
Wan-Move: Motion-controllable Video Generation via Latent Trajectory Guidance

Ruihang Chu^{1,2*†‡} Yefei He^{1*} Zhekai Chen^{3*} Shiwei Zhang^{1†} Xiaogang Xu⁴
Bin Xia⁴ Dingdong Wang⁴ Hongwei Yi Xihui Liu³ Hengshuang Zhao³
Yu Liu¹ Yingya Zhang¹ Yujiu Yang^{2†}

¹Tongyi Lab, Alibaba Group ²Tsinghua University ³HKU ⁴CUHK

Github: <https://github.com/ali-vilab/Wan-Move>

Abstract

We present Wan-Move, a simple and scalable framework that brings motion control to video generative models. Existing motion-controllable methods typically suffer from coarse control granularity and limited scalability, leaving their outputs insufficient for practical use. We narrow this gap by achieving precise and high-quality motion control. Our core idea is to directly make the original condition features motion-aware for guiding video synthesis. To this end, we first represent object motions with dense point trajectories, allowing fine-grained control over the scene. We then project these trajectories into latent space and propagate the first frame’s features along each trajectory, producing an aligned spatiotemporal feature map that tells how each scene element should move. This feature map serves as the updated latent condition, which is naturally integrated into the off-the-shelf image-to-video model, *e.g.*, Wan-I2V-14B, as motion guidance without any architecture change. It removes the need for auxiliary motion encoders and makes fine-tuning base models easily scalable. Through scaled training, Wan-Move generates 5-second, 480p videos whose motion controllability rivals Kling 1.5 Pro’s commercial Motion Brush, as indicated by user studies. To support comprehensive evaluation, we further design MoveBench, a rigorously curated benchmark featuring diverse content categories and hybrid-verified annotations. It is distinguished by larger data volume, longer video durations, and high-quality motion annotations. Extensive experiments on MoveBench and the public dataset consistently show Wan-Move’s superior motion quality. Code, models, and benchmark data are made available.

1 Introduction

Motion lies at the heart of video generation as it fundamentally transforms static images into dynamic visual narratives. Recognizing its importance, both the research community [1, 2, 3] and commercial players [4, 5, 6] have devoted considerable effort to controlling motion in video generative models.

The essence of motion control lies in injecting a motion guidance signal into the video generation process. Thus, the two key choices are (i) how to represent the guidance signal and (ii) how to integrate it into the generator. First, existing motion guidance representations can be broadly classified into sparse and dense types. Sparse representations include bounding boxes [7, 8] and segmentation masks [1, 9, 10]. Although these signals can steer an object’s global movement, they fail to control

*Equal contribution

†Corresponding authors

‡Project leader

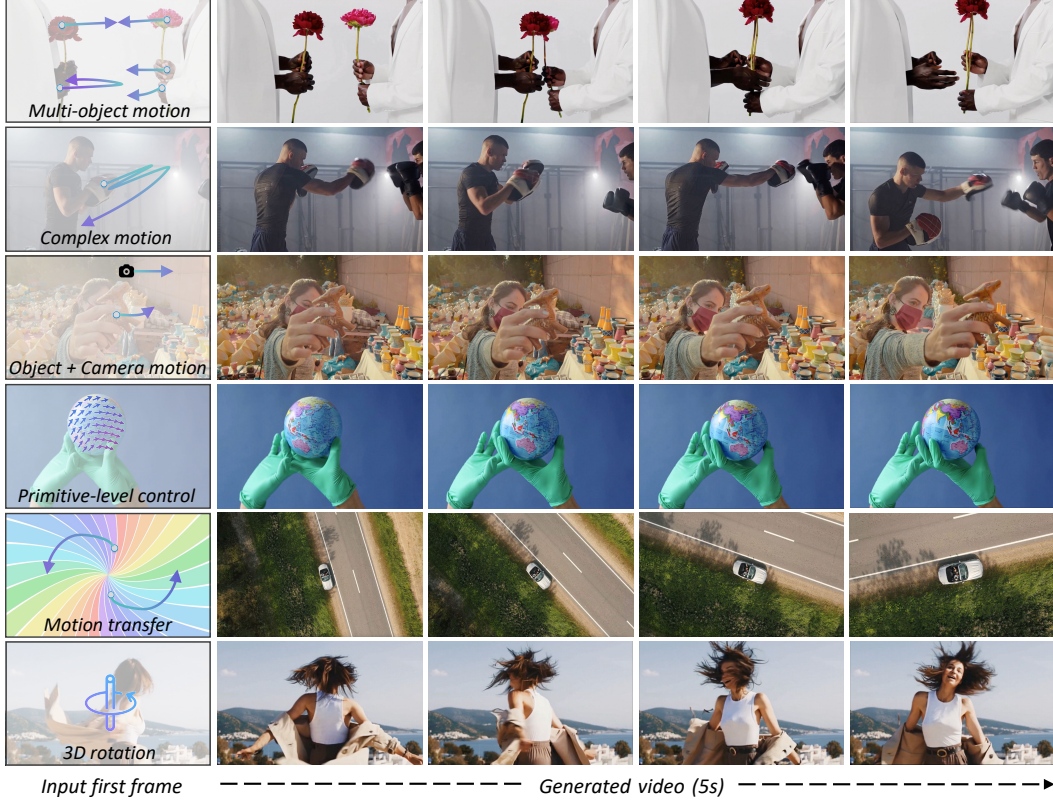


Figure 1: Wan-Move is a image-to-video generation framework that supports diverse motion control applications. The generated samples (832×480 p, 5s) exhibits high visual fidelity and accurate motion.

local motions. In contrast, dense representations, such as pixel-wise optical flow [2, 11, 12, 3] and point trajectories [13, 14], offer more fine-grained control ability. Yet, optical flow requires an additional model for flow estimation during inference, which adds cumulative errors across frames and hampers scalability. While point trajectories are easy to specify during inference, each track is only a single-pixel thread and lacks surrounding spatial context. This makes it hard to align textures and motion patterns across neighboring regions. Second, to inject the guidance signal into generative models, a range of motion encoders have been designed [14, 15, 1, 13, 16, 17], with ControlNet [18] being a popular way to fuse motion cues. However, all of these pipelines introduce extra motion-aware modules, which may degrade the motion signal during processing and make it harder to fine-tune the video-generation backbone at scale.

To tackle these challenges, we present Wan-Move, a novel motion-control framework that builds on the existing image-to-video (I2V) generation model without adding auxiliary motion-processing modules. Our core idea is to inject motion information by directly editing the image condition features. We turn it into an updated latent guide that conveys both appearance and motion throughout video generation. Thanks to this simple and effective design, Wan-Move delivers high-quality motion control and scales easily by fine-tuning the powerful I2V backbone.

Specifically, we represent motion trajectories using point tracks [13] as they capture fine-grained local and global movement. Unlike prior work [13] that embeds point trajectories into latent features, we transfer each trajectory from pixel space into latent coordinates. As I2V generation aims to animate the first frame, we guide this process by copying the first-frame feature at each tracked position to its corresponding location in later frames along the latent trajectory. Each copied feature preserves rich context, thus the propagated signal drives more natural local motion, as verified in Sec. 5.3. Moreover, since motion guidance is injected by editing the image condition features, we add no extra modules. As a result, Wan-Move can plug straight into the I2V backbone, such as Wan-I2V-14B [19], and support scalable fine-tuning with fast convergence. Fig. 1 shows that Wan-Move generates high-fidelity video clips (832×480 p, 5s) with precise motion control, enabling a diverse set of applications, as illustrated in Sec. 5.4. To our best knowledge, it is the first research model (*to be open-sourced*) to match the visual quality of commercial products such as Kling 1.5 Pro’s Motion Brush [4].

To set a rigorous, comprehensive evaluation for motion-control methods, we introduce a free-license benchmark termed MoveBench. Compared with existing benchmarks [20, 21, 22] that offer fewer clips, shorter durations, and incomplete motion annotations, MoveBench provides more data, greater diversity, and reliable motion annotations (Fig. 5). Concretely, we design a curation pipeline to categorize the video library into 54 content categories, 10-25 videos each, giving rise to over 1000 cases to ensure a broad scenario coverage. All video clips maintain a 5-second duration to facilitate evaluation of long-range dynamics. Every clip is paired with detailed motion annotations for single or multiple objects. They include both precise point trajectories and sparse segmentation masks to fit a wide range of motion-control models. We ensure annotation quality by developing an interactive labeling pipeline. It combines human labeling with SAM [23] predictions, marrying annotation precision with automated scalability. In summary, our contributions are as follows:

- We propose Wan-Move for motion control in image-to-video generation. Unlike prior approaches that require motion encoding, it injects the motion guidance by editing condition features, adding no new modules and allowing the easy fine-tuning of base models at scale.
- We introduce MoveBench, a comprehensive and well-curated benchmark to assess motion control. A hybrid human+SAM labeling pipeline ensures annotation quality.
- Extensive experiments on MoveBench and public datasets show that Wan-Move supports diverse motion-control tasks and delivers commercial-grade results with scaled training.

2 Related Work

Video generation models. Video diffusion models [24] pioneer the extension of denoising diffusion probabilistic models (DDPMs) to video generation through a 3D U-Net architecture. Subsequent advancements, such as Imagen Video [25] and Phenaki [26], enhance this framework to produce longer and higher-resolution sequences. Nevertheless, these CNN-based approaches [27, 28, 29] face limitations in capturing long-range spatiotemporal dependencies. Transformer-based architectures [30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46] overcome this bottleneck and greatly improve training scalability. Recent innovations, including CogVideoX [47] and HunyuanVideo [6], further validate the efficacy of spatio-temporal attention mechanisms for coherent video synthesis. Notably, Wan [19] introduces an efficient framework for both text-to-video and image-to-video generation, setting a new standard for open-source video models. Our Wan-Move, introduces how to leverage latent trajectory guidance to enable motion control upon the image-to-video diffusion model, enabling precise motion control while preserving visual fidelity.

Motion-controllable video generation. To adapt pretrained video generation models for motion-controllable synthesis, training-free methods [48, 49, 50, 51] optimize input noisy latents or manipulate attention mechanisms, enabling zero-shot control. However, these approaches often exhibit performance degradation when controlling fine-grained or multi-object motion. In contrast, fine-tuning based methods [52, 2, 15, 1, 3, 14, 20, 53, 16, 54, 55, 56, 10, 12, 57, 58, 59] leverage diverse motion signals and introduce various techniques to integrate them into the base model. While these methods significantly enhance output quality, they typically require auxiliary encoders or fusion modules, complicating the model architecture and limiting training scalability. Among these studies, the most relevant to our work is Motion Prompting [13], as both employ point trajectories to represent motion guidance. However, we differ in two key aspects. First, Motion Prompting [13] encodes point tracks via random embeddings in pixel space, where the guidance is pixel-level threads that lack surrounding context to offer local control. We express point trajectories in latent space using the image feature, providing rich local information and finer control. Second, Motion Prompting [13] integrates motion guidance through a separate ControlNet [18], whereas we directly use the pretrained base model without architectural modifications, which facilitates its scalable fine-tuning. Sec. 5.3 provides quantitative and qualitative evidence of our advantages.

