FlexAM: Flexible Appearance-Motion Decomposition for Versatile Video Generation Control

Mingzhi Sheng[1,*] Zekai Gu[2,*] Peng Li[2] Cheng Lin[3]
Hao-Xiang Guo[4] Ying-Cong Chen[1,2,†] Yuan Liu[2,†]

[1]The Hong Kong University of Science and Technology (Guangzhou), Guangzhou, China
[2]The Hong Kong University of Science and Technology, Hong Kong SAR, China
[3]Macau University of Science and Technology, Macau SAR, China
[4]Tsinghua University, Beijing, China
msheng758@connect.hkust-gz.edu.cn, {zekai.gu,plibp}@connect.ust.hk,
chenglin@must.edu.mo, guohaoxiangxiang@gmail.com,
{yingcongchen,yuanly}@ust.hk

Abstract. Effective and generalizable control in video generation remains a significant challenge. While many methods rely on ambiguous or task-specific signals, we argue that a fundamental disentanglement of "appearance" and "motion" provides a more robust and scalable pathway. We propose FlexAM, a unified framework built upon a novel 3D control signal. This signal represents video dynamics as a point cloud, introducing three key enhancements: multi-frequency positional encoding to distinguish fine-grained motion, depth-aware encoding, and a flexible control signal for balancing precision and generalization. This representation allows FlexAM to effectively disentangle appearance and motion, enabling a wide range of tasks including I2V/V2V editing, camera control, and spatial object editing. Extensive experiments demonstrate that FlexAM achieves superior performance across all evaluated tasks. Codes are available at https://github.com/IGL-HKUST/FlexAM.

Keywords: video generation · video editing · generative rendering

1 Introduction

The field of video generation has witnessed remarkable breakthroughs in recent years. Beyond the typical text-to-video [13, 22, 43] and image-to-video [13, 33, 34] paradigms, controllable video generation [5, 8, 11, 18, 21, 23, 25, 26] has emerged as a significant area of research, spawning a multitude of downstream tasks. These include, for instance, reference-image-based [5, 7, 16, 24] and reference-audio-based [9, 35, 42] video synthesis. This rapid evolution underscores the dynamic and intricate nature of the field.

*Equal contribution. †Corresponding authors.

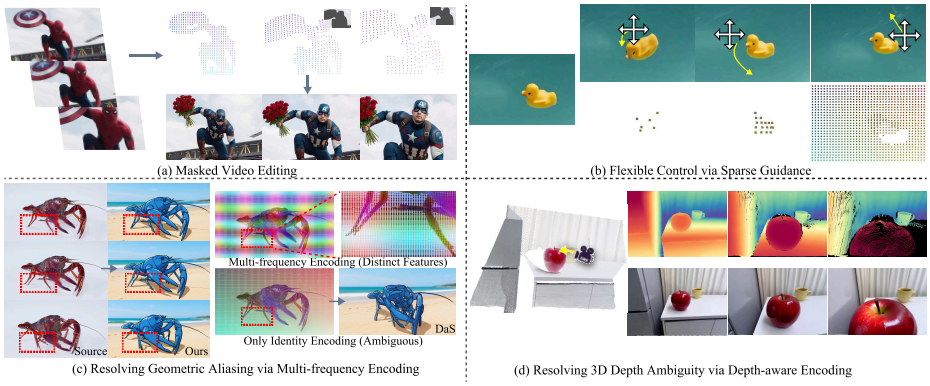

Fig. 1: FlexAM treats controllable video generation as a fundamental disentanglement of appearance and motion. By introducing a multi-attribute 3D point cloud as a robust motion representation, our framework enables: **(a)** masked video editing that transfers appearance while preserving motion; **(b)** flexible control balancing precision and generalization via varying point densities; **(c)** multi-frequency encoding to resolve geometric aliasing; and **(d)** depth-aware encoding to resolve 3D depth ambiguity.

To enhance controllability, researchers have explored decomposing videos into various elemental signals. Prior works have often focused on specific tasks by isolating particular signals, such as skeletons [6,42], optical flow [3,11], depth [8,21], camera motion [2, 4, 14, 36, 39], and object trajectories [10, 12, 20, 41, 44]. While some studies have attempted to combine these disparate modalities for compositional control [6,18,19,25,26], we argue that such approaches are often inefficient. They typically require bespoke training data and model designs for each new control signal, which complicates the pipeline and limits scalability. For instance, while VACE employs a unified architecture, it still requires distinct types of control inputs tailored to specific tasks, such as depth maps for structural control and human skeletons for pose transfer. This necessity for task-specific data preprocessing to extract various modalities—and the subsequent requirement for multi-condition training—complicates the data pipeline and limits the model's ability to generalize to unseen control types.

A more fundamental approach is to decompose video into "appearance" and "motion." We posit that this disentanglement is essential for robust controllable generation, as "motion" is a generalizable concept that can encapsulate human pose, camera movement, and object dynamics. Recent pioneering works have adopted this philosophy; for instance, Tora [44] models motion via 2D trajectories, while MagicMotion [20] utilizes object bounding boxes. More recently, Diffusion as Shader (DaS) [12] introduced a 3D point cloud as a more versatile motion signal.

However, current appearance-motion decomposition methods face four critical limitations: (1) **First-frame Dependency**: Reliance on initial-frame conditioning prevents these models from describing or maintaining the appearance

of regions that emerge later in the sequence. (2) **Inflexible Control**: Existing methods typically operate at a fixed granularity, forcing a trade-off between dense structural precision and sparse generative flexibility (Figure 1b). Most frameworks fail to adaptively bridge these two control regimes. (3) **Geometric Aliasing**: Existing motion representations cannot distinguish between the motions of closely adjacent elements, often leading to identity confusion and failure in scenarios like Figure 1c. (4) **Depth Ambiguity**: The lack of explicit depth information hinders accurate reasoning about spatial occlusions and object proximity relative to the camera, resulting in physically implausible dynamics.

To address these challenges, we propose FlexAM, a unified framework built upon a novel appearance-motion decomposition. For appearance representation, FlexAM moves beyond first-frame conditioning by accepting arbitrarily masked video (or individual frames) as the appearance condition. This allows it to seamlessly execute tasks including image-to-video synthesis, masked video-to-video editing, and arbitrary combinations thereof.

