

AnimateAnything: Consistent and Controllable Animation for Video Generation

Guojun Lei^{1*}, Chi Wang^{1*}, Hong Li^{3,5*}, Rong Zhang⁴, Yikai Wang², Weiwei Xu^{1†}

¹ State Key Lab of CAD&CG, Zhejiang University ² Tsinghua University

³ Beihang University ⁴ Zhejiang Gongshang University ⁵ ShengShu

guojunlei@zju.edu.cn, wangchi1995@zju.edu.cn, link0502@buaa.edu.cn

zhangrong@zjgsu.edu.cn, yikaiw@outlook.com, xww@cad.zju.edu.cn

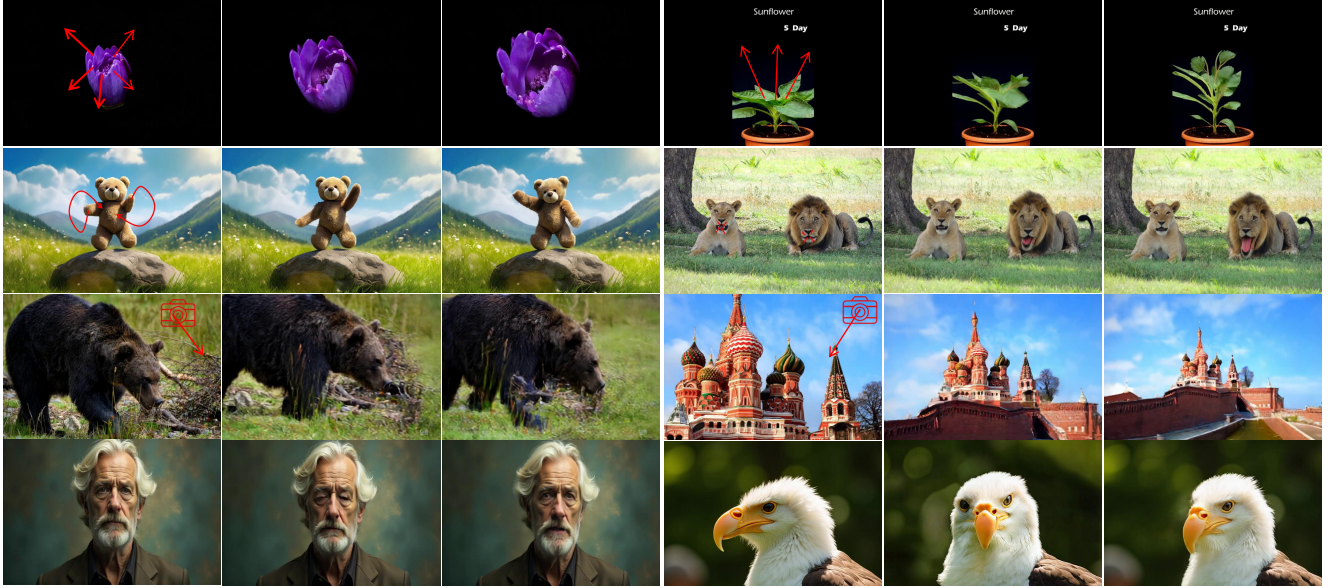


Figure 1. Animate anything. Consistent and controllable animation for different kinds of control signals. Given a reference image and corresponding user prompts, our approach can animate arbitrary characters, generating clear stable videos while maintaining consistency with the appearance details of the reference object.

Abstract

We present a unified controllable video generation approach *AnimateAnything* that facilitates precise and consistent video manipulation across various conditions, including camera trajectories, text prompts, and user motion annotations. Specifically, we carefully design a multi-scale control feature fusion network to construct a common motion representation for different conditions. It explicitly converts all control information into frame-by-frame optical flows. Then we incorporate the optical flows as motion priors to guide the final video generation. In addition, to reduce the flickering issues caused by large-scale motion,

we propose a frequency-based stabilization module. It can enhance temporal coherence by ensuring the video’s frequency domain consistency. Experiments demonstrate that our method outperforms the state-of-the-art approaches. For more details and videos, please refer to the anonymous webpage: <https://yu-shaonian.github.io/AnimateAnything/>.

1. Introduction

The emergence of Sora [5] has led to a breakthrough in large-scale video generation. Recently, controllable video generation [17, 22, 45, 51], *i.e.* controlling camera trajectories and object movements, has gained significant attention. It has expanded applications of video generation, making

*Joint first authors.

†Corresponding author.

them directly applicable to film production and virtual reality. However, due to the high complexity of large-scale camera and object movements, achieving precise control over video generation in such cases remains challenging.

MotionCtrl [45] and CameraCtrl [17] support camera trajectory manipulation for dynamic video generation, but they rely solely on text input. Since text descriptions provide only the overall characteristics of a video and cannot convey specific details precisely, it is insufficient to manipulate the video generation process only using text prompts. In contrast, image guidance, such as user-annotated trajectories or reference videos, can present more detailed visual cues. Motion-I2V [38] allows for image-based guidance but only enables slight object movements through user drag annotations, such as adjusting eye direction or indicating leg motion. It is not capable of manipulating the camera trajectory of a video. MOFA-Video [31] achieves control over detailed, pixel-level movements but is also limited to small-scale camera movement. To ensure global consistency in dynamic videos, MOFA-Video requires users to specify the movement direction for each local region of the input image that may move, making the process overly complex for user interaction.

This paper focus on image-to-video (I2V) generation that simultaneously processes dynamic control signals, such as arrow-based motion annotations, camera movements, and reference videos. However, the integration of these signals is challenging due to their different modalities, making the direct combination with a single video generation model difficult. Current approaches [15] typically attempt to train each control signal individually (for instance, through Lora [24]), and then collaboratively apply these signals to enhance video generation results. Nevertheless, these methods often necessitate careful parameter tuning or denoising strategies, making it difficult to maintain video stability, which can lead to flickering effects or incoherent pixel motion caused by different control signals. Some methods [31, 38] aim to facilitate the controllable generation of local motion in videos by introducing optical flow fields; however, they are ineffective in addressing camera motion signals, as camera movement introduces global motion information that is independent of the subject’s motion. Based on the insights presented above, we speculate that if the local motion of the subject and the global motion of the camera can be unified into a representation of frame-by-frame pixel movements, namely optical flow, it would support the guidance of the video generation model’s behavior, thereby possessing the potential to achieve synchronous control of various signals.

The key challenge is to handle incoherent pixel motion caused by different control signals. For instance, as Fig. 2 shows, the optical flow generated from different control signals varies greatly. Camera motion involves global move-

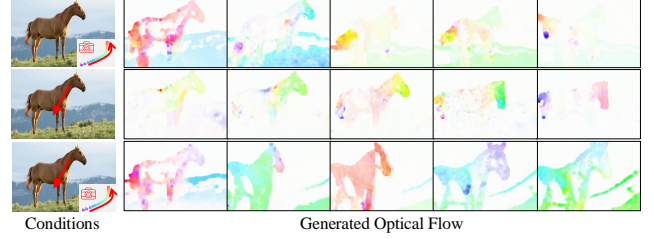


Figure 2. The generated optical flow by our method with different condition signals. Given a specific image, from top to bottom are optical flows generated with camera trajectory, arrow-based motion annotation, and both conditions, respectively.

ment, affecting both foreground and background pixels (the first row of Fig. 2), while a single motion annotation mainly influences localized foreground pixel motion (the second row of Fig. 2). Simultaneously introducing these two conditions directly may result in control signal conflicts, making the model confused about the following movements (the third row of Fig. 2). It is difficult to directly integrate these conditions through existing methods, since it demands considering both the individual impact of each condition and their complex interactions, such as projection transformations and occlusion completion. Therefore, we strategically design various condition-injection modules based on the representation and correlations of different control signals to enable unified optical flow generation.

Therefore, we propose an innovative two-stage video generation method to achieve multi-condition joint control. In the first stage, we convert various motion control signals into a unified optical flow, which is then used to guide the final video generation in the second stage. To further enhance video stability, we convert features from the time domain to the frequency domain and introduce a spectral attention mechanism to improve the overall quality of the generated videos.

The main contributions are summarized as follows:

- We introduce a two-stage pipeline to achieve stable and flexible video generation with different kinds of control signals. In the first stage, all control signals will be unified into frame-by-frame optical flow, which is then fed into the second stage to synchronize with text controls for high-quality video generation.
- We utilize an adaptive feature refinement in the frequency domain. This operation effectively suppresses instability and flickering in the generated video by modifying the temporal frequency features within the video.
- We perform extensive experiments to demonstrate the superiority of our method over state-of-the-art methods both quantitatively and qualitatively.

2. Related Work

Controllable Video Generation. Text-to-video (T2V) generation has received significant attention in recent years, especially after the emergence of Sora [24]. Typically, in this area, the text-based control information is injected via cross-attention mechanisms [17, 45]. While MagicTime [54] introduces a novel GPT [1]-based “Magic Text-Encoder” to enhance text comprehension ability. However, text-driven approaches often fail to convey video details precisely. As a result, methods driven by text and image simultaneously have become popular, significantly addressing these limitations. To achieve more effective video generation, pioneers [10, 13, 25, 43, 59] have explored generating videos under the guidance of some easily obtainable signals such as edges, depth, optical flow, or bounding boxes. Taking reference video as motion guidance, MotionClone [29] enables motion cloning by treating temporal-attention weights as motion representation. Recently, with the immense potential of the film industry and virtual reality applications, precise control over camera and object motion trajectories has gained increasing interest.

Camera Trajectory Driven Video Generation. To facilitate camera trajectory control, AnimateDiff [15] trains additional motion LoRA [21] modules for each specific camera path. However, this method lacks precise control of camera trajectory and cannot generate videos for unseen camera trajectories. A straightforward solution to these issues is to treat camera parameters as additional conditions for video generation. MotionCtrl [45] employs 12 pose matrix parameters as frame-level conditions to explicitly introduce camera trajectory. Nevertheless, this method still shows limitations in capturing the necessary geometric information for precise camera control. CameraCtrl [17] enhances camera information integration by using plücker embeddings to represent camera trajectories. To integrate the camera embedding more effectively, VD3D [2] and CamCo [49] introduce a ControlNet [57]-like conditioning mechanism and an epipolar attention module, respectively.

Object Motion Trajectory Driven Video Generation. CameraCtrl [17] and MotionCtrl [45] support basic motion control through text description, which is rough and imprecise. Compared to T2V’s object motion control based on motion descriptions, the Image-to-video (I2V) method generates videos from object motion trajectories. It allows for a more precise and user-friendly description of object positions, movement directions, and motion amplitude within the scene. Yin et al., Hao et al. and Shi et al. introduce explicit optical flow as an intermediate representation to guide video generation. Similarly, MOFA-Video takes sparse motion hints as input and generates dense optical flows to warp multi-scale features to guide video generation.

Motion-I2V [38] is the most relevant method to ours, as both use a two-stage framework and select optical flow

as an intermediate motion representation. In the first stage of optical flow generation, Motion-I2V derives optical flow from text and reference motion field images. In contrast, we treat optical flow as a unified visual motion representation that incorporates reference images, drag actions, and view-point changes. This allows for a more comprehensive and balanced handling of diverse visual conditions. In the second stage, while Motion-I2V directly integrates optical flow with text and images, we employ the Expert AdaLN [51] to promote feature space alignment adaptively, enabling a deep fusion of multimodal features. This enhancement improves the capability and generalization of this stage, allowing it to perform effectively even when the image and optical flow are not fully aligned.

3. Methods

In this section, we present AnimateAnything, a unified controllable video generation approach for precise and consistent video customization across various conditions. As the pipeline illustrated in Fig. 3, we convert all visual control signals into a unified optical flow representation and then utilize it to guide the final video generation. In the following subsections, we will provide a detailed explanation of preliminary knowledge, each module of the pipeline, and the corresponding training strategies. Firstly, we provide a brief overview of Video Diffusion Models in Sec. 3.1. Afterward, we present the architecture of converting all control signals into unified flows in Sec. 3.2. Then, we introduce how the flows guide the final video generation in Sec. 3.3. Finally, we give detailed descriptions of the frequency stabilization module 3.4 and training strategy 3.5.

3.1. Video Diffusion Models

The video diffusion model builds on the concept of image diffusion probabilistic models, extending into the temporal dimension. It captures the dynamic relationships between frames in a video sequence, allowing for the generation of continuous and high-quality video content. By learning to reverse the added noise, it ensures temporal consistency and coherence in the generated videos. Let $x_0 \in R^{f \times h \times w \times c}$ represent a video latent variable, where f is the total number of frames, each of size $h \times w$ with c channels. The forward diffusion process is modeled as a chain that incrementally adds Gaussian noise to the original video, defined as follows:

$$x_t = \sqrt{\bar{\alpha}_t}x_{t-1} + \sqrt{(1 - \bar{\alpha}_t)}\epsilon, \epsilon \sim N(0, 1), \quad (1)$$

where $t \in \{1, \dots, T\}$ denotes the timestep, $\bar{\alpha}_t$ regulates the intensity of noise added at each t , and ϵ is drawn from standard Gaussian noise. In the reverse process, a denoising model is learned to estimate $p(x_{t-1}|x_t)$, typically parameterized by a neural network θ . The optimization ob-

