AKiRa: Augmentation Kit on Rays for optical video generation

Xi Wang[1], Robin Courant[1], Marc Christie[2], Vicky Kalogeiton[1]

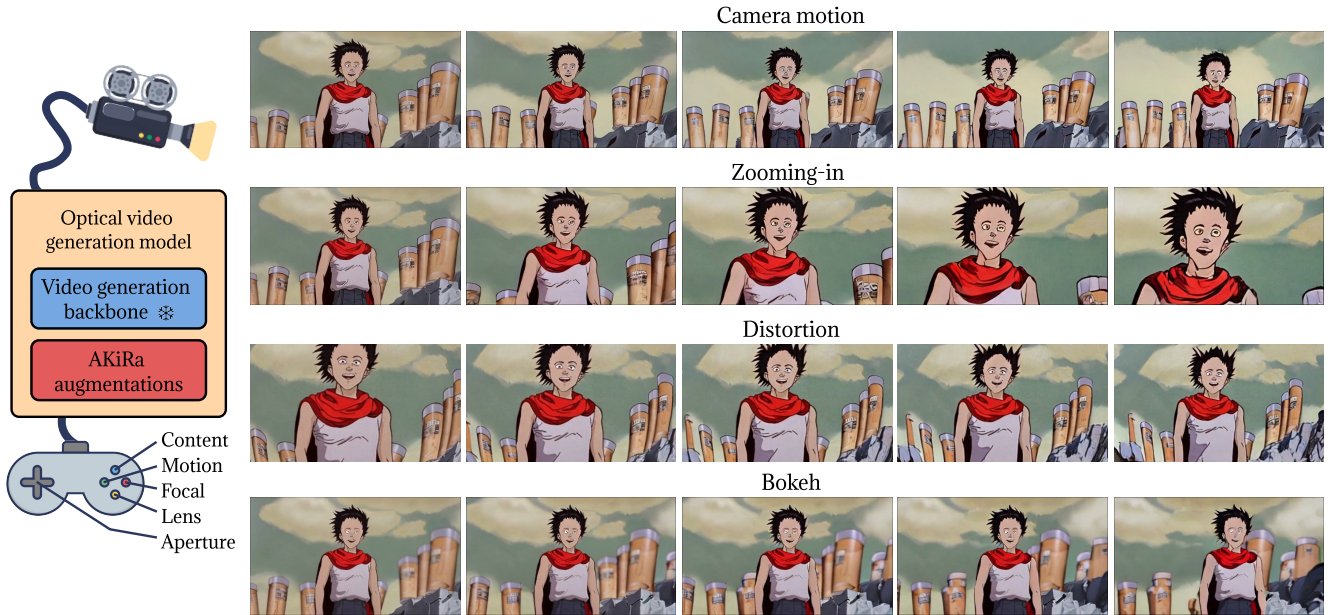[1]LIX, École Polytechnique, IP Paris   [2]Univ Rennes, IRISA, Inria, CNRS

Figure 1. While current state-of-the-art video generation approaches offer limited control to users on camera motion, we propose a dedicated data augmentation framework —AKiRa — to train an optical video generation model that provides users with a panel of controls on camera motions (top row), camera focal length (second row), lens distortion (third row), or bokeh (camera aperture an in/out of focus regions in bottom row). See more in our project page.

Abstract

*Recent advances in text-conditioned video diffusion have greatly improved video quality. However, these methods offer limited or sometimes no control to users on camera aspects, including dynamic camera motion, zoom, distorted lens and focus shifts. These motion and optical aspects are crucial for adding controllability and cinematic elements to generation frameworks, ultimately resulting in visual content that draws focus, enhances mood, and guides emotions according to filmmakers' controls. In this paper, we aim to close the gap between controllable video generation and camera optics. To achieve this, we propose AKiRa (Augmentation Kit on Rays), a novel augmentation framework that builds and trains a camera adapter with a complex camera model over an existing video generation backbone. It enables fine-tuned control over camera motion as well as complex optical parameters (focal length, distortion, aperture) to achieve cinematic effects such as zoom, fish-eye effect, and bokeh. Extensive experiments demonstrate AKiRa's effectiveness in combining and composing camera optics while outperforming all state-of-the-art methods. This work sets a new landmark in **controlled** and **optically enhanced** video generation, paving the way for future optical video generation methods.*

1. Introduction

Creating high-quality video sequences has always been a complex interplay between the content itself and the means, the camera, by which it is portrayed. A powerful narrative or stunning visual concept can only reach its full po-

1

tential when the camera effectively translates it into moving images, capturing nuances through cinematic effects such as dynamic camera motions, zooms, distorted lenses, or intentional focus shifts. Such techniques add texture and depth, drawing the viewer's eye to specific details, enhancing mood, and subtly guiding emotions. When mastered, they allow filmmakers to transform raw content into a rich, immersive experience that feels artful and purposeful.

Recently, video generation have gained notable popularity in the community [5, 18]. Researchers have focused on various aspects, such as improving generation quality [13, 40, 70], increasing resolution [24, 43] or enhancing the efficiency [1, 6, 54]. Despite the remarkable results of modern work in most aforementioned aspects, they often overlook the cinematic techniques, crucial for achieving realistic and immersive storytelling. To tackle this, some recent approaches [21, 56] provide some control over camera motion. Specifically, given raw camera poses and text, these methods generate video following the specified camera trajectory. However, all these approaches simplify the camera model to its *motion* alone and neglect crucial **optical effects** like *focal length (for zoom), lens distortion (fisheye effect), aperture and focus point (for focus shifts)* for three reasons. First, most methods do not use any underlying camera model [13, 40, 70], and instead, they treat videos as a sequence of pixels; hence, despite excessive training, the underlying representations still lack visual coherency. Second, some recent works [21, 56] use simplistic camera models, treating videos as a sequence of posed cameras without lens effects; hence, lack control. Third, there is no adequate training data with optical effects or optical parameters meaning that methods cannot use the data to train for optical camera effects directly. Accounting for such optical parameters attached to the camera is essential, and their absence limits the generation of optically coherent video content and reduces the potential for cinematic quality.

To address the above limitations, we introduce the concept of *optical* video generation models — video generation models which are optically coherent and over which users can control the motion of the camera as well as its optical parameters such as focal length, lens distortion, aperture and focus point. Such a model requires the integration of optical effects into the generation pipeline so that camera and optical parameters are directly leveraged by the model.

To achieve this, we first propose to rely on a Plücker coordinates to represent an image as a collection of camera rays [39, 64] to ensure coherency. We extend the diffusion-based video generation model of [21] by proposing an optically-enhanced camera representation with lens distortion and a dedicated aperture map to encode in- and out-of-focus regions in the screen. Second, we propose a dedicated framework, named AKiRa (Augmentation Kit on Rays), which exploits these extended camera ray rep-

resentations to augment the training dataset, by simulating optical effects on input frames paired with corresponding optical parameters. With this data augmentation method, we then train a camera adapter on top of a pre-trained and frozen video generation backbone. As a result, we provide the first *optical* video generation framework capable of controlling optical camera parameters in addition to camera trajectories, allowing the creation of complex cinematographic effects. The work contributes to closing the gap between the level of control offered to designers in computer graphics worlds and the visual quality of video generative frameworks. We evaluate the generalization of our method across various video generation backbones and on a comprehensive benchmark with robust metrics. We show that our method disentangles camera parameters, specifically separating zoom from translational motion—an achievement not possible with other approaches, where these parameters remain intertwined. Finally, AKiRa outperforms state-of-the-art methods —MotionCtrl [56] and CameraCtrl [21].

Our contributions are: (1) the **first *optical* video generation framework**, which offers the ability to control both camera motions and optics, enabling the generation of videos with complex optical effects (e.g. zoom, fisheye, focus shifts), (2) the design of a **camera model representation including optical parameters** expressed in a Plücker map, extended with an aperture map to model in- and out-of-focus effect; (3) a **joint camera-frame Augmentation Kit on Rays (AKiRa)** modelling optical effects to enable the training of more controllable video generation models.

2. Related work

Our work contributes to the field of *controllable video generation*, which can be approached in two ways: in the first, the generation is guided by text, while in the second way, the generation is guided by camera and text. Another related area is virtual cinematography, which does not generate videos but instead focuses on entities within an existing environment, e.g. camera angles, lighting, and composition.

Text-to-video generation (T2V). The first video diffusion model is introduced by [25], building on the success of image diffusion [14, 23, 45, 46]. Later, Imagen-Video [24] and Make-A-Video [43] introduced cascaded pixel-based diffusion models for high-definition video generation. To reduce training costs, some works [1, 6, 54] perform diffusion in latent space. Others [5, 7, 18, 51] fine-tune temporal adapters on 2D layers of pre-trained text-to-image models [41] using video datasets [3]. Recently, [34, 35, 70] explored transformer backbones (DiT) for scalability, while RIVER [13] and MovieGen [40] moved to flow matching, achieving state-of-the-art performance. These methods rely *solely* on text guidance, which is often sufficient for image generation. However, video generation requires additional complexity, incorporating temporal dynamics and camera

behavior (motion and optics). To address this, we focus on enhancing camera controllability for video generation.

Camera-based T2V. First approaches of T2V [5, 18] fine-tune the video generation model using LoRA [26] to achieve categorical camera motion. Drawing inspiration from the conditional generation problem with auxiliary modules [15, 16, 50, 63, 65], some works [27, 53] condition models on motion maps to manage camera motion, while [57, 61] are restricted to predefined motion directions. Recent works [2, 10, 21, 56, 58, 69] directly integrate precise camera pose parameters into video generation: MotionCtrl [56] initiated by feeding raw camera poses into the model with a trainable adapter; CameraCtrl [21] expanded this work with Plücker coordinates, effectively linking camera intrinsic and extrinsic parameters to image content through ray-based modelling. However, most research [21, 56, 58] focuses solely on camera motion, neglecting factors like focal length, distortion, aperture and focus point. Ignoring focal length, for example, causes current methods to confuse zoom effects with forward and backward translation, resulting in geometric differences [19]. Moreover, most work assumes a distortion-free camera model, limiting generation capacity, as lens distortions (e.g., fisheye effects) are common in artistic videos [4]. Finally, the aperture and focus point are also neglected, despite being powerful effects for guiding image focus. To address these, we propose an *optical video generation model* that incorporates a complex camera model into a video generation backbone and enables control of camera motion, focal length, distortion, aperture and focus point.

Virtual cinematography. Recent works explore methods for integrating cinematic and aesthetics into content generation, via camera control. For instance, [29, 30] synthesize camera trajectories with cinematic styles for 3D animations using reference clips. Cinematic transfer adapts cinematic features from reference clips to new scenes. JAWS [52] pioneered this by optimizing camera trajectories directly within a neural radiation field [36] to match visual cinematic features from a reference clip; follow-up works [8, 32, 55] further refined this, notably adding character re-targeting. Other recent works [12, 31] focus on camera trajectory generation using diffusion models conditioned on text prompts and character motion. In [12] they introduce a dataset of camera trajectories with text extracted from movies, grounding generated trajectories in cinematic features. Yet, these approaches require existing content (e.g. 3D environments [29, 30], clips [52]), thus limiting creative possibilities. This may prevent creators from generating new content, limiting stylistic exploration by encouraging replicating cinematic features from references. Instead, we propose a method that enables cinematic control directly within video generation, removing the need for pre-existing content and expanding users' creative freedom.
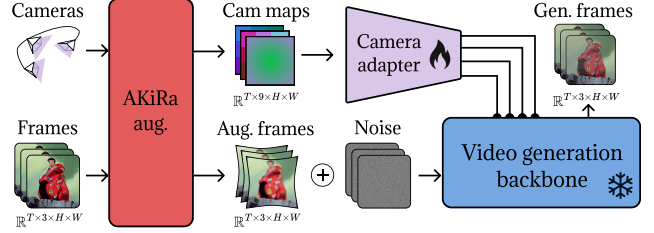

Figure 2. **Overview of AKiRa training.** The camera adapter is trained by jointly augmenting camera data and frames using AKiRa augmentations. The adapter processes multiple camera parameters—motion, focal length, distortion, aperture, and focus point. This adapter is integrated into a pre-trained, frozen backbone, resulting in an optical video generation model.

3. Method

In this section, we present our approach for training an *optical* video generation model, enabling users to manipulate the camera's motion as well as optics to produce cinematic effects such as zoom, distortion, and bokeh.

Overview. To train an optical video generation model capable of controlling both camera motion and optics, we require training data with associated camera parameters. Here, *optics* refer to the camera's optical characteristics, including focal length, lens distortion, aperture and focus point, which together shape how the camera captures field-of-view (zoom), lens characteristics (distortion), and light exposure (in- or out-of-focus). Although some recent datasets pair videos with camera trajectories [12, 71], to our knowledge, there are no datasets that include videos with rich varying optical information. Hence, we propose a set of data augmentations based on a complex camera model that parameterizes *optics* (Section 3.1). To better disentangle each parameter, we design a highly expressive representation for our model (Section 3.2). Finally, we describe the augmentations to generate videos paired with optical parameters (Section 3.3). As illustrated in Figure 2, with these augmentations, we extend prior works [21, 56], by training a camera adapter that controls camera motion and *optics* on top of a pre-trained, frozen video generation backbone.

3.1. Camera model

To train an *optical* video generation model, we need a camera model to represent the camera parameters. Here we propose a camera model by relying on and extending the pinhole camera model to represent not only camera motion and focal length (pinhole camera model) but also the lens distortion and the aperture and focus point (distorted pinhole camera model with aperture), as described below.

Pinhole camera model. We build our camera model upon the standard pinhole camera model [20], which includes extrinsic and intrinsic camera parameters. The *extrinsic*
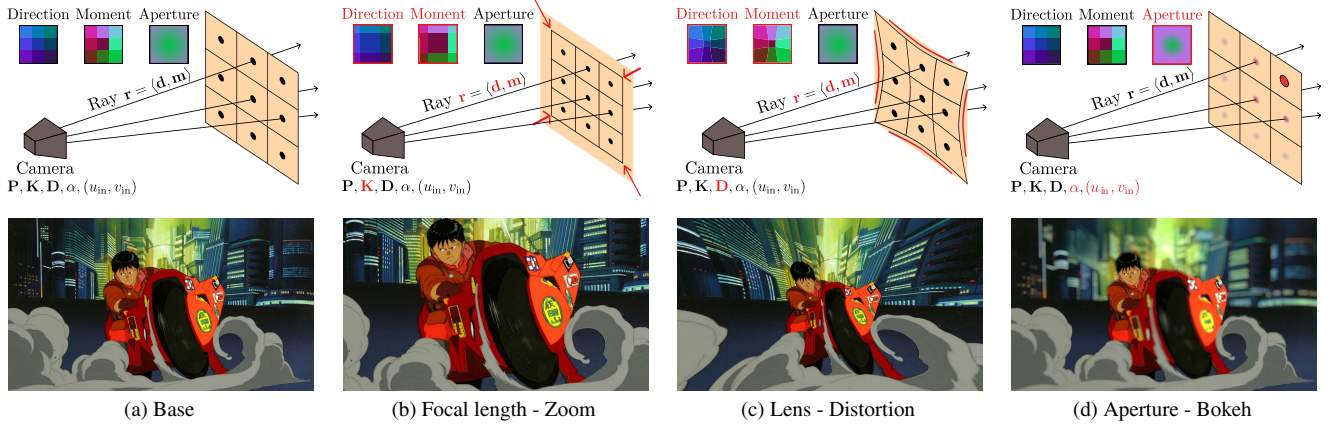
Figure 3. **Optical effect overview**. Visualization of various optical effects proposed in our system —zoom, distortion, and bokeh—and their impacts on both the camera parameters (top row) and visual output (bottom row). In addition, as with state-of-art techniques, we enable the control of the camera motion (not displayed here).

parameters $\mathbf{P} \in \mathbb{R}^{3 \times 4}$ describe the position and orientation of the camera. They are composed of a rotation matrix $\mathbf{R} \in SO(3)$ and translation vector $\mathbf{t} \in \mathbb{R}^3$, giving $\mathbf{P} = [\mathbf{R}|\mathbf{t}] \in SE(3)$. The *intrinsic parameters* $\mathbf{K} \in \mathbb{R}^{3 \times 3}$ include the principal point (c_x, c_y) and the focal length $f \in \mathbb{R}^+$. Specifically, f controls the *zoom-in* (high f) and *zoom-out* (low f) capabilities of the camera (different from forward or backward translation), as shown in Figure 3b. Using this pinhole model, a 3D world point $\mathbf{X} \in \mathbb{R}^3$ is projected to 2D pixel coordinates $(u, v) \in \mathbb{R}^2$ as follows:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}}_{\text{Intrinsic matrix } \mathbf{K}} \underbrace{\begin{bmatrix} \mathbf{R} & \mathbf{t} \\ 0 & 1 \end{bmatrix}}_{\text{Extrinsic matrix } \mathbf{P}} \begin{bmatrix} X \\ 1 \end{bmatrix}. \quad (1)$$

Distorted pinhole camera model. The pinhole model does not inherently account for lens distortion, as it assumes a lens-free setup. To represent more realistic cameras, we extend this model by adding radial distortion parameters $\mathbf{D} \in \mathbb{R}^3$ [44] (e.g., the well-known *fisheye lens* with ultra-wide-angle distortion as shown in Figure 3c). These parameters adjust the pixel coordinates (u, v) radially to $(u_{\mathbf{D}}, v_{\mathbf{D}}) \in \mathbb{R}^2$ with $r = \sqrt{(u - c_x)^2 + (v - c_y)^2}$, follow:

$$\begin{bmatrix} u_{\mathbf{D}} \\ v_{\mathbf{D}} \end{bmatrix} = \begin{bmatrix} u \\ v \end{bmatrix} \left(\mathbf{1}_3 + \mathbf{D} \left[r^2, r^4, r^6 \right] \right) , \quad (2)$$

Distorted pinhole camera model with aperture. The standard pinhole model, with infinitely small aperture, does not capture depth-of-field effects. To simulate the *bokeh effect*—the appearance of in-focus and out-of-focus regions, as illustrated in Figure 3d—we introduce an aperture parameter $\alpha \in \mathbb{R}$. This parameter, along with focus point $(u_{\text{in}}, v_{\text{in}}) \in \mathbb{R}^2$, controls bokeh intensity and location.

3.2. Camera model representation

Ray-based camera model representation. To train our *optical* video generation model, we require an effective camera representation that connects the optical properties of the camera to the generated visual content. For this, we map the geometric camera model $(\mathbf{P}, \mathbf{K}, \mathbf{D})$ to screen pixels $(u, v) \in \mathbb{R}^2$ (or patches in practice) using a ray representation $\mathbf{r}$, where each pixel is associated with a ray (a line) passing through the camera's centre $\mathbf{O} \in \mathbb{R}^3$.

Plücker map. We adopt the Plücker coordinates [39] to represent our camera model, as in [21, 64]. A ray $\mathbf{r} = \langle \mathbf{d}, \mathbf{m} \rangle \in \mathbb{R}^6$ is represented by its direction $\mathbf{d}$ and its moment $\mathbf{m}$ about any point $\mathbf{p}$ on the ray, such that $\mathbf{m} = \mathbf{p} \times \mathbf{d}$. The direction $\mathbf{d}$ is computed by reprojecting the pixel coordinates (u, v) with camera parameters $\mathbf{P}$ and $\mathbf{K}$, and the moment $\mathbf{m}$ is calculated by taking the camera centre $\mathbf{O}$ as the point $\mathbf{p}$ since all rays pass through $\mathbf{O}$:

$$\mathbf{d} = \mathbf{R}^\top \mathbf{K}^{-1} \left[u_{\mathbf{D}}, v_{\mathbf{D}}, 1 \right]^\top , \quad \mathbf{m} = (-\mathbf{R}^\top \mathbf{t}) \times \mathbf{d} . \quad (3)$$

Plücker coordinates $\mathbf{m}$ and $\mathbf{d}$ encode both information about the focal length $\mathbf{K}$ and lens distortion $\mathbf{D}$, as these parameters are inherently linked to ray orientations, as shown in Figures 3b and 3c. For each frame, we derive direction and moment maps with the same dimensions as the frame, associating each pixel with a specific moment and direction.

Aperture map. The camera's aperture parameter α is not captured by Plücker coordinates, as it is not directly ray-related in the Pinhole model, as shown in Figure 3d. To address this, we introduce an aperture map with the same structure and dimensions as the direction and moment maps, assigning an aperture parameter to each pixel in the frame. To achieve this, we first define the coordinates of the focus point $(u_{\text{in}}, v_{\text{in}})$ representing the sharpest point in the frame.
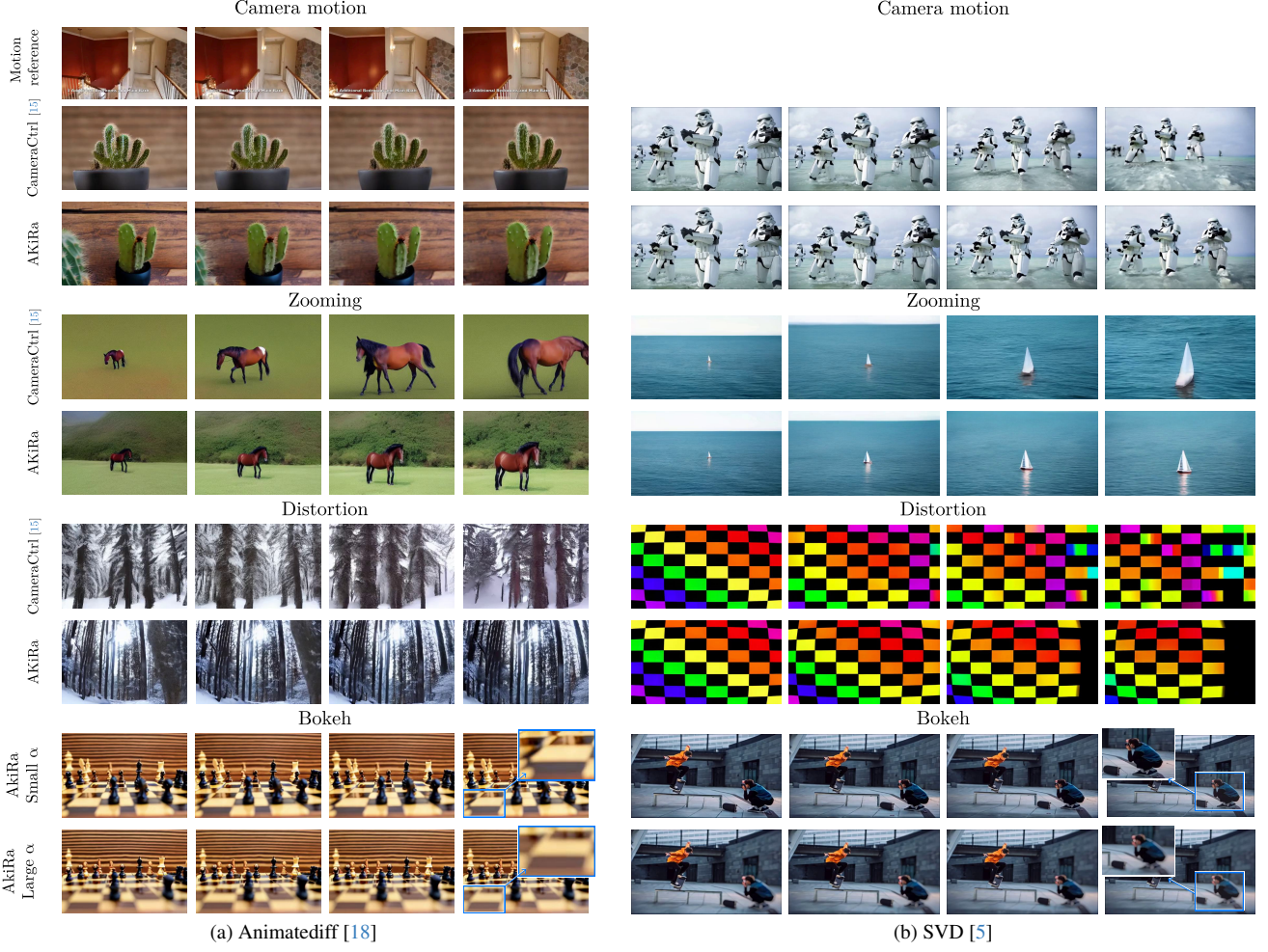
4

Figure 4. **Qualitative results** of AKiRa on Animatediff [18] and SVD [5] backbones. We recommend viewing the supplementary video.

We then define the per-pixel aperture map as $\mathbf{a} \in \mathbb{R}^3$ for any point (u, v) on the frame as:

$$\mathbf{a} = \left[(u - u_{\text{in}}), (v - v_{\text{in}}), \|(u, v) - (u_{\text{in}}, v_{\text{in}})\|^{\frac{1}{\sigma(\alpha)}}\right]^\top. \tag{4}$$

where σ is the sigmoid function. We will discuss the aperture map in the supplementary materials.

Finally, each frame encodes camera information — direction, moment, and aperture $(\mathbf{d}, \mathbf{m}, \mathbf{a})$ — into a 9-dimensional camera map $\mathbb{R}^{9 \times H \times W}$ matching the video frame dimensions.

3.3. AKiRa: Augmentation Kit on Rays

To augment and disentangle optical features in our extended Plücker camera model, we propose AKiRa, an Augmentation Kit on Rays. It contains augmentation techniques for both video frames and corresponding optical parameters:

Zooming - focal length. For the zooming effect, we augment the focal length f in Equation 1 using a **zooming factor** $s \in \mathbb{R}^+$: $s > 1$ represents a zoom-in effect, and $s < 1$ represents zoom-out, both proportional to s.

In image space, changing the focal length can be simulated by a center cropping and resizing the image back to its original resolution. This transformation modifies the image coordinates to $(u_{\text{fl}}, v_{\text{fl}})$ after the focal length change:

$$u_{\text{fl}} = s\,(u - c_x) + c_x, \quad v_{\text{fl}} = s\,(v - c_y) + c_y, \tag{5}$$

where (c_x, c_y) is the principal point of Equation 1. With the Plücker map aligned to image pixels, its augmentation can be performed using Equation 5 (see Figure 3b).

Focal length changes (zoom) are often mistaken for forward/backward movement. While zoom affects only the cropping area, translation induces **perspective changes** (see Figure 5). Such ambiguities can hinder the accurate interpretation of focal length changes and translational motion during training. Thus, augmenting focal length is essential
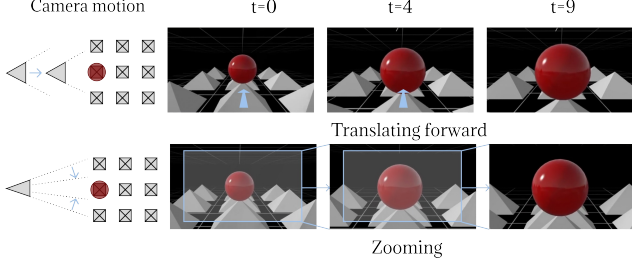
Figure 5. **Difference between zoom and push forward.** Zooming (change of focal length) is similar to image cropping and resizing while pushing forward changes the perspective of the scene.

to **disambiguate** these effects, reducing confusion between zooming and movement, and enabling richer optical compositionality: e.g. moving right while zooming in or moving forward while zooming out (commonly referred to as a dolly zoom), which are popular in modern cinematography.

Distortion - lens. Augmentation on distortion involves modifying the radial distortion coefficients **D** as defined in Equation 2. In alignment with focal-length augmentation, this transformation is applied simultaneously to both the image and Plücker coordinates, as shown in Figure 3c.

Directly augmenting distortion in image space can create undefined areas as the image stretches to a non-rectangular shape. To resolve this, we compute a zooming factor s to ensure the image can be cropped without undefined borders and incorporate this into the zooming augmentation.

Bokeh - aperture. The Bokeh effect depends on the aperture α, the depth information of the scene and a focus point $(u_{\text{in}}, v_{\text{in}})$. More precisely, larger depth distances result in a larger blur radius $b_r \in \mathbb{R}$, which can be approximately described as [38, 62] with d_{in} the disparity value (invert depth) of the focus point: $b_r = \alpha |d - d_{\text{in}}|$. We estimate depth [60] and then use virtual bokeh rendering [38, 62] based on b_r to augment both the aperture α and focus point $(u_{\text{in}}, v_{\text{in}})$ to simulate bokeh effects (Figure 3d).

Augmentation Algorithm *Sampling.* In our augmentation kit, we randomly sample parameters for every frame independently. However, direct random sampling may introduce flickering artifacts as between frames the number may change too fast, which is uncommon in the real-world and can harm the learning. To prevent this, we limit the rate of change between frames and ensure smooth transitions by applying spline interpolation to the sampled parameters.

Dropout. We also apply augmentation dropout to ensure that: (1) a certain percentage of the original video frames are used during training, and (2) the model learns specific combinations of augmentations effectively.

Algorithm. Our augmentation starts with bokeh using the pre-estimated depth, then applies distortion on the original images to use the maximum resolution, and followed by focal length changes, see more in the supplementary material.

4. Experiments

4.1. Evaluation metrics

For video quality of generated videos, we report the commonly used Fréchet Inception Distance (*FID*) [22], Fréchet Video Distance (*FVD*) [49]. Given that recent works [17, 59] point out that the *FVD* tends to be biased toward content while overlooking the temporal aspects, we propose to report the Content-Debiased FVD (*CD-FVD*) [17], considered more relevant.

Evaluating camera motion fidelity is challenging for generated content. Some approaches use SLAM and pose estimation methods [37, 42, 67] to evaluate the estimated trajectory from generated video content (MotionCtrl [56], CameraCtrl [21]). However, these often assume (partial) static consistency, which is hard to maintain in the generated content due to flickering or unrealistic motion artefacts but irrelevant to motion quality; (ii) the absence of precise camera model definitions (e.g., focal length, distortion) further complicates trajectory-based assessments [33, 47] given that optic parameters greatly influence motion interpretation; (iii) trajectory-based methods are computationally intensive (can reach ∼10 minutes per 16-frame video [67]) and struggle to scale efficiently for large-scale video assessments. See motion evaluation in supplementary materials.

Therefore, to show the **camera motion fidelity towards the control** we report two metrics: first, following [21, 56] reporting trajectory errors, we estimate dense camera poses using ParticleSfM [67] but calculate scale-corrected rotational and translational relative pose errors between frames (*RPE-R* and *RPE-t*, respectively). The choice of using scale correction and relative pose aims at reducing the unstable estimation of trajectory-based metrics [33, 47].

Additionally, to measure **motion alignment** between reference and generated videos we propose a flow similarity metric (*FlowSim*). This metric relies less on camera model parameters and focuses primarily on frame-to-frame motion. Moreover, the optical flow-based approach is computationally efficient and scalable with GPU parallelism. For this, we estimate the optical flow of both videos $\mathbf{F}_r$ and $\mathbf{F}_g$ using RAFT [48], then extract the flow magnitude $\|\mathbf{F}\|$ and direction φ. To filter out residual noise, we consider only flow components with a magnitude above a set threshold t, and we compute the cosine similarity of the directional components as follows, with $\mathbb{1}$ an indicator function:

$$\text{FlowSim}(\mathbf{F}_r, \mathbf{F}_g) = \mathbb{1}_{\|\mathbf{F}_r\| \& \|\mathbf{F}_r\| > t} \cdot \varphi_r \cdot \varphi_g, \quad (6)$$

Dynamic consistency (VBench). We also assess the **temporal dynamic consistency** of generated videos with the subject consistency (*Consistency*), smoothness (*Smoothness*) and temporal flickering (*Flickering*) metrics proposed in the video benchmark suite VBench [28].

6

	Method	Video quality			Camera motion fidelity			Dynamic consistency (VBench)		
Backbone	Camera control	FID ↓	FVD ↓	CD-FVD ↓	RPE-R (deg)↓	RPE-t (cm)↓	FlowSim ↑	Consistency ↑	Smoothness ↑	Flickering↑
AnimateDiff [18]	-	65.09	835.2	768.6	4.354	3.048	0.00	0.845	0.843	0.792
	MotionCtrl [56]	104.64	822.9	449.7	0.443	1.857	58.50	0.961	0.965	0.939
	CameraCtrl [21]	61.62	384.5	355.2	0.730	1.807	53.74	0.965	0.960	0.942
	AKiRa (ours)	61.52	332.3	328.7	0.438	1.480	70.97	0.969	0.974	0.957
SVD [5]	-	29.94	351.0	440.8	1.837	2.541	0.56	0.916	0.959	0.930
	MotionCtrl [56]	57.09	217.3	460.8	1.705	1.343	11.96	0.916	0.979	0.950
	CameraCtrl [21]	32.32	173.8	424.9	0.329	1.322	79.64	0.972	0.991	0.969
	AKiRa (ours)	32.58	162.8	398.3	0.295	1.321	80.11	0.981	0.993	0.975

Table 1. **Comparison with the state-of-the-art.** Comparison of AKiRa and concurrent methods with different backbones on WebVid dataset, evaluating video quality, camera motion fidelity, and dynamic consistency. First best and second best .

Optical consistency. To evaluate zoom (*ZoomSim*) and distortion (*DistortSim*), we use *FlowSim* to measure the similarity between generated and theoretical optical flow (see supplementary materials). For evaluating bokeh effects, inspired by [11], we use an off-the-shelf defocus detector [68] and report the in-focus area (*FocusArea*) across varying aperture levels: a low aperture results in a wide in-focus area (everything appears in focus), while a high aperture produces a narrow in-focus area, showing selective focus.

4.2. Quantitative comparison

Comparison to the state of the art. Table 1 reports the performance of AKiRa against state-of-the-art camera control approaches for video generation: MotionCtrl [56] and CameraCtrl [21]. We evaluate video quality, camera motion fidelity and temporal dynamic consistency metrics (Section 4.1) for text-to-video (T2V) backbone Animatediff [18] and image-to-video (I2V) backbone SVD [5], also reporting baselines without any camera control module (-). All metrics are computed on $1,000$ generated samples using random text prompts or conditioning frames from the WebVid dataset [3]. For AKiRa, as we explicitly control the bokeh, to improve the realism of the generated results, we set $\alpha = 50$ (see discussion in supplementary materials).

In video quality, AKiRa achieves leading performance across quality, camera motion fidelity, and dynamic consistency metrics, outperforming other state-of-the-art methods. Especially, for FVD and CD-FVD, AKiRa scores best with both AnimateDiff (332.3, 328.7) and SVD (162.8, 398.3), outperforming CameraCtrl's 384.5 and 355.2 on AnimateDiff, and 173.8 and 424.9 on SVD.

For camera motion fidelity, we observe that: (1) AKiRa achieves the highest FlowSim scores (AnimateDiff: 70.97; SVD: 80.11), ahead of MotionCtrl 58.50 with AnimateDiff, and CameraCtrl 79.64 with SVD, and lowest motion errors, both translational and rotational (0.438 and 1.480 on Animatediff); (2) the proposed metric, FlowSim, assigned close to zero values to random motion, thus proportionally reflecting camera pose errors with improved interpretability.

For dynamic consistency, AKiRa leads in VBench scores, e.g. achieving 0.969, 0.974, 0.957 on AnimateDiff, and 0.981, 0.993, 0.975 on SVD. These results validate that disentangling camera components during training enables high-quality and consistent video synthesis.

Optical comparison. In Table 3, we show the optical consistency performances (Section 4.1). We focus on CameraCtrl [21] since its ray-based representation supports zoom and distortion effects, though does not account for bokeh, whereas MotionCtrl [56] lacks these effects due to its camera model. We also report raw Animatediff [18] as the baseline without any control module (-). We generate dedicated datasets of 1,000 samples for each category. For zoom and distortion, we create static motion videos with random zoom-in/zoom-out effects and distortions. For bokeh, we generate 1,000 samples with varying apertures (α=0/30/100). Table 3 highlights that AKiRa outperforms both AnimateDiff and CameraCtrl in optical coherence metrics, achieves the highest *ZoomSim* and *DistortSim* scores (86.82 and 81.19), showing superior zoom and distortion handling compared to CameraCtrl. In terms of bokeh, we observe a decreasing trend of the *FocusArea* with increasing aperture α from 0 to 100, confirming the controllability of the aperture feature (a larger aperture results in a bigger bokeh area). These results highlight AKiRa's ability to control zoom, distortion, and focus effects.

User study. We conducted a user study with 25 participants, evaluating (1) video, (2) text-to-video, (3) motion, (4) zoom, (5) distortion effect, and (6) bokeh effect qualities. Participants viewed 10 samples, with each generated by: the baseline (no camera control module), MotionCtrl [56], CameraCtrl [21], and our AKiRa, both the Animatediff [18] and SVD [5] backbones. For each criterion, participants ranked the methods from 1 (best) to 4 (worst).

In Table 2 we report the proportion of participants ranking each method as 1st across various criteria (with 25% as a random baseline). Results show that AKiRa consistently ranks highest, with an average ranking proportion of 44.2%, compared to 29.2% for the next best method, CameraCtrl [21]. Specifically, 43.6% of participants preferred CameraCtrl's zoom effect, followed closely by AKiRa at 35.6%.

Figure 6. **Comparison of dolly zoom** effect on SVD (I2V) [5] backbone between AKiRa, CameraCtrl[21] and the reference from *Guardians of the Galaxy Vol. 2* (*Gunn, 2017*). Note that the background recedes while the character remains static.

Method	Video	Text	Motion	Zoom	Distortion	Bokeh	Average
Baseline [5, 18]	13.2	11.2	7.2	4.0	8.4	7.2	8.5
MotionCtrl [56]	27.6	6.4	21.6	16.8	18.8	17.6	18.1
CameraCtrl [21]	19.6	27.2	23.6	43.6	29.6	31.6	29.2
AKiRa (ours)	39.6	55.2	47.6	35.6	43.2	43.6	44.2

Table 2. **Results of user study.** % participants ranking each method as 1st for: *Video* quality, *Text-to-video* consistency, motion fidelity, and optical coherence (*Zoom*, *Distortion*, *Bokeh*). First best and second best.

Method	ZoomSim ↑	DistortSim ↑	FocusArea		
			$\alpha=0$	$\alpha=30$	$\alpha=100$
AnimateDiff [18]	0.00	0.00	0.90	-	-
CameraCtrl [21]	78.98	62.13	0.69	-	-
AKiRa (ours)	86.82	81.19	0.72	0.64	0.61

Table 3. **Optical comparison** of AKiRa and CameraCtrl [21] with Animatediff [18] backbone on WebVid, evaluating optical coherence (zoom, distortion and bokeh).

Focal	Lens	Aperture	CD-FVD ↓	FlowSim ↑
			355.2	53.74
✓			351.2	67.25
	✓		360.1	71.82
		✓	375.8	73.27
✓	✓		359.4	66.86
✓		✓	332.7	62.93
	✓	✓	331.0	71.97
✓	✓	✓	328.7	70.97

Table 4. **Ablation of augmentations.** Ablation of each augmentation combination in AKiRa. All models are trained on Animatediff [18] backbone.

This preference likely stems from CameraCtrl's overly fast and impactful zoom effect, though it lacks subject consistency upon closer inspection (see Section 4.3 and Figure 4). Overall, AKiRa notably enhances the aesthetic quality of the backbone, with 39.6% and 55.2% of participants ranking it 1st for video quality and text-video consistency, compared to only 13.2% and 11.2% for the baselines.

Ablation study. In Table 4, we analyze the impact of each augmentation in AKiRa on overall model performance. For each augmentation combination, we train a model using the Animatediff [18] backbone. The results show that applying all augmentations yields the best performances in both visual (*CD-FVD*) and motion (*FlowSim*) quality. The strong performance of distortion with bokeh (second-to-last row) is due to the cropping and rescaling process intrinsic to distortion augmentation (see Sec. 3), which implicitly works as a focal length adjustment, leading to comparable results. Additionally, the high *FlowSim* score for bokeh-only augmentation (fourth row) might be due to overly smoothed content, which implicitly improves the similarity measure.

4.3. Qualitative results

Qualitative comparison. In Figure 4 we present a qualitative comparison of the various effects—motion, zoom, distortion, and bokeh—between AKiRa and CameraCtrl [21], using both the Animatediff [18] and SVD [5] backbones. For motion (first part), AKiRa accurately reproduces reference motion from the RealEstate dataset [71] across both backbones. In zooming (second part), our method demonstrates better consistency, while CameraCtrl struggles; for example, in the SVD examples (second part, second col-

umn), AKiRa maintains the boat's position and orientation, unlike CameraCtrl, where the boat shows deformation. In terms of distortion (third part), CameraCtrl fails to achieve this effect, whereas AKiRa successfully applies it; for instance, in the Animatediff example (third part, first column), the tree trunks visibly distort. Lastly, for the bokeh effect (fourth part), AKiRa adds adjustable out-of-focus blur. In the Animatediff example (fourth part, first column), for instance, the foreground of the chessboard displays varying blur levels depending on the aperture.

A special case: dolly zoom. A dolly zoom is an iconic cinematic effect that produces a dramatic perspective shift, achieved by simultaneously pulling while zooming in reverse direction. We show in Figure 6 that AKiRa successfully reproduces the dolly zoom effect, whereas CameraCtrl [21] achieves only a pure zoom-out effect. This is due to AKiRa 's precise disentangling of key camera parameters, especially focal length and camera position. We recommend viewing the supplementary video.

5. Conclusion

In this paper, we introduce the concept of optical video generation, a framework that allows users to control camera motions as well as optical parameters. We trained a dedicated camera adapter using a set of data augmentation techniques —AKiRa — that pairs camera/lens parameters with corresponding videos. Results show that our framework generates optically coherent content, outperforming state-of-the-art approaches while offering extra control. AKiRa expands the possibilities of video generation and bridges the gap between synthetic and real-world capabilities.

Acknowledgments

References

[1] Jie An, Songyang Zhang, Harry Yang, Sonal Gupta, Jia-Bin Huang, Jiebo Luo, and Xi Yin. Latent-shift: Latent diffusion with temporal shift for efficient text-to-video generation. *arXiv preprint arXiv:2304.08477*, 2023. 2

[2] Sherwin Bahmani, Ivan Skorokhodov, Aliaksandr Siarohin, Willi Menapace, Guocheng Qian, Michael Vasilkovsky, Hsin-Ying Lee, Chaoyang Wang, Jiaxu Zou, Andrea Tagliasacchi, et al. Vd3d: Taming large video diffusion transformers for 3d camera control. *arXiv preprint 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. 2, 7, 11, 12

[4] Marcelo Bertalmío. *Image processing for cinema*. CRC Press, 2014. 3

[5] Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023. 2, 3, 5, 7, 8, 11, 12, 15

[6] 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. In *CVPR*, 2023. 2

[7] Haoxin Chen, Menghan Xia, Yingqing He, Yong Zhang, Xiaodong Cun, Shaoshu Yang, Jinbo Xing, Yaofang Liu, Qifeng Chen, Xintao Wang, et al. Videocrafter1: Open diffusion models for high-quality video generation. *arXiv preprint arXiv:2310.19512*, 2023. 2

[8] Weiliang Chen, Fangfu Liu, Diankun Wu, Haowen Sun, Haixu Song, and Yueqi Duan. Dreamcinema: Cinematic transfer with free camera and 3d character. *arXiv preprint arXiv:2408.12601*, 2024. 3

[9] Jiyu Cheng, Yuxiang Sun, and Max Q-H Meng. Improving monocular visual slam in dynamic environments: an optical-flow-based approach. *Advanced Robotics*, 2019. 16

[10] Soon Yau Cheong, Duygu Ceylan, Armin Mustafa, Andrew Gilbert, and Chun-Hao Paul Huang. Boosting camera motion control for video diffusion transformers. *arXiv preprint arXiv:2410.10802*, 2024. 3

[11] Robin Courant, Christophe Lino, Marc Christie, and Vicky Kalogeiton. High-level features for movie style understanding. In *ICCV-W*, 2021. 7

[12] Robin Courant, Nicolas Dufour, Xi Wang, Marc Christie, and Vicky Kalogeiton. E.T. the Exceptional Trajectories: text-to-camera-trajectory generation with character awareness. In *ECCV*, 2024. 3

[13] Aram Davtyan, Sepehr Sameni, and Paolo Favaro. Efficient video prediction via sparsely conditioned flow matching. In *ICCV*, 2023. 2

[14] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *NeurIPS*, 2021. 2

[15] Nicolas Dufour, David Picard, and Vicky Kalogeiton. Scam! transferring humans between images with semantic cross attention modulation. In *ECCV*. Springer, 2022. 3

[16] Jon Gauthier. Conditional generative adversarial nets for convolutional face generation. *Class project for Stanford CS231N: convolutional neural networks for visual recognition, Winter semester*, 2014. 3

[17] Songwei Ge, Aniruddha Mahapatra, Gaurav Parmar, Jun-Yan Zhu, and Jia-Bin Huang. On the content bias in fréchet video distance. In *CVPR*, 2024. 6

[18] Yuwei Guo, Ceyuan Yang, Anyi Rao, Zhengyang Liang, Yaohui Wang, Yu Qiao, Maneesh Agrawala, Dahua Lin, and Bo Dai. Animatediff: Animate your personalized text-to-image diffusion models without specific tuning. In *ICLR*, 2023. 2, 3, 5, 7, 8, 11, 12, 13, 15

[19] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003. 3

[20] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2004. 3, 15

[21] 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, 3, 4, 6, 7, 8, 11, 12, 14

[22] 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. *NeurIPS*, 2017. 6

[23] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *NeurIPS*, 2020. 2

[24] Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P Kingma, Ben Poole, Mohammad Norouzi, David J Fleet, et al. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022. 2

[25] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *NeurIPS*, 2022. 2

[26] Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. In *ICLR*, 2022. 3, 11

[27] Teng Hu, Jiangning Zhang, Ran Yi, Yating Wang, Hongrui Huang, Jieyu Weng, Yabiao Wang, and Lizhuang Ma. Motionmaster: Training-free camera motion transfer for video generation. *arXiv preprint arXiv:2404.15789*, 2024. 3

[28] Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, et al. Vbench: Comprehensive benchmark suite for video generative models. In *CVPR*, 2024. 6

[29] Hongda Jiang, Bin Wang, Xi Wang, Marc Christie, and Baoquan Chen. Example-driven virtual cinematography by learning camera behaviors. *ACM TOG*, 2020. 3

[30] H. Jiang, M. Christie, X. Wang, L. Liu, B. Wang, and B. Chen. Camera keyframing with style and control. *ACM TOG*, 2021. 3

[31] Hongda Jiang, Xi Wang, Marc Christie, Libin Liu, and Baoquan Chen. Cinematographic camera diffusion model. In *Computer Graphics Forum*, 2024. 3

[32] Xuekun Jiang, Anyi Rao, Jingbo Wang, Dahua Lin, and Bo Dai. Cinematic behavior transfer via nerf-based differentiable filming. In *CVPR*, 2024. 3

[33] Li Jinyu, Yang Bangbang, Chen Danpeng, Wang Nan, Zhang Guofeng, and Bao Hujun. Survey and evaluation of monocular visual-inertial slam algorithms for augmented reality. *Virtual Reality & Intelligent Hardware*, 2019. 6

[34] Xin Ma, Yaohui Wang, Gengyun Jia, Xinyuan Chen, Ziwei Liu, Yuan-Fang Li, Cunjian Chen, and Yu Qiao. Latte: Latent diffusion transformer for video generation. *arXiv preprint arXiv:2401.03048*, 2024. 2

[35] Willi Menapace, Aliaksandr Siarohin, Ivan Skorokhodov, Ekaterina Deyneka, Tsai-Shien Chen, Anil Kag, Yuwei Fang, Aleksei Stoliar, Elisa Ricci, Jian Ren, et al. Snap video: Scaled spatiotemporal transformers for text-to-video synthesis. In *CVPR*, 2024. 2

[36] B. Mildenhall, P. Srinivasan, M. Tancik, J. Barron, R. Ramamoorthi, and R. Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020. 3

[37] Linfei Pan, Dániel Baráth, Marc Pollefeys, and Johannes L Schönberger. Global structure-from-motion revisited. In *ECCV*, 2024. 6

[38] Juewen Peng, Zhiguo Cao, Xianrui Luo, Hao Lu, Ke Xian, and Jianming Zhang. Bokehme: When neural rendering meets classical rendering. In *CVPR*, 2022. 6

[39] Julius Plücker. *Analytisch-geometrische Entwicklungen*. GD Baedeker, 1828. 2, 4

[40] Adam Polyak, Amit Zohar, Andrew Brown, Andros Tjandra, Animesh Sinha, Ann Lee, Apoorv Vