

# ATI: Any Trajectory Instruction for Controllable Video Generation

Angtian Wang Haibin Huang Jacob Zhiyuan Fang Yiding Yang Chongyang Ma  
 ByteDance Intelligent Creation  
<https://anytraj.github.io/>

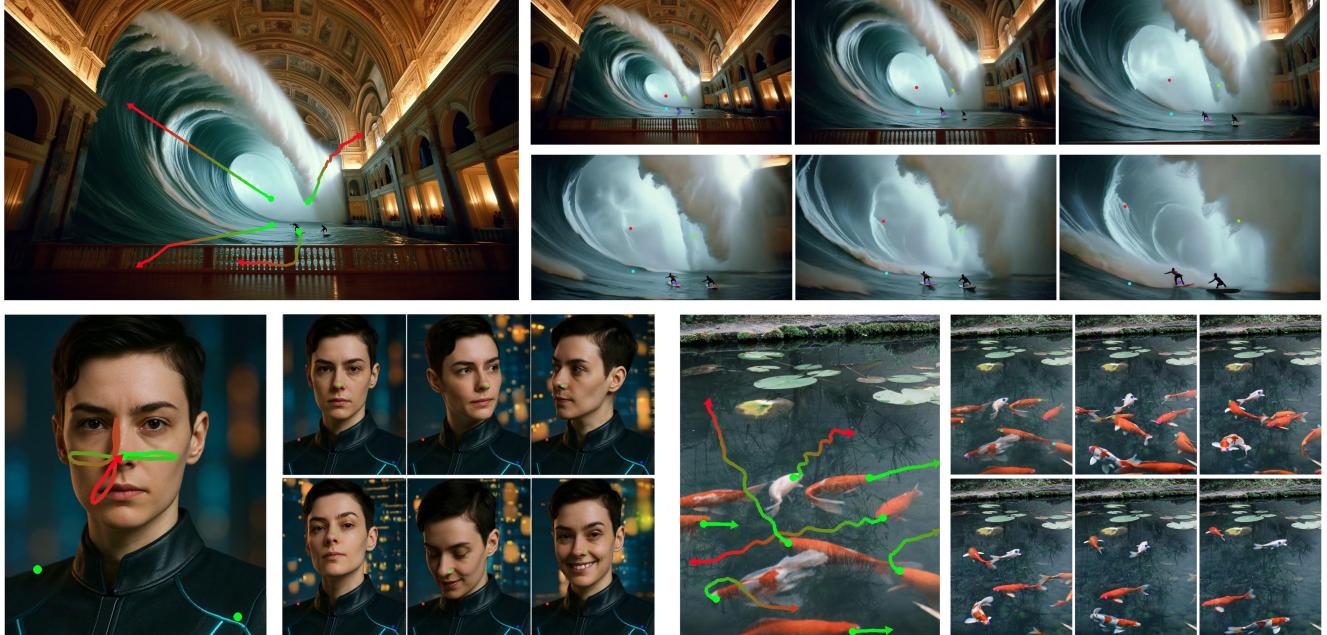


Figure 1. ATI is able to generate a video given an initial frame (left) and a set of user-specified trajectories. Green dots denote the starting points, and red dots indicate the ending points of each trajectory. On the right, we show uniformly sampled frames from the generated video, with colored dots tracking the position of each trajectory point over time.

## Abstract

We propose a unified framework for motion control in video generation that seamlessly integrates camera movement, object-level translation, and fine-grained local motion using trajectory-based inputs. In contrast to prior methods that address these motion types through separate modules or task-specific designs, our approach offers a cohesive solution by projecting user-defined trajectories into the latent space of pre-trained image-to-video generation models via a lightweight motion injector. Users can specify keypoints and their motion paths to control localized deformations, entire object motion, virtual camera dynamics, or combinations of these. The injected trajectory signals guide the generative process to produce temporally consistent and semantically aligned motion sequences. Our

framework demonstrates superior performance across multiple video motion control tasks, including stylized motion effects (e.g., motion brushes), dynamic viewpoint changes, and precise local motion manipulation. Experiments show that our method provides significantly better controllability and visual quality compared to prior approaches and commercial solutions, while remaining broadly compatible with various state-of-the-art video generation backbones.

## 1. Introduction

Recent advances in video generation models [1–4, 9, 11, 14, 17, 18, 23, 24, 28, 34, 37, 44] have demonstrated remarkable capabilities in synthesizing realistic and diverse video content. However, precise control over motion remains a significant challenge, particularly when users re-

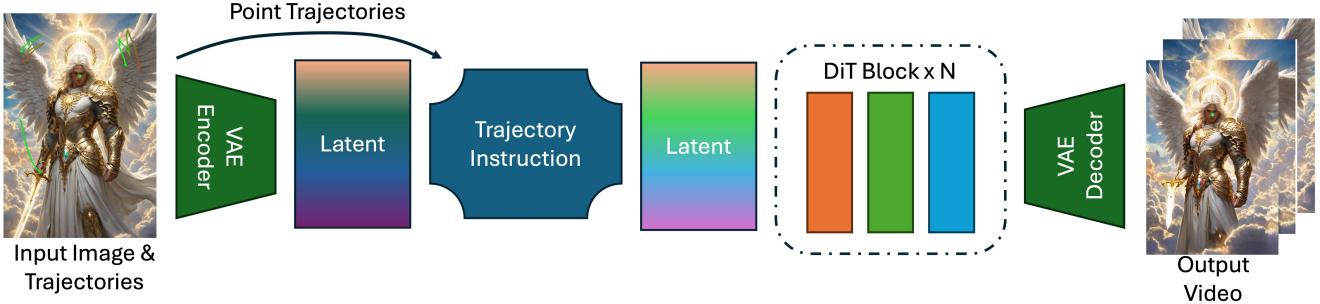


Figure 2. ATI takes an image and user specified trajectories as inputs. The point-wise trajectories are injected into the latent condition for the generation. Videos are decoded from the latent denoised from the DiT.

quire fine-grained direction over how specific elements move within the generated sequence. Current approaches [21, 26, 31, 32, 41] typically address different types of motion control—such as camera movement, object translation, or local deformation—through separate specialized modules, leading to fragmented workflows and inconsistent results. Motion control in video generation encompasses a spectrum of manipulations, from camera operations (panning, zooming, rotation) to object-level translations and localized movements of specific regions. These motion types are inherently related and often need to be coordinated to achieve desired visual effects. The separation of these controls in existing systems limits creative expression and requires users to navigate multiple interfaces or models.