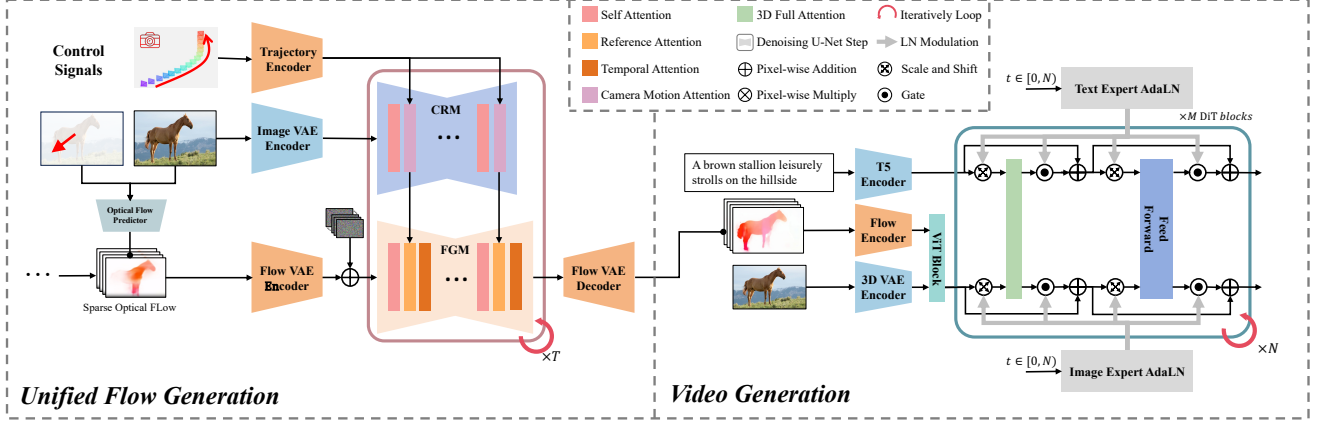


Figure 3. **AnimateAnything Pipeline.** The pipeline consists of two stages: 1) Unified Flow Generation, which creates a unified optical flow representation by leveraging visual control signals through two synchronized latent diffusion models, namely the Flow Generation Model (FGM) and the Camera Reference Model (CRM). The FGM accepts sparse or coarse optical flow derived from visual signals other than camera trajectory. The CRM inputs the encoded reference image and camera trajectory embedding to generate multi-level reference features. These features are fed into a reference attention layer to progressively guide the FGM’s denoising process in each time step, producing a unified dense optical flow. 2) Video Generation, which compresses the generated unified flow with a 3D VAE encoder and integrates it with video latents from the image encoder using a single ViT block. The final output is then combined with text embeddings to generate the final video using the DiT blocks.

jective is to minimize the following loss function: A denoising model, parametrized by neural network θ , estimates $p(x_{t-1}|x_t)$ in the reverse process, minimizing the given loss function:

$$\mathcal{L}(\theta) = \mathbb{E}_{x_0, \epsilon, \mathcal{C}, t} [\|\epsilon - \hat{\epsilon}_\theta(x_t, \mathcal{C}, t)\|_2^2], \quad (2)$$

where \mathcal{C} denotes the guidance conditions, like text. To train a video generation diffusion model using images, the image encoding is typically concatenated with the x_t , enabling the model to efficiently use its semantic features.

3.2. Stage 1: Unified Flow Generation

In this stage, we carefully design different injection modules based on the characteristics of each control signal and their relationship to achieve unified optical flow generation. In detail, we categorize the injection modules into explicit and implicit injection based on the attributes of visual control signals. The explicit injection module is proposed to control signals that can be directly converted into sparse optical flow for one or some frames, such as arrow-based motion annotation on specific pixels. The implicit injection is to incorporate control signals that are difficult to directly convert to pixel-level optical flow like camera trajectory. Finally, since information in the reference image, such as semantic categories, is directly related to various control signals, the image is involved in both implicit and explicit injection methods. In the following, we will explain the detailed operations for explicit and implicit injection, and further discuss the unified control signal at this stage.

Explicit Injection. As shown in Fig. 3, we explicitly convert different explicit control signals into initial sparse optical flow, and then apply a classical latent diffusion model [34], namely the Flow Generation Model (FGM), to transform it into dense optical flows. For these signals like arrow-based motion annotation, we use the following pipeline for conversion. Given a reference image, the user can label various motion trajectories on the image to represent the desired movements of objects and the environment. Take one trajectory for example, the trajectory can be regarded as a 2D point set, $\mathcal{M} \in \mathbb{R}^{P \times 2} = [(x_0, y_0), (x_1, y_1), \dots, (x_{P-1}, y_{P-1})]$, where P is the point number. The sparse control points can be extracted from \mathcal{M} with bicubic interpolation and then used to generate a point-wise sparse motion flow F^s in the following equation, the same as MOFA-Video [31], allowing us to guide the object motions and environmental changes effectively.

$$F_{l-1}^s(x_i, y_i) = \hat{\mathcal{T}}_l(x_i, y_i) - \hat{\mathcal{T}}_0(x_i, y_i) \quad (3)$$

where $l \in \{1, 2, \dots, L-1\}$, i denotes each pixel in the image. We also use CMP [56] to enhance sparse optical flows. Theoretically, any visual control signals convertible to sparse optical flow through extraction [50] or generation [9, 52], such as audio [9, 14], videos, and object landmarks [55], etc., can be input to FGM.

Implicit Injection. For implicit control signals like camera trajectory condition, we adopt the progressive condition injection design of AnimateAnyone [22] and implicitly embed it into the FGM denoising process through the Camera

Reference Model (CRM) progressively. The CRM employs a pre-trained image generation network based on the U-Net [35] architecture (SD1.5¹) and is initialized with original weights. It integrates camera trajectories with the reference image to get multi-scale reference features at specific time steps through camera motion attention, which uses image latents as query, camera features as both the key and value in the original cross-attention part. Then these features are utilized to guide the generation of dense optical flow via the reference attention layer in the FGM the same as AnimateAnyone. In order to better describe the camera pose, we use Plücker embeddings [39] as the representation of camera trajectory. Given the extrinsic and intrinsic camera parameters $\mathbf{R}, \mathbf{t}, \mathbf{K}_f$ for the f -th frame, we derive a Plücker embedding $\ddot{\mathbf{p}}_{f,h,w} \in \mathbb{R}^6$ for each pixel located at (h, w) . This embedding represents the vector from the camera center to the pixel’s position as:

$$\ddot{\mathbf{p}}_{f,h,w} = \left(\mathbf{t}_f \times \hat{\mathbf{d}}_{f,h,w}, \hat{\mathbf{d}}_{f,h,w} \right) \quad (4)$$

$$\hat{\mathbf{d}}_{f,h,w} = \frac{\mathbf{d}}{\|\mathbf{d}_{f,h,w}\|}, \quad \mathbf{d}_{f,h,w} = \mathbf{R}_f \mathbf{K}_f [w, h, 1]^\top + \mathbf{t}_f \quad (5)$$

Computing Plücker embedding for each pixel results in a representation $\ddot{\mathbf{P}} \in \mathbb{R}^{6 \times F \times H \times W}$ for a specified trajectory. To inject the trajectory representation into the reference motion network, we designed a trajectory encoder structurally similar to the camera encoder in CameraCtrl [17]. However, we improved the architecture: after each 2D ResNet block, we replaced the temporal attention with self-attention and output multi-scale trajectory features.

