames per sample at a resolution of 704×1280 for both RGB frames and optical flow maps. For the control branch architecture, we adopt VACE [27]. Optical flow for real-world videos is extracted using WAFT [49], an off-the-shelf optical flow estimator. Additional training details are provided in the supplementary document.

Evaluation datasets and metrics. We evaluate our framework under two scenarios: object motion synthesis and camera-controlled video synthesis. For each scenario, we prepare both real and synthetic datasets: the real datasets contain common scenes, while the synthetic datasets include highly dynamic motions.

For the object motion synthesis scenario, we use Pexels [2] for common scenes and the DynaVid-Human test set for highly dynamic scenes. For the camera-controlled synthesis scenario, we use the RE10K test set [61] for common scenes and the DynaVid-Camera test set for highly dynamic camera motion control. We directly use the ground-truth camera parameters provided in the RE10K and DynaVid-Camera test sets. From the real datasets (Pexels and RE10K), we randomly sample 100 video clips for evaluation. For the synthetic datasets (DynaVid-Human and DynaVid-Camera), we additionally render 100 videos using our dataset generation pipeline (Sec. 3.1).

We evaluate our models in terms of visual and motion quality, camera controllability, and motion fidelity. For visual quality, we report Fréchet Video Distance (FVD) [47], aesthetic quality (A-Qual), and imaging quality (I-Qual) [25]. For motion quality, we measure motion smoothness (M-Smooth) and temporal flickering (T-Flick) [25]. The four metrics, A-Qual, I-Qual, M-Smooth, and T-Flick, are computed using VBench [25]. To assess camera controllability, we compute the mean rotation error (mRotErr) following CameraCtrl [22]. Finally, we evaluate motion error (M-Err) of the motion-guided video generator

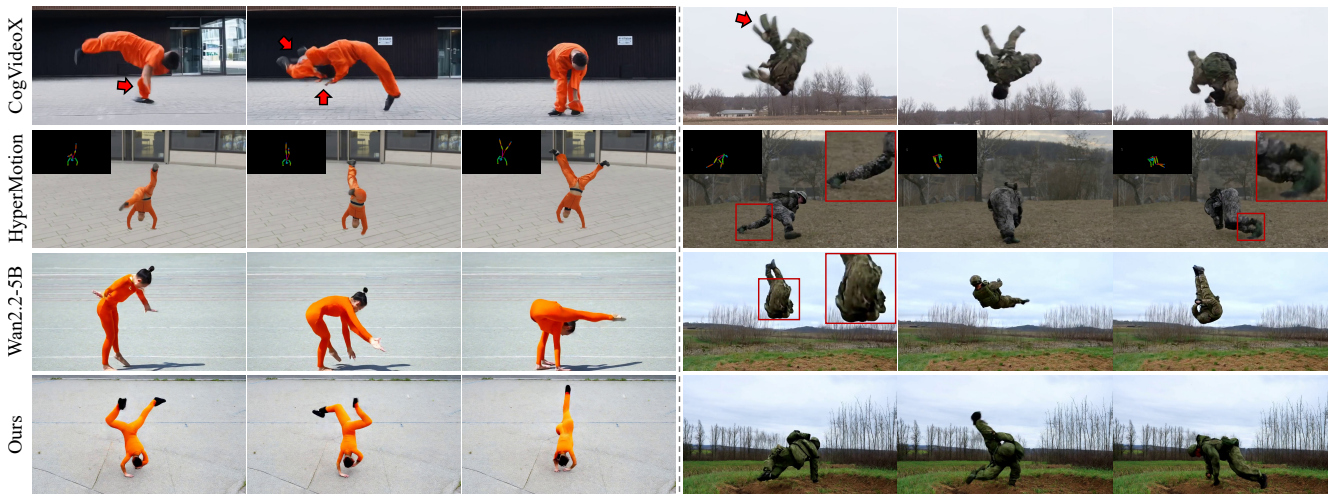


Figure 4. Qualitative comparison of dynamic object motion generation. CogVideoX [56] and Wan2.2-5B [48] often produce distorted or unrealistic human motions with visual artifacts. HyperMotion [52] produces unnatural appearances because it relies on the first frame as input. In contrast, our method generates realistic videos with natural and highly dynamic motions.