In this paper, we propose a unified trajectory-based framework that addresses this limitation by treating all forms of motion control through a common lens. Our key insight is that diverse motion effects can be represented as trajectories of specific points within the scene, whether these points are anchored to local features, whole objects, or used to indicate camera perspective changes. By defining motion as trajectory paths for user-selected key points, we establish a consistent, intuitive interface for motion specification. Our approach builds upon state-of-the-art image-to-video generation models [24, 28], augmenting them with a specialized motion injector module. This module processes trajectory information and projects it into the latent space of the pre-trained video generation model, effectively guiding the synthesis process to follow the specified motion paths. Importantly, our method does not require retraining the base video model, making it adaptable to different generation architectures. We demonstrate the versatility of our framework through extensive experiments across various motion control tasks. Our results show that the unified approach not only simplifies the user workflow but also produces higher quality motion than previous methods that handle different motion types separately. The framework excels in challenging scenarios that require coordination between camera movement and object motion, outperforming both special-

ized academic methods and commercial video generation products in both control precision and visual quality. The main contributions of our work include:

- A unified framework for motion control in video generation that seamlessly integrates camera movements, object-level motion, and local deformations through trajectory-based guidance.
- A motion injector module that effectively projects user-specified trajectory controls into the latent space of pre-trained video generation models.
- Comprehensive evaluation demonstrating superior performance across various motion control tasks compared to previous methods and commercial products.
- Demonstration of compatibility with different base video generation models, highlighting the approach’s flexibility and broad applicability.

## 2. Related Work

Motion-controlled video generation aims to synthesize temporally coherent videos with user-defined motion guidance, which aims to manipulate the camera motion and object movement. Methods such as CamI2V [42], CameraC-trl [10], and CamCo [35] encode camera trajectories using Plücker coordinates to achieve fine-grained camera path conditioning. Others like ViewCrafter [40] and I2VControl-Camera [7] leverage 3D scene reconstruction from a single image to generate point cloud renderings that guide camera perspectives during generation. Additionally, there exist some training-free approach like [12, 39]. [15] proposes collaborative diffusion methods which address consistent multi-view synthesis with controllable cameras.

Another important aspect of Motion-controlled video generation is object motion control. Various strategies are used to guide object trajectories, *e.g.*, optical flow-based methods, such as DragNUWA [38], Image Conductor [16], DragAnything [33], and MotionBridge [27], utilize sparse or dense flow to control object displacement. Others like MOFA-Video [21] and Motion-I2V [25] directly learn dense motion fields to guide generation. Bounding-box-

based control is employed in Boximator [30] and Direct-a-Video [36], while methods like LeviTor [29] incorporate depth and clustering for accurate 3D motion guidance. Newer works like ReVideo [20], Peekaboo [13], and Trailblazer [19] expand this paradigm with interactive or trajectory-aware modules. Particularly, training-free method like [39] injects motion trajectories by decomposing the task into ‘out-of-place’ and ‘in-place’ motion animation and leverage layout-conditioned image generation for motion generation.

Recent works further advances to motion control that simultaneously handles camera and object motion. MotionC-trl [31] introduces explicit modules to support concurrent control, while Motion Prompting [8] encodes motion tracks to guide both scene and subject dynamics. Perception-as-Control [5] proposes a 3D-aware representation that fuses motion perception with generation. VidCraft3 [43] builds unified, disentangled control across multiple motion modalities.

### 3. Method

We propose ATI (Figure 2), a diffusion-based video generation framework that enables Fine-grained feature-level Instruction of Trajectories. Specifically, ATI introduces a Gaussian-based motion injector to encode trajectory signals, spanning local, object-level, and camera motion, directly into the latent space of a pretrained image-to-video diffusion model. This enables unified and continuous control over both object and camera dynamics.

#### 3.1. Conditional Video Generation Models

Recent advances in diffusion models have revolutionized generative modeling for both images and videos. In the video domain, these models aim to synthesize realistic, temporally consistent sequences. When extended to conditional generation, the objective is to generate videos based on specific inputs such as text, images, or motion cues, enabling fine-grained control over content, appearance, and motion. Prominent architectures like Diffusion Transformers (DiT) achieve state-of-the-art performance by integrating spatiotemporal modeling with conditional guidance. Let a video be denoted as  $\mathbf{x}_0 \in \mathbb{R}^{T \times H \times W \times C}$ , where  $T$  is the number of frames. Let  $\mathbf{c}$  represent a conditioning signal (e.g., a text prompt, an image, or a trajectory).

The diffusion model defines a forward noising process that progressively corrupts the video with Gaussian noise:

$$q(\mathbf{x}_t | \mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1 - \beta_t} \mathbf{x}_{t-1}, \beta_t \mathbf{I}) \quad (1)$$

$$q(\mathbf{x}_0 | \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_0; \sqrt{\bar{\alpha}_t} \mathbf{x}_0, (1 - \bar{\alpha}_t) \mathbf{I}) \quad (2)$$

where  $\bar{\alpha}_t = \prod_{i=1}^t (1 - \beta_i)$ .

A neural network  $\epsilon_\theta$  learns to reverse the noising process by predicting the added noise, conditioned on  $\mathbf{c}$ :

$$\mathcal{L}_\epsilon = \|\epsilon - \epsilon_\theta(\mathbf{x}_t, t, \mathbf{c})\|^2 \quad (3)$$

In this report, we choose Seaweed-7B [24] and Wan2.1-14B [28] as the base video generation models, where the denoising model  $\epsilon_\theta$  is implemented using a DiT architecture [22].

#### 3.2. Gaussian Model for Trajectory Instruction of Feature

We propose a Gaussian model for feature-level instruction of point trajectories. Specifically, for each trajectory point, we assign a weight  $P(f | l_{i,j,t})$  to every pixel  $(i, j)$  in the latent space.

For a point trajectory  $\phi_t = (x_t, y_t)$  at frame  $t$ , we define:

$$P(f | l_{i,j,t}) = \exp\left(-\frac{\|\phi_t - (i, j)\|^2}{2\sigma}\right),$$

where  $\sigma$  is a predefined constant. In practice, we set  $\sigma = \frac{1}{440}$  so that the Gaussian weight decays to half its maximum at the nearest diagonal pixel.

As illustrated in Figure 3, we first pass the input image  $I$  through the VAE encoder  $\Phi$  to obtain a latent feature map

$$L_I = \Phi(I) \in \mathbb{R}^{H \times W \times C}. \quad (4)$$

We then extract, for each trajectory point, a  $C$ -dimensional feature vector  $f$  at its initial position  $\phi_0 = (x_0, y_0)$  by bilinearly sampling from  $L_I$  whenever  $(x_0, y_0)$  does not lie exactly on an integer grid coordinate.

In Figure 3, the bottom-left panel depicts the latent feature grid as colored cells; arrows indicate the precise sub-pixel sampling locations for each trajectory point. The inset in the middle shows how these sampled values assemble into the latent feature vector  $f$ . Finally, the bottom-right panels visualize the spatial Gaussian masks—centered at the trajectory locations  $\phi_t$  in subsequent frames—computed as

$$P(f | l_{i,j,t}) = \exp\left(-\|\phi_t - (i, j)\|^2/(2\sigma)\right), \quad (5)$$

which softly distributes the feature  $f$  across neighboring latent pixels to guide the image-to-video generator with fine-grained control.

#### 3.3. Tail Dropout Regularization

In practice, when a user-specified point trajectory terminates before the end of the video, the model often hallucinates spurious occluders around the final annotated frame. We attribute this to our training labels: any point that falls off its ground-truth track is marked as “occluded” or “out of frame,” which inadvertently teaches the model to introduce occlusions whenever a trajectory ends.

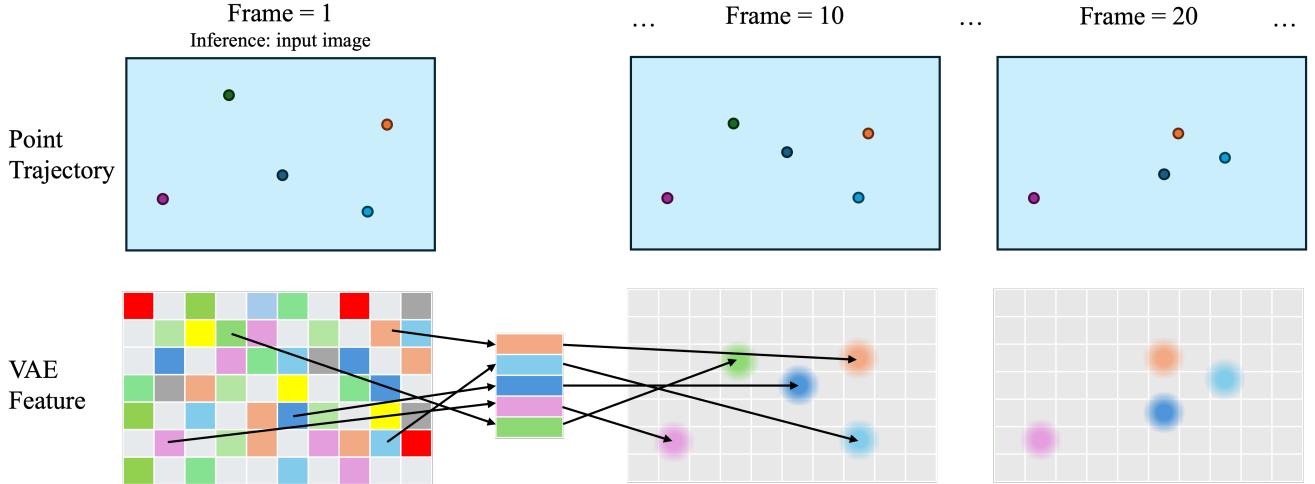


Figure 3. Trajectory Instruction module computes a latent feature from a point’s trajectory. During inference, given the point’s location in the first frame (*i.e.*, the input image), we sample the feature at that location using bilinear interpolation. We then compute a spatial Gaussian distribution for each visible point on its corresponding location in every subsequent frame.

To mitigate this, we introduce a *Tail Dropout Regularizer*. During training, with probability  $p$  (set to 0.2 in our experiments), we sample a dropout frame

$$t_d \sim \mathcal{U}\{0, 1, \dots, T\},$$

where  $T$  is the full trajectory length. We then truncate the trajectory via set the visibility of that point to 0 after frame  $t_d$ , effectively simulating an early termination. This encourages the model to learn that missing future points do *not* imply occlusion.

Empirically, applying Tail Dropout significantly reduces visual distortions and the appearance of unintended occluders when trajectories end before the final frame at inference time.

### 3.4. Data Collection

We create our training dataset by first processing 5 million high-quality video clips, which are filtered to contain no scene cuts and to meet strict aesthetic criteria—and then selecting 2.4 million clips exhibiting strong object motion. To generate point trajectory annotation, we apply TAP-Net [6] to each selected clip as follows:

1. On the first frame, uniformly sample  $N = 120$  points such that initial pairwise distances are approximately equal.
2. Track these seed points throughout the clip using TAP-Net.
3. Record the trajectory of each point tracker on each frame  $t$ :
  - **Trajectory**  $(x_t, y_t)$ , the 2D coordinates.
  - **Visibility**  $v_t \in \{0, 1\}$ , indicating if the point is visible.

All trajectories and visibility flags are stored to support downstream model training and evaluation. During training, for each video clip, we randomly select 1 to 20 points.

## 4. Experiments

We integrate ATI into two different video generation frameworks: Seaweed-7B [24] and Wan2.1-14B [28]. In Sec. 4.1, we detail our training and inference setups. We evaluate ATI on both frameworks, providing qualitative comparisons in Sec. 4.2 and quantitative analyses in Sec. 4.3.

### 4.1. Implementation Details

We integrate ATI into two video generation frameworks: Seaweed-7B [24] and Wan2.1-14B [28]. Our implementations build on the pre-trained I2V model by injecting the trajectory instruction between the preprocessing stage and the patchify layer. For both models, we fine-tune all DiT parameters for 50,000 iterations using 64 GPUs with 80 GB of VRAM each. All other training hyperparameters follow the standard I2V fine-tuning setup.

**Training and inference time.** Incorporating the ATI module into our video generation pipeline does not significantly affect training or inference times. After 15,000 iterations, both models achieve satisfactory trajectory-following performance. During inference, both the Seaweed-7B ATI and Wan2.1-14B ATI models generate a five-second, 480p video in approximately 8 GPU-minutes.

**Wan2.1 ATI model details.** In the Wan2.1 variant, we handle first-frame conditioning by inserting black frames

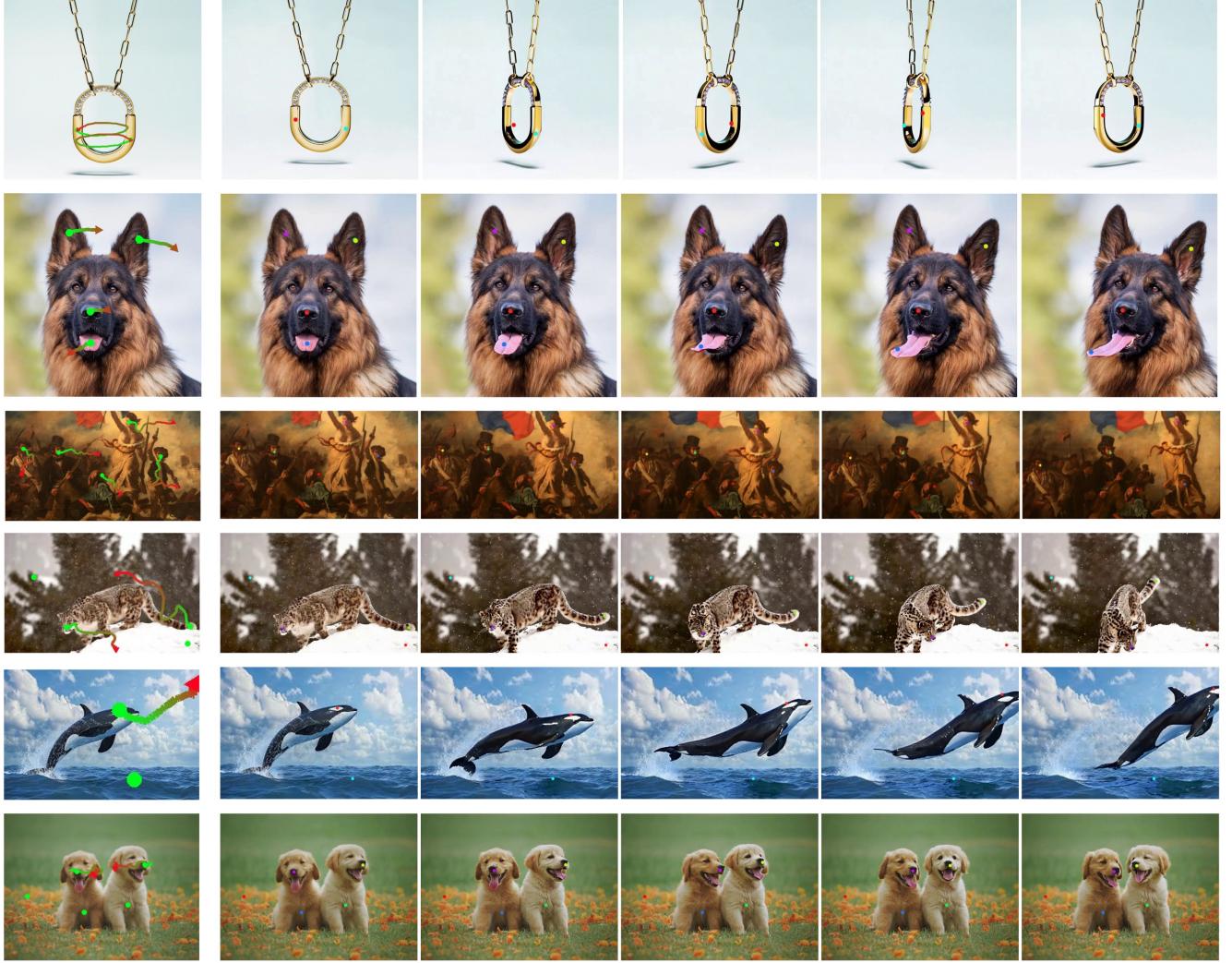


Figure 4. Object Motion Control. Left: the input image overlaid with user-specified trajectories—green dots mark each trajectory’s start point, and arrows mark each end point. Endpoint color encodes trajectory length, indicating that some trajectories span only part of the generated video. Right: five frames uniformly sampled from the generated video. Dot colors serve only to distinguish between trajectories.

immediately after the initial RGB image and feeding this RGB sequence into the VAE encoder to extract latent features. As a result, the latent stream includes features corresponding to the black frames. We then blend the trajectory instruction features with these black-frame VAE features according to the probability scheme described in Sec. 3.2.

**Interactive trajectory editor.** We provide an interactive trajectory editor for creating and refining point trajectories on a single input image. The tool allows users to draw and adjust trajectories, place static points to denote stationary objects, and apply global camera motions such as horizontal panning or zooming in and out.

## 4.2. Qualitative Results

We present video generation results from our ATI model using trajectories created with the tools described in Sec 4.1. Unless otherwise stated, all examples use the Seaweed-7B ATI model.

Figure 4 illustrates outputs for trajectories that emphasize **object motion** and deformation. In the left-hand insets, we overlay the initial frame with the user-defined point trajectories: green dots mark each trajectory’s start point, and arrows mark its end. The color of each endpoint also encodes trajectory length, since some paths span only part of the generated video. On the right, we show five frames uniformly sampled from the generated video. Dot colors in each frame serve only to distinguish between trajectories.

Figure 5 demonstrates the **camera control** capabilities

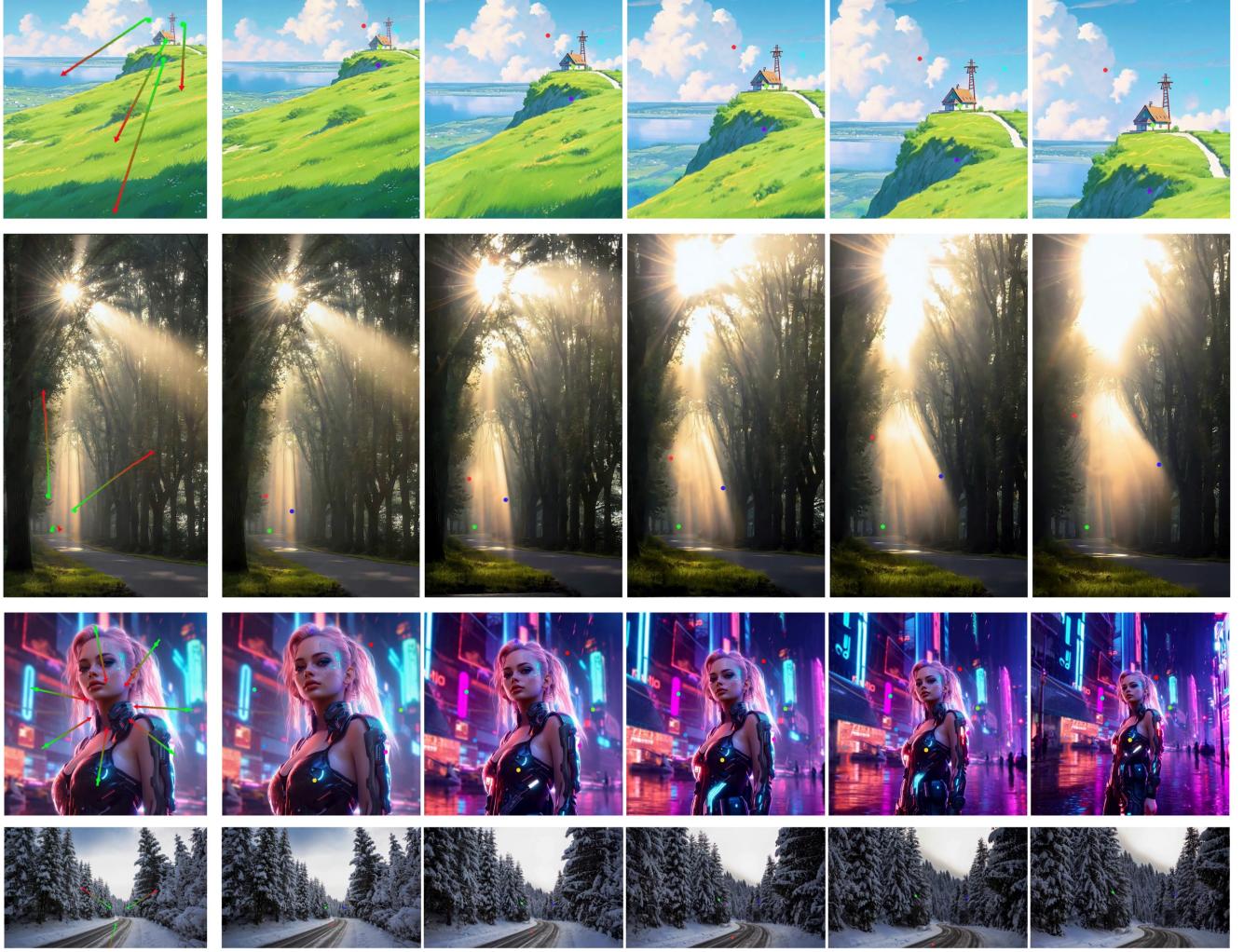


Figure 5. Video generation results with camera control. Left: Input image superimposed with user specified trajectories. Right: Five frames uniformly sampled from the generated video.

of our ATI model. Moving a set of points radially outward from the image center at a constant speed creates a smooth zoom-in effect. Combining this radial motion with a uniform horizontal translation lets us target the zoom on a specific region. By anchoring a static trajectory on the subject while applying the zoom to the background, we reproduce a classic dolly-zoom (as the last example shown). However, if all trajectories consist solely of planar horizontal shifts or zooms, the generated video’s content may remain static, exhibiting only 2D camera movement.

Figure 6 illustrates videos generated under **simultaneous camera and object motion control**. You can create these trajectories by first drawing camera-movement paths and then editing selected ones for object motion, or by defining object-motion trajectories first and subsequently applying the camera-movement transformation.

Figure 7 shows a qualitative comparison between the

Wan2.1 ATI model and the Seaweed ATI model. Overall, we observe that Seaweed ATI demonstrates slightly better trajectory-instruction-following ability, which may be attributable to differences in how the input latent is zero-conditioned (see Sec. 4.1). On the other hand, we observe richer motion for the Wan2.1 ATI model on those unconstrained locations.

Overall, we observe that ATI achieves a high success rate in generating videos that follow the user-specified trajectories, except in the following cases:

- Very rapid movements (*e.g.*, when a point travels half the image width in two frames), which can prevent the model from accurately following the trajectory.
- Trajectories requiring object disassembly (*e.g.*, forcing an object to split into multiple parts), leading to either failure to follow the trajectory or unnatural distortions (*e.g.*, generating an extra cat head).

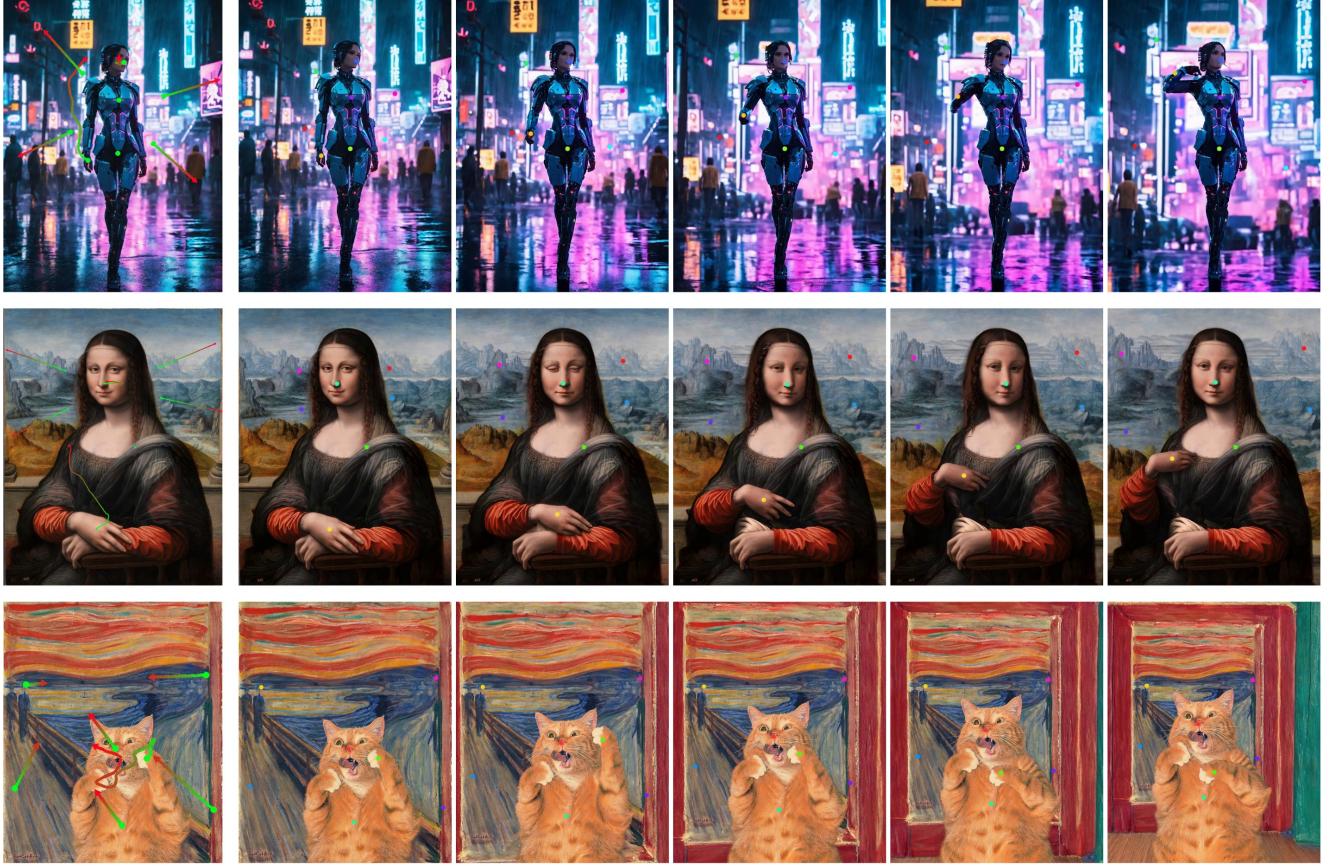


Figure 6. Video generation results with coherent control of camera and object motion. Left: Input image superimposed with user specified trajectories. Right: Five frames uniformly sampled from the generated video.

| ATI Base Model  | Acc@0.05 | Acc@0.01 | App. Rate |
|-----------------|----------|----------|-----------|
| Seaweed-7B [24] | 59.0     | 36.0     | 67.9      |
| Wan2.1-14B [28] | 55.9     | 34.7     | 65.5      |

Table 1. Quantitative results for trajectory instruction following ability of ATI using different base models.

Notably, our approach handles intersecting point trajectories successfully, even when tracking points overlap at certain time steps. We also observe an interesting phenomenon: the ATI model often finds alternative, realistic solutions to satisfy the user’s trajectory instructions (for example, rotating the camera rather than applying implausible object deformations).

### 4.3. Quantitative Results

As shown in Table 1, we quantitatively evaluate the ability of ATI models to follow user-specified point trajectories. First, we collect 100 image–trajectory pairs; for each image, we manually draw between one and ten point trajectories. We then evaluate all ATI models on this test

set. For each generated video, we use TAP-Net to track the points from the first-frame user inputs and compute the per-frame error distance  $d$  between the TAP-Net outputs and the ground-truth trajectories. We introduce three metrics to assess tracking accuracy: **Acc@0.01**, the percentage of frames where the point distance is less than  $0.01 \times$  the image diagonal; **Acc@0.05**, the percentage of frames where the point distance is less than  $0.05 \times$  the image diagonal; and **Appearance Rate**, the proportion of frames in which the tracker correctly predicts a point as visible whenever the user-specified trajectory is present. We report the average value of each metric over the entire test set.

## 5. Conclusion

In this paper, we introduce ATI, a unified trajectory-based control framework that seamlessly integrates camera movement, object translation, and fine-grained local motion within a single latent-space injection module. Our experiments demonstrate that this cohesive approach not only outperforms prior modular methods and commercial systems in both controllability and visual quality, but also remains agnostic to the choice of underlying video

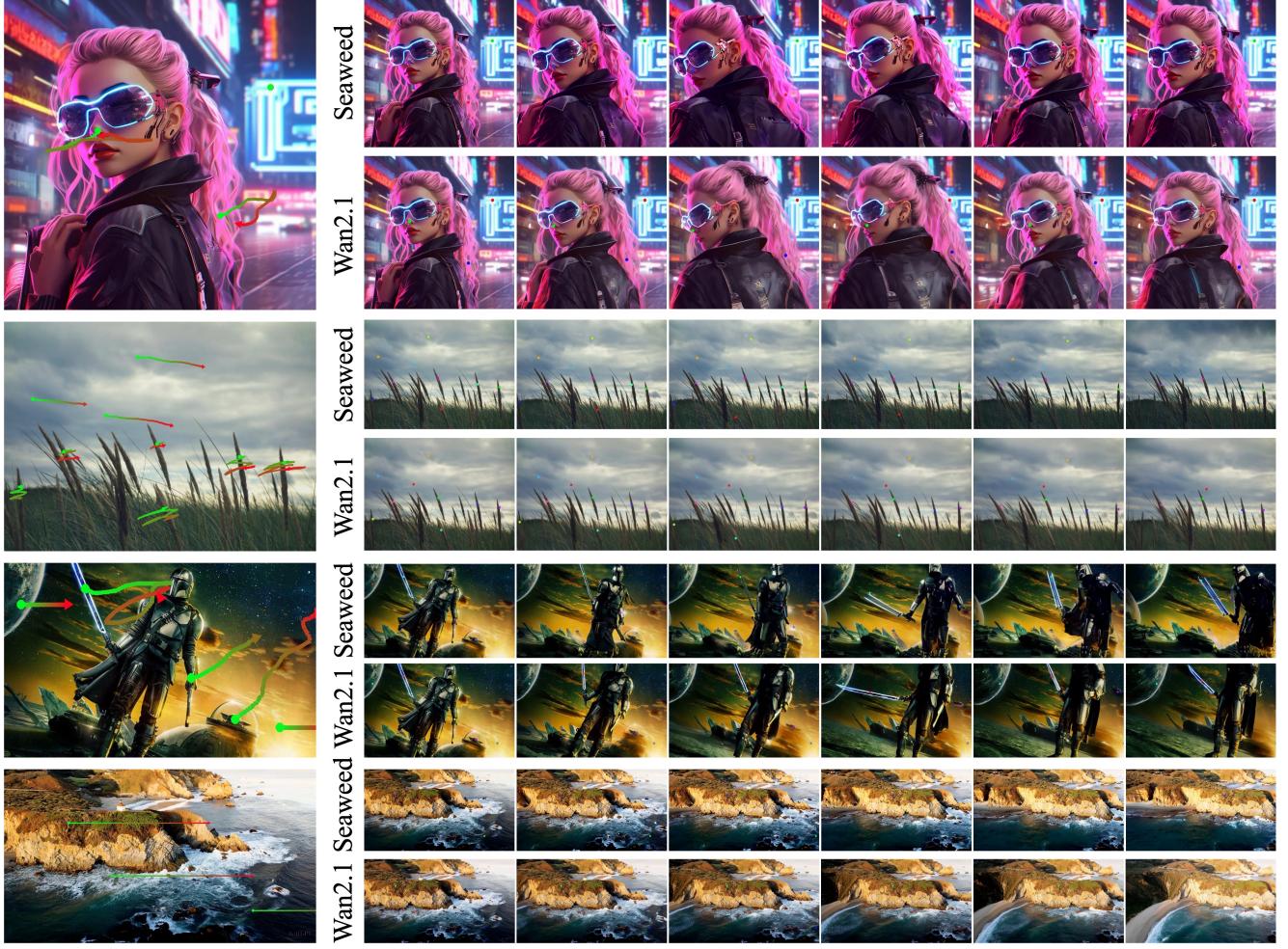


Figure 7. Qualitative comparison for ATI video generation with different backend models.

generation model. In the future, we will further enhance the control capabilities to ensure that object motion better follows both real-world physics and user inputs.

## References

- [1] Omer Bar-Tal, Hila Chefer, Omer Tov, Charles Herrmann, Roni Paiss, Shiran Zada, Ariel Ephrat, and et.al. Lumiere: A space-time diffusion model for video generation. 2024. 1
- [2] Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, Varun Jampani, and Robin Rombach. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint*, 2023.
- [3] Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. Align your latents: High-resolution video synthesis with latent diffusion models. 2023.
- [4] Haoxin Chen, Yong Zhang, Xiaodong Cun, Menghan Xia, Xintao Wang, Chao Weng, and Ying Shan. Videocrafter2: Overcoming data limitations for high-quality video diffusion models. 2024. 1
- [5] Yingjie Chen, Yifang Men, Yuan Yao, Miaomiao Cui, and Liefeng Bo. Perception-as-control: Fine-grained controllable image animation with 3d-aware motion representation. *arXiv preprint arXiv:2501.05020*, 2025. 3
- [6] Carl Doersch, Ankush Gupta, Larisa Markeeva, Adria Recasens, Lucas Smaira, Yusuf Aytar, Joao Carreira, Andrew Zisserman, and Yi Yang. TAP-vid: A benchmark for tracking any point in a video. *Advances in Neural Information Processing Systems*, 35:13610–13626, 2022. 4
- [7] Wanquan Feng, Jiawei Liu, Pengqi Tu, Tianhao Qi, Mingzhen Sun, Tianxiang Ma, Songtao Zhao, Siyu Zhou, and Qian He. I2vcontrol-camera: Precise video camera control with adjustable motion strength. *arXiv preprint arXiv:2411.06525*, 2024. 2
- [8] Daniel Geng, Charles Herrmann, Junhwa Hur, Forrester Cole, Serena Zhang, Tobias Pfaff, Tatiana Lopez-Guevara,

- Carl Doersch, Yusuf Aytar, Michael Rubinstein, et al. Motion prompting: Controlling video generation with motion trajectories. *arXiv preprint arXiv:2412.02700*, 2024. 3
- [9] Xun Guo, Mingwu Zheng, Liang Hou, Yuan Gao, Yufan Deng, Pengfei Wan, Di Zhang, Yufan Liu, Weiming Hu, Zhengjun Zha, et al. I2v-adapter: A general image-to-video adapter for diffusion models. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–12, 2024. 1
- [10] 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 preprint arXiv:2404.02101*, 2024. 2
- [11] Wenyi Hong, Ming Ding, Wendi Zheng, Xinghan Liu, and Jie Tang. Cogvideo: Large-scale pretraining for text-to-video generation via transformers. *arXiv preprint arXiv:2205.15868*, 2022. 1
- [12] Chen Hou, Guoqiang Wei, Yan Zeng, and Zhibo Chen. Training-free camera control for video generation. *arXiv preprint arXiv:2406.10126*, 2024. 2
- [13] Yash Jain, Anshul Nasery, Vibhav Vineet, and Harkirat Behl. Peekaboo: Interactive video generation via masked-diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8079–8088, 2024. 3
- [14] Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuozhuo Dai, Jin Zhou, Jiangfeng Xiong, Xin Li, and et.al. Hunyuanyideo: A systematic framework for large video generative models, 2025. 1
- [15] Zhengfei Kuang, Shengqu Cai, Hao He, Yinghao Xu, Hongsheng Li, Leonidas J Guibas, and Gordon Wetzstein. Collaborative video diffusion: Consistent multi-video generation with camera control. *Advances in Neural Information Processing Systems*, 37:16240–16271, 2024. 2
- [16] Yaowei Li, Xintao Wang, Zhaoyang Zhang, Zhouxia Wang, Ziyang Yuan, Liangbin Xie, Ying Shan, and Yuexian Zou. Image conductor: Precision control for interactive video synthesis. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 5031–5038, 2025. 2
- [17] Bin Lin, Yunyang Ge, Xinhua Cheng, Zongjian Li, Bin Zhu, Shaodong Wang, Xianyi He, Yang Ye, Shanghai Yuan, Luhuan Chen, et al. Open-sora plan: Open-source large video generation model. *arXiv preprint arXiv:2412.00131*, 2024. 1
- [18] Guoqing Ma, Haoyang Huang, Kun Yan, Liangyu Chen, Nan Duan, Shengming Yin, , and et.al. Step-video-t2v technical report: The practice, challenges, and future of video foundation model, 2025. 1
- [19] Wan-Duo Kurt Ma, John P Lewis, and W Bastiaan Kleijn. Trailblazer: Trajectory control for diffusion-based video generation. In *SIGGRAPH Asia 2024 Conference Papers*, pages 1–11, 2024. 3
- [20] Chong Mou, Mingdeng Cao, Xintao Wang, Zhaoyang Zhang, Ying Shan, and Jian Zhang. Revideo: Remake a video with motion and content control. *Advances in Neural Information Processing Systems*, 37:18481–18505, 2024. 3
- [21] Muyao Niu, Xiaodong Cun, Xintao Wang, Yong Zhang, Ying Shan, and Yinqiang Zheng. Mofa-video: Controllable image animation via generative motion field adaptions in frozen image-to-video diffusion model. *arXiv preprint arXiv:2405.20222*, 2024. 2
- [22] William Peebles and Saining Xie. Scalable diffusion models with transformers. *arXiv preprint arXiv:2212.09748*, 2022. 3
- [23] Adam Polyak, Amit Zohar, Andrew Brown, Andros Tjandra, Animesh Sinha, Ann Lee, , and et.al. Movie gen: A cast of media foundation models, 2025. 1
- [24] Team Seaweed, Ceyuan Yang, Zhijie Lin, Yang Zhao, Shanchuan Lin, Zhibei Ma, Haoyuan Guo, Hao Chen, Lu Qi, Sen Wang, et al. Seaweed-7b: Cost-effective training of video generation foundation model. *arXiv preprint arXiv:2504.08685*, 2025. 1, 2, 3, 4, 7
- [25] Xiaoyu Shi, Zhaoyang Huang, Fu-Yun Wang, Weikang Bian, Dasong Li, Yi Zhang, Manyuan Zhang, Ka Chun Cheung, Simon See, Hongwei Qin, et al. Motion-i2v: Consistent and controllable image-to-video generation with explicit motion modeling. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–11, 2024. 2
- [26] Xiaoyu Shi, Zhaoyang Huang, Fu-Yun Wang, Weikang Bian, and Dasong andand et.al Li. Motion-i2v: Consistent and controllable image-to-video generation with explicit motion modeling. *SIGGRAPH 2024*, 2024. 2
- [27] Maham Tanveer, Yang Zhou, Simon Niklaus, Ali Mahdavi Amiri, Hao Zhang, Krishna Kumar Singh, and Nanxuan Zhao. Motionbridge: Dynamic video inbetweening with flexible controls. *arXiv preprint arXiv:2412.13190*, 2024. 2
- [28] Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, and et.al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025. 1, 2, 3, 4, 7
- [29] Hanlin Wang, Hao Ouyang, Qiuyu Wang, Wen Wang, Ka Leong Cheng, Qifeng Chen, Yujun Shen, and Limin Wang. Levitor: 3d trajectory oriented image-to-video synthesis. *arXiv preprint arXiv:2412.15214*, 2024. 3
- [30] Jiawei Wang, Yuchen Zhang, Jiaxin Zou, Yan Zeng, Guoqiang Wei, Liping Yuan, and Hang Li. Boximator: Generating rich and controllable motions for video synthesis. *arXiv preprint arXiv:2402.01566*, 2024. 3
- [31] Zhouxia Wang, Ziyang Yuan, Xintao Wang, Tianshui Chen, Menghan Xia, Ping Luo, and Ying Shan. Motionctrl: A unified and flexible motion controller for video generation. *arXiv preprint*, 2024. 2, 3
- [32] Jianzong Wu, Xiangtai Li, Yanhong Zeng, Jiangning Zhang, Qianyu Zhou, Yining Li, Yunhai Tong, and Kai Chen. Motionbooth: Motion-aware customized text-to-video generation. *arXiv preprint*, 2024. 2
- [33] Weijia Wu, Zhuang Li, Yuchao Gu, Rui Zhao, Yefei He, David Junhao Zhang, Mike Zheng Shou, Yan Li, Tingting Gao, and Di Zhang. Draganything: Motion control for anything using entity representation. In *European Conference on Computer Vision*, pages 331–348. Springer, 2024. 2
- [34] Desai Xie, Zhan Xu, Yicong Hong, Hao Tan, Difan Liu, Feng Liu, Arie Kaufman, and Yang Zhou. Progressive autoregressive video diffusion models, 2024. 1

- [35] Dejia Xu, Weili Nie, Chao Liu, Sifei Liu, Jan Kautz, Zhangyang Wang, and Arash Vahdat. Camco: Camera-controllable 3d-consistent image-to-video generation. *arXiv preprint arXiv:2406.02509*, 2024. [2](#)
- [36] Shiyuan Yang, Liang Hou, Haibin Huang, Chongyang Ma, Pengfei Wan, Di Zhang, Xiaodong Chen, and Jing Liao. Direct-a-video: Customized video generation with user-directed camera movement and object motion. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–12, 2024. [3](#)
- [37] Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazhen Xu, and et.al Yang. Cogvideox: Text-to-video diffusion models with an expert transformer. *arXiv preprint arXiv:2408.06072*, 2024. [1](#)
- [38] Shengming Yin, Chenfei Wu, Jian Liang, Jie Shi, Houqiang Li, Gong Ming, and Nan Duan. Dragnuwa: Fine-grained control in video generation by integrating text, image, and trajectory. *arXiv preprint arXiv:2308.08089*, 2023. [2](#)
- [39] Shoubin Yu, Jacob Zhiyuan Fang, Jian Zheng, Gunnar Sigurdsson, Vicente Ordóñez, Robinson Piramuthu, and Mohit Bansal. Zero-shot controllable image-to-video animation via motion decomposition. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 3332–3341, 2024. [2, 3](#)
- [40] Wangbo Yu, Jinbo Xing, Li Yuan, Wenbo Hu, Xiaoyu Li, Zhipeng Huang, Xiangjun Gao, Tien-Tsin Wong, Ying Shan, and Yonghong Tian. Viewcrafter: Taming video diffusion models for high-fidelity novel view synthesis. *arXiv preprint arXiv:2409.02048*, 2024. [2](#)
- [41] Rui Zhao, Yuchao Gu, Jay Zhangjie Wu, David Junhao Zhang, Jiawei Liu, Weijia Wu, Jussi Keppo, and Mike Zheng Shou. Motiondirector: Motion customization of text-to-video diffusion models. *arXiv preprint*, 2023. [2](#)
- [42] Guangcong Zheng, Teng Li, Rui Jiang, Yehao Lu, Tao Wu, and Xi Li. Cami2v: Camera-controlled image-to-video diffusion model. *arXiv preprint arXiv:2410.15957*, 2024. [2](#)
- [43] Sixiao Zheng, Zimian Peng, Yanpeng Zhou, Yi Zhu, Hang Xu, Xiangru Huang, and Yanwei Fu. Vidcraft3: Camera, object, and lighting control for image-to-video generation. *arXiv preprint arXiv:2502.07531*, 2025. [3](#)
- [44] 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. *arXiv preprint arXiv:2412.20404*, 2024. [1](#)