Through the combined use of explicit and implicit injection, we can effectively mitigate the incoherent pixel motion caused by different control signals. In addition, we used Unimatch [50] to extract a high-quality optical flow from training videos as ground truth during training. Similar to Motion-I2V [38], we trained a Flow Variational Autoencoder (VAE) to compress the optical flow of the pixel space into a latent flow space reducing computational resources.

3.3. Stage 2: Video Generation

In the second stage, we aim to use the unified dense optical flow representation from the previous stage to guide the video generation model in creating a final video that aligns with the semantics of the reference image and annotations, as shown in Fig. 3. For the video generation model, we inherit from CogVideoX framework [19, 51]. However, we introduce optical flow as a conditional guidance. Specifically, we use a flow encoder to encode the flows as the flow latent z_f . The flow encoder adopts four symmetrically arranged stages, respectively performing 2× downsampling and upsampling by the interleaving of ResNet block stacked

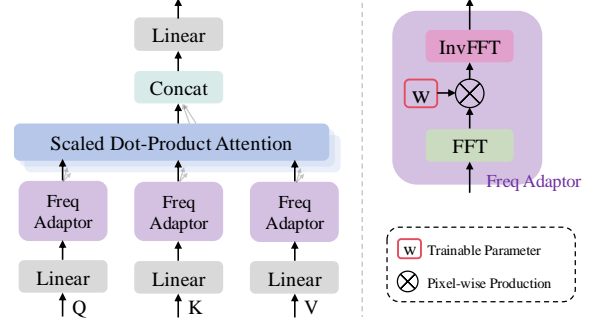


Figure 4. Video Stabilization Module

stages achieving a 4× compression in the temporal dimension and an 8×8 compression in the spatial dimension similar to the 3D VAE encoder [51]. Then we use a single basic Vision Transformer (ViT) block [12] to query video latents z_v from the flow latents z_f before calculating self-attention in the ViT block.

$$z'_v = \text{Attention}(Q, K, V) = \text{Softmax}(QK^T) V, \quad (6)$$

where $Q = W^Q z_v, K = W^K z_f, V = W^V z_f$. Notably, the flow feature maps only serve as key and value features. At the same time, the text prompts go through the text encoder using the google-research T5 model [33]. The result of the transformer block is then concatenated with text embedding before going through the full-attention transformer block. As shown in Fig. 3, we only train the optical flow encoder, input transformer block, and our video smoothing module (detailed in Sec. 3.4), keeping the parameters of other parts fixed to reduce the training difficulty.

3.4. Frequency Stabilization

In the previous two subsections, we effectively incorporated a large amount of motion control into our network through a two-stage design, supporting the generation of significant motion variations. In such cases, the corresponding optical flow may change drastically, making it prone to flickering and instability in the final video. To solve this problem, we review the video generation task from the perspective of information encoding. Treating the generated video as a sequence of images, flickering typically occurs due to misalignment of features between frames. This stems from the training of video generation models, where noise added to different frames at the same time step is independent. Despite temporal interactions among features, it remains challenging to prevent noise from negatively affecting the continuity and stability of video features. This instability will greatly affect the video generation quality. Compared to temporal features, frequency-domain features can more directly reveal some essential video-level information from a different perspective with individual frequency components, which is important for suppressing flickering issues.

¹<https://huggingface.co/stable-diffusion-v1-5/stable-diffusion-v1-5>

Thus, we adaptively modify the frequency-domain features extracted with the Fast Fourier Transform (FFT) [11] to maintain temporal stability. Specifically, as shown in Fig. 4, we modify the attention mechanism in the Diffusion Transformer (DiT) [32] architecture by first applying an FFT to each weight matrix to obtain its spectral features. We then multiply these features by a parameterized weight matrix W , followed by an inverse FFT (InvFFT) to restore the original temporal-domain feature. This is then used to compute dot-product attention, ensuring the consistency of scene features along the temporal direction during video generation.

3.5. Training Strategy

We conducted experiments on a server equipped with $8 \times$ NVIDIA Tesla A800 80G GPUs. In the first stage, for optical flow generation, we primarily use the Real10K [62] and DL3DV10K [28] datasets for training. In the second stage, for video generation, we utilize the WebVid10M [4] and OpenVid [30] datasets. Both datasets are large and diverse, covering various aspects of daily life from multiple sources, ensuring strong generalization.

Currently, achieving large-scale camera trajectory control for video generation remains a significant challenge. One of the major difficulties is the limited availability of video data with camera trajectory poses. The datasets available are Real10K [62] and DL3DV10K [28], but both are primarily indoor or static scene datasets. The video model trained on these datasets is unsuitable for dynamic scenes, while the generation is prone to failure. Another difficulty is that the dynamic video datasets available rarely contain pose information due to the difficulty of camera pose estimation in dynamic scenes using structure-from-motion (SfM) methods like COLMAP [37]. So, we organize and augment the data, and we further decompose the training of the first stage to achieve multi-condition controllable network training with limited data. Through careful search, we found that many videos on OpenVid [30] are shot from fixed camera positions, which can serve as a good starting point to boost dynamic training. We select a batch of videos with roughly fixed camera positions by evaluating the motion magnitude of the global optical flow, totaling around 10,000 videos. Our model is to first train the initial model on the Real10K dataset, then set the camera viewpoint of this batch of videos to be fixed at the origin, and subsequently train our model using the selected dynamic videos.

Given that the optical flow data required for the second stage is directly sourced from the video, both stages can be trained independently, with connection only needed during the inference process. Additionally, we apply noise to the optical flow in the training of the second stage to enhance the learning capability of the video generation model.

Table 1. Quantitative comparisons (Pose got by DUST3R, VggSfM, and ParticleSfM). We compare against prior works on basic trajectory and random trajectory respectively. T-Err, R-Err represent *translation error* and *rotation error*.

	Basic Trajectory						Difficult Trajectory					
	DUST3R		VggSfM		ParticleSfM		DUST3R		VggSfM		ParticleSfM	
	T-Err↓	R-Err↓	T-Err↓	R-Err↓	T-Err↓	R-Err↓	T-Err↓	R-Err↓	T-Err↓	R-Err↓	T-Err↓	R-Err↓
CameraCtrl	0.090	0.300	1.405	0.177	2.277	0.825	0.082	0.306	1.559	0.144	2.172	0.722
MotionCtrl	0.057	0.233	1.324	0.258	1.811	0.868	0.060	0.267	0.875	0.137	2.424	0.756
Ours	0.041	0.159	1.036	0.125	1.648	0.685	0.053	0.203	0.447	0.119	2.042	0.572

4. Experiments

We evaluate our method through quantitative metrics that confirm both the generation quality and its alignment with control signals, alongside visualizing the generated results for further qualitative comparison.