For more specific robotic scenarios, works [60, 61] rely on pretrained DINOv2 features [62] to transfer object representations across frames for motion control in generated videos, yet both delivers DINOv2’s limitation in representing motion signals. DINOv2 excels at high-level semantic encoding but lacks fine-grained object details. Thus, GEM [1] employs additional identity embeddings to distinguish objects and train an ObjectNet to bridge the domain gap between DINOv2 and the UNet’s feature space. Moreover, the 14 patch size in DINOv2 may restrict the granularity of the proposed

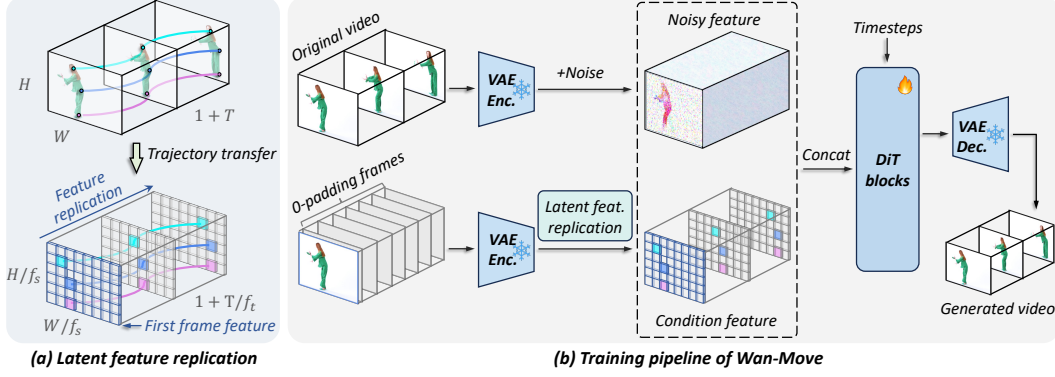


Figure 2: (a) To inject motion guidance, we transfer point trajectories from videos to latent space, then replicate the first frame feature into subsequent zero-padded frames along each latent trajectory. (b) Wan-Move is trained upon an existing image-to-video generation model (e.g., work [19, 29]), with an efficient latent feature replication step (as in (a)) to update the condition feature. The CLIP [65] image encoder and umT5 [66] text encoder from the base model are omitted for simplicity.

control. In contrast, our approach mitigates these limitations by employing native VAE in I2V foundation models without relying on any auxiliary modules (e.g., identity embeddings).

Benchmark for motion-controllable video generation. Most motion-controllable work evaluates on small, task-specific datasets, which are typically ad hoc, limited in scope, or insufficient for evaluating long-range dynamics and multi-object interactions. For example, datasets such as DAVIS [21] and VIPSeg [22] have been repurposed for trajectory control methods, yet their short clip durations and sparse annotations make them inadequate for assessing long-term consistency or complex interactions. While MagicBench [20] expands to 600 video clips, it categorizes samples solely by object count and relies on automatically generated labels from noisy pipelines, limiting the annotation precision. To address these limitations, we introduce MoveBench, a comprehensive benchmark for motion-controllable video generation. It includes carefully selected 1018 videos with extensive annotations, long-range dynamics, and 54 well-classified content patterns.

3 Method

3.1 Preliminary

Video diffusion models [24, 19, 6, 63] apply Gaussian noise to clean data during the forward process and learn a reverse process to denoise and generate videos. To reduce computational costs, the denoising network typically operates on latent video representations obtained from a pretrained VAE [64]. Given an input video $\mathbf{V} \in \mathbb{R}^{(1+T) \times H \times W \times 3}$, the encoder \mathcal{E} compresses both the temporal and spatial dimensions with compression ratios f_t (temporal) and f_s (spatial), while expanding the channel dimension to C , yielding $\mathbf{x} = \mathcal{E}(\mathbf{V}) \in \mathbb{R}^{(1+\frac{T}{f_t}) \times \frac{H}{f_s} \times \frac{W}{f_s} \times C}$. The decoder \mathcal{D} then reconstructs the video from the latent representation as $\hat{\mathbf{V}} = \mathcal{D}(\mathbf{x})$.

Our work focuses on motion-controllable image-to-video (I2V) generation, where models are required to generate motion-coherent videos based on the input first-frame image and motion trajectories. While the first frame will be encoded into the condition feature $\mathbf{z}_{\text{image}}$ by the VAE, motion trajectories, which can be represented in diverse formats, remain in pixel space. Thus, the key challenge lies in effectively encoding motion trajectories into the condition feature $\mathbf{z}_{\text{motion}}$ and injecting it into the generative model. To avoid the signal degradation and training difficulties associated with additional motion encoder and fusion modules, we aim to develop a motion-control framework that leverages existing I2V models without architectural modifications, as detailed below.

3.2 Latent Trajectory Guidance

To enable video generation conditioned on the first frame, an effective approach of popular I2V models [19, 29] concatenates the latent noise \mathbf{x}_t and the first-frame condition feature $\mathbf{z}_{\text{image}}$ along

the channel dimension. $\mathbf{z}_{\text{image}}$ is obtained by encoding the first frame \mathbf{I} along with zero-padded subsequent frames $\mathbf{0}_{T \times H \times W \times C}$ using a pretrained VAE encoder \mathcal{E} :

$$\mathbf{z}_{\text{image}} = \mathcal{E}(\text{concat}[\mathbf{I}, \mathbf{0}_{T \times H \times W \times 3}]) \in \mathbb{R}^{(1+\frac{T}{f_t}) \times \frac{H}{f_s} \times \frac{W}{f_s} \times C}. \quad (1)$$

For motion guidance representation, we adopt point trajectories, following prior studies [13, 14], as they provide fine-grained control and capture both local and global motion. Formally, a point trajectory of length $1 + T$ can be represented as $\mathbf{p} \in \mathbb{R}^{(1+T) \times 2}$, where $\mathbf{p}[n] = (x_n, y_n)$ specifies the trajectory location in the n -th frame in the pixel space. Existing methods often employ auxiliary modules to encode and integrate trajectories into the backbone. Yet, this approach may degrade motion signals during motion encoding. In addition, training extra modules increases the complexity of fine-tuning the base model at scale. This raises a key question: *Can we inject pixel-space motion guidance without auxiliary modules?*

Intuitively, I2V generation aims to animate the first frame, while motion trajectories specify object positions in each subsequent frame. Given the translation equivariance of VAE models, latent features at corresponding trajectory positions should closely resemble those in the first frame. Motivated by this, we propose encoding trajectories directly into latent space via spatial mapping, eliminating the need for an extra motion encoder:

$$\tilde{\mathbf{p}}[n] = \begin{cases} \frac{\mathbf{p}[n]}{f_s} & \text{if } n = 0, \\ \frac{\sum_{i=(n-1) \cdot f_t + 1}^{n \cdot f_t} \mathbf{p}[i]}{f_t \cdot f_s} & 1 \leq n \leq \frac{T}{f_t}, \end{cases} \quad (2)$$

The latent trajectory position at the first frame is derived by spatial mapping, while for subsequent frames, it is averaged over each consecutive f_t frames. This deterministically transforms pixel-space trajectories into latent space. To inject the obtained latent trajectories, we extract the latent features of the first frame at the initial trajectory point $\tilde{\mathbf{p}}[0]$ and replicate them across subsequent frames according to $\tilde{\mathbf{p}}$, leveraging the translation equivariance of latent features, as shown in Fig. 2 (a):

$$\mathbf{z}_{\text{image}}[n, \tilde{\mathbf{p}}[n, 0], \tilde{\mathbf{p}}[n, 1], :] = \mathbf{z}_{\text{image}}[0, \tilde{\mathbf{p}}[0, 0], \tilde{\mathbf{p}}[0, 1], :] \quad \text{for } n = 1, \dots, \frac{T}{f_t}. \quad (3)$$

Here, $\mathbf{z}_{\text{image}}[t, h, w, :]$ denotes the feature vector at temporal index t , height h , and width w . This operation efficiently injects motion guidance into the condition feature by updating $\mathbf{z}_{\text{image}}$, eliminating the need for explicit motion condition features and injection modules. An overview of the Wan-Move generation framework is presented in Fig. 2(b). When multiple visible point trajectories coincide at a given spatiotemporal position, we randomly select one trajectory’s corresponding first-frame feature.

3.3 Training and Inference

Training data. We curate a high-quality training dataset, which undergoes rigorous two-stage filtering to ensure both visual quality and motion consistency. First, we manually annotate the visual quality of 1,000 samples and use them to train an expert scoring model for initial quality assessment. To further enhance temporal coherence, we introduce a motion quality filtering stage. Specifically, for each video, we extract SigLIP [67] features from the first frame and compute the mean SigLIP features for the remaining frames. The cosine similarity between these features serves as our stability metric. Based on empirical analysis of 10,000 samples, we establish a threshold to retain only videos where the content remains consistent with the initial frame. This two-stage pipeline produces a final dataset of 2 million high-quality 720p videos with strong visual quality and motion coherence. Additional details on the training data sources are provided in the supplementary material.

Modeling training. Based on our training dataset, we use CoTracker [68] to track the trajectories of a dense 32×32 grid of points. For each training iteration, we sample k trajectories from a mixed distribution: with 5% probability, no trajectory is used ($k = 0$); with 95% probability, k is uniformly sampled from 1 to 200. Notably, we retain a 5% probability of dropping motion conditions, which effectively preserves the model’s original image-to-video generation capability. For the selected trajectories, we extract the first-frame features and replicate them to subsequent zero-padded frames, as formalized by Eq. (3). Since CoTracker distinguishes between visible and

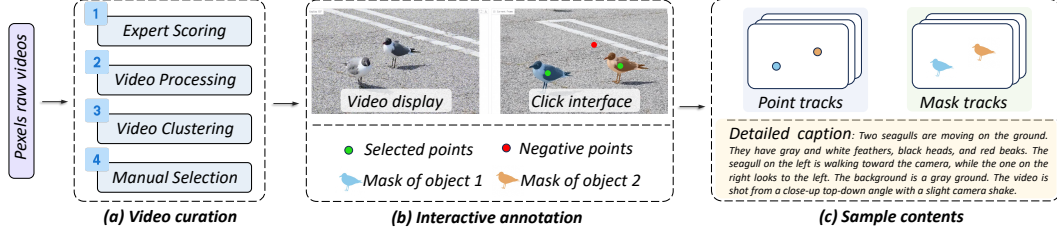


Figure 3: Construction pipeline of MoveBench to obtain high-quality samples with rich annotations.