For motion representation, we construct a 3D-aware "motion video" composed of a set of trajectories, where the initial XYZ coordinates of the points define the trajectory colors to ensure temporal consistency. This motion video incorporates three core design elements: (1) As shown in Figure 1c, to achieve high precision and distinguish adjacent elements, the pixel color is determined by the point's 3D coordinates in the first frame's camera system, encoded using multi-frequency positional encoding across multiple layers. (2) To ensure the signal is depth-aware, we introduce additional explicit layers that represent the depth variations of these points over time, enabling the model to better reason about 3D structure and occlusion. (3) FlexAM supports varying point densities to represent motion, enabling it to flexibly balance precision and generalization to meet diverse task requirements. This is achieved through a training strategy that involves varying degrees of downsampling on the 3D control signal. In summary, our contributions are as follows:

1. We propose a comprehensive and general-purpose 3D control signal. By abstracting video motion into a colored dynamic point cloud and using precise multi-frequency positional encodings to distinguish adjacent elements, it explicitly represents the 3D spatio-temporal motion of elements. This enables our model to precisely control the motion and trajectories of elements during generation and editing, facilitating tasks such as camera control and object manipulation.
2. Beyond single-frame constraints, our method leverages masked video to provide a unified appearance condition for I2V and V2V tasks. By effectively disentangling appearance and motion, we enable flexible, independent editing of these components for any part of the input video or image.
3. We employ a training strategy that involves varying degrees of downsampling on the 3D control signal. This allows the model to learn to balance precision and generalization, keeping flexible and adapting to the demands of different tasks.

We conduct extensive experiments comparing FlexAM with baseline methods on tasks including controllable video generation and editing, camera control, and spatial object editing. The results demonstrate that FlexAM surpasses all baselines and achieves state-of-the-art (SOTA) performance across all evaluated tasks.

2 Related Work

2.1 Controllable Video Generation

Recent advancements in video diffusion models have achieved high-fidelity, temporally coherent video generation [22, 34, 43], spurring research into controllable video synthesis. Early efforts focused on specific modalities [4, 8, 11, 25, 39, 44], such as using poses or skeletons for human-centric animation [5,6,27], or audio for talking heads and sound-synchronized scenes [9, 35, 42]. More recent paradigms include trajectory-based control, where users specify 2D motion paths [10,41,44], and 3D-aware control, targeting cinematic camera or unified human-camera motion [2, 4, 14, 36, 39]. Despite this progress, most existing methods are designed for a single type of control and often rely on 2D signals, which can be ambiguous when representing complex 3D spatial dynamics.

2.2 Appearance Editing

Appearance editing aims to modify visual attributes, such as object identity or style, while preserving the original motion. A key challenge in this task is effectively decoupling the appearance and motion. Previous approaches use various strategies to achieve this. Video-P2P [25] and follow-your-motion [26] adapt an image model using cross-attention control. MagicEdit [21], Motion Prompting [10], and MotionCtrl [39] explicitly disentangle content, structure, and motion during training. However, these methods often rely on intermediate 2D representations. These 2D signals are usually lossy and lack 3D spatial context, leading to ambiguities and artifacts during complex motion. More comprehensive frameworks like VACE [18] aim for versatility but are not truly unified, requiring users to prepare a cumbersome and diverse set of conditions, such as depth maps, skeletons, or masks. FlexAM addresses these limitations by using the motion video as an explicit representation of the underlying dynamics. This single, 3D-aware signal avoids both the ambiguity of 2D representations and the complexity of multiple conditions. By conditioning on this unified signal, our model better preserves the target motion while propagating the new appearance to the masked regions.

2.3 Camera Control

Precise camera control is essential for cinematic storytelling and requires a strong understanding of 3D scene geometry. Recent works, however, often specialize in

distinct sub-tasks. For instance, CineMaster [36] and CameraCtrl [14] present 3D-aware frameworks for I2V/T2V generation with controllable trajectories, while ReCamMaster [2] focuses on V2V re-rendering. This specialization means I2V-focused models cannot perform V2V re-cinematography, and vice versa. Other methods have explored different avenues for 3D-aware control. DaS [12] aimed for versatile control, including camera manipulation, using a 3D tracking video. However, these methods suffer from limitations, including early 3D-aware models, which implicitly infer 3D structure, leading to potential depth ambiguity. The 3D tracking signal lacked explicit, time-varying depth information, which resulted in instabilities during precise camera control. This highlights the need for a framework that not only unifies I2V and V2V camera control but also relies on a 3D signal that is robust against geometric ambiguity.

2.4 Spatial Object Editing

Spatial object editing involves manipulating the 3D position, rotation, and trajectory of specific objects within a scene. Some 3D-aware methods approach this task by focusing on explicit geometry. GeoDiffuser [31] enables object manipulation by directly editing the scene's depth maps. While this provides explicit 3D control, such an approach can lack flexibility. DaS [12] introduced 3D tracking videos as a promising 3D-aware control signal. However, this earlier 3D representation lacked sufficient depth detail to resolve complex 3D relationships, leading to spatial inconsistencies. FlexAM addresses this limitation by proposing a more comprehensive and flexible 3D control signal, encoding richer spatial and depth information to achieve fine-grained, spatially-consistent object manipulation.

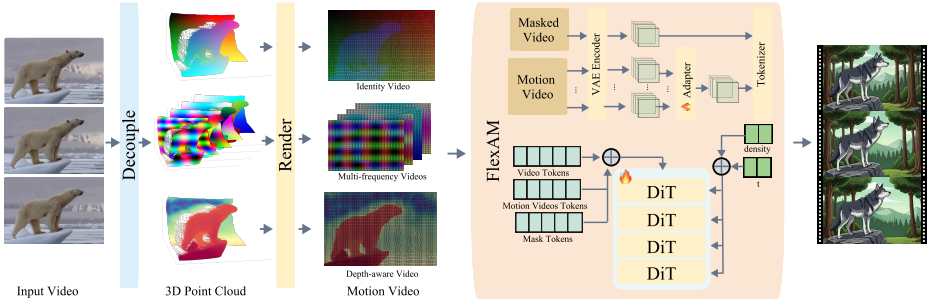

Fig. 2: The FlexAM pipeline. FlexAM decomposes controllable video generation into appearance and motion control. Given an input video, we construct a masked video as the appearance condition and render the 3D point cloud into an 18-channel motion video as the motion condition. These decoupled signals are encoded by the VAE, fused through the Adapter and tokenizer, and injected into the DiT backbone to generate the edited video while preserving the source motion dynamics.

3 Method

Our approach, FlexAM, is built upon the key insight of disentangling video into two fundamental components: appearance and motion. This separation is pivotal, as it permits the independent editing of either part. FlexAM can then utilize these edited representations as control signals to the generative model, enabling targeted adjustments to the video's appearance or motion while keeping another part of the video consistent. We first detail our representations for appearance (Sec. 3.1) and our novel motion video (Sec. 3.2), followed by the architecture that integrates them (Sec. 3.3).