4.1. Image-to-Video Generation Ability.

As shown in Tab. 2, four classical image-level quality metrics, including Fréchet Inception Distance (FID) [18], SSIM [44], PSNR [20] and LPIPS [58], are used to evaluate the quality of the generated video frames with Motion-I2V [38], MOFA-Video [31], DynamiCrafter [48], CogVideoX[51], PyramidFlow [26], and OpenSora [61], and video-level metric Fréchet Video Distance (FVD) [40] is applied to assess video-level quality, similar to previous video generation methods [6–8, 27, 46]. Following CogVideo [51] and PyramidFlow [26], we employed several metrics from VBench [23] to evaluate the Subject Consistency (SubC), Motion Smoothness (MoS) and Aesthetic Quality (AesQ) of Our Video as shown in Tab. 3 and Fig. 9 on OpenVid [30] and WebVid [3] datasets. With optical flow guidance, our methods can achieve better performance especially when the generated video contains human motions and animal motions.

4.2. Control Signals Driven I2V Generation.

Camera Trajectory. The distance between predicted and ground-truth camera trajectories is used to measure the camera alignment. Here, we use ParticleSfM[60], VggSfM[41] and DUST3R[42] to evaluate our camera trajectory with CameraCtrl[17] and MotionCtrl[45] on basic trajectory (sample every 8 frames) and difficult trajectory (sample every max frame we can sample) in Real10K [63] shown in Tab. 5 and Fig. 5. Specifically, we estimate the camera trajectories for both the generated and real videos using the same methods to eliminate the potential scale differences caused by different Structure-from-Motion (SfM) techniques. And, we evaluate the quality of these trajectories by evaluating the scale and differences in the rotation and translation parameters of the camera matrix using *rotation error* and *translation error*, as outlined in He et al. [17], Wang et al. [45].

User Arrow Annotation. As shown in Fig. 1, we can turn

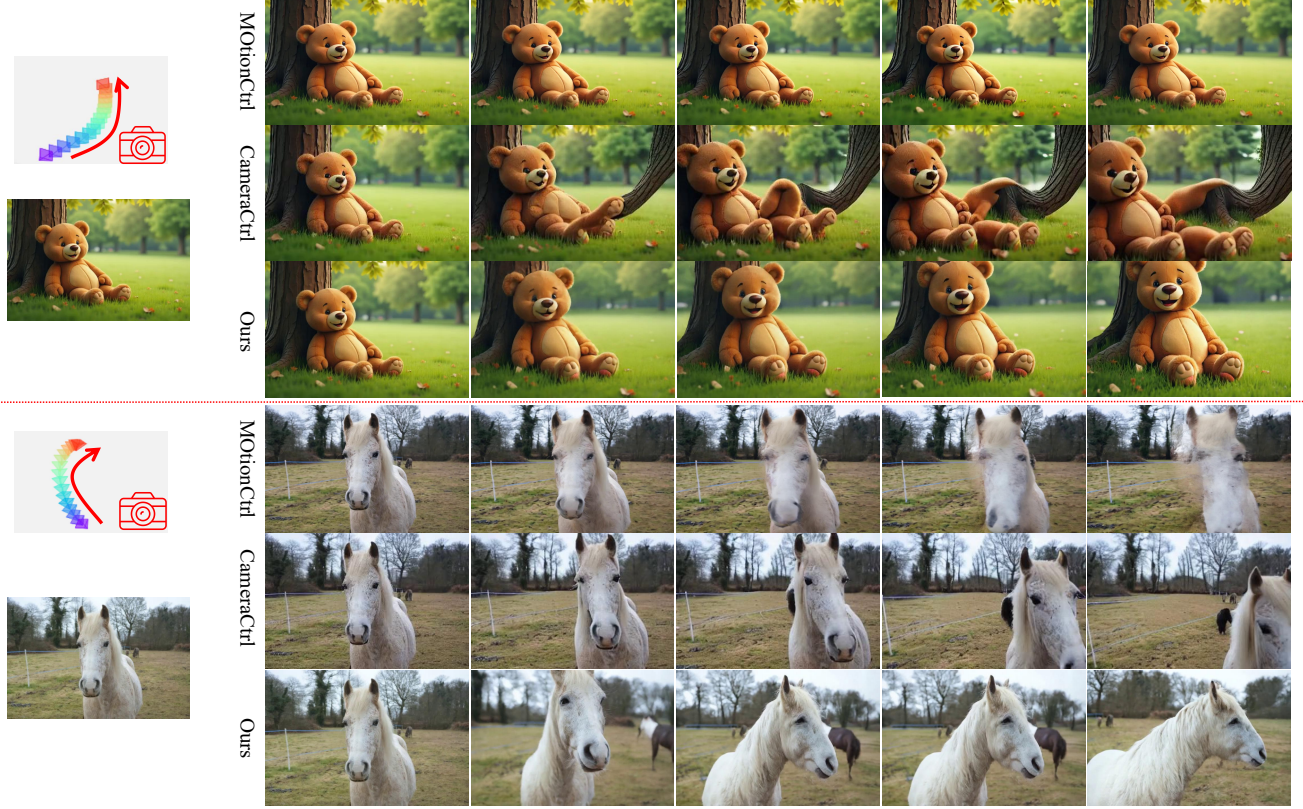


Figure 5. Camera trajectory comparison with other trajectory-based methods

Table 2. Video quality comparison.

	webvid					OpenVid				
	LPIPS↓	PSNR↑	SSIM↑	FID↓	FVD↓	LPIPS↓	PSNR↑	SSIM↑	FID↓	FVD↓
Motion-I2V	0.375	16.14	0.487	94.77	720	0.329	15.28	0.488	72.14	704
MOFA-Video	0.351	18.43	0.603	57.12	524	0.300	19.64	0.655	52.66	654
DynamiCrafter	0.268	18.56	0.532	63.73	685	0.393	13.83	0.402	59.61	751
CogVideoX+image	0.147	24.22	0.762	59.20	486	0.164	22.61	0.762	43.29	547
Pyramid-Flow	0.152	24.99	0.792	55.78	470	0.122	23.37	0.789	39.48	453
Open-Sora	0.179	23.21	0.725	58.33	552	0.117	22.78	0.760	44.48	512
Ours	0.135	25.22	0.810	48.11	380	0.113	25.04	0.836	33.12	322

Table 3. Video consistency quality comparison. SubC: Subject Consistency; MoS: Motion Smoothness; AesQ: Aesthetic Quality.

	webvid			OpenVid		
	SubC ↑	MoS ↑	Aesq ↑	SubC ↑	MoS ↑	Aesq ↑
DynamiCrafter	0.832	0.958	0.443	0.910	0.964	0.536
CogVideoX+image	0.855	0.984	0.443	0.929	0.987	0.567
Pyramid-Flow	0.906	0.991	0.438	0.941	0.991	0.537
Open-Sora	0.897	0.989	0.438	0.954	0.990	0.524
Ours	0.928	0.991	0.474	0.971	0.993	0.600

any kind of user drags into corresponding optical flows, which are then treated as the unified guidance for the final video generation. For this part, we compare with cur-

rent state-of-the-art user drag animation methods MOFA-Video [31], DragAnything [47], Motion-I2V [38] shown in Fig. 7. Our method can achieve more stable and consistent video generation on specific user drags. More results can be seen on the anonymous project webpage.