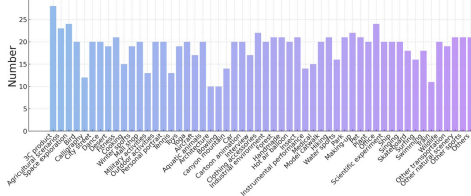


Figure 4: Balanced sample number per class.

Benchmark	Statistics			Trajectory annotations	
	Videos	Frames	Video categorization	Mask track	Point track
DAVIS	50	35-100	✗	✓	✗
VIPSeg (val)	343	24	✗	✓	✗
MagicBench	600	49	✗	✓	✗
MoveBench	1018	81	✓	✓	✓

Figure 5: Comparison with related benchmarks.

occluded point trajectories, we perform feature replication only along the visible trajectories. During training, the model parameters θ is initialized from the I2V model [19] and fine-tuned to predict the vector field $\mathbf{v}_t(\mathbf{x}_t)$ that transports samples from the noise distribution to the data distribution [69]:

$$\mathcal{L}_{\text{FM}}(\theta) = \mathbb{E}_{t, \mathbf{x}_t, \mathbf{c}} [\|\mathbf{v}_\theta(\mathbf{x}_t, t, \mathbf{c}) - \mathbf{v}_t(\mathbf{x}_t)\|^2], \quad (4)$$

where \mathbf{c} denotes the union of the generation condition.

Inference with Wan-Move. The inference process closely resembles the original I2V model, with an additional latent feature replication operation. Specifically, Wan-Move conditions generation on three inputs: (1) a text prompt, (2) an input image as the first frame, and (3) sparse or dense point trajectories for motion control. Pretrained umT5 [66] and CLIP [65] models are employed to encode global context from the text prompt and first frame, respectively. The resulting image embedding $\mathbf{z}_{\text{global}}$ and text embeddings \mathbf{z}_{text} are then injected into the DiT backbone via decoupled cross-attention [18]. Additionally, a VAE is used to extract the first-frame condition feature $\mathbf{z}_{\text{image}}$, which will be injected through latent feature replication (as detailed in Sec. 3.2). Classifier-free guidance is applied to enhance alignment with conditional information. Formally, let unconditional vector field $\mathbf{v}_{\text{uncond}} = \mathbf{v}_\theta(\mathbf{x}_t, t, \mathbf{z}_{\text{image}}, \mathbf{z}_{\text{global}})$, and conditional vector field $\mathbf{v}_{\text{cond}} = \mathbf{v}_\theta(\mathbf{x}_t, t, \mathbf{z}_{\text{image}}, \mathbf{z}_{\text{global}}, \mathbf{z}_{\text{text}})$. The guided vector field $\tilde{\mathbf{v}}_\theta(\mathbf{x}_t, t, \mathbf{z}_{\text{image}}, \mathbf{z}_{\text{global}}, \mathbf{z}_{\text{text}})$ is a weighted combination of the conditional and unconditional outputs, with the guidance scale w :

$$\tilde{\mathbf{v}}_\theta(\mathbf{x}_t, t, \mathbf{z}_{\text{image}}, \mathbf{z}_{\text{global}}, \mathbf{z}_{\text{text}}) = \mathbf{v}_{\text{uncond}} + w(\mathbf{v}_{\text{cond}} - \mathbf{v}_{\text{uncond}}) \quad (5)$$

4 MoveBench

Current benchmarks for motion-controllable video generation suffer from small scale, short duration, and lack precise, comprehensive motion annotations, thus introducing bias and limiting granularity. To address these gaps, we introduce MoveBench, a high-quality benchmark with 1018 videos (480×832 resolution, 5-second duration), designed for comprehensive evaluation of motion-controllable generation, as illustrated in Fig. 3-5. The evaluation videos are selected from Pexels [70], a large-scale, high-quality dataset containing about 400K videos, all released under a free license.

MoveBench combines algorithmic curation with human expertise to ensure diverse, representative, and precisely annotated motion data. Compared to prior works, it offers three key features: **(i) High quality.** We curate videos through a rigorous four-stage pipeline, as illustrated in Fig. 3(a). We first utilize the expert scoring model obtained from Sec. 3.3 to score videos based on visual quality, filtering out low-quality content. Then, the selected videos are cropped to 480p and uniformly sampled to 81 frames to ensure temporal consistency. Finally, videos are clustered into 54 content categories and we manually select the 15-25 most representative examples for each category, balancing diversity and quality (Fig. 4). **(ii) Precise annotations.** As shown in Fig. 5, we provide both point and mask annotations, so that methods using mask guidance signals can also be evaluated using our benchmark.

Table 1: Performance comparisons on MoveBench and DAVIS. Wan-Move consistently yields substantial improvements in both visual fidelity and motion quality across all metrics.

Method	MoveBench					DAVIS				
	FID↓	FVD↓	PSNR↑	SSIM↑	EPE↓	FID↓	FVD↓	PSNR↑	SSIM↑	EPE↓
ImageConductor [53]	34.5	424.0	13.4	0.49	15.6	54.2	513.6	11.6	0.47	14.8
LeviTor [16]	18.1	98.8	15.6	0.54	3.4	22.0	115.4	13.3	0.51	3.7
Tora [14]	22.5	100.4	15.7	0.55	3.3	25.9	129.2	13.7	0.49	3.5
MagicMotion [20]	17.5	96.7	14.9	0.56	3.2	24.2	113.4	12.8	0.53	3.5
Wan-Move (Ours)	12.2	83.5	17.8	0.64	2.6	14.7	94.3	16.5	0.61	2.5

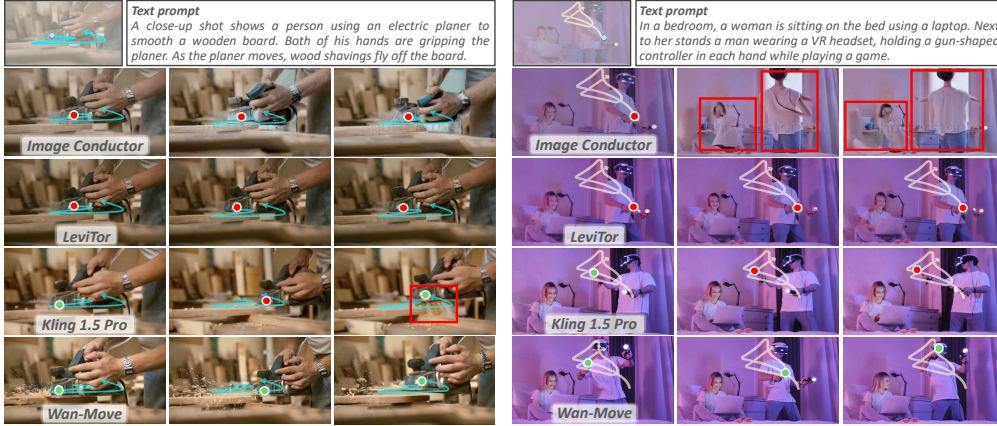


Figure 6: Qualitative comparisons between Wan-Move and recent approaches, including both academic methods [53, 16] and commercial solutions [4]. Motions that deviate from the specified trajectories and major visual artifacts are marked with red dots and boxes, respectively.

To precisely annotate motion regions, we design an interactive annotation interface as shown in Fig. 3. Annotators click on a target region in the first frame, prompting SAM [23] to generate an initial mask. When the mask exceeds the desired area, annotators add negative points to exclude irrelevant regions. This is critical for isolating articulated motions or small objects in cluttered scenes. After annotation, each video contains at least one point indicating a representative motion, with 192 videos additionally including multiple-object motion trajectories. **(iii) Detailed captions.** We use Gemini [71] to generate dense descriptions covering objects, actions, and camera dynamics. Unlike segmentation datasets like DAVIS [21], our captions are tailored for video generation tasks.

5 Experiment

5.1 Experimental Setup

Wan-Move is implemented on top of Wan-I2V-14B [19], a state-of-the-art image-to-video (I2V) generation model. As described in Sec. 3.3, we fine-tune Wan-Move on a high-quality dataset consisting of 2M high-quality videos. Only the DiT backbone is trainable, while the image and text encoders remain frozen. During inference, we use a classifier-free guidance scale w of 5.0 unless otherwise specified. Detailed training configurations are provided in the supplementary material.

To quantitatively evaluate the fidelity of generated videos, we compute standard video quality metrics including FID [72], FVD [73], PSNR, and SSIM [74]. To assess motion accuracy, we measure the L2 distance between ground truth tracks and those estimated from generated videos, following [13] in denoting this metric as end-point error (EPE). All evaluations are performed at a resolution of 480p.

5.2 Main Results

Single-object motion control. We present an extensive comparison between Wan-Move and recent motion-controllable video generation methods [53, 16, 14, 20, 4]. Quantitative results on MoveBench and the public DAVIS [21] are shown in Table 1. Qualitative visualizations are presented in Fig. 6. Unlike other methods that rely on point tracks for motion guidance, MagicMotion [20] takes as input

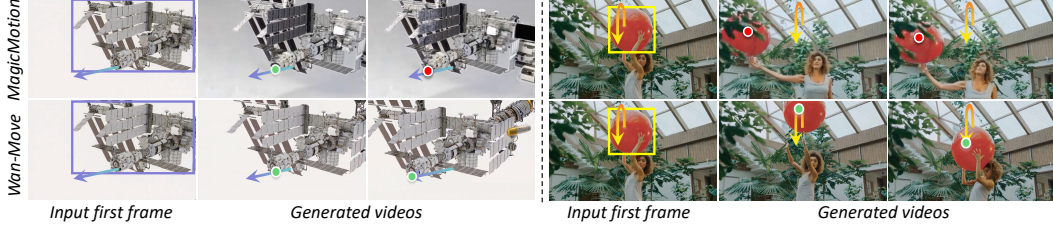


Figure 7: Qualitative comparisons with MagicMotion [20], controlling motions using sparse signals (i.e., bounding boxes) as input. Motions that break the guidance are marked with red dots.

Table 2: MoveBench multi-object motion results.

Method	FID↓	FVD↓	PSNR↑	SSIM↑	EPE↓
ImageConductor	77.5	764.5	13.9	0.51	9.8
Tora	53.2	350.0	14.5	0.54	3.5
Wan-Move (Ours)	28.8	226.3	16.7	0.62	2.2

Table 3: Our win rates in 2AFC human study.

Method	Motion accuracy	Motion quality	Visual quality
LeviTor	98.2	98.0	98.8
Tora	96.2	93.8	98.4
MagicMotion	89.4	96.4	98.2
Kling 1.5 Pro	47.8	53.4	50.2

Table 4: Ablation on motion guidance strategies.

Motion guidance	FID↓	FVD↓	PSNR↑	SSIM↑	EPE↓
Pixel replication	17.3	91.0	15.3	0.56	3.7
Random track embedding	15.4	89.2	16.1	0.59	2.7
Latent feature replication	12.2	83.5	17.8	0.64	2.6

Table 5: Ablation on condition fusion strategies.