3.1 Appearance Control Signal Representation

We define the appearance broadly to accommodate diverse generation and editing tasks. Unlike methods conditioned solely on a single first frame, FlexAM accepts a partially masked video $\mathbf{V}_{\text{masked}} \in \mathbb{R}^{T\times H\times W\times 3}$ alongside a corresponding binary edit mask $\mathbf{f}_{\text{mask}}$ as the appearance condition. In $\mathbf{V}_{\text{masked}}$, unedited regions retain their original pixel values, while regions targeted for editing are filled with a gray value of 127. The binary mask $\mathbf{f}_{\text{mask}} \in \mathbb{R}^{T\times H\times W\times 1}$ explicitly delineates these targeted areas, where a value of 1 exactly aligns with the 127-filled regions in $\mathbf{V}_{\text{masked}}$. $\mathbf{f}_{\text{mask}}$ serves as an additional condition to apply distinct denoising timesteps for edited versus preserved regions, ensuring accurate modification while strictly maintaining unedited contexts. This flexible representation allows the model to naturally handle tasks ranging from image-to-video, where all frames except the first are masked, to local video-to-video editing, all within a unified framework.

3.2 Motion Control Signal Representation

Inspired by the 3D point cloud representation for video dynamics [12], we propose a multi-attribute motion representation. Unlike prior single-dimensional tracking signals, our points are endowed with high-dimensional attributes designed for precise, depth-aware, and flexible control. As shown in Figure 2, this attributed point cloud is rendered into an 18-channel temporal signal, which we term the motion video, to serve as our unified motion control interface. The 18 channels are organized as six RGB-like streams: one identity stream, four multi-frequency streams, and one depth-aware stream.

Specifically, the motion video describes the dynamics of a set of 3D points $\{\mathbf{p}_i(t) \in \mathbb{R}^3\}$, where t is the frame index. Each point $\mathbf{p}_i(t) = (x_i(t), y_i(t), z_i(t))$ corresponds to a specific spatial region. To create the final control signal, these points are projected onto the 2D screen space. At each projected location $(u_i(t), v_i(t))$, we populate attribute vector $\mathbf{A}_i(t)$. All locations without projected points are set to zero. This attribute vector $\mathbf{A}_i(t)$ is composed of three components, strategically designed to be precise, depth-aware, and flexible.

Precise Positional Encoding To precisely distinguish adjacent points and capture fine-grained spatial relationships, we encode the initial 3D coordinates.

This includes an identity positional encoding $\mathbf{f}_{\text{identity}} \in \mathbb{R}^3$, which stores the normalized initial coordinates $(x'_i(0), y'_i(0), z'_i(0))$, where z' represents inverse depth. Additionally, we compute a multi-frequency positional encoding $\mathbf{f}_{\text{freq}} \in \mathbb{R}^{12}$ by concatenating four RGB-like streams, where each stream applies $\gamma_l(v) = \cos(2^l \pi v)$, $l \in \{0, 1, 2, 3\}$, to the three normalized initial coordinates.

Depth-aware Representation To explicitly encode 3D structure and resolve depth ambiguity, we include a depth-aware encoding $\mathbf{f}_{\text{depth}} \in \mathbb{R}^3$. This component represents the current normalized depth $z''_i(t)$ at time t, which is mapped to an RGB vector using the Spectral colormap $\mathcal{S}(\cdot)$. This offers richer gradients than grayscale, enhancing sensitivity to fine-grained depth variations and accelerating convergence.

Control Flexibility and Masked Editing During 2D rendering, points projected onto unedited regions ($\mathbf{f}_{\text{mask}} = 0$) are zeroed out, confining the motion signal to the target areas. For the edited regions, the full attribute vector $\mathbf{A}_i(t)$ at the projected location $(u_i(t), v_i(t))$ is the concatenation of these components:

$$\mathbf{A}_i(t) = \left(\mathbf{f}^{(i)}_{\text{identity}}, \mathbf{f}^{(i)}_{\text{freq}}, \mathbf{f}^{(i,t)}_{\text{depth}}\right) \in \mathbb{R}^{18}, \tag{1}$$

where

$$\mathbf{f}^{(i)}_{\text{identity}} = (x'_i(0), y'_i(0), z'_i(0)) \in \mathbb{R}^3, \tag{2}$$

$$\mathbf{f}^{(i)}_{\text{freq}} = \text{Concat}^3_{l=0}\left(\gamma_l(x'_i(0)), \gamma_l(y'_i(0)), \gamma_l(z'_i(0))\right) \in \mathbb{R}^{12}, \tag{3}$$

$$\mathbf{f}^{(i,t)}_{\text{depth}} = \mathcal{S}(z''_i(t)) \in \mathbb{R}^3. \tag{4}$$

Sparse-to-Dense Flexibility. A key feature of the motion video is its density flexibility. The signal can represent motion densely or sparsely, determined by the number of points $\{\mathbf{p}_i(t)\}$ used. This allows FlexAM flexibly to balance precision and generalization. As detailed in Sec. 3.3, we train the model to handle this variability by stochastically down-sampling the point clouds during training.

3.3 Architecture and Training Integration

To inject our decoupled signals into a pre-trained latent video diffusion model (e.g., Wan2.2Fun 5B Control), we encode both appearance and motion conditions into the backbone's latent space. Specifically, the appearance condition $\mathbf{V}_{\text{masked}}$, together with its binary mask $\mathbf{f}_{\text{mask}}$, is encoded into appearance latents, while the 18-channel motion video is first partitioned into six RGB-like streams and then encoded independently using the same pre-trained VAE. Since the diffusion backbone operates in the VAE latent space, this shared encoding scheme makes the motion latents directly compatible with the denoising network, without introducing an additional task-specific motion encoder. A lightweight CNN-based Adapter then fuses the motion latents into a unified motion control signal $\mathbf{C}$. To support flexible sparse-to-dense control, we introduce a density embedding for the point density d. Following the design of the diffusion timestep embedding, d is mapped to an embedding vector and injected into the denoising network, explicitly indicating the spatial density of the motion control signal. Detailed training configurations are provided in the supplementary material.

3.4 Controllable Video Generation and Editing

In this section, we elaborate on how FlexAM achieves controllable video generation and video editing.

