Learning to Generate Highly Dynamic Videos using Synthetic Motion Data

Supplementary Material

In this supplemental document, we provide:

- Additional details of datasets,
- Additional implementation details of DynaVid,
- Additional experimental details of DynaVid,
- Additional analysis and discussion,
- Experimental details of baseline methods, and
- Additional qualitative results.

A. Additional Details of Datasets

This section provides additional statistics and characteristics of the real-world datasets (i.e., the internal dataset and RealEstate10K [10]) as well as the synthetic datasets (DynaVid-Human and DynaVid-Camera). As discussed in Sec. 1 of the main paper, commonly used training datasets contain scarce examples of highly dynamic motions, limiting the ability of video diffusion models to learn such behaviors. The DynaVid datasets are designed to fill this gap by providing highly dynamic human actions and fast moving camera motions.

A.1. Real-World Datasets

Internal dataset. We use an internal large-scale video dataset to train the motion-guided video generator and to provide real motion supervision for the motion generator in the object-motion scenario. As shown in Fig. S1(a), the dataset covers diverse real-world scenes such as human activities, natural environments, and man-made structures, and contains about 86K video clips.

Fig. S2(a) visualizes the histogram of average optical flow magnitudes extracted from the internal dataset. Most videos exhibit very small motion, with over 90% of samples having a average flow magnitude below 4.91 pixels. This indicates that the dataset predominantly contains videos with relatively subtle motion. In Sec. D.2, we curate the internal dataset to mainly contain dynamic motions and train an additional model on the curated dataset, confirming that the internal dataset contains almost no examples of vigorous human motions such as breakdancing.

RealEstate10K dataset. For camera-controlled video synthesis, we employ the RealEstate10K (RE10K) dataset [10]. As shown in Fig. S1(b), RE10K primarily consists of indoor scenes with moderate camera movement, and we use 33K video clips for training.

Fig. S2(b) shows the histogram of average flow magnitudes. Compared with the internal dataset, RE10K exhibits larger motions, but the majority of clips still show limited dynamics, with over 90% of samples below 8.54 pixels. This indicates that RE10K mainly includes videos with modest camera motion and lacks examples involving rapid rotations or large translations.

Figure S1. Visual examples from each dataset.

A.2. Synthetic Datasets

DynaVid-Human. We train our model on DynaVid-Human for highly dynamic object motion generation. As shown in Fig. S1(c), although the rendered videos have artificial appearance, they contain highly dynamic human movements that are rarely found in real datasets. We use a total of 500 motion-rich scenes rendered using our dataset generation pipeline.

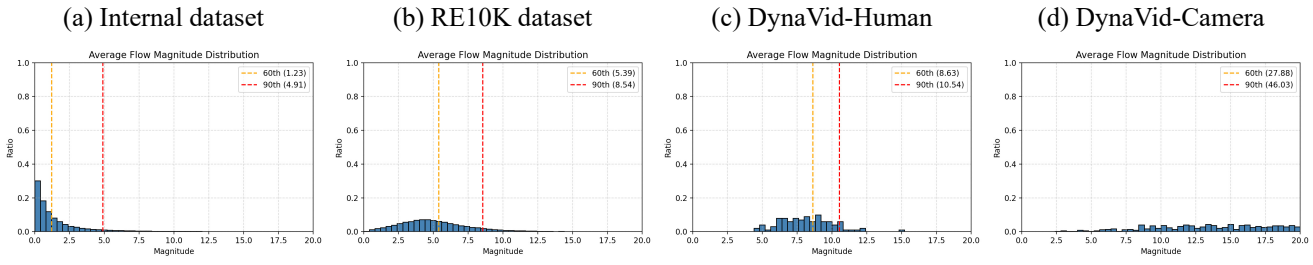
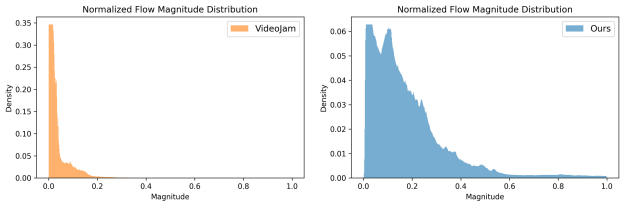


Figure S2. Histograms of optical flow magnitudes across datasets. The magnitude axis is limited to the range [0,20], and most flow magnitudes in DynaVid-Camera lie beyond this range.



(a) VideoJAM (b) Ours

Figure S3. Normalized flow magnitude histogram. Our normalization method spreads flow magnitudes more evenly over the [0,1] range compared with the normalization used in VideoJAM [3].