	Pexels					DynaVid-Human test				
	FVD (↓)	A-Qual (↑)	I-Qual (↑)	M-Smooth (↑)	T-Flick (↑)	FVD (↓)	A-Qual (↑)	I-Qual (↑)	M-Smooth (↑)	T-Flick (↑)
CogVideoX-5B	1519.54	0.5646	0.6613	0.9844	0.9673	2238.68	0.5071	0.5562	0.9779	0.9565
Wan2.2-5B	1172.02	0.5779	0.7235	**0.9928**	**0.9883**	1775.99	**0.5389**	0.6974	0.9904	0.9791
HyperMotion	420.82*	0.5578	0.6972	0.9922	0.9850	391.22*	0.5092	0.6265	**0.9939**	**0.9914**
Ours	**1126.38**	**0.5807**	**0.7342**	0.9900	0.9748	**1351.94**	0.5312	**0.7352**	0.9931	0.9864

Table 1. Quantitative evaluation of object motion generation on the Pexels [2] and DynaVid-Human test datasets, representing common and highly dynamic scenes, respectively. Our method achieves comparable or superior results on both datasets compared to other baseline methods. *: HyperMotion [52] uses a first frame as input, resulting in lower FVD scores due to the appearance similarity.

using the mean squared error between the input flow maps and those estimated from the generated video frames.

4.1. Comparison on Dynamic Object Motion

We compare our model with CogVideoX-5B [56], Wan2.2-5B [48], and HyperMotion [52]. CogVideoX and Wan2.2 are recent text-to-video generation models. HyperMotion is a Wan2.1-14B-based human-pose-controlled video generation method that takes a text prompt, a first frame, and a 2D human pose sequence as inputs, where the poses are extracted using an off-the-shelf pose estimator [54].

Figs. 1 and 4 present qualitative comparisons on highly dynamic object motion generation. While CogVideoX [56] and Wan2.2 [48] often fail to produce natural or dynamic movements due to the limited training data, our method successfully captures motion dynamics in highly dynamic video synthesis scenarios. HyperMotion [52] tends to generate frames with an artificial appearance, partly because it relies on the first frame as a visual prior.

Tab. 1 reports quantitative comparisons on datasets featuring common scenes (Pexels) and highly dynamic motions (DynaVid-Human). Our method achieves comparable or superior visual and motion quality across both datasets compared to the baselines. Although HyperMotion [52], which leverages a larger backbone and additional input human pose sequences, performs comparably in dynamic scenarios, it exhibits degraded visual quality on common scenes, as it is primarily trained on human-centric video datasets. Note that HyperMotion reports lower FVD scores as it directly uses the first frame as input, which increases appearance similarity with the evaluation datasets and consequently leads to lower FVD scores.

4.2. Comparison on Extreme Camera Control

We compare our model with AC3D [5] and GEN3C [40]. Both AC3D and our method are text-to-video diffusion models that use Plücker embedding as control signals for camera-controlled video synthesis. GEN3C, in contrast, is an image-to-video diffusion model that estimates depth from the input image to reconstruct 3D caches (e.g., point clouds) for 3D-aware camera control.

Figs. 1 and 5 present qualitative comparisons with the baseline methods. AC3D fails to follow extreme camera trajectories involving rapid viewpoint changes (e.g., 180° rotations), while GEN3C produces artifacts in regions unseen from the input image. In contrast, our method generates realistic frames that accurately follow challenging camera paths, benefiting from training with synthetic motion data.

Tab. 2 reports quantitative comparisons with the baseline models. While our model shows comparable camera con-