Appearance Editing. Appearance editing aims to modify the visual style or specific regions of an input video (e.g., style transfer or character replacement) while preserving its original motion dynamics. We achieve this by leveraging FlexAM's decoupled representation: extracting the source dynamics as the motion interface, and using the masked video to guide the appearance generation. Specifically, we first detect 3D points and their trajectories using DELTA [28] from the source video to generate a motion video. Next, we construct the appearance condition by masking the editable regions (filling them with gray) and prepending a repainted first frame to the sequence. Finally, FlexAM integrates these inputs to transfer the target appearance to the masked regions according to the preserved dynamics, generating the target video.

Camera Control. This task aims to synthesize videos from static images or existing videos that follow a user-specified camera trajectory. Our core idea is to construct a 3D or 4D dynamic point cloud of the scene and re-project these points from the novel camera views to form the motion video.

For the image-to-video generation task with a specific camera trajectory, we first estimate the depth map of the input image, which is then encoded using $\mathbf{A}_i(t)$. These points are then projected onto the given camera trajectory to construct a motion video, allowing FlexAM to control the camera motion precisely.

For the video-to-video camera control task, we first detect 3D points and trajectories and estimate the camera parameters and trajectory of the input video. Then we combine them to build a 4D dynamic point cloud. Subsequently, these points are projected onto the new camera trajectory to construct a new motion video for FlexAM to achieve high-precision camera re-cinematography.

Spatial Object Editing. Spatial object editing involves manipulating the 3D position, rotation, and scale of specific objects within a scene while maintaining geometric plausibility. Our framework accomplishes this by isolating the 3D point cloud of the target object, applying explicit geometric transformations in 3D space, and rendering the updated scene geometry into a new motion video. Given an input image, we first estimate its depth map and segment the target object. We then apply the desired geometric manipulations directly to the point cloud of this isolated object. Finally, utilizing the depth-aware encoding attributes to avoid depth ambiguity, we project the manipulated point cloud back to the image plane to construct the motion video for the final video generation.

4 Experiment

We conduct experiments across three tasks, appearance editing, camera control, and spatial object editing, to demonstrate the versatility of FlexAM in video editing and controlling video generation.

4.1 Appearance Editing

We primarily compared FlexAM and the baseline on two subtasks: motion transfer and partial editing. For a fair comparison, all models use the same reference images generated by Qwen Image Edit [40] as the unified appearance guidance.
Motion Transfer. *Baselines.* We compare FlexAM with representative motion transfer methods, including DaS [12], VACE [18], and Wan2.2Fun (5B) Control [1]. Specifically, VACE and Wan2.2Fun are conditioned on depth maps, while DaS utilizes 3D tracking videos. In contrast, our method is conditioned on the proposed motion video, which provides a more comprehensive 3D-aware representation.
Metrics. We evaluate text-video alignment (Tex-Ali) and temporal consistency (Tem-Con) using CLIP [15,29]. Additionally, we employ VBench++ [17] to comprehensively assess subject/background consistency (*i2v_sub*, *i2v_bg*), motion smoothness (*mot_sm*), and aesthetic/imaging quality (*aes_ql*, *img_ql*).
Results. As shown in Table 1, FlexAM exhibits overall superior performance. It significantly outperforms VACE in subject and background fidelity (0.99 vs. 0.86), alongside a notable improvement in aesthetic scores (*aes_ql*: 0.64). Qualitatively (Figure 3), VACE repaints the reference image and loses character identity, Wan2.2Fun distorts facial features due to over-alignment with depth, and

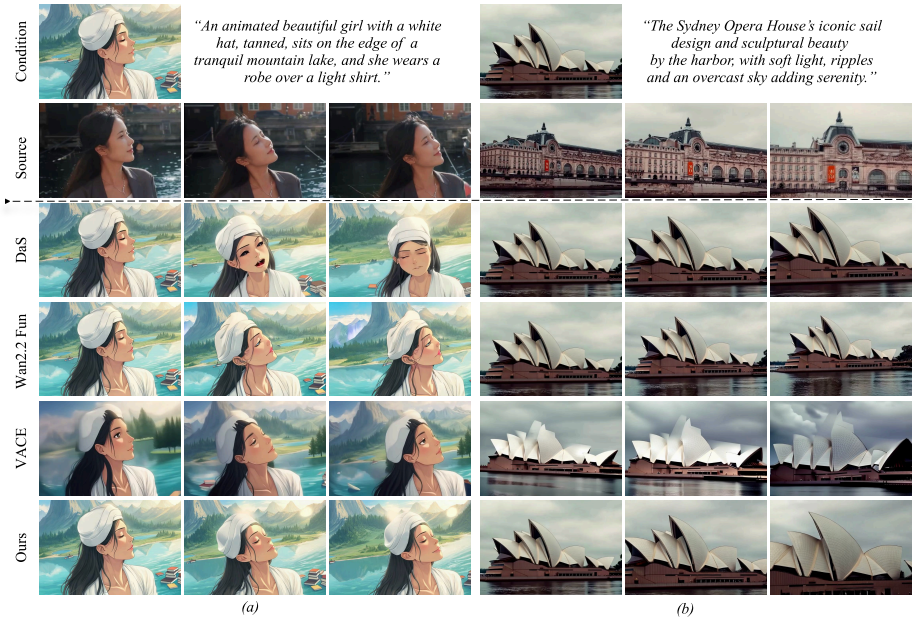

Fig. 3: Qualitative comparison on motion transfer between our method, DaS, Wan2.2 Fun, and VACE. We compare the results of different methods in transferring the human motion from the Source to a new appearance. Compared to the baseline, our method accurately transfers the motion.

Table 1: Quantitative comparison of Motion Transfer. FlexAM achieves the highest text-video alignment, temporal consistency, and visual aesthetics while maintaining high subject and background fidelity.

	CLIP Metrics		VBench++ Metrics				
Method	Tex-Ali ↑	Tem-Con ↑	i2v_sub ↑	i2v_bg ↑	mot_sm ↑	aes_ql ↑	img_ql ↑
DaS [12]	32.14	0.968	0.98	0.98	0.98	0.60	**0.68**
Wan2.2 Fun [1]	32.39	0.971	0.98	**0.99**	0.98	0.61	**0.68**
VACE [18]	32.38	0.970	0.86	0.86	**0.99**	0.57	0.63
FlexAM (Ours)	**32.55**	**0.976**	**0.99**	**0.99**	**0.99**	**0.64**	**0.68**

DaS fails to reconstruct the pose. In contrast, FlexAM accurately transfers motion with superior visual harmony and temporal coherence.