Reference Video. We demonstrate the capability of our Stage 2 in generating animations driven by reference videos, where a dense optical flow can be extracted. First, we test the case that the optical flow and given images are well aligned. Specifically, we generated the reference image by replacing or stylizing the subject in the first frame of the video as the reference image [36]. As shown in Fig. 6, it can be seen that Motion-I2V lacks sensitivity to a wide range of motions. Although MotionClone and MOFA-Video can

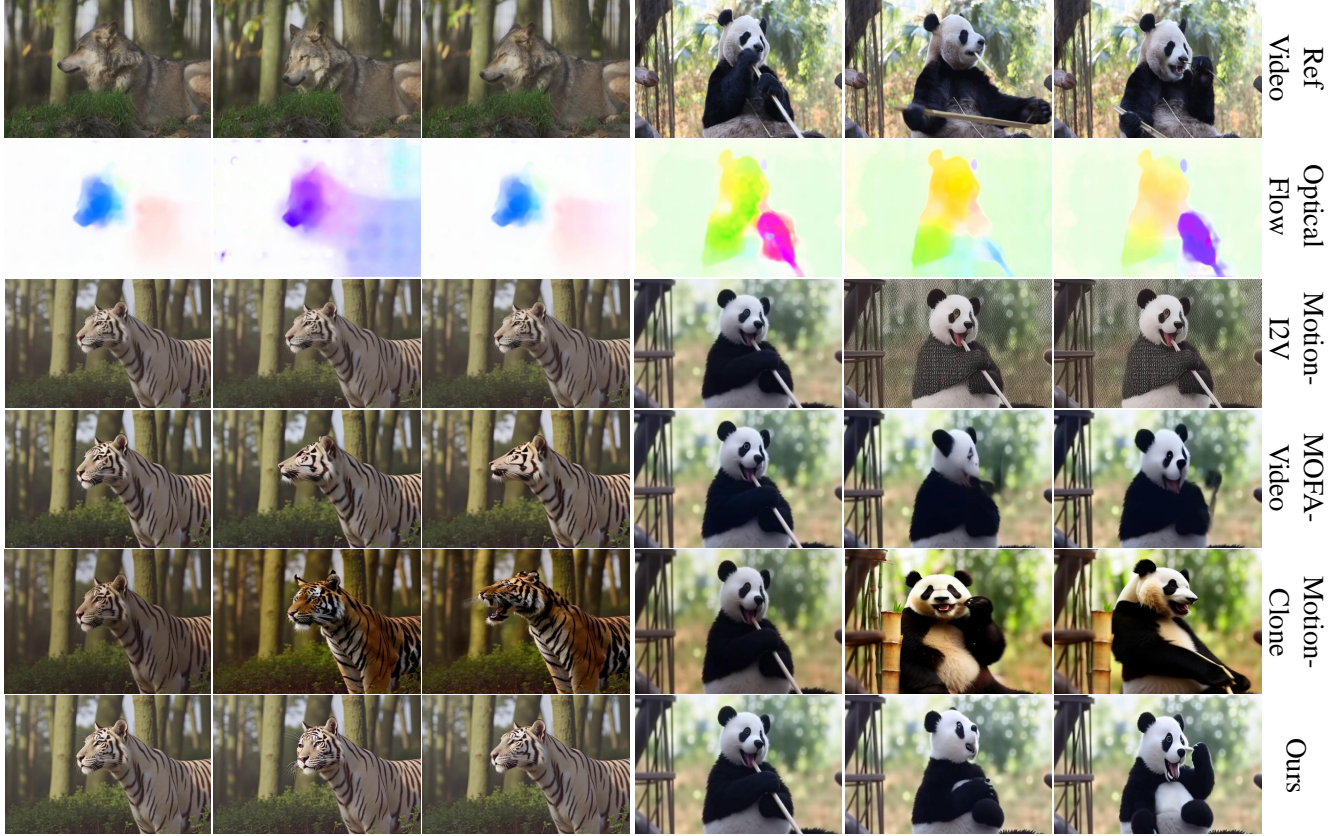


Figure 6. Motion Transfer comparison with state-of-the-art methods.



Figure 7. Users drag animation comparison with other animation methods.

achieve significant video motions, they result in style inconsistency and artifacts. Our generated results maintain significant motion alignment without a skeleton or facial keypoint extraction, while achieving optimal subject consistency. To better evaluate the generalization, we further experiment on the facial replacement task with the setting that the image and the unified optical flow are not perfectly aligned. The dense optical flow here inputted to Stage 2 is directly extracted from another facial motion video. As shown in Fig. 8, our Stage 2 can tolerate some misalignments while still performing effectively, producing consistent expressions and lip motions. This provides our method



Figure 8. Human face animation with optical flow extracted from reference video

with greater flexibility and robustness.

4.3. Ablation Study And Analysis

Setups. To verify the effectiveness of the components in our video generation pipeline, we designed several sets of ablation experiments on Real10K [63]. (1) The multi-frame camera encoding is added to the latent variables and used as input to the DiT blocks. (2) We replicate the first half of



Figure 9. Image to video generation comparison with current state-of-the-art methods.

Table 4. Ablation study.

	Visual Quality					Trajectory Alignment	
	LPIPS↓	PSNR↑	SSIM↑	FID↓	FVD↓	TransErr↓	RotErr↓
Camera embedding	0.401	14.22	0.531	52.46	346	0.551	0.048
ControlNet-Like	0.400	14.21	0.528	50.96	356	0.737	0.050
w/o FS	0.241	17.88	0.615	46.85	311	0.671	0.059
w/o noise	0.228	19.32	0.654	49.38	474	0.425	0.048
Full Model	0.142	23.22	0.796	41.67	168	0.354	0.047

the blocks of FGM as a reference network and then add the output of each block of the reference network to the output of the corresponding original block, like ControlNet. (3) and (4) both use the globally estimated optical flow from the video data as input, with the distinction that (3) removes Frequency Stabilization (FS), while (4) does not apply noise before feeding the global optical flow into the Flow Encoder.