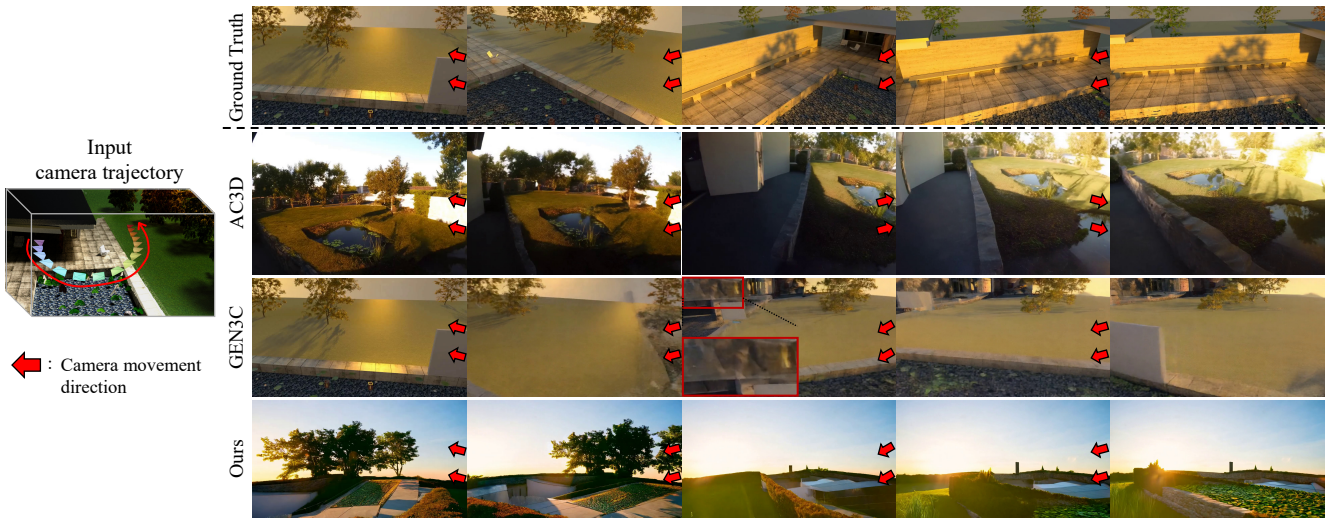


Figure 5. Qualitative comparison of camera-controlled video synthesis. Red arrows indicate the directions of camera motion. AC3D [5] fails to follow the extreme 180° rotation, while GEN3C [40] produces noticeable artifacts in regions unseen from the initial view (zoomed-in red box). In contrast, our method produces natural-looking videos that faithfully follow the input camera trajectory. For fair comparison, the same camera parameters are used for all methods.

	RE10K test						DynaVid-Camera test					
	mRotErr (↓)	FVD (↓)	A-Qual (↑)	I-Qual (↑)	M-Smooth (↑)	T-Flick (↑)	mRotErr (↓)	FVD (↓)	A-Qual (↑)	I-Qual (↑)	M-Smooth (↑)	T-Flick (↑)
AC3D	0.1347	685.05	0.5184	0.6636	**0.9918**	0.9651	1.1529	782.01	0.4483	0.5407	0.9727	0.9403
GEN3C	**0.0809**	566.99*	0.4579	0.6079	0.9899	**0.9731**	1.1852	237.15*	0.3889	0.5659	**0.9844**	**0.9611**
Ours	0.1136	**664.21**	**0.5425**	**0.7224**	0.9888	0.9699	**0.9289**	**674.72**	**0.4501**	**0.6713**	0.9760	0.9487

Table 2. Quantitative evaluation of camera-controlled video synthesis on RE10K [61] and DynaVid-Camera, representing common and dynamic scenes, respectively. While achieving comparable rotation errors on common scenes, our method significantly outperforms other baseline methods on scenes with rapidly changing viewpoints. *: GEN3C [40] uses a first frame as input, resulting in lower FVD scores.

trollability on RE10K [61], which includes moderate viewpoint changes, it significantly outperforms both baselines on DynaVid-Camera, which contains rapidly changing camera trajectories. Moreover, our method demonstrates higher visual quality across the datasets. Additional qualitative examples are provided in the supplementary document.

4.3. Analysis

Importance of synthetic motion data. We first investigate the importance of synthetic motion data in training, which provides highly dynamic motion patterns to the model (Sec. 3.4). As shown in Tab. 3, removing synthetic motion data leads to a significant degradation in both visual and motion quality on DynaVid-Human, resulting in much higher FVD scores for highly dynamic video generation. In contrast, only a minor change in FVD is observed on Pexels.