Partial Editing. This task targets modifying the appearance of specific regions while preserving their original motion trajectories, a capability essential for applications such as virtual try-on and cinematic customization.

Baselines. We compare FlexAM with VACE on both foreground and background editing. Specifically, we use SAM2 [30] to mask the foreground or background of the input videos. The models are expected to accurately transfer the appearance from the reference images to the masked regions, seamlessly integrate it with the original motion, and faithfully preserve the unmasked areas.

Results. As shown in Figure 4(a), during foreground editing, VACE frequently redraws the first-frame appearance and exhibits temporal drift at limb boundaries. In contrast, FlexAM better follows the reference rhythm while maintaining more

Fig. 4: Qualitative comparison on foreground and background editing. We transfer the motion from the source videos while replacing the foreground/ background appearance using the reference prompt/image. (a) Replace bear with Godzilla; compared to VACE, our method better follows the reference poses and preserves identity and color details. (b) Airplane wing over mountains at sunset; While VACE maintains foreground consistency but loses background motion, our method integrates the input video's background motion with the new appearance, preserving coherent dynamics.

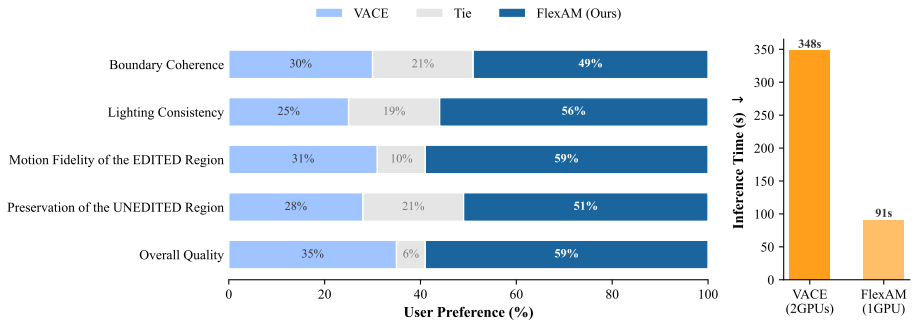

Fig. 5: Quantitative comparison on partial editing. We compare the user preference (%) and inference time between our method and VACE. (Left) FlexAM is consistently preferred across all dimensions. (Right) Compared to VACE, FlexAM achieves a significant speedup, demonstrating superior efficiency.

stable identity and local details. For background editing (Figure 4(b)), VACE fails to preserve background motion, while FlexAM integrates the reference background appearance with the source background dynamics and maintains foreground consistency. We further conduct a randomized A/B user study with 27 participants on 8 video pairs. As shown in Figure 5 (Left), FlexAM is consistently preferred, particularly in lighting consistency (56% vs. 25%) and preservation of the unedited region (51% vs. 28%). Efficiency evaluation in Figure 5 (Right) further highlights our advantages: despite a 14B baseline, our 5B architecture requires only a single GPU and significantly reduces inference time from 348s to 91s. Additional quantitative results using VBench++ [17] and VE-Bench [32] are provided in the supplementary material.

4.2 Camera Control

We evaluate camera control to examine whether the proposed general-purpose motion representation can achieve competitive geometric precision against task-specific camera-control models. We conduct quantitative evaluation on image-to-video (I2V) camera control for trajectory accuracy, and further provide qualitative results on the more challenging video-to-video (V2V) camera re-rendering setting.

Baselines. We compare FlexAM with three representative camera-control methods: DaS [12], Wan2.2 Fun Control Camera [1], and ReCamMaster [2]. All methods are given the same input frame or video and the same target camera trajectory whenever applicable. Since Wan2.2 Fun Control Camera only supports image-to-video generation, it is excluded from the qualitative V2V comparison.

Metrics. To evaluate camera-control accuracy, we measure the discrepancy between the target camera trajectory and the trajectory estimated from the generated video. Specifically, for each generated video, we use Pi3 [38] to reconstruct the relative camera pose of each frame with respect to the first frame. We then

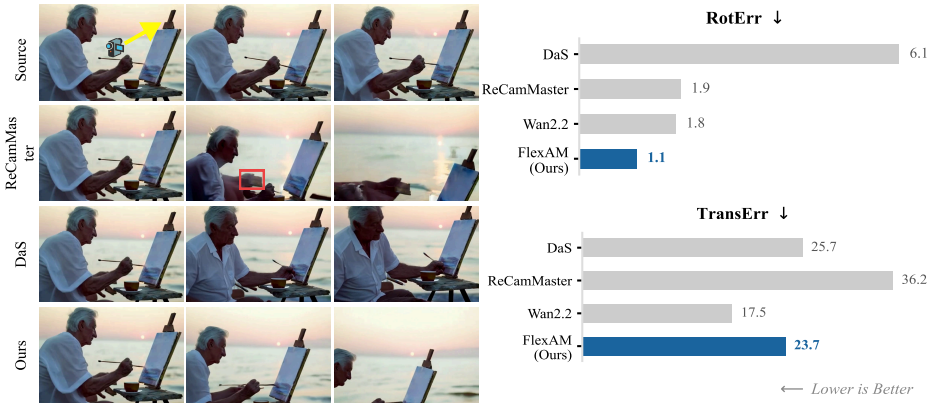

Fig. 6: Camera control comparison. (Left) Qualitative V2V camera re-rendering with a prescribed "pan up-right" trajectory. (Right) Quantitative I2V camera-control evaluation on RealEstate10K.

extract the normalized rotation quaternion and translation vector, and compute the rotation and translation errors as:

$$\text{RotErr} = \arccos\Big(\frac{1}{T-1}\sum_{i=2}^{T}\langle \mathbf{q}^i_{\text{gen}}, \mathbf{q}^i_{\text{gt}}\rangle\Big), \quad \text{TransErr} = \arccos\Big(\frac{1}{T-1}\sum_{i=2}^{T}\langle \mathbf{t}^i_{\text{gen}}, \mathbf{t}^i_{\text{gt}}\rangle\Big). \tag{5}$$

All methods are evaluated using the same Pi3-based pose estimation protocol and compared against the same ground-truth trajectories.

Results. For I2V task, we evaluate all methods on 100 randomly sampled sequences from RealEstate10K [45]. Each method receives the initial frame and the ground-truth camera trajectory as input, and we compare the estimated camera poses of the generated videos against the ground-truth trajectory. As shown in Figure 6 (Right), FlexAM achieves the lowest rotation error among all methods. For translation error, the task-specific Wan2.2 Fun Control Camera obtains a slightly lower score due to its explicit pose-conditioned supervision, while FlexAM remains competitive without relying on specialized pose-control modules.