Cond. fusion	FID↓	FVD↓	EPE↓	Latency (s)
ControlNet	12.4	84.6	2.5	987 (+225)
Concat. (Ours)	12.2	83.5	2.6	765 (+3)

sparse masks and bounding boxes. Since boxes can be directly derived from segmentation masks, they are inherently compatible with MoveBench that covering mask annotations. Hence, we also conduct comparisons using box-based inputs in Fig. 7. Among these methods, ImageConductor [53] exhibits poor performance in both image and motion quality, which can be attributed to its reliance on direct pixel-level trajectory injection, where single-pixel features lack sufficient semantic and texture information. The remaining methods report similar EPE (3.2–3.4), despite differing motion guidance approaches: Levitor [16] and MagicMotion [20] utilize the complex ControlNet [18], while Tora [14] adopts the lightweight adaLN [30]. Notably, our method achieves the **best motion control performance** (lowest EPE) and **video quality** (highest PSNR and SSIM) through latent trajectory replication without introducing additional parameters. This underscores the effectiveness of our latent trajectory guidance in adhering to motion constraints. Consistent results on the DAVIS dataset further validate the robustness of our approach.

Multi-object motion control. As MoveBench includes 192 cases with annotated multi-object motion, we further evaluate Wan-Move against baselines [53, 14] on this challenging setting, as presented in Table 2. Our method achieves significantly lower FVD and reduced EPE compared to other methods, highlighting its precise adherence to motion constraints in more complex scenarios.

Human study. We conduct a two-alternative forced-choice (2AFC) human evaluation comparing Wan-Move with SOTA approaches [14, 4, 16, 20]. Each method generated 50 conditioned samples, which are evaluated by 20 participants. The results, presented in Table 3, report Wan-Move’s win rates across three metrics: motion accuracy, motion quality, and visual quality. Compared to Tora [14], Wan-Move achieves win rates exceeding 96% in all categories. When evaluated against the commercial model Kling 1.5 Pro, our method demonstrates competitive performance, with superior win rates in motion quality. This narrows the gap between research-oriented and commercial models.

5.3 Ablation Study

Trajectory guidance strategy. We investigate the impact of motion guidance strategies on video quality and motion consistency. Quantitative and qualitative results as presented in Table 4 and Fig. 8, respectively. Pixel replication applies pixel-level copy-paste along the original trajectory, followed by VAE encoding. Yet, since single-pixel features contain limited semantic and texture information, the resulting motion control is weak, as reflected by a high EPE value of 3.7 and generation



Figure 8: Visualization of various guidance strategies.

Table 6: Ablation on maximum number of point trajectories (see Sec. 3.3) during training.

<i>Number</i>	FID↓	FVD↓	PSNR↑	SSIM↑	EPE↓
$N = 10$	12.8	86.6	17.6	0.62	3.3
$N = 100$	12.9	84.7	17.7	0.65	2.7
$N = 200$	12.2	83.5	17.8	0.64	2.6
$N = 500$	13.3	83.9	17.6	0.63	3.0
$N = 1024$	13.4	83.7	17.2	0.61	3.9

Table 7: Ablation on actual number of point trajectories during inference.

<i>Number</i>	FID↓	FVD↓	PSNR↑	SSIM↑	EPE↓
$N = 0$	12.8	87.9	17.9	0.64	12.4
$N = 1$	12.2	83.5	17.8	0.64	2.6
$N = 16$	10.6	78.3	18.2	0.67	2.2
$N = 512$	7.7	51.0	20.3	0.75	1.5
$N = 1024$	6.2	45.2	21.9	0.79	1.1

Table 8: Ablation on different I2V backbones and training data scale. Wan-Move attains better results under the same setting.

<i>Method</i>	<i>Backbone</i>	<i>Data scale</i>	FID↓	FVD↓	PSNR↑	SSIM↑	EPE↓
MagicMotion	CogVideoX-5B	23K	17.5	96.7	14.9	0.56	3.2
Wan-Move-Cog-23K	CogVideoX-5B	23K	16.0	92.3	16.8	0.59	2.8
Tora	CogVideoX-5B	630K	22.5	100.4	15.7	0.55	3.3
Wan-Move-Cog-630K	CogVideoX-5B	630K	14.1	87.3	17.2	0.61	2.8
Wan-Move	Wan2.1-I2V-14B	2000K	12.2	83.5	17.8	0.64	2.6

Table 9: Large-motion and out-of-distribution-motion subset.

<i>Subset</i>	<i>Method</i>	FID↓	FVD↓	EPE↓
Large	Tora	29.1	126.3	4.3
	MagicMotion	24.6	119.3	4.1
	Wan-Move	14.5	86.6	3.0
OOD	Tora	28.9	120.2	4.0
	MagicMotion	23.5	115.7	3.9
	Wan-Move	13.5	86.0	2.8

failures (see Fig. 8). The random track embedding approach, originally proposed for pixel space representations [13], is adapted to assign randomly initialized embeddings in latent space for injecting motion guidance. While effective for rigid single-region control, this approach fails to incorporate contextual information from surrounding regions, resulting in suboptimal video quality (lower PSNR and SSIM) and stiff motion near tracked points. For example, the hand moves, but surrounding bread remains static in Fig. 8. In contrast, our proposed latent feature replication method achieves superior video quality (highest PSNR of 17.8) and precise motion control (lowest EPE of 2.6).

Condition fusion strategy. We compare different motion condition approaches, namely ControlNet [18] and direct concatenation (our approach). The results are presented in Table 5. Notably, simple concatenation of motion conditions with input noise achieves performance comparable to ControlNet in motion-controllable generation. Yet, ControlNet introduces significant additional modules, substantially increasing inference latency by 225 seconds over the original I2V model. In contrast, Wan-Move preserves the base model architecture and only adds a one-time trajectory extraction process, increasing just 3-second inference time.

Number of point trajectories during training. Table 6 evaluates the impact of the maximum number of point tracks (N) during training. As N increases from 10 to 200, the model’s motion-following capability improves progressively, evidenced by the decreasing EPE. The optimal performance, in terms of both structural similarity (SSIM) and EPE, is achieved at $N=200$. However, further increasing the number of point tracks leads to a rise in EPE. This can be attributed to the mismatch between the dense point tracks in the training and the sparse point tracks during evaluation.

Number of point tracks during inference.

Table 12 ablates the performance of Wan-Move across varying numbers of point trajectories over MoveBench. As the number of tracks increases, EPE drops significantly, indicating better motion guidance and enhanced temporal coherence. When reaching the maximum number of point trajectories extracted by CoTracker, Wan-Move achieves the lowest EPE of 1.1. Though it is trained with at most 200 tracks, the model shows strong generalization capability. Notably, naive I2V inference (with no point tracks) yields PSNR and SSIM scores comparable to motion-controlled generation, confirming that our model strongly retains its inherent I2V quality. Naive I2V samples generated by Wan-Move are presented in Fig. 9.

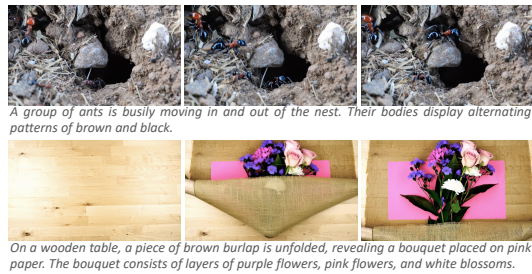


Figure 9: I2V results of Wan-Move (no point tracks).

Backbones and data scale. In pursuit of the best generation quality, we initially train Wan-Move with a large-scale dataset and a strong backbone. To ensure a fair comparison with the leading approaches MagicMotion and Tora, we align Wan-Move’s backbone and training data scale with them. This yields two variants, *i.e.*, Wan-Move-Cog-23K and Wan-Move-Cog-630K, which are trained on 23K and 630K data samples respectively, using CogVideoX1.5-5B-I2V [47] as backbone. The

detailed comparison on MoveBench is shown in Table 8. Under the same backbone and data scale, Wan-Move still outperforms these two powerful methods.

Evaluation on large-motion and out-of-distribution motion scenarios. To further verify the model generalizability, we curate subsets from MoveBench containing high-amplitude and out-of-distribution motion control cases. For each video, its motion amplitude score is computed as the average of the top 2% largest optical flow values extracted by RAFT [75]. The top 20% highest-score videos are selected as large-motion videos. Besides, we manually curate 50 uncommon motion cases as out-of-distribution subset, including complex foreground-background interactions, objects moving out of frame, and rare camera motions. Evaluation results on these challenging examples are shown in Table 9. Notably, Wan-Move consistently outperforms two leading baselines, with performance gaps further widening under these difficult condition. In addition, Wan-Move’s performance only marginally drops compared to its results on the full benchmark, demonstrating its robustness.

5.4 Motion Control Applications

As point trajectories can flexibly represent various types of motion, Wan-Move supports a wide range of motion control applications, as showcased in Fig. 1. First, rows 1–2 show object control using single or multiple point trajectories. For camera control (row 3), we can either drag background elements directly or follow the approach of work [13]. The latter estimates a point cloud from a monocular depth predictor [76], projects it along a camera pose trajectory, and applies z-buffering to obtain camera-aligned 2D trajectories. Following work [13], we perform primitive-level control by rotating a virtual sphere to generate projected 2D trajectories for globe motion (row 4). In row 5, we enable motion transfer by applying trajectories extracted from one video to update the condition features of a different image. Row 6 shows 3D rotation control by estimating depth-based positions, applying a rotation, and projecting the results to 2D. We refer readers to the supplementary file for more visualizations and full videos.

6 Conclusion and Discussion

We propose Wan-Move, a simple and scalable framework for precise motion control in video generation. It represents motion with point trajectories and transfers them into latent coordinates through spatial mapping, requiring no extra motion encoder. We then inject trajectory guidance into first-frame condition features via latent feature replication, achieving effective motion control without architectural changes. For rigorous evaluation, we further present MoveBench, a comprehensive and well-curated benchmark featuring diverse content categories with hybrid-verified annotations. Extensive experiments on MoveBench and public datasets show that Wan-Move generates high-quality, long-duration (5s, 480p) videos with motion controllability on par with commercial tools like Kling 1.5 Pro’s Motion Brush. We believe our open-sourced solution offers an efficient path to scale motion-controllable video generation and will empower a wide range of creators.

Limitations and broader impacts. Wan-Move uses point trajectories to guide motion, which can be unreliable when tracks are missing due to occlusion. While we observe that short-term occlusions can be recovered once the point reappears, showing a degree of generalization, prolonged absence may lead to loss of control (see Appendix). As with other generative models, Wan-Move carries dual-use potential. Its ability to produce realistic, controllable videos can benefit creative industries, education, and simulation, but also risks misuse for generating misleading or harmful content.

7 Acknowledgment

This work was supported by the National Natural Science Foundation of China (Grant No. 62576191).