Importance of batch mixture. During fine-tuning of the motion generator, we employ mixed training batches containing both real and synthetic flows (Sec. 3.4). To analyze the effect of this strategy, we train a variant that uses only synthetic motion data during fine-tuning, excluding real data from the batch mixture. As reported in Tab. 3, this variant exhibits substantial performance degradation with FVD scores on the Pexels dataset. This indicates that real flow data is necessary to preserve general motion priors, while training with synthetic flow data alone leads to overfitting to synthetic motion patterns.

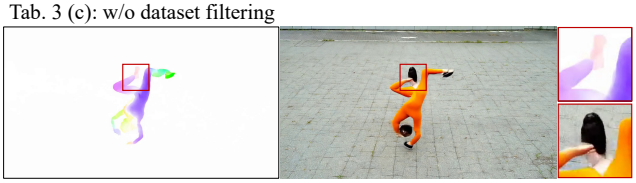
Importance of dataset filtering. We apply a simple dataset filtering technique based on flow cycle consistency to reduce optical flow estimation errors in the real dataset (Sec. 3.4). To evaluate its effect, we train a motion-guided video generator without filtering. Although this variant shows similar quantitative results (Tab. 3), it produces unnatural videos in highly dynamic scenarios. As shown in Fig. 6, this model generates misaligned body parts (e.g., incorrect head positions highlighted in red), revealing inconsistency between video frames and input motions. This is also validated with the motion error (M-Err), where the variant reports 4.734 compared to 4.287 for our final model.

Effect of using rendered videos. We further examine the impact of using rendered synthetic videos instead of synthetic motion data, which represents the most straightforward way to leverage synthetic datasets. To this end, we train an additional model that uses rendered videos together with batch mixture training on real videos. As shown in Tab. 3, this model exhibits degraded performance on Pexels, reporting higher FVD scores. Moreover, despite the use of batch mixture, it produces videos with an artificial ap-

	Pexels					DynaVid-Human test				
	FVD (↓)	A-Qual (↑)	I-Qual (↑)	M-Smooth (↑)	T-Flick (↑)	FVD (↓)	A-Qual (↑)	I-Qual (↑)	M-Smooth (↑)	T-Flick (↑)
Ours	1126.38	0.5807	0.7342	0.9900	0.9748	1351.94	0.5312	0.7352	0.9931	0.9864
w/o synthetic motion data (a)	(-49.85)	(-0.0199)	(-0.0156)	(-0.0002)	(-0.0019)	(+527.04)	(+0.0106)	(-0.0403)	(-0.0062)	(-0.0249)
w/o batch mixture (b)	(+759.36)	(-0.0073)	(-0.0070)	(+0.0030)	(+0.0144)	(-122.24)	(-0.0050)	(-0.0006)	(-0.0022)	(-0.0013)
w/o dataset filtering† (c)	(-33.79)	(+0.0011)	(+0.0035)	(+0.0007)	(+0.0009)	(+32.44)	(+0.0055)	(-0.0140)	(+0.0008)	(+0.0012)
w/ synthetic video data (d)	(+104.43)	(+0.0163)	(+0.0124)	(-0.0012)	(+0.0005)	(-653.98)*	(+0.0112)	(-0.0487)	(-0.0000)	(+0.0049)

Table 3. Ablation study of our main components, evaluating video and motion synthesis quality on the Pexels and DynaVid-Human test datasets. †: Our model without data filtering achieves comparable quantitative results but produces noticeably degraded visual quality in highly dynamic motion synthesis, as shown in Fig. 6. *: Our model trained with rendered synthetic videos reports lower FVD scores, because it produces artificial appearances resembling synthetic data, as shown in Fig. 6.

Tab. 3 (c): w/o dataset filtering



"An orange-suited figure performs a series of breakdancing movements …"

Tab. 3 (d): w/ synthetic video data

"A person dressed in a shiny blue outfit performs acrobatic movements …"

Figure 6. Visual examples of the ablation study. Our model without dataset filtering produces video frames that are inconsistent with the input motion (top). Our model trained using the synthetic video dataset reproduces the artificial look of rendered videos (bottom).

pearance as shown in Fig. 6, which leads to lower FVD scores on DynaVid-Human due to the synthetic visual characteristics of that dataset. These results demonstrate the limitations of directly using rendered videos for training.