Fig. S2(c) presents the histogram of average flow magnitudes. Unlike the internal dataset, DynaVid-Human includes substantially stronger motion, with magnitudes concentrated approximately between 5–10 pixels, and 90% of samples below 10.54 pixels. This demonstrates that DynaVid-Human effectively complements the lack of highly dynamic human motion in real-world datasets.

DynaVid-Camera. We train our model on DynaVid-Camera for highly dynamic camera motion control scenario. As shown in Fig. S1(d), despite its artificial appearance, the dataset includes large rotations, fast translations, and complex trajectories that are difficult to capture in real footage. We use 1,000 synthetically generated scenes for training.

Fig. S2(d) shows the distribution of average flow magnitudes. Compared to RE10K, DynaVid-Camera contains vastly stronger camera movements, with 90% of samples below 46.03 pixels. This demonstrates that DynaVid-Camera provides large-scale rotations and translations, offering dynamic and precise control signals for camera-motion conditioning.

B. Implementation Details of DynaVid

B.1. Flow-to-RGB Conversion

We provide additional details of the Flow-to-RGB conversion described in Sec. 3.2 of the main paper. During the normalization step in Eq. 1 of the main paper, we use a dataset-level scale factor of 29.5, which is a stable statistic estimated from sufficient samples (Sec. D.1). The square-root-based normalization spreads the flow magnitude distribution more evenly across the [0,1] range. As shown in Fig. S2(a), the original flow magnitudes are heavily concentrated near zero. After applying our normalization, the magnitudes become more broadly distributed, as illustrated in Fig. S3(b).

We compare our normalization method with the linear normalization approach used in VideoJAM [3]. Because a linear transform does not sufficiently expand the low-magnitude range, VideoJAM still produces values concentrated near zero, as shown on Fig. S3(a). To quantify the impact of the normalization methods, we evaluate flow reconstruction error using the VAE from Wan2.2-5B [7]. Specifically, we convert flow maps into RGB, encode and decode them through the VAE, convert the decoded RGB back to flow, and measure the mean squared error between the original and reconstructed flow maps. Our method achieves a reconstruction error of 0.0486, whereas VideoJAM yields 0.1338, demonstrating the effectiveness of our normalization scheme.

B.2. Control Branch

As described in Sec. 3.3 of the main paper, we employ a control branch when the model takes an additional control signal beyond the noisy input and text prompt. Specifically, the control branch is used in two cases: (1) in the motion generator for the camera-control scenario, where the control signal is a Plücker embedding, and (2) in the motion-guided video generator, where the control signal is the optical flow.

We adopt the VACE [4] design for injecting control signals into a pre-trained video diffusion model (Fig. 3(b) of the main paper). The control branch is composed of transformer blocks that follow the same architecture as those in the base video diffusion model. However, the number of blocks is reduced: the control branch uses 10 transformer blocks, whereas the base video diffusion model contains 30 blocks.

For feature injection, the context features produced by each block of the control branch are added sequentially to

	Pexels					DynaVid-Human test				
	FVD (↓)	A-Qual (↑)	I-Qual (↑)	M-Smooth (↑)	T-Flick (↑)	FVD (↓)	A-Qual (↑)	I-Qual (↑)	M-Smooth (↑)	T-Flick (↑)
Wan2.2-5B variant	**1071.26**	0.5801	0.7121	0.9917	0.9848	1883.50	**0.5598**	0.7092	0.9906	0.9768
Wan2.2-5B	1172.02	0.5779	0.7235	**0.9928**	**0.9883**	1775.99	0.5389	0.6974	0.9904	0.9791
Ours	1126.38	**0.5807**	**0.7342**	0.9900	0.9748	**1351.94**	0.5312	**0.7352**	**0.9931**	**0.9864**

Table S1. Quantitative evaluation of object motion generation using the variant model trained on the curated real dataset.

the latent features of every third transformer block in the base model, specifically blocks [0,3,6,...,27]. This design efficiently incorporates the control signal while maintaining compatibility with the base diffusion architecture. During training, we update only the parameters of the control branch while keeping the parameters of the base video diffusion model fixed.

C. Experimental Details of DynaVid

C.1. Training Details

DynaVid consists of a motion generator and a motion-guided video generator, which are trained separately. We adopt the same training strategy for both models. All experiments are conducted on 4 A100-80GB GPUs, using gradient accumulation of 4, resulting in the total batch size of 16. For the motion generator, we pre-train and fine-tune the model sequentially, each for 5K iterations. We use the AdamW optimizer and a learning rate of 1e-5, with a warm-up schedule over the first 500 iterations. For the motion-guided video generator, we also train for 5K iterations using AdamW with the same learning rate 1e-5 and the same 500-iteration warm-up schedule.

C.2. Details on Robustness Experiment

In Sec. 4.3 of the main paper, we analyze the robustness of the motion-guided video generator to perturbations in the input flow, simulating motion generation errors from the motion generator. To this end, we create noisy flow variants by adding Gaussian noise to the optical flow maps and adjusting the signal-to-noise ratio (SNR) to 25, 20, 15, 10, and 5 dB.

Specifically, we first estimate the variance of optical flow from the real-world flow dataset, which is 2.59, and use this value to determine the corresponding noise variance for each target SNR. The resulting noisy flow maps are then fed into the motion-guided video generator to evaluate how performance degrades as the reliability of the motion signal decreases.

D. Additional Analysis and Discussion

D.1. Stability of the Scale Factor

In Sec. 3.2 of the main paper, we introduce a dataset-level scale factor s_f for the Flow-to-RGB conversion. This scale factor is estimated from a large number of samples and exhibits low variance across different dataset splits. To evaluate its stability, we randomly sample 100 and 1,000 videos from the internal dataset and compute the 99th percentile of flow magnitude, obtaining 28.5 and 27.5 pixels, respectively. These values are close to the original estimate of 29.5 pixels computed from 10,000 samples. This result indicates that the scale factor remains stable across different splits, and minor variations in s_f do not materially affect training.

Figure S4. Qualitative results of the variant model trained on the curated real dataset.

D.2. Training with Curated Real Dataset

As explained in Sec. A.1, real-world video datasets generally contain modest motions. However, because these datasets are large-scale, they may still include a small number of videos with highly dynamic motion, such as breakdancing.

To examine this possibility, we construct a curated subset by selecting videos whose optical-flow magnitudes fall within the top 60–90% range, ensuring that the subset contains comparatively faster motions. We then fine-tune a variant model, based on the Wan2.2-5B text-to-video diffusion backbone, using only this curated real dataset so that the model becomes aligned with the most dynamic motion patterns of the real data.

Fig. S4 shows qualitative results produced by this variant model. As expected, the model still fails to synthesize natural-looking videos with highly dynamic motions. Furthermore, as reported in Tab. S1, its performance on the DynaVid-Human dataset is significantly degraded, especially in terms of FVD [6]. These results highlight that

Figure S5. Additional visual examples of the limitations.

the commonly used training datasets contain almost no examples of highly dynamic motions, making it difficult for models trained solely on real data to learn such motion patterns.

D.3. Additional Limitations

While our framework is robust to local motion errors introduced by the motion generator (Sec. 4.3 of the main paper), it may still produce artifacts when global motion structures are incorrect. Because the motion generator focuses exclusively on object motion, it does not explicitly model the interactions between a moving subject and surrounding objects.

For example, in boxing-ring scenes, the model struggles to capture physical interactions between a person and the surrounding ropes, leading to motion that violates physical constraints. These erroneous motions are subsequently propagated to the motion-guided video generator, resulting in unnatural video synthesis (Fig. S5).

Such limitations could be mitigated by jointly modeling motion and appearance, allowing cross-modal feedback between the two representations. We leave this direction as future work.

E. Experimental Details of Baseline Methods

For highly dynamic object motion generation, we compare our method against Wan2.2-5B [7] and CogVideoX-5B [9], which are text-to-video generation models, and HyperMotion [8], which is a pose-to-video generation model [8]. For camera-controlled video synthesis with large viewpoint changes, we compare our method against AC3D [2] and GEN3C [5], both designed for camera motion control. For all baseline methods, we use the official checkpoints and inference code provided in their respective GitHub repositories.

Wan2.2-5B. We use the Diffusers-based implementation (https://github.com/huggingface/diffusers). We also utilize Diffusers-based checkpoints, which are converted from the official Wan2.2-5B checkpoint (https://github.com/Wan-Video/Wan2.2).

CogVideoX-5B. We use the Diffusers-based implementation (https://github.com/huggingface/diffusers). We also utilize Diffusers-based checkpoints, which are converted from the official CogVideoX-5B checkpoint (https://github.com/zai-org/CogVideo).

HyperMotion. We use the PyTorch implementation and checkpoints from the official Github repository of HyperMotion (https://github.com/vivoCameraResearch/HyperMotion).

AC3D. We use the PyTorch implementation and checkpoints from the official Github repository of AC3D (https://github.com/snap-research/ac3d).

GEN3C. We use the PyTorch implementation and checkpoints from the official Github repository of GEN3C (https://github.com/nv-tlabs/GEN3C).

F. Additional Qualitative Results

F.1. Highly Dynamic Object Motion Generation

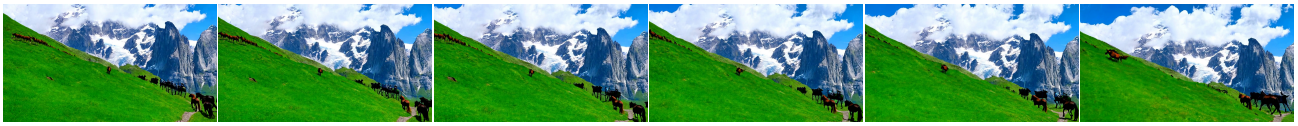
We present additional qualitative results of our method. Fig. S6(a) shows synthesis results on common scenes (e.g., Pexels [1]), where our method produces high-quality video frames. Fig. S6(b) provides examples involving highly dynamic motions, demonstrating that our approach consistently generates natural-looking videos even under vigorous motion.

F.2. Highly Dynamic Camera Motion Control

Fig. S7(a) shows results under moderate viewpoint changes (e.g., RE10K [10]), where our method maintains visual realism. Fig. S7(b) presents challenging scenarios with large rotations and fast translations, confirming that our method can synthesize stable, high-quality videos while accurately following highly dynamic camera trajectories.

(a) Common Scenes

“A fire burns brightly, with flames dancing over logs and embers …”



“A group of horses and riders traverse a lush green mountain slope, set against a backdrop of majestic snow-capped peaks …”

“A man sits comfortably in a yellow armchair, engrossed in reading a book …”

“A man stands confidently, dressed in a light blue shirt and a brown blazer, with a lanyard around his neck …”

“A vibrant underwater scene unfolds, showcasing a coral reef teeming with life …”

(b) Highly Dynamic Object Motions

“A man wearing headphones and athletic gear runs along a concrete riverside path …”

“A person in a futuristic, form-fitting suit and helmet is seen performing a series of dynamic movements in an open, sandy area …”

“A man dressed in a muted brown shirt and fitted dark pants runs across an open, empty space …”

“A stage performer dressed in a sleek white performance outfit performs a series of acrobatic maneuver …”

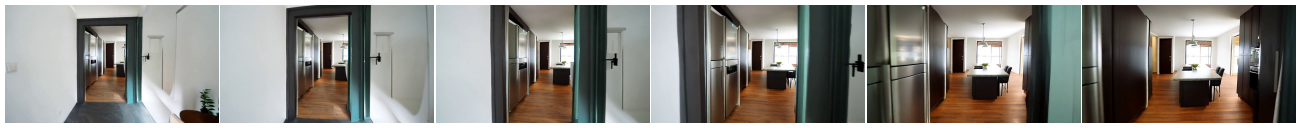
“A fox performs a series of breakdance movements in an open courtyard …”

Figure S6. Additional video synthesis results on common scenes (a) and on highly dynamic object motions (b).

(a) Moderate Camera Motion Control

"A cozy bedroom with a large bed dressed in white linens and a single purple pillow …"

"A man stands in a grassy yard surrounded by large stone sculptures …"



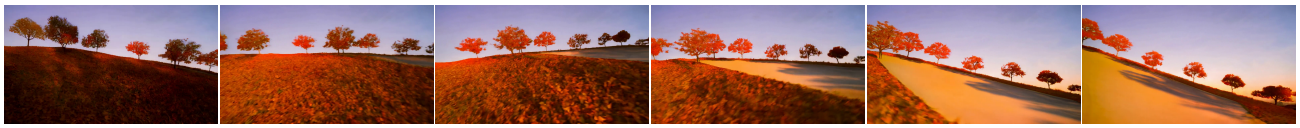
"A modern kitchen and dining area, starting from a hallway leading into the space …"

"A spacious and well-lit bathroom featuring a large corner bathtub, a double sink vanity with dark cabinets …"

"A cozy bedroom with a neatly made bed adorned with a blue throw blanket and patterned pillows …"

(b) Highly Dynamic Camera Motion Control

"A serene outdoor setting during sunset, revealing a modern building with a pool area adorned with floating lily pads …"



"A serene landscape featuring a series of trees with autumn-colored leaves scattered across a gently sloping terrain …"

"In a desolate urban landscape, a mechanical creature with a rugged, armored design races through the streets …"

"A serene outdoor setting during golden hour, featuring a tranquil pond surrounded by a stone wall and scattered with lily pads …"

"A serene outdoor setting during sunset, featuring a stone wall and a small pond adorned with floating lily pads…"

Figure S7. Additional camera-controlled video synthesis results under moderate viewpoint changes (a) and highly dynamic viewpoint changes, including 180° rotations and large translations (b).

References

[1] Pexels, royalty-free stock footage website. https://www.pexels.com. Accessed: 2024-09-30. 4

[2] 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. 4

[3] 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. 2

[4] 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. 2

[5] 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. 4

[6] 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. 3

[7] 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. 2, 4

[8] 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. 4

[9] 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. 4

[10] 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. 1, 4