For V2V task, we perform a qualitative comparison using an internet video with a prescribed "pan up-right" trajectory. As shown in Figure 6 (Left), ReCamMaster [2] produces rendering artifacts and trajectory deviations, while DaS [12] struggles to align the generated video with the target camera motion. In contrast, FlexAM better preserves the source content while following the prescribed camera trajectory, demonstrating its robustness in video re-rendering.

4.3 Spatial Object Editing

For spatial object editing, we adopt the SAM2 [30] and MoGe [37] to get the object points. We compare FlexAM with DaS and GeoDiffuser [31] in two kinds

Method	CLIP Score ↑
GeoDiffuser [31]	0.9110
DaS [12]	0.9437
FlexAM (Ours)	**0.9536**

Fig. 7: Results for object manipulation. *Left:* Qualitative comparison of object translation (top) and rotation (bottom). *Right:* Quantitative evaluation using CLIP Scores.

of tasks: translation and rotation. Since GeoDiffuser is an image-based manipulation method, we compare its results with the final video frame generated by FlexAM and DaS, and use the CLIP [29] to evaluate semantic alignment between the generated results and text prompts.

Results. Figure 7 demonstrates that FlexAM achieves accurate object manipulation and produces photorealistic videos with strong multiview consistency. Compared to baselines, FlexAM exhibits better occlusion reasoning and editing capability, while also achieving the highest CLIP score. Further quantitative evaluation of spatial object editing is provided in the supplementary material.

4.4 Ablation Study

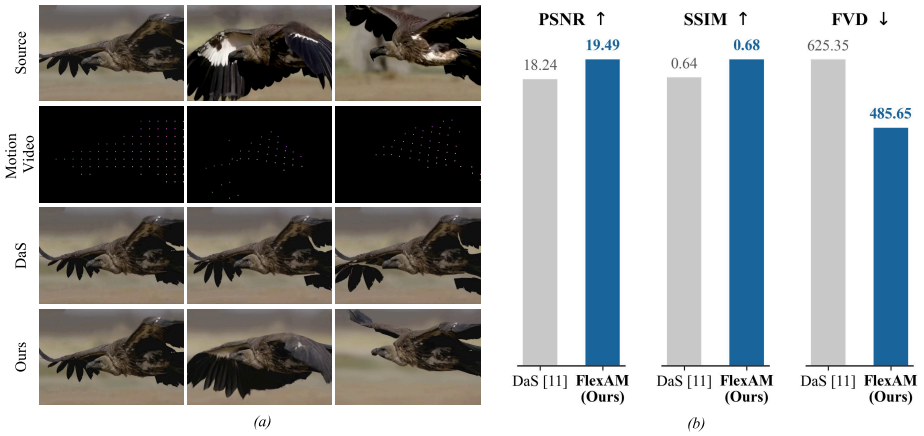

Fig. 8: Qualitative and quantitative comparison under sparse motion signals. (a) Qualitatively, FlexAM maintains accurate motion and appearance even under sparse control, whereas the baseline DaS [12] exhibits motion drift and shape distortion. (b) Quantitatively, FlexAM consistently outperforms DaS across all three reconstruction metrics.

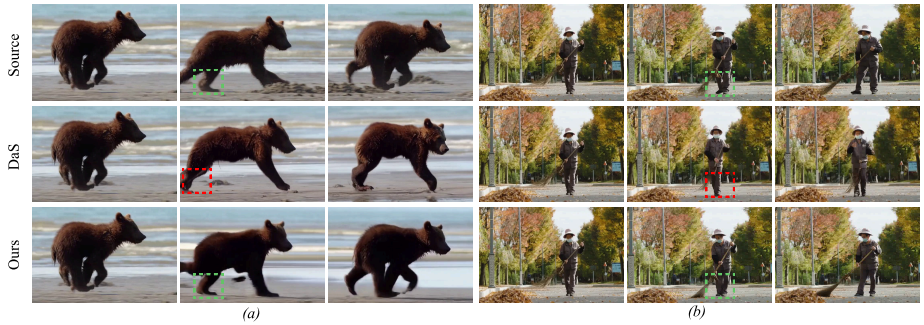

Fig. 9: Qualitative comparison of precise positional encoding. FlexAM adds multi-frequency encodings to keep nearby points separable and motions consistent. The areas marked with color boxes demonstrate FlexAM's better ability in reconstructing dynamic details compared to the baseline.

We evaluate the core components of our motion video by comparing FlexAM against a baseline that utilizes only vanilla 3D Tracking Video. Specifically, we analyze the necessity of our multi-attribute design—including control flexibility, precise positional encoding, and depth-aware representation—in maintaining motion precision and spatial consistency.

Control Flexibility. We evaluate robustness to sparsity via a video reconstruction task, requiring models to recover original videos using only the first frame and extracted sparse motion signals. As presented in Figure 8, FlexAM outperforms the baseline DaS [12] in quantitative reconstruction fidelity (PSNR, SSIM, FVD). Qualitatively, our method maintains accurate motion control and source appearance even under sparse conditions, demonstrating FlexAM's capability to balance flexibility and precision across varying control densities.

Precise Positional Encoding. As shown in Figure 9, the baseline suffers from motion aliasing when screen-space trajectories overlap. This ambiguity causes the model to confuse or merge adjacent points, such as the bear's left and right feet. Our multi-frequency positional encoding resolves this, enabling the model to preserve limb identity throughout the gait cycle. This yields temporally consistent contacts and plausible kinematics, in clear contrast to the baseline which often swaps the feet.

Depth-aware Encoding. Figure 10 illustrates the necessity of video depth as a conditioning factor. Encoding 3D point clouds solely based on 2D screen-space coordinates introduces ambiguity, as distinct 3D spatial movements can project to identical 2D results. Consequently, baselines relying on static depth fail to distinguish between scale variations and actual depth movements, causing physically implausible occlusions. In contrast, FlexAM incorporates time-varying depth to accurately recover 3D trajectories and ensure physically consistent dynamics.

Fig. 10: Qualitative comparison of depth-aware encoding. Without time-varying depth, scale changes can be mistaken for depth motion, leading to implausible occlusions. FlexAM better preserves camera-object spatial relationships and produces more physically plausible dynamics.

Fig. 11: Robustness and failure cases. FlexAM handles moderate geometric mismatch but degrades under severe mismatch.

5 Limitations and Conclusions

Limitations. FlexAM still faces several limitations. First, its control precision depends on the quality of the estimated 3D trajectories; inaccurate tracking, heavy occlusion or fast motion may affect fine-grained motion control. Second, although FlexAM is robust to moderate geometric mismatch between the source motion and target appearance, severe mismatch can lead to implausible deformation, as shown in Figure 11. Third, due to limited lighting diversity in the current training data, occasional illumination mismatches may occur during generation. Constructing more diverse lighting-aware datasets and scaling to larger video data are important directions for improving robustness.

Conclusions. We introduce FlexAM, a unified framework for controllable video generation that encodes scene dynamics as a dynamic 3D point cloud augmented with multi-frequency positional encodings, depth-aware encoding and flexible control. This control representation cleanly disentangles appearance from motion, enabling a single model to handle I2V/V2V editing, camera trajectory control, and object manipulation. Extensive experiments show consistent improvements over strong baselines, indicating that FlexAM achieves high precision without sacrificing generalization.

References

1. AIGC-Apps: VideoX-Fun: A video generation pipeline for diffusion transformer (2026), https://github.com/aigc-apps/VideoX-Fun, Accessed: 30 June 2026
2. Bai, J., Xia, M., Fu, X., Wang, X., Mu, L., Cao, J., Liu, Z., Hu, H., Bai, X., Wan, P., Zhang, D.: ReCamMaster: Camera-controlled generative rendering from a single video. arXiv preprint arXiv:2503.11647 (2025)
3. Burgert, R., Xu, Y., Xian, W., Pilarski, O., Clausen, P., He, M., Ma, L., Deng, Y., Li, L., Mousavi, M., Ryoo, M., Debevec, P., Yu, N.: Go-with-the-flow: Motion-controllable video diffusion models using real-time warped noise. In: CVPR (2025)
4. Cao, C., Zhou, J., Li, S., Liang, J., Yu, C., Wang, F., Xue, X., Fu, Y.: Uni3C: Unifying precisely 3d-enhanced camera and human motion controls for video generation. arXiv preprint arXiv:2504.14899 (2025)
5. Chen, L., Ma, T., Liu, J., Li, B., Chen, Z., Liu, L., He, X., Li, G., He, Q., Wu, Z.: HuMo: Human-centric video generation via collaborative multi-modal conditioning. arXiv preprint arXiv:2509.08519 (2025)
6. Cheng, G., Gao, X., et al.: Wan-Animate: Unified character animation and replacement with holistic replication. arXiv preprint arXiv:2509.14055 (2025)
7. Deng, Y., Yin, Y., Guo, X., Wang, Y., Fang, J.Z., Yuan, S., Yang, Y., Wang, A., Liu, B., Huang, H., Ma, C.: MAGREF: Masked guidance for any-reference video generation with subject disentanglement. arXiv preprint arXiv:2505.23742 (2025)
8. Feng, R., Weng, W., Wang, Y., Yuan, Y., Bao, J., Luo, C., Chen, Z., Guo, B.: Ccedit: Creative and controllable video editing via diffusion models. arXiv preprint arXiv:2309.16496 (2024)
9. Gao, X., Hu, L., et al.: Wan-S2V: Audio-driven cinematic video generation. arXiv preprint arXiv:2508.18621 (2025)
10. Geng, D., Herrmann, C., Hur, J., Cole, F., Zhang, S., Pfaff, T., Lopez-Guevara, T., Doersch, C., Aytar, Y., Rubinstein, M., Sun, C., Wang, O., Owens, A., Sun, D.: Motion prompting: Controlling video generation with motion trajectories. arXiv preprint arXiv:2412.02700 (2025)
11. Geyer, M., Bar Tal, O., Bagon, S., Dekel, T.: Tokenflow: Consistent diffusion features for consistent video editing. In: International Conference on Learning Representations. vol. 2024, pp. 1608–1620 (2024)
12. Gu, Z., Yan, R., Lu, J., Li, P., Dou, Z., Si, C., Dong, Z., Liu, Q., Lin, C., Liu, Z., et al.: Diffusion as shader: 3d-aware video diffusion for versatile video generation control. In: Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers. pp. 1–12 (2025)
13. HaCohen, Y., Chiprut, N., et al.: Ltx-video: Realtime video latent diffusion. arXiv preprint arXiv:2501.00103 (2024)
14. He, H., Xu, Y., Guo, Y., Wetzstein, G., Dai, B., Li, H., Yang, C.: CameraCtrl: Enabling camera control for text-to-video generation. arXiv preprint arXiv:2404.02101 (2025)
15. Hessel, J., Holtzman, A., Forbes, M., Bras, R.L., Choi, Y.: Clipscore: A reference-free evaluation metric for image captioning. arXiv preprint arXiv:2104.08718 (2022)
16. Hu, T., Yu, Z., Zhou, Z., Liang, S., Zhou, Y., Lin, Q., Lu, Q.: HunyuanCustom: A multimodal-driven architecture for customized video generation. arXiv preprint arXiv:2505.04512 (2025)
17. Huang, Z., Zhang, F., Xu, X., He, Y., Yu, J., Dong, Z., Ma, Q., Chanpaisit, N., Si, C., Jiang, Y., Wang, Y., Chen, X., Chen, Y.C., Wang, L., Lin, D., Qiao, Y., Liu, Z.: VBench++: Comprehensive and versatile benchmark suite for video generative

models. IEEE Transactions on Pattern Analysis and Machine Intelligence (2025). https://doi.org/10.1109/TPAMI.2025.3633890
18. Jiang, Z., Han, Z., Mao, C., Zhang, J., Pan, Y., Liu, Y.: VACE: All-in-one video creation and editing. arXiv preprint arXiv:2503.07598 (2025)
19. Ju, X., Wang, T., Zhou, Y., Zhang, H., Liu, Q., Zhao, N., Zhang, Z., Li, Y., Cai, Y., Liu, S., Pakhomov, D., Lin, Z., Kim, S.Y., Xu, Q.: Editverse: Unifying image and video editing and generation with in-context learning. arXiv preprint arXiv:2509.20360 (2025)
20. Li, Q., Xing, Z., Wang, R., Zhang, H., Dai, Q., Wu, Z.: Magicmotion: Controllable video generation with dense-to-sparse trajectory guidance. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 12112–12123 (October 2025)
21. Liew, J.H., Yan, H., Zhang, J., Xu, Z., Feng, J.: Magicedit: High-fidelity and temporally coherent video editing. arXiv preprint arXiv:2308.14749 (2023)
22. Lin, B., Ge, Y., Cheng, X., Li, Z., Zhu, B., Wang, S., He, X., Ye, Y., et al.: Open-Sora Plan: Open-source large video generation model. arXiv preprint arXiv:2412.00131 (2024)
23. Ling, P., Bu, J., Zhang, P., Dong, X., Zang, Y., Wu, T., Chen, H., Wang, J., Jin, Y.: Motionclone: Training-free motion cloning for controllable video generation. arXiv preprint arXiv:2406.05338 (2024)
24. Liu, L., Ma, T., Li, B., Chen, Z., Liu, J., Li, G., Zhou, S., He, Q., Wu, X.: Phantom: Subject-consistent video generation via cross-modal alignment. arXiv preprint arXiv:2502.11079 (2025)
25. Liu, S., Zhang, Y., Li, W., Lin, Z., Jia, J.: Video-p2p: Video editing with cross-attention control. arXiv preprint arXiv:2303.04761 (2023)
26. Ma, Y., Liu, Y., Zhu, Q., Yang, A., Feng, K., Zhang, X., Li, Z., Han, S., Qi, C., Chen, Q.: Follow-your-motion: Video motion transfer via efficient spatial-temporal decoupled finetuning. arXiv preprint arXiv:2506.05207 (2025)
27. Meng, R., Zhang, X., Li, Y., Ma, C.: EchoMimicV2: Towards striking, simplified, and semi-body human animation. arXiv preprint arXiv:2411.10061 (2025)
28. Ngo, T.D., Zhuang, P., Gan, C., Kalogerakis, E., Tulyakov, S., Lee, H.Y., Wang, C.: DELTA: Dense efficient long-range 3d tracking for any video. arXiv preprint arXiv:2410.24211 (2025)
29. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., Sutskever, I.: Learning transferable visual models from natural language supervision. arXiv preprint arXiv:2103.00020 (2021)
30. Ravi, N., Gabeur, V., Hu, Y.T., Hu, R., Ryali, C., Ma, T., Khedr, H., Rädle, R., Rolland, C., Gustafson, L., Mintun, E., Pan, J., Alwala, K.V., Carion, N., Wu, C.Y., Girshick, R., Dollár, P., Feichtenhofer, C.: SAM 2: Segment anything in images and videos. arXiv preprint arXiv:2408.00714 (2024)
31. Sajnani, R., Vanbaar, J., Min, J., Katyal, K., Sridhar, S.: GeoDiffuser: Geometry-based image editing with diffusion models. arXiv preprint arXiv:2404.14403 (2025)
32. Sun, S., Liang, X., Fan, S., Gao, W., Gao, W.: VE-Bench: Subjective-aligned benchmark suite for text-driven video editing quality assessment. arXiv preprint arXiv:2408.11481 (2024)
33. Team, M.L., Bayan, Li, B., Lei, B., Wang, B., et al.: Longcat-flash technical report. arXiv preprint arXiv:2509.01322 (2025)
34. Wan, T., Wang, A., Ai, B., Wen, B., Mao, C., et al.: Wan: Open and advanced large-scale video generative models. arXiv preprint arXiv:2503.20314 (2025)

35. Wang, M., Wang, Q., Jiang, F., Fan, Y., Zhang, Y., Qi, Y., Zhao, K., Xu, M.: FantasyTalking: Realistic talking portrait generation via coherent motion synthesis. arXiv preprint arXiv:2504.04842 (2025)
36. Wang, Q., Luo, Y., Shi, X., Jia, X., Lu, H., Xue, T., Wang, X., Wan, P., Zhang, D., Gai, K.: CineMaster: A 3D-aware and controllable framework for cinematic text-to-video generation. arXiv preprint arXiv:2502.08639 (2025)
37. Wang, R., Xu, S., Dai, C., Xiang, J., Deng, Y., Tong, X., Yang, J.: MoGe: Unlocking accurate monocular geometry estimation for open-domain images with optimal training supervision. arXiv preprint arXiv:2410.19115 (2025)
38. Wang, Y., Zhou, J., Zhu, H., Chang, W., Zhou, Y., Li, Z., Chen, J., Pang, J., Shen, C., He, T.: π^3: Permutation-equivariant visual geometry learning. arXiv preprint arXiv:2507.13347 (2025)
39. Wang, Z., Yuan, Z., Wang, X., Chen, T., Xia, M., Luo, P., Shan, Y.: MotionCtrl: A unified and flexible motion controller for video generation. arXiv preprint arXiv:2312.03641 (2024)
40. Wu, C., Li, J., Zhou, J., Lin, J., Gao, K., Yan, K., ming Yin, S., Bai, S., Xu, X., Chen, Y., Chen, Y., Tang, Z., Zhang, Z., Wang, Z., Yang, A., Yu, B., Cheng, C., Liu, D., Li, D., Zhang, H., Meng, H., Wei, H., Ni, J., Chen, K., Cao, K., Peng, L., Qu, L., Wu, M., Wang, P., Yu, S., Wen, T., Feng, W., Xu, X., Wang, Y., Zhang, Y., Zhu, Y., Wu, Y., Cai, Y., Liu, Z.: Qwen-Image technical report. arXiv preprint arXiv:2508.02324 (2025)
41. Wu, W., Li, Z., Gu, Y., Zhao, R., He, Y., Zhang, D.J., Shou, M.Z., Li, Y., Gao, T., Zhang, D.: Draganything: Motion control for anything using entity representation. arXiv preprint arXiv:2403.07420 (2024)
42. Yang, S., Kong, Z., Gao, F., Cheng, M., Liu, X., Zhang, Y., Kang, Z., Luo, W., Cai, X., He, R., Wei, X.: InfiniteTalk: Audio-driven video generation for sparse-frame video dubbing. arXiv preprint arXiv:2508.14033 (2025)
43. Yang, Z., Teng, J., Zheng, W., Ding, M., Huang, S., Xu, J., Yang, Y., Hong, W., Zhang, X., Feng, G., Yin, D., Zhang, Y., Wang, W., Cheng, Y., Xu, B., Gu, X., Dong, Y., Tang, J.: CogVideoX: Text-to-video diffusion models with an expert transformer. arXiv preprint arXiv:2408.06072 (2025)
44. Zhang, Z., Liao, J., Li, M., Dai, Z., Qiu, B., Zhu, S., Qin, L., Wang, W.: Tora: Trajectory-oriented diffusion transformer for video generation. arXiv preprint arXiv:2407.21705 (2025)
45. Zhou, T., Tucker, R., Flynn, J., Fyffe, G., Snavely, N.: Stereo magnification: Learning view synthesis using multiplane images. In: SIGGRAPH (2018)