Robustness to motion generation error. Because the motion-guided video generator directly takes the motion synthesized by the motion generator as input, it can be affected by motion generation errors. Thus, we validate whether the motion-guided video generator can synthesize high-quality video frames while faithfully following motion when the input flow is noisy. We prepare clean optical flow and noisy variants by adding Gaussian noise to achieve target signal-to-noise ratios of 25, 20, 15, 10, and 5 dB. Then, the motion-guided video generator synthesizes videos conditioned on each flow variant. Tab. 4 reports the motion error and aesthetic quality. Our framework remains robust at 20dB and above, with gradual degradation as the noise level increases. Further experimental details are provided in the supplementary document.

Behavior beyond the training domain. Although the DynaVid-Human dataset contains only single-person scenes, our method can synthesize motions that extend beyond this training domain. As shown in Fig. 7, the model generates a dynamically moving panda whose body structure differs from that of humans, despite being trained solely on human-centric data. This indicates that our approach can produce plausible motions for certain unseen object types, even though it is not explicitly trained for such cases.

	Clean	25dB	20dB	15dB	10dB	5dB
M-Err (↓)	4.287	4.371	4.374	4.323	4.391	4.434
A-Qual (↑)	0.5312	0.5233	0.5239	0.5148	0.5071	0.4632

Table 4. Error robustness of the motion-guided video generator.

Figure 7. Capability to synthesize non-human dynamic motions.

Figure 8. Limitations. Our method is less effective in generating videos with multiple people performing dynamic motions.

Limitations. Our method is not free from limitations. Since our synthetic dataset primarily consists of single-person scenes, the model often struggles to generate videos involving multiple people with highly dynamic motions (Fig. 8). This issue could be mitigated by increasing the diversity and scale of the synthetic dataset, which we leave for future work. Additional discussions of these limitations are provided in the supplementary document.

5. Conclusion

We presented DynaVid, a video synthesis framework for generating realistic videos with highly dynamic motions and controllable camera movements. Our key idea is to leverage *synthetic motion data* to provide dynamic and precise motion supervision while avoiding appearance-domain gaps. Through a two-stage design that separately models motion generation and motion-guided video synthesis, our method effectively leverages synthetic and real-world data to achieve both visual realism and dynamic controllability. Extensive experiments demonstrate that DynaVid outperforms existing video diffusion models in dynamic motion generation and camera-controlled synthesis, while maintaining robustness to motion generation errors and moderate out-of-domain scenarios.

Acknowledgment. This work was supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) under the following programs: RS-2024-00395401 (Development of VFX creation and combination using generative AI), RS-2019-II191906 (Artificial Intelligence Graduate School Program at POSTECH), RS-2024-00457882 (AI Research Hub Project), and RS-2024-0045788. This work was also supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (RS-2023-00211658 and RS-2024-00438532). This work was also supported by Microsoft Research Asia.

References

[1] Blendswap. https://www.blendswap.com/. 4

[2] Pexels, royalty-free stock footage website. https://www.pexels.com. Accessed: 2025-10-21. 2, 5, 6

[3] *Mixamo*. Adobe. 3

[4] Sherwin Bahmani, Ivan Skorokhodov, Aliaksandr Siarohin, Willi Menapace, Guocheng Qian, Michael Vasilkovsky, Hsin-Ying Lee, Chaoyang Wang, Jiaxu Zou, Andrea Tagliasacchi, et al. Vd3d: Taming large video diffusion transformers for 3d camera control. *arXiv preprint arXiv:2407.12781*, 2024. 3

[5] Sherwin Bahmani, Ivan Skorokhodov, Guocheng Qian, Aliaksandr Siarohin, Willi Menapace, Andrea Tagliasacchi, David B Lindell, and Sergey Tulyakov. Ac3d: Analyzing and improving 3d camera control in video diffusion transformers. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 22875–22889, 2025. 2, 3, 6, 7

[6] Jianhong Bai, Menghan Xia, Xiao Fu, Xintao Wang, Lianrui Mu, Jinwen Cao, Zuozhu Liu, Haoji Hu, Xiang Bai, Pengfei Wan, et al. Recammaster: Camera-controlled generative rendering from a single video. *arXiv preprint arXiv:2503.11647*, 2025. 3

[7] Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. Frozen in time: A joint video and image encoder for end-to-end retrieval. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1728–1738, 2021. 2