References

- [1] 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 *ECCV*, 2024.
- [2] 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:2308.08089*, 2023.
- [3] Ryan Burgert, Yuancheng Xu, Wenqi Xian, Oliver Pilarski, Pascal Clausen, Mingming He, Li Ma, Yitong Deng, Lingxiao Li, Mohsen Mousavi, Michael Ryoo, Paul Debevec, and Ning Yu. Go-with-the-flow: Motion-controllable video diffusion models using real-time warped noise. In *CVPR*, 2025.
- [4] Kuaishou. Kling ai. <https://klingai.kuaishou.com>, 2024.06.
- [5] Runway. Gen-3. <https://runwayml.com>, 2024.06.
- [6] Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuozhuo Dai, Jin Zhou, Jiangfeng Xiong, Xin Li, Bo Wu, Jianwei Zhang, et al. Hunyuanvideo: A systematic framework for large video generative models. *arXiv:2412.03603*, 2024.
- [7] Jianzong Wu, Xiangtai Li, Yanhong Zeng, Jiangning Zhang, Qianyu Zhou, Yining Li, Yunhai Tong, and Kai Chen. Motionbooth: Motion-aware customized text-to-video generation. *NeurIPS*, 2024.
- [8] Jiawei Wang, Yuchen Zhang, Jiaxin Zou, Yan Zeng, Guoqiang Wei, Liping Yuan, and Hang Li. Boximator: Generating rich and controllable motions for video synthesis. *arXiv:2402.01566*, 2024.
- [9] Zuozhuo Dai, Zhenghao Zhang, Yao Yao, Bingxue Qiu, Siyu Zhu, Long Qin, and Weizhi Wang. Animateanything: Fine-grained open domain image animation with motion guidance. *arXiv:2311.12886*, 2023.
- [10] Haitao Zhou, Chuang Wang, Rui Nie, Jinlin Liu, Dongdong Yu, Qian Yu, and Changhu Wang. Trackgo: A flexible and efficient method for controllable video generation. In *AAAI*, 2025.
- [11] Mathis Koroglu, Hugo Caselles-Dupré, Guillaume Jeanneret Sanmiguel, and Matthieu Cord. On-lyflow: Optical flow based motion conditioning for video diffusion models. *arXiv:2411.10501*, 2024.
- [12] 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 *SIGGRAPH*, 2024.
- [13] Daniel Geng, Charles Herrmann, Junhwa Hur, Forrester Cole, Serena Zhang, Tobias Pfaff, Tatiana Lopez-Guevara, Carl Doersch, Yusuf Aytar, Michael Rubinstein, et al. Motion prompting: Controlling video generation with motion trajectories. *arXiv:2412.02700*, 2024.
- [14] Zhenghao Zhang, Junchao Liao, Menghao Li, Zuozhuo Dai, Bingxue Qiu, Siyu Zhu, Long Qin, and Weizhi Wang. Tora: Trajectory-oriented diffusion transformer for video generation. *arXiv:2407.21705*, 2024.
- [15] 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 *SIGGRAPH*, 2024.

- [16] Hanlin Wang, Hao Ouyang, Qiuyu Wang, Wen Wang, Ka Leong Cheng, Qifeng Chen, Yujun Shen, and Limin Wang. Levitor: 3d trajectory oriented image-to-video synthesis. *arXiv:2412.15214*, 2024.
- [17] Yuqing Chen, Junjie Wang, Lin Liu, Ruihang Chu, Xiaopeng Zhang, Qi Tian, and Yujiu Yang. O-disco-edit: Object distortion control for unified realistic video editing. *arXiv preprint arXiv:2509.01596*, 2025.
- [18] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *ICCV*, 2023.
- [19] Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwei Yu, Haiming Zhao, Jianxiao Yang, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.
- [20] Quanhao Li, Zhen Xing, Rui Wang, Hui Zhang, Qi Dai, and Zuxuan Wu. Magicmotion: Controllable video generation with dense-to-sparse trajectory guidance. *arXiv:2503.16421*, 2025.
- [21] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alex Sorkine-Hornung, and Luc Van Gool. The 2017 davis challenge on video object segmentation. *arXiv:1704.00675*, 2017.
- [22] Jiaxu Miao, Xiaohan Wang, Yu Wu, Wei Li, Xu Zhang, Yunchao Wei, and Yi Yang. Large-scale video panoptic segmentation in the wild: A benchmark. In *CVPR*, 2022.
- [23] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment Anything. In *ICCV*, 2023.
- [24] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *arXiv:2204.03458*, 2022.
- [25] Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P. Kingma, Ben Poole, Mohammad Norouzi, David J. Fleet, and Tim Salimans. Imagen video: High definition video generation with diffusion models. *arXiv:2210.02303*, 2022.
- [26] Ruben Villegas, Mohammad Babaeizadeh, Pieter-Jan Kindermans, Hernan Moraldo, Han Zhang, Mohammad Taghi Saffar, Santiago Castro, Julius Kunze, and Dumitru Erhan. Phenaki: Variable length video generation from open domain textual description. *arXiv:2210.02399*, 2022.
- [27] Haoxin Chen, Yong Zhang, Xiaodong Cun, Menghan Xia, Xintao Wang, Chao Weng, and Ying Shan. Videocrafter2: Overcoming data limitations for high-quality video diffusion models. In *CVPR*, 2024.
- [28] Jiuniu Wang, Hangjie Yuan, Dayou Chen, Yingya Zhang, Xiang Wang, and Shiwei Zhang. Modelscope text-to-video technical report. *arXiv:2308.06571*, 2023.
- [29] Omer Bar-Tal, Hila Chefer, Omer Tov, Charles Herrmann, Roni Paiss, Shiran Zada, Ariel Ephrat, Junhwa Hur, Guanghui Liu, Amit Raj, et al. Lumiere: A space-time diffusion model for video generation. In *SIGGRAPH Asia*, 2024.
- [30] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *ICCV*, 2023.
- [31] Wenyi Hong, Ming Ding, Wendi Zheng, Xinghan Liu, and Jie Tang. Cogvideo: Large-scale pretraining for text-to-video generation via transformers. *arXiv:2205.15868*, 2022.
- [32] Junsong Chen, Jincheng Yu, Chongjian Ge, Lewei Yao, Enze Xie, Yue Wu, Zhongdao Wang, James Kwok, Ping Luo, Huchuan Lu, et al. Pixart- α : Fast training of diffusion transformer for photorealistic text-to-image synthesis. In *ICLR*, 2024.
- [33] Yang Jin, Zhicheng Sun, Ningyuan Li, Kun Xu, Kun Xu, Hao Jiang, Nan Zhuang, Quzhe Huang, Yang Song, Yadong Mu, and Zhouchen Lin. Pyramidal flow matching for efficient video generative modeling. In *ICLR*, 2025.

- [34] Yoav HaCohen, Nisan Chiprut, Benny Brazowski, Daniel Shalem, Dudu Moshe, Eitan Richardson, Eran Levin, Guy Shiran, Nir Zabari, Ori Gordon, Poriya Panet, Sapir Weissbuch, Victor Kulikov, Yaki Bitterman, Zeev Melumian, and Ofir Bibi. Ltx-video: Realtime video latent diffusion. *arXiv:2501.00103*, 2024.
- [35] Xin Ma, Yaohui Wang, Gengyun Jia, Xinyuan Chen, Ziwei Liu, Yuan-Fang Li, Cunjian Chen, and Yu Qiao. Latte: Latent diffusion transformer for video generation. *TMLR*, 2025.
- [36] Bin Lin, Yunyang Ge, Xinhua Cheng, Zongjian Li, Bin Zhu, Shaodong Wang, Xianyi He, Yang Ye, Shenghai Yuan, Liuhan Chen, et al. Open-sora plan: Open-source large video generation model. *arXiv:2412.00131*, 2024.
- [37] OpenAI. Video generation models as world simulators, 2024.
- [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:2410.13720*, 2024.
- [39] Fan Bao, Chendong Xiang, Gang Yue, Guande He, Hongzhou Zhu, Kaiwen Zheng, Min Zhao, Shilong Liu, Yaole Wang, and Jun Zhu. Vidu: a highly consistent, dynamic and skilled text-to-video generator with diffusion models. *arXiv:2405.04233*, 2024.
- [40] MiniMax. Hailuo ai. <https://hailuoai.com/video>, 2024.09.
- [41] Zangwei Zheng, Xiangyu Peng, Tianji Yang, Chenhui Shen, Shenggui Li, Hongxin Liu, Yukun Zhou, Tianyi Li, and Yang You. Open-sora: Democratizing efficient video production for all, March 2024.
- [42] GenmoTeam. Mochi 1. <https://github.com/genmoai/models>, 2024.
- [43] Guoqing Ma, Haoyang Huang, Kun Yan, Liangyu Chen, Nan Duan, Shengming Yin, Changyi Wan, Ranchen Ming, Xiaoni Song, Xing Chen, et al. Step-video-t2v technical report: The practice, challenges, and future of video foundation model. *arXiv:2502.10248*, 2025.
- [44] Junsong Chen, Yuyang Zhao, Jincheng Yu, Ruihang Chu, Junyu Chen, Shuai Yang, Xianbang Wang, Yicheng Pan, Daquan Zhou, Huan Ling, et al. Sana-video: Efficient video generation with block linear diffusion transformer. *arXiv preprint arXiv:2509.24695*, 2025.
- [45] Shuai Yang, Wei Huang, Ruihang Chu, Yicheng Xiao, Yuyang Zhao, Xianbang Wang, Muyang Li, Enze Xie, Yingcong Chen, Yao Lu, et al. Longlive: Real-time interactive long video generation. *arXiv preprint arXiv:2509.22622*, 2025.
- [46] Shentong Mo, Enze Xie, Ruihang Chu, Lanqing Hong, Matthias Niessner, and Zhenguo Li. Dit-3d: Exploring plain diffusion transformers for 3d shape generation. *Advances in neural information processing systems*, 36:67960–67971, 2023.
- [47] Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, Da Yin, Xiaotao Gu, Yuxuan Zhang, Weihan Wang, Yean Cheng, Ting Liu, Bin Xu, Yuxiao Dong, and Jie Tang. CogVideoX: Text-to-Video Diffusion Models with An Expert Transformer. In *ICLR*, 2025.
- [48] 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:2411.04989*, 2024.
- [49] Haonan Qiu, Zhaoxi Chen, Zhouxia Wang, Yingqing He, Menghan Xia, and Ziwei Liu. Freetraj: Tuning-free trajectory control in video diffusion models. *arXiv:2406.16863*, 2024.
- [50] Wan-Duo Kurt Ma, John P Lewis, and W Bastiaan Kleijn. Trailblazer: Trajectory control for diffusion-based video generation. In *SIGGRAPH Asia*, 2024.
- [51] Yash Jain, Anshul Nasery, Vibhav Vineet, and Harkirat Behl. Peekaboo: Interactive video generation via masked-diffusion. In *CVPR*, 2024.