Analysis. As shown in Tab. 4, From the first two rows, we can see the superiority of using a unified optical flow representation, which surpasses other camera control signal guidance methods in terms of visual quality and camera rule prediction. The third row reflects that not using Frequency Stability during the training process leads to a significant drop in performance. However, it is still better than the other two camera signal guidance methods. The fourth row illustrates that the noise application conducted before feeding the Optical Flow into the Flow encoder effectively enhances the robustness of the generation.

5. Conclusion

In this paper, we present a unified controllable video generation approach enabling precise and consistent video manipulation across various conditions. We unified Flow as a joint control signal by converting diverse visual control signals (e.g., object motion, camera motion) into a joint optical flow representation. Then the unified flows are used to guide the final video generation. This strategy reduces the complexity of handling multiple, isolated control signals and promotes consistency in the generated video. In addition, we propose a frequency-based stabilization module to preserve the key features in the frequency domain and reduce the flickering issues caused by large-scale motion. Experiments demonstrate that this two-stage pipeline

can control the video generation precisely and have impressive generalization capabilities.

References

- [1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv:2303.08774*, 2023. 3
- [2] Sherwin Bahmani, Ivan Skorokhodov, Aliaksandr Siarohin, Willi Menapace, Guocheng Qian, Michael Vasilkovsky, Hsin-Ying Lee, Chaoyang Wang, Jiaxu Zou, Andrea Tagliasacchi, DavidB Lindell, and Sergey Tulyakov. Vd3d: Taming large video diffusion transformers for 3d camera control. *arXiv:2407.12781*, 2024. 3
- [3] Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. Frozen in time: A joint video and image encoder for end-to-end retrieval. In *ICCV*, 2021. 6
- [4] Max Bain, Arsha Nagrani, Gul Varol, and Andrew Zisserman. Frozen in time: A joint video and image encoder for end-to-end retrieval. In *ICCV*, 2021. 6
- [5] Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, Clarence Ng, Ricky Wang, and Aditya Ramesh. Video generation models as world simulators. 2024. 1
- [6] Shengqu Cai, Duygu Ceylan, Matheus Gadelha, Chun-HaoPaul Huang, TuanfengYang Wang, and Gordon Wetzstein. Generative rendering: Controllable 4d-guided video generation with 2d diffusion models. In *CVPR*, 2023. 6
- [7] Shengqu Cai, EricRyan Chan, Songyou Peng, Mohamad Shahbazi, Anton Obukhov, LucVan Gool, and Gordon Wetzstein. Diffdreamer: Towards consistent unsupervised single-view scene extrapolation with conditional diffusion models. In *ICCV*, 2023.
- [8] Duygu Ceylan, Chun-HaoP Huang, and NiloyJ Mitra. Pix2video: Video editing using image diffusion. In *ICCV*, 2023. 6
- [9] Moitreyia Chatterjee and Anoop Cherian. Sound2sight: Generating visual dynamics from sound and context. In *ECCV*, 2020. 4
- [10] Xinyuan Chen, Yaohui Wang, Lingjun Zhang, Shaobin Zhuang, Xin Ma, Jiashuo Yu, Yali Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. Seine: Short-to-long video diffusion model for generative transition and prediction. In *ICLR*, 2023. 3
- [11] James W Cooley, Peter AW Lewis, and Peter D Welch. Historical notes on the fast fourier transform. *Proceedings of the IEEE*, 55(10):1675–1677, 1967. 6
- [12] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICCV*, 2021. 5
- [13] Patrick Esser, Johnathan Chiu, Parmida Atighehchian, Jonathan Granskog, and Anastasis Germanidis. Structure

- and content-guided video synthesis with diffusion models. In *ICCV*, 2023. 3
- [14] Shuheng Ge, Haoyu Xing, Li Zhang, and Xiangqian Wu. Opflowtalker: Realistic and natural talking face generation via optical flow guidance. *arXiv:2405.14709*, 2024. 4
- [15] Yuwei Guo, Ceyuan Yang, Anyi Rao, Yaohui Wang, Yu Qiao, Dahua Lin, and Bo Dai. Animatediff: Animate your personalized text-to-image diffusion models without specific tuning. In *ICLR*, 2023. 2, 3
- [16] Zekun Hao, Xun Huang, and Serge Belongie. Controllable video generation with sparse trajectories. In *CVPR*, 2018. 3
- [17] Hao He, Yinghao Xu, Yuwei Guo, Gordon Wetzstein, Bo Dai, Hongsheng Li, and Ceyuan Yang. Cameractrl: Enabling camera control for text-to-video generation. *arXiv:2404.02101*, 2024. 1, 2, 3, 5, 6
- [18] 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. 6
- [19] 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. 5
- [20] Alain Hore and Djemel Ziou. Image quality metrics: Psnr vs. ssim. In *ICPR*, 2010. 6
- [21] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv:2106.09685*, 2021. 3
- [22] Li Hu, Xin Gao, Peng Zhang, Ke Sun, Bang Zhang, and Liefeng Bo. Animate anyone: Consistent and controllable image-to-video synthesis for character animation. *arXiv:2311.17117*, 2023. 1, 4
- [23] Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, Yaohui Wang, Xinyuan Chen, Limin Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. VBench: Comprehensive benchmark suite for video generative models. In *CVPR*, 2024. 6
- [24] HuEdward J., Yulong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *ICLR*, 2022. 2, 3
- [25] Yash Jain, Anshul Nasery, Vibhav Vineet, and Harkirat Behl. Peekaboo: Interactive video generation via masked-diffusion. In *CVPR*, 2024. 3
- [26] 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. *arXiv:2410.05954*, 2024. 6
- [27] Levon Khachatryan, Andranik Movsisyan, Vahram Tadevosyan, Roberto Henschel, Zhangyang Wang, Shant Navasardyan, and Humphrey Shi. Text2video-zero: Text-to-image diffusion models are zero-shot video generators. In *ICCV*, 2023. 6
- [28] Lu Ling, Yichen Sheng, Zhi Tu, Wentian Zhao, Cheng Xin, Kun Wan, Lantao Yu, Qianyu Guo, Zixun Yu, Yawen Lu, et al. DI3dv-10k: A large-scale scene dataset for deep learning-based 3d vision. In *CVPR*, 2024. 6
- [29] Pengyang Ling, Jiazi Bu, Pan Zhang, Xiaoyi Dong, Yuhang Zang, Tong Wu, Huaian Chen, Jiaqi Wang, and Yi Jin. Motionclone: Training-free motion cloning for controllable video generation. *arXiv:2406.05338*, 2024. 3
- [30] Kepan Nan, Rui Xie, Penghao Zhou, Tiehan Fan, Zhenheng Yang, Zhijie Chen, Xiang Li, Jian Yang, and Ying Tai. Openvid-1m: A large-scale high-quality dataset for text-to-video generation. *arXiv:2407.02371*, 2024. 6
- [31] Muyao Niu, Xiaodong Cun, Xintao Wang, Yong Zhang, Ying Shan, and Yinqiang Zheng. Mofa-video: Controllable image animation via generative motion field adaptations in frozen image-to-video diffusion model. *arXiv:2405.20222*, 2024. 2, 4, 6, 7
- [32] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *ICCV*, 2023. 6
- [33] Adam Roberts, Hyung Won Chung, Anselm Levskaya, Gaurav Mishra, James Bradbury, Daniel Andor, Sharan Narang, Brian Lester, Colin Gaffney, Afroz Mohiuddin, Curtis Hawthorne, Aitor Lewkowycz, Alex Salcianu, Marc van Zee, Jacob Austin, Sebastian Goodman, Livio Baldini Soares, Haitang Hu, Sasha Tsvyashchenko, Aakanksha Chowdhery, Jasmijn Bastings, Jannis Bulian, Xavier Garcia, Jianmo Ni, Andrew Chen, Kathleen Keane, Jonathan H. Clark, Stephan Lee, Dan Garrette, James Lee-Thorp, Colin Raffel, Noam Shazeer, Marvin Ritter, Maarten Bosma, Alexandre Passos, Jeremy Maitin-Shepard, Noah Fiedel, Mark Omernick, Brennan Saeta, Ryan Sepassi, Alexander Spiridonov, Joshua Newlan, and Andrea Geminio. Scaling up models and data with t5x and seqio. *arXiv:2203.17189*, 2022. 5
- [34] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Bjorn Ommer. High-resolution image synthesis with latent diffusion models. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 4
- [35] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention—MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III* 18. Springer, 2015. 5
- [36] L Rout, Y Chen, N Ruiz, C Caramanis, S Shakkottai, and W Chu. Semantic image inversion and editing using rectified stochastic differential equations. *arXiv:2410.10792*, 2024. 7
- [37] Johannes L. Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, 2016. 6
- [38] Xiaoyu Shi, Zhaoyang Huang, Fu-Yun Wang, Weikang Bian, Dasong Li, Yi Zhang, Manyuan Zhang, Ka Chun Cheung, Simon See, Hongwei Qin, Jifeng Dai, and Hongsheng Li. Motion-i2v: Consistent and controllable image-to-video generation with explicit motion modeling. In *SIGGRAPH*, 2024. 2, 3, 5, 6, 7
- [39] Vincent Sitzmann, Semon Rezchikov, Bill Freeman, Josh Tenenbaum, and Frédo Durand. Light field networks: Neural scene representations with single-evaluation rendering. In *NeurIPS*, 2021. 5

- [40] Thomas Unterthiner, Sjoerdvan Steenkiste, Karol Kurach, Raphaël Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric & challenges. *arXiv:2303.14207*, 2018. 6
- [41] Jianyuan Wang, Nikita Karaev, Christian Rupprecht, and David Novotny. Vggsfm: Visual geometry grounded deep structure from motion. In *CVPR*, 2024. 6
- [42] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *CVPR*, 2024. 6
- [43] Xiang Wang, Hangjie Yuan, Shiwei Zhang, Dayou Chen, Jiniu Wang, Yingya Zhang, Yujun Shen, Deli Zhao, and Jingren Zhou. Videocomposer: Compositional video synthesis with motion controllability. *NeurIPS*, 2024. 3
- [44] Z. Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *TIP*, 2004. 6
- [45] 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. 1, 2, 3, 6
- [46] JayZhangjie Wu, Yixiao Ge, Xintao Wang, Weixian Lei, Yuchao Gu, Wynne Hsu, Ying Shan, Xiaohu Qie, and MikeZheng Shou. Tune-a-video: One-shot tuning of image diffusion models for text-to-video generation. In *ICCV*, 2023. 6
- [47] 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. 7
- [48] Jinbo Xing, Menghan Xia, Yong Zhang, Haoxin Chen, Wangbo Yu, Hanyuan Liu, Xintao Wang, Tien-Tsin Wong, and Ying Shan. Dynamicrafter: Animating open-domain images with video diffusion priors. *arXiv:2310.12190*, 2023. 6
- [49] Dejjia Xu, Weili Nie, Chao Liu, Sifei Liu, Jan Kautz, Zhangyang Wang, and Arash Vahdat. Camco: Camera-controllable 3d-consistent image-to-video generation. *arXiv:2406.02509*, 2024. 3
- [50] Haofei Xu, Jing Zhang, Jianfei Cai, Hamid Rezatofighi, Fisher Yu, Dacheng Tao, and Andreas Geiger. Unifying flow, stereo and depth estimation. *IEEE TPAMI*, 2023. 4, 5
- [51] Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. Cogvideox: Text-to-video diffusion models with an expert transformer. *arXiv:2408.06072*, 2024. 1, 3, 5, 6
- [52] Zhonghua Yi, Hao Shi, Kailun Yang, Qi Jiang, Yaozu Ye, Ze Wang, Huajian Ni, and Kaiwei Wang. Focusflow: Boosting key-points optical flow estimation for autonomous driving. *IEEE Trans. Intell. Veh.*, 2023. 4
- [53] 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
- [54] Shenghai Yuan, Jinfa Huang, Yujun Shi, Yongqi Xu, Ruijie Zhu, Bin Lin, Xinhua Cheng, Li Yuan, and Jiebo Luo. Mag-ictime: Time-lapse video generation models as metamorphic simulators. *arXiv:2404.05014*, 2024. 3
- [55] Bohan Zeng, Boyu Liu, Hong Li, Xuhui Liu, Jianzhuang Liu, Dapeng Chen, Wei Peng, and Baochang Zhang. Fnevr: neural volume rendering for face animation. In *NeurIPS*, 2024. 4
- [56] Xiaohang Zhan, Xingang Pan, Ziwei Liu, Dahua Lin, and Chen Change Loy. Self-supervised learning via conditional motion propagation. In *CVPR*, 2019. 4
- [57] Lvmin Zhang and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *ICCV*, 2023. 3
- [58] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 6
- [59] Yabo Zhang, Yuxiang Wei, Dongsheng Jiang, Xiaopeng Zhang, Wangmeng Zuo, and Qi Tian. Controlvideo: Training-free controllable text-to-video generation. *arXiv:2305.13077*, 2023. 3
- [60] Wang Zhao, Shaohui Liu, Hengkai Guo, Wenping Wang, and Yong-Jin Liu. Particlesfm: Exploiting dense point trajectories for localizing moving cameras in the wild. In *ECCV*, 2022. 6
- [61] 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, 2024. 6
- [62] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification. *ACM TOG*, 2018. 6
- [63] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: learning view synthesis using multiplane images. *ACM TOG*, 2018. 6, 8