[8] Philip J. Ball, Jakob Bauer, Frank Belletti, Bethanie Brownfield, Ariel Ephrat, Shlomi Fruchter, Agrim Gupta, Kristian Holsheimer, Aleksander Holynski, Jiri Hron, Christos Kaplanis, Marjorie Limont, Matt McGill, Yanko Oliveira, Jack Parker-Holder, Frank Perbet, Guy Scully, Jeremy Shar, Stephen Spencer, Omer Tov, Ruben Villegas, Emma Wang, Jessica Yung, Cip Baetu, Jordi Berbel, David Bridson, Jake Bruce, Gavin Buttimore, Sarah Chakera, Bilva Chandra, Paul Collins, Alex Cullum, Bogdan Damoc, Vibha Dasagi, Maxime Gazeau, Charles Gbadamosi, Woohyun Han, Ed Hirst, Ashyana Kachra, Lucie Kerley, Kristian Kjems, Eva Knoepfel, Vika Koriakin, Jessica Lo, Cong Lu, Zeb Mehring, Alex Moufarek, Henna Nandwani, Valeria Oliveira, Fabio Pardo, Jane Park, Andrew Pierson, Ben Poole, Helen Ran, Tim Salimans, Manuel Sanchez, Igor Saprykin, Amy Shen, Sailesh Sidhwani, Duncan Smith, Joe Stanton, Hamish Tomlinson, Dimple Vijaykumar, Luyu Wang, Piers Wingfield, Nat Wong, Keyang Xu, Christopher Yew, Nick Young, Vadim Zubov, Douglas Eck, Dumitru Erhan, Koray Kavukcuoglu, Demis Hassabis, Zoubin Gharamani, Raia Hadsell, Aäron van den Oord, Inbar Mosseri, Adrian Bolton, Satinder Singh, and Tim Rocktäschel. Genie 3: A new frontier for world models. 2025. 3

[9] Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023. 2

[10] Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, Clarence Ng, Ricky Wang, and Aditya Ramesh. Video generation models as world simulators. 2024. 2, 3

[11] Ryan Burgert, Yuancheng Xu, Wenqi Xian, Oliver Pilarski, Pascal Clausen, Mingming He, Li Ma, Yitong Deng, Lingxiao Li, Mohsen Mousavi, et al. Go-with-the-flow: Motion-controllable video diffusion models using real-time warped noise. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 13–23, 2025. 3

[12] Daniel J Butler, Jonas Wulff, Garrett B Stanley, and Michael J Black. A naturalistic open source movie for optical flow evaluation. In *European conference on computer vision*, pages 611–625. Springer, 2012. 3

[13] Hila Chefer, Uriel Singer, Amit Zohar, Yuval Kirstain, Adam Polyak, Yaniv Taigman, Lior Wolf, and Shelly Sheynin. Videojam: Joint appearance-motion representations for enhanced motion generation in video models. *arXiv preprint arXiv:2502.02492*, 2025. 4

[14] Tsai-Shien Chen, Chieh Hubert Lin, Hung-Yu Tseng, Tsung-Yi Lin, and Ming-Hsuan Yang. Motion-conditioned diffusion model for controllable video synthesis. *arXiv preprint arXiv:2304.14404*, 2023. 3

[15] Tsai-Shien Chen, Aliaksandr Siarohin, Willi Menapace, Ekaterina Deyneka, Hsiang-wei Chao, Byung Eun Jeon, Yuwei Fang, Hsin-Ying Lee, Jian Ren, Ming-Hsuan Yang, et al. Panda-70m: Captioning 70m videos with multiple cross-modality teachers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13320–13331, 2024. 2

[16] Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018. 3

[17] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2758–2766, 2015. 3

[18] Epic Games. Unreal engine. 3

[19] Daniel Geng, Charles Herrmann, Junhwa Hur, Forrester Cole, Serena Zhang, Tobias Pfaff, Tatiana Lopez-Guevara,

Yusuf Aytar, Michael Rubinstein, Chen Sun, et al. Motion prompting: Controlling video generation with motion trajectories. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 1–12, 2025. 3

[20] Klaus Greff, Francois Belletti, Lucas Beyer, Carl Doersch, Yilun Du, Daniel Duckworth, David J Fleet, Dan Gnanapragasam, Florian Golemo, Charles Herrmann, et al. Kubric: A scalable dataset generator. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3749–3761, 2022. 3