- [52] Xiang Wang, Hangjie Yuan, Shiwei Zhang, Dayou Chen, Jiuniu Wang, Yingya Zhang, Yujun Shen, Deli Zhao, and Jingren Zhou. VideoComposer: Compositional video synthesis with motion controllability. In *NeurIPS*, 2023.
- [53] Yaowei Li, Xintao Wang, Zhaoyang Zhang, Zhouxia Wang, Ziyang Yuan, Liangbin Xie, Ying Shan, and Yuexian Zou. Image conductor: Precision control for interactive video synthesis. In *AAAI*, 2025.
- [54] Xiao Fu, Xian Liu, Xintao Wang, Sida Peng, Menghan Xia, Xiaoyu Shi, Ziyang Yuan, Pengfei Wan, Di Zhang, and Dahua Lin. 3dtrajmaster: Mastering 3d trajectory for multi-entity motion in video generation. In *ICLR*, 2024.
- [55] Qinghe Wang, Yawen Luo, Xiaoyu Shi, Xu Jia, Huchuan Lu, Tianfan Xue, Xintao Wang, Pengfei Wan, Di Zhang, and Kun Gai. Cinemaster: A 3d-aware and controllable framework for cinematic text-to-video generation. *arXiv:2502.08639*, 2025.
- [56] 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. *arXiv:2501.03847*, 2025.
- [57] Yingjie Chen, Yifang Men, Yuan Yao, Miaomiao Cui, and Liefeng Bo. Perception-as-control: Fine-grained controllable image animation with 3d-aware motion representation. *arXiv:2501.05020*, 2025.
- [58] Xiang Wang, Shiwei Zhang, Haonan Qiu, Ruihang Chu, Zekun Li, Yingya Zhang, Changxin Gao, Yuehuan Wang, Chunhua Shen, and Nong Sang. Replace anyone in videos. *arXiv preprint arXiv:2409.19911*, 2024.
- [59] Bin Xia, Jiyang Liu, Yuechen Zhang, Bohao Peng, Ruihang Chu, Yitong Wang, Xinglong Wu, Bei Yu, and Jiaya Jia. Dreamve: Unified instruction-based image and video editing. *arXiv preprint arXiv:2508.06080*, 2025.
- [60] Mariam Hassan, Sebastian Stapf, Ahmad Rahimi, Pedro Rezende, Yasaman Haghighi, David Brüggemann, Isinsu Katircioglu, Lin Zhang, Xiaoran Chen, Suman Saha, et al. Gem: A generalizable ego-vision multimodal world model for fine-grained ego-motion, object dynamics, and scene composition control. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 22404–22415, 2025.
- [61] Aram Davtyan, Sepehr Sameni, Björn Ommer, and Paolo Favaro. Cage: Unsupervised visual composition and animation for controllable video generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 16163–16171, 2025.
- [62] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.
- [63] 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:2311.15127*, 2023.
- [64] Diederik P Kingma, Max Welling, et al. Auto-encoding variational bayes, 2013.
- [65] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021.
- [66] Hyung Won Chung, Noah Constant, Xavier Garcia, Adam Roberts, Yi Tay, Sharan Narang, and Orhan Firat. Unimax: Fairer and more effective language sampling for large-scale multilingual pretraining. *arXiv:2304.09151*, 2023.
- [67] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *ICCV*, 2023.

- [68] Nikita Karaev, Ignacio Rocco, Benjamin Graham, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Cotracker: It is better to track together. In *ECCV*, 2024.
- [69] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv:2210.02747*, 2022.
- [70] JovianZM. Pexels-400k dataset. <https://huggingface.co/datasets/jovianzm/Pexels-400k>, 2024.
- [71] Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. Gemini: a family of highly capable multimodal models. *arXiv:2312.11805*, 2023.
- [72] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *NeurIPS*, 2017.
- [73] Thomas Unterthiner, Sjoerd Van Steenkiste, Karol Kurach, Raphal Marinier, Marcin Michalski, and Sylvain Gelly. Fvd: A new metric for video generation. In *ICLR*, 2019.
- [74] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *TIP*, 2004.
- [75] 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.
- [76] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. In *NeurIPS*, 2024.
- [77] 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, and Sergey Tulyakov. Panda-70m: Captioning 70m videos with multiple cross-modality teachers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.
- [78] Pixabay. Discover and download free videos - pixabay, 2025.
- [79] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibao Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.
- [80] PyTorch Team. Pytorch fully sharded data parallel (fsdp). <https://pytorch.org/docs/stable/fsdp.html>, 2021. Accessed: [Insert Date].
- [81] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.
- [82] Sam Ade Jacobs, Masahiro Tanaka, Chengming Zhang, Minjia Zhang, Shuaiwen Leon Song, Samyam Rajbhandari, and Yuxiong He. Deepspeed ulyssees: System optimizations for enabling training of extreme long sequence transformer models. *arXiv preprint arXiv:2309.14509*, 2023.
- [83] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

NeurIPS Paper Checklist

1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

Answer: [\[Yes\]](#)

Justification: In the abstract and introduction, we detail our task objectives, weaknesses in the current methods, and the improvements we propose for these weaknesses.

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [\[Yes\]](#)

Justification: In the end part of our main paper, we discuss the limitations of this paper.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. Theory assumptions and proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [\[NA\]](#)

Justification: The paper does not include theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. Experimental result reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [\[Yes\]](#)

Justification: We fully disclose all the experimental information in Section 5.1 and supplementary material.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
 - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
 - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
 - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
 - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The code will be open-sourced upon acceptance.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. Experimental setting/details

Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: We specify the training and test details in Section 5.1 and supplementary material.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. Experiment statistical significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [NA]

Justification: For motion-controllable video generation tasks, the error bar is uncommon, and there is currently no mature error evaluation system in this community.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. Experiments compute resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: We provide training details in the supplementary material.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. Code of ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines>?

Answer: [Yes]

Justification: We have carefully reviewed the NeurIPS Code of Ethics to ensure that our submission complies with all regulations.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. Broader impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: We have discussed broader impacts in Section 6.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: The pre-trained model used in our paper is an image-to-video model, and the content of the generated video is specified by the input image. The paper poses no such risks.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: The models and data used in this paper are open-sourced and authorized.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

Justification: The paper does not release new assets.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. Declaration of LLM usage

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (<https://neurips.cc/Conferences/2025/LLM>) for what should or should not be described.

Wan-Move: Motion-controllable Video Generation via Latent Trajectory Guidance

Supplementary Material

Contents

8 Implementation Details	23
8.1 Training Data Details	23
8.2 MoveBench Construction Details	23
8.3 Training and Inference Configuration	24
9 Additional Experiments	25
9.1 Choice of Feature Replication Strategies under Trajectory Overlap	25
9.2 Choice of Different Training Strategies	25
9.3 Model Performance Under Trajectory Disappearance	25
9.4 Failure Cases	25
10 Qualitative Visualizations	26
10.1 More Qualitative Comparisons	26
10.2 More Camera Control Results	26
10.3 More Motion Transfer Results	26
10.4 More 3D Rotation Results	27

8 Implementation Details

8.1 Training Data Details

Table 10 presents the composition of the filtered training datasets, which are sourced from Panda70M [77], Pixabay [78], Pexels [70], and YouTube. YouTube videos are independently collected for this study. To prevent data leakage, the videos from Pexels are strictly separated from those in the proposed MoveBench.

All videos for training are captioned using Qwen2.5-VL [79], with the prompt structure illustrated in Fig. 10. This prompt emphasizes motion and camera attributes while preserving the fundamental scene descriptions, ensuring that the model semantically understands the context and generates physically plausible motions. The same captioning prompt is applied to the videos in MoveBench.

8.2 MoveBench Construction Details

Video content clustering. Following the initial filtering stage, we conduct a rigorous content clustering process to ensure broad scenario coverage in our benchmark. Specifically, we sample 16 frames per filtered video and compute the average of their SigLip [67] features. Using k-means clustering, we group these features into 54 distinct content categories. Each category label, *e.g.*, Tennis, is then automatically captioned using Qwen2.5-VL [79]. Finally, we manually select the 15–25 most representative videos per category to maintain a balance of diversity.

Interactive labeling. Existing models often fail to accurately identify representative motion regions in videos, as the most prominent motion may not be optimal, and many motions terminate prematurely.

Table 10: The statistics of the training datasets.

Dataset source	Number	Captioner
Panda70M [77]	0.56M	Qwen2.5-VL
Pixabay [78]	0.42M	Qwen2.5-VL
Pexels [70]	0.25M	Qwen2.5-VL
YouTube	0.75M	Qwen2.5-VL

Figure 10: Prompt for video caption.

Video Caption Prompt
VIDEO_PROMPT = "Please describe the video in a concise and natural paragraph. Your description should follow these rules:\n\"
 "a) Focus primarily on the motion and behavior of main subjects in the video, such as people or animals. Describe their actions in chronological order.\n\"
 "b) Briefly describe the appearance and number of these subjects, including details like color, size, and orientation.\n\"
 "c) Mention spatial relationships between subjects if relevant (e.g., in front of, to the left of, etc.).\n\"
 "d) You should describe the camera perspective and movement at the end of the description, including the shooting angle (e.g., top-down, frontal, side view) and camera motion (e.g., pan left, zoom in, dolly out, slight shake), especially if they contribute to the perception of motion.\n\"
 "e) Briefly describe the background or scene, but keep it minimal unless it's important for understanding the motion.\n\"
 "f) Do not include text recognition, named characters, or style analysis (e.g., realistic, animated), unless they are essential for understanding the motion.\n\"
 "Keep your description concise and fluent, ideally within 2-5 sentences. Your description: "

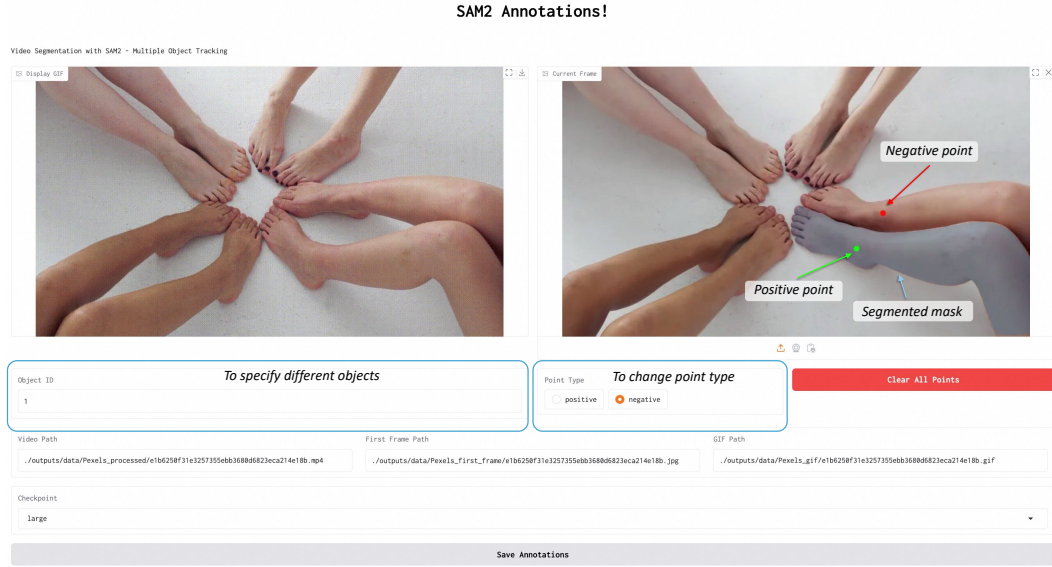


Figure 11: The interactive annotation interface displays the video (left) and its first frame (right). Users click green positive points to specify the start point of a motion trajectory, and red negative points to exclude irrelevant regions if needed. SAM segments the mask of moving objects & regions for user review. To annotate multiple motion trajectories, users must assign different object IDs.

To facilitate precise annotation of motion regions, we introduce an interactive labeling interface (Fig. 11) for selecting the initial motion point and its corresponding mask in the first frame. Annotators begin by selecting a target point in the initial frame, prompting SAM [23] to generate a preliminary segmentation mask. If the mask extends beyond the desired area, negative points can be added to exclude irrelevant regions. This method effectively isolates articulated motions or small objects. For subsequent frames, point trajectories are automatically extracted using CoTracker [68].

8.3 Training and Inference Configuration

As illustrated in Sec. 5.1 of the main paper, we employ Wan-I2V-14B [19] as the base I2V model. During training, both the DiT and umT5 components of Wan are wrapped with Fully Sharded Data Parallel (FSDP) [80], with parameters cast to `torch.bfloat16` for memory efficiency. The training employs the AdamW optimizer [81] with a weight decay of $1e-3$ and a base learning rate of $5e-6$. The first 2,000 steps are used for linear warm-up to enable a smooth transition from the initial I2V generation (corresponding to 0 point trajectories) to motion-controllable video generation. We adopt flow matching objective for optimization, where the number of time sampling steps is set to 1,000 during training. To enable large-scale training with long sequences (e.g., 5s video clip), we adopt the Ulysses sequence parallelism strategy [82] following Wan, setting the sequence parallel size to 4. We train our model using 64 NVIDIA A100 GPUs, with each GPU processing a quarter of

Table 11: Impact of feature replication strategies when multiple motion trajectories overlap.

strategy	FID↓	FVD↓	PSNR↑	SSIM↑	EPE↓
Average	13.1	83.4	17.5	0.63	2.7
Random	12.2	83.5	17.8	0.64	2.6

Table 12: Impact of using a dense-to-sampling training strategy.

Strategy	FID↓	FVD↓	PSNR↑	SSIM↑	EPE↓
Dense-to-Sampling	12.9	84.2	17.5	0.62	2.6
Sampling	12.2	83.5	17.8	0.64	2.6

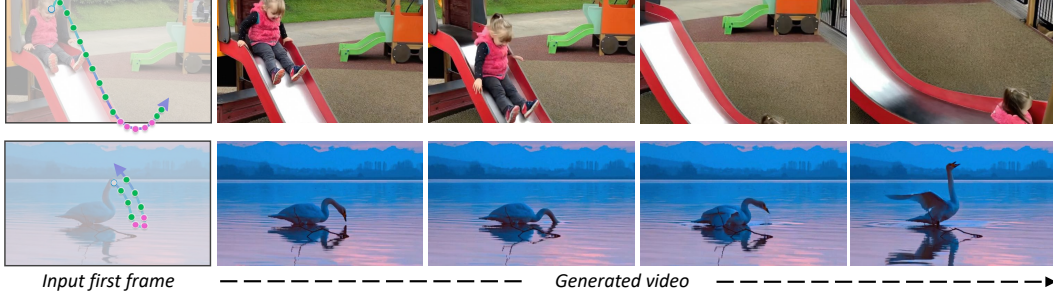


Figure 12: Wan-Move generalizes to continue controlling motion trajectories when they temporarily disappears. Green circles indicate visible segments, while red circles mark invisible segments, *e.g.*, occluded or out-of-frame parts.

the sequence length, for a total of 30,000 steps. During inference, we follow Wan’s sampling scheme with 50 sampling steps.

9 Additional Experiments

9.1 Choice of Feature Replication Strategies under Trajectory Overlap

We analyze the impact of feature replication strategies in cases of trajectory overlap. The results, presented in Table 11, demonstrate that randomly selecting a single trajectory’s first-frame feature for replication when multiple trajectories coincide yields superior video quality and motion control. This is evidenced by lower FVD and EPE compared to feature averaging. We hypothesize that averaging features from overlapping trajectories leads to information loss, thereby degrading performance.

9.2 Choice of Different Training Strategies

This subsection evaluates the performance differences between dense-to-sampling and direct sampling training strategies, as presented in Table 12. Prior work [14, 15, 2, 53] commonly employs a two-stage dense-to-sampling pipeline, where the first stage uses dense motion trajectories to enhance motion control followed by sparse trajectories in the second stage. However, we find that our model, trained with randomly sampling of 1-200 points (as refer to Sec. 3.3 in the main paper), achieves comparable EPE and lower FVD compared to the two-stage approach. These results demonstrate that our method provides generalization capability in point trajectory numbers while simplifying the training process. This generalization ability is also verified in Table 12 of the main paper.

9.3 Model Performance Under Trajectory Disappearance

Fig. 12 illustrates our model’s capability to generate motion-coherent videos when handling temporarily invisible trajectories. The Wan-Move maintains stable generation quality in these challenging scenarios, which we attribute to both the presence of similar cases in the training data and the model’s inherent generalization capacity.

9.4 Failure Cases

This subsection analyzes and visualizes three primary failure modes of Wan-Move, as illustrated in Fig. 13. First, control degradation occurs when motion trajectories remain invisible for extended durations, causing the model to lose conditional guidance. Second, performance deteriorates in visually complex scenes with multiple interacting objects in crowded environments. Third, implausible

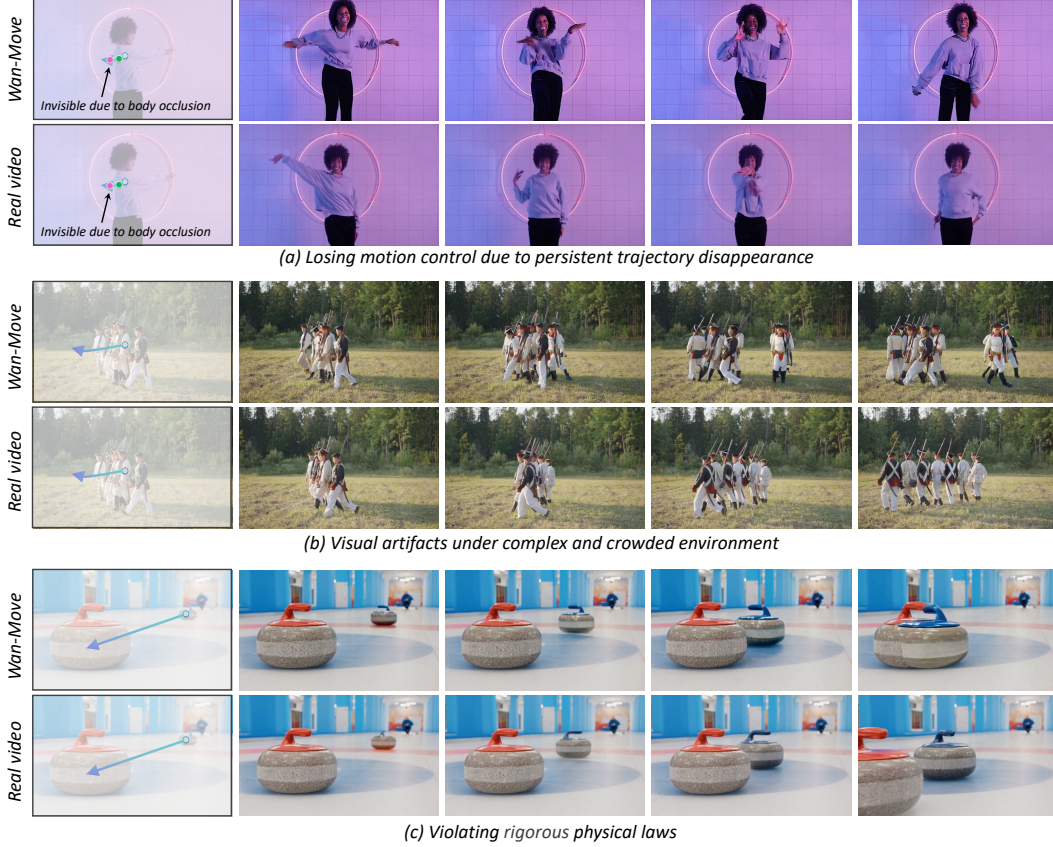


Figure 13: Three primary failure modes of Wan-Move. (a) Loss of motion control due to persistent trajectory disappearance; (b) Visual artifacts in overly complex, crowded environments; and (c) motion outputs that violate rigorous physical laws.

motion trajectories that violate fundamental physical laws result in out-of-distribution predictions. Furthermore, erroneous tracking points identified by CoTracker [68] may compound these failure modes.

10 Qualitative Visualizations

10.1 More Qualitative Comparisons

We present additional qualitative comparisons with state-of-the-art academic [14] and commercial [4] approaches, as shown in Fig. 14.

10.2 More Camera Control Results

As demonstrated in Fig. 15, Wan-Move enables camera control. This can be accomplished, following the work [13], by estimating a point cloud using a monocular depth predictor [76], projecting it along a predefined camera trajectory, and applying z-buffering to derive occlusion flags and camera-aligned 2D trajectories.