[21] Zekai Gu, Rui Yan, Jiahao Lu, Peng Li, Zhiyang Dou, Chenyang Si, Zhen Dong, Qifeng Liu, Cheng Lin, Ziwei Liu, et al. Diffusion as shader: 3d-aware video diffusion for versatile video generation control. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers*, pages 1–12, 2025. 3

[22] Hao He, Yinghao Xu, Yuwei Guo, Gordon Wetzstein, Bo Dai, Hongsheng Li, and Ceyuan Yang. Cameractrl: Enabling camera control for video diffusion models. In *The Thirteenth International Conference on Learning Representations*, 2025. 2, 3, 5

[23] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 2

[24] Li Hu. Animate anyone: Consistent and controllable image-to-video synthesis for character animation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8153–8163, 2024. 3

[25] Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, et al. Vbench: Comprehensive benchmark suite for video generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21807–21818, 2024. 5

[26] Yash Jain, Anshul Nasery, Vibhav Vineet, and Harkirat Behl. Peekaboo: Interactive video generation via masked-diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8079–8088, 2024. 3

[27] Zeyinzi Jiang, Zhen Han, Chaojie Mao, Jingfeng Zhang, Yulin Pan, and Yu Liu. Vace: All-in-one video creation and editing. *arXiv preprint arXiv:2503.07598*, 2025. 4, 5

[28] Wonjoon Jin, Qi Dai, Chong Luo, Seung-Hwan Baek, and Sunghyun Cho. Flovd: Optical flow meets video diffusion model for enhanced camera-controlled video synthesis. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 2040–2049, 2025. 3

[29] KlingAI. Kling ai. 2025. 2

[30] Jingyun Liang, Yuchen Fan, Kai Zhang, Radu Timofte, Luc Van Gool, and Rakesh Ranjan. Movideo: Motion-aware video generation with diffusion model. In *European Conference on Computer Vision*, pages 56–74. Springer, 2024. 3

[31] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022. 2, 5

[32] Qihao Liu, Ju He, Qihang Yu, Liang-Chieh Chen, and Alan Yuille. Revision: High-quality, low-cost video generation with explicit 3d physics modeling for complex motion and interaction. *arXiv preprint arXiv:2504.21855*, 2025. 3

[33] Nikolaus Mayer, Eddy Ilg, Philip Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4040–4048, 2016. 3

[34] Koichi Namekata, Sherwin Bahmani, Ziyi Wu, Yash Kant, Igor Gilitschenski, and David B Lindell. Sg-i2v: Self-guided trajectory control in image-to-video generation. *arXiv preprint arXiv:2411.04989*, 2024. 3

[35] Haomiao Ni, Changhao Shi, Kai Li, Sharon X Huang, and Martin Renqiang Min. Conditional image-to-video generation with latent flow diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 18444–18455, 2023. 3

[36] Karran Pandey, Yannick Hold-Geoffroy, Matheus Gadelha, Niloy J Mitra, Karan Singh, and Paul Guerrero. Motion modes: What could happen next? In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 2030–2039, 2025. 3

[37] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4195–4205, 2023. 2

[38] Adam Polyak, Amit Zohar, Andrew Brown, Andros Tjandra, Animesh Sinha, Ann Lee, Apoorv Vyas, Bowen Shi, Chih-Yao Ma, Ching-Yao Chuang, et al. Movie gen: A cast of media foundation models. *arXiv preprint arXiv:2410.13720*, 2024. 2

[39] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE transactions on pattern analysis and machine intelligence*, 44(3):1623–1637, 2020. 3

[40] Xuanchi Ren, Tianchang Shen, Jiahui Huang, Huan Ling, Yifan Lu, Merlin Nimier-David, Thomas Müller, Alexander Keller, Sanja Fidler, and Jun Gao. Gen3c: 3d-informed world-consistent video generation with precise camera control. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 6121–6132, 2025. 1, 3, 6, 7

[41] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 4

[42] Prafull Sharma, Varun Jampani, Yuanzhen Li, Xuhui Jia, Dmitry Lagun, Fredo Durand, Bill Freeman, and Mark Matthews. Alchemist: Parametric control of material properties with diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24130–24141, 2024. 3

[43] Xiaoyu Shi, Zhaoyang Huang, Fu-Yun Wang, Weikang Bian, Dasong Li, Yi Zhang, Manyuan Zhang, Ka Chun Cheung, Simon See, Hongwei Qin, et al. Motion-i2v: Consistent and

controllable image-to-video generation with explicit motion modeling. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–11, 2024. 3

[44] Xincheng Shuai, Henghui Ding, Zhenyuan Qin, Hao Luo, Xingjun Ma, and Dacheng Tao. Free-form motion control: A synthetic video generation dataset with controllable camera and object motions. *arXiv preprint arXiv:2501.01425*, 2025. 2, 3

[45] The Sora team. Sora2. 2025. 2, 3

[46] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *European conference on computer vision*, pages 402–419. Springer, 2020. 3

[47] Thomas Unterthiner, Sjoerd Van Steenkiste, Karol Kurach, Raphael Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric & challenges. *arXiv preprint arXiv:1812.01717*, 2018. 5

[48] Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025. 1, 2, 3, 4, 5, 6

[49] Yihan Wang and Jia Deng. Waft: Warping-alone field transforms for optical flow. *arXiv preprint arXiv:2506.21526*, 2025. 3, 5

[50] Zhouxia Wang, Ziyang Yuan, Xintao Wang, Yaowei Li, Tianshui Chen, Menghan Xia, Ping Luo, and Ying Shan. Motionctrl: A unified and flexible motion controller for video generation. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–11, 2024. 2, 3

[51] Weijia Wu, Zhuang Li, Yuchao Gu, Rui Zhao, Yefei He, David Junhao Zhang, Mike Zheng Shou, Yan Li, Tingting Gao, and Di Zhang. Draganything: Motion control for anything using entity representation. In *European Conference on Computer Vision*, pages 331–348. Springer, 2024.

[52] Shuolin Xu, Siming Zheng, Ziyi Wang, HC Yu, Jinwei Chen, Huaqi Zhang, Bo Li, and Peng-Tao Jiang. Hypermotion: Dit-based pose-guided human image animation of complex motions. *arXiv preprint arXiv:2505.22977*, 2025. 6

[53] Zhongcong Xu, Jianfeng Zhang, Jun Hao Liew, Hanshu Yan, Jia-Wei Liu, Chenxu Zhang, Jiashi Feng, and Mike Zheng Shou. Magicanimate: Temporally consistent human image animation using diffusion model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1481–1490, 2024. 3

[54] Jie Yang, Ailing Zeng, Ruimao Zhang, and Lei Zhang. X-pose: Detecting any keypoints. In *European Conference on Computer Vision*, pages 249–268. Springer, 2024. 6

[55] Shiyuan Yang, Liang Hou, Haibin Huang, Chongyang Ma, Pengfei Wan, Di Zhang, Xiaodong Chen, and Jing Liao. Direct-a-video: Customized video generation with user-directed camera movement and object motion. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–12, 2024. 3

[56] Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. Cogvideox: Text-to-video diffusion models with an expert transformer. *arXiv preprint arXiv:2408.06072*, 2024. 2, 6

[57] Shengming Yin, Chenfei Wu, Jian Liang, Jie Shi, Houqiang Li, Gong Ming, and Nan Duan. Dragnuwa: Fine-grained control in video generation by integrating text, image, and trajectory. *arXiv preprint arXiv:2308.08089*, 2023. 3

[58] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3836–3847, 2023.

[59] Yuang Zhang, Jiaxi Gu, Li-Wen Wang, Han Wang, Junqi Cheng, Yuefeng Zhu, and Fangyuan Zou. Mimicmotion: High-quality human motion video generation with confidence-aware pose guidance. *arXiv preprint arXiv:2406.19680*, 2024. 3

[60] Qi Zhao, Xingyu Ni, Ziyu Wang, Feng Cheng, Ziyan Yang, Lu Jiang, and Bohan Wang. Synthetic video enhances physical fidelity in video synthesis. *arXiv preprint arXiv:2503.20822*, 2025. 2, 3

[61] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. *arXiv preprint arXiv:1805.09817*, 2018. 5, 7