10.3 More Motion Transfer Results

This subsection presents dense motion transfer visualizations generated by Wan-Move using dense point trajectories (1,024 in our implementation). As illustrated in Fig. 16, Wan-Move achieves nearly identical appearance quality and motion alignment compared to the original videos given dense

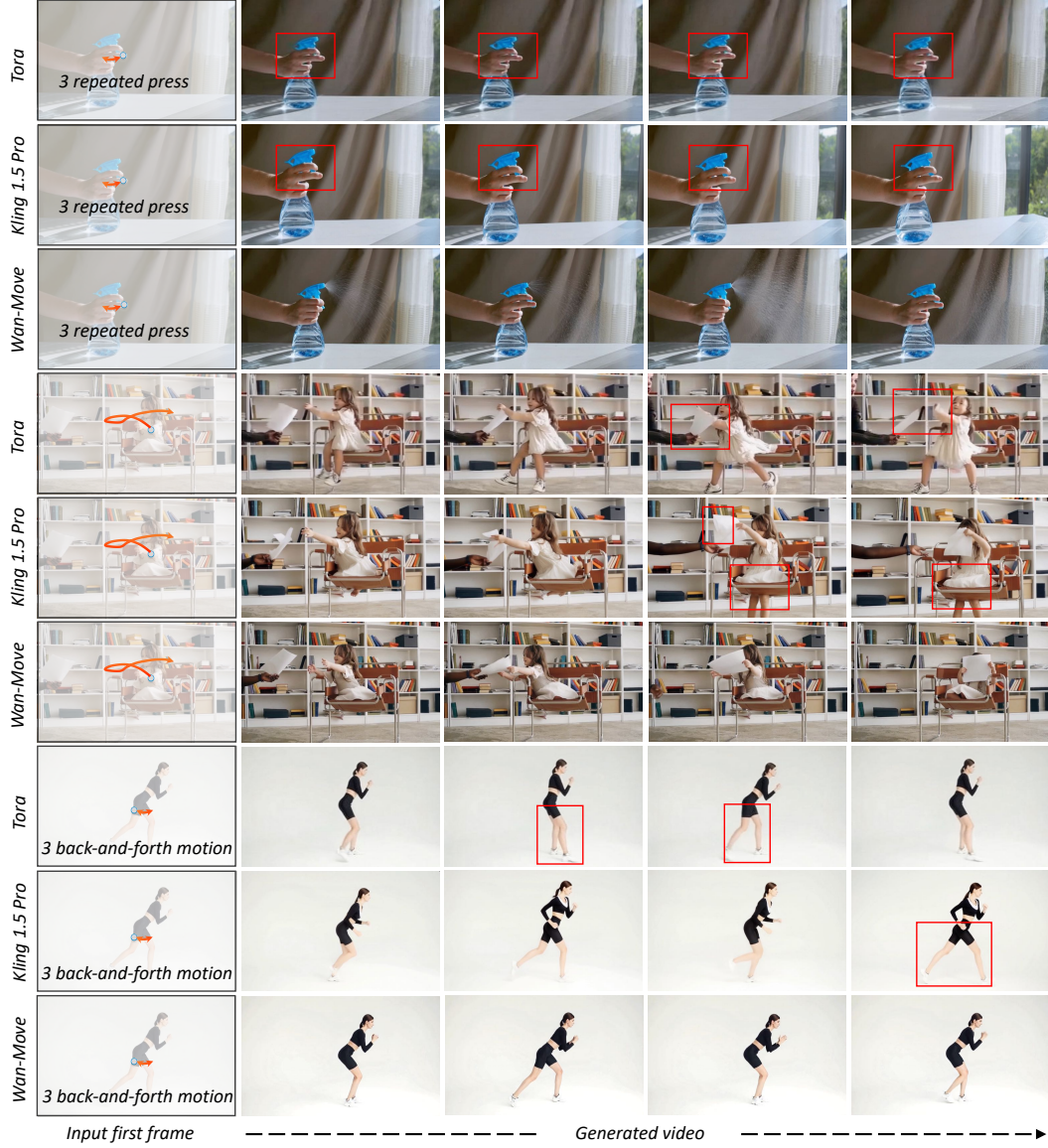


Figure 14: Additional qualitative comparisons with Tora [14] and the commercial model Kling 1.5 Pro [4]. Wan-Move demonstrates superior motion accuracy and visual quality. Major motion control failures or visual artifacts are denoted with red boxes.

trajectory conditions and the same first frame. Moreover, Wan-Move also enables video editing by copying the motion while using an additional image editing model to modify the content in the first frame, maintaining the original video’s motion trajectories, as shown in Fig. 17.

10.4 More 3D Rotation Results

As illustrated in Fig. 18, Wan-Move additionally supports 3D object rotation. This capability is realized by first estimating depth-based 3D positions, applying a rotational transformation, and then reprojecting the results into 2D trajectories. These trajectories subsequently serve as conditioning inputs for our model to rotate the objects in videos.

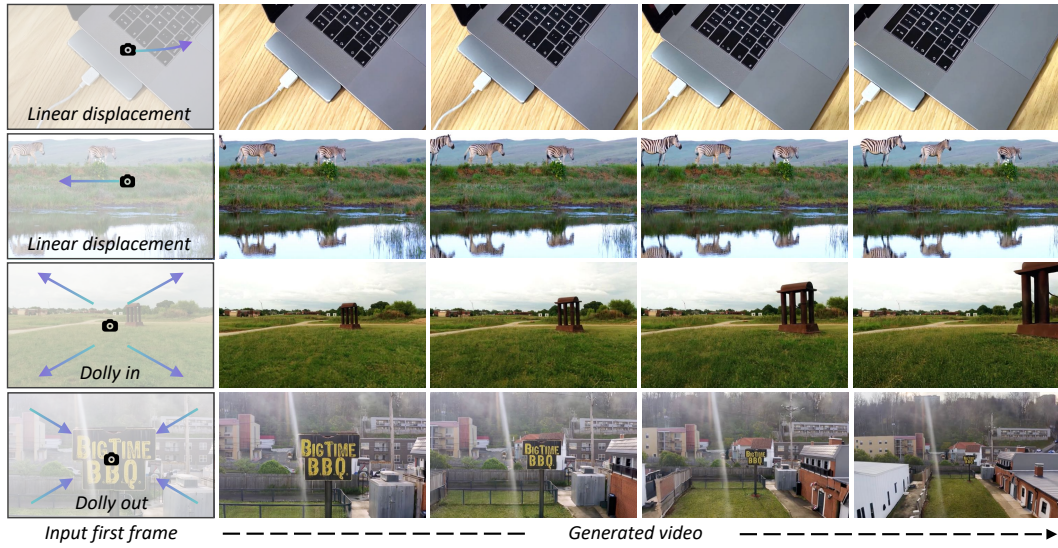


Figure 15: Wan-Move enables effective and flexible camera control through different point trajectories, such as linear displacement, dolly in, and dolly out.

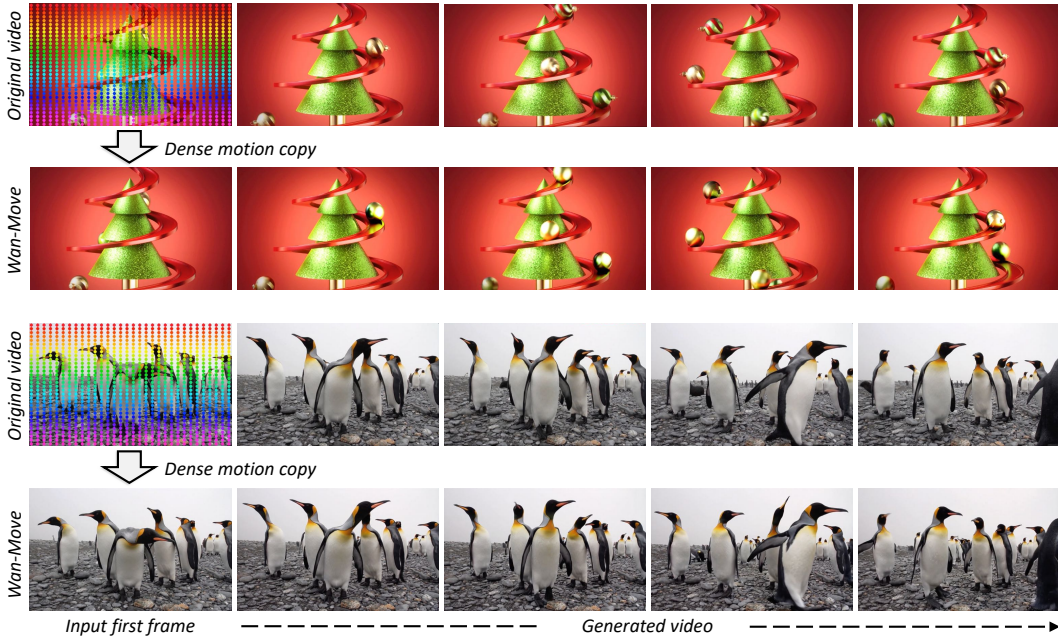


Figure 16: Wan-Move enables accurate video motion copy using dense point trajectories (*e.g.*, 1024 points). The synthesized video preserves high fidelity in both appearance and object-level motion alignment with the original video, even under complex environmental conditions.

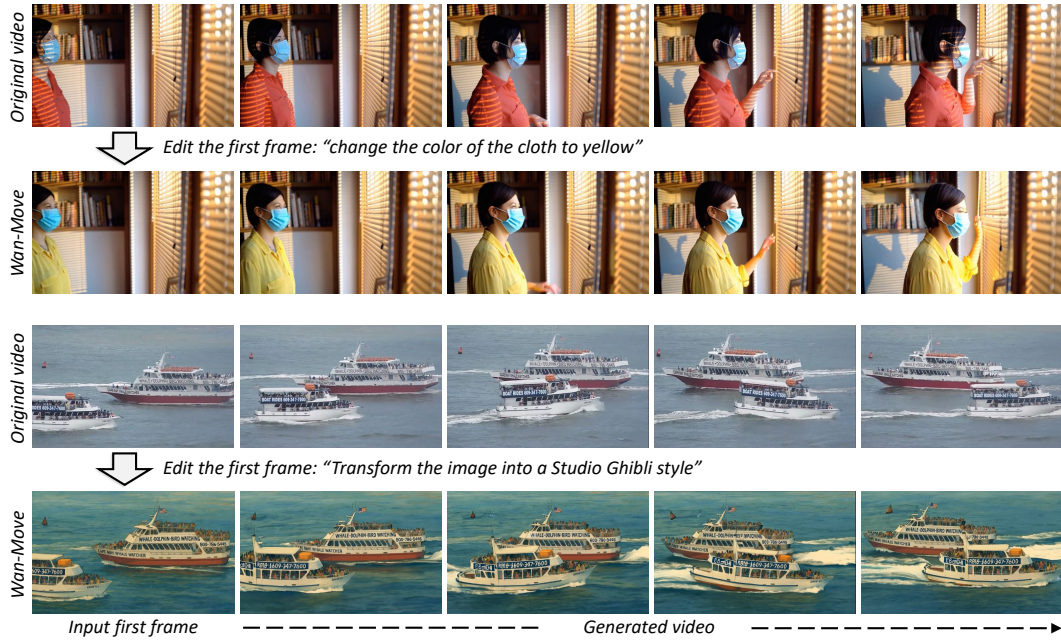


Figure 17: Wan-Move enables video editing through motion copy and additional image editing models. It first applies the image editing model (e.g., ControlNet [18], GPT-4o [83]) to modify the style or content of the first frame, then uses the original video’s motion trajectories to animate the edited image frame.



Figure 18: Wan-Move enables object 3D rotation by estimating depth-based positions, applying a rotation, and projecting the results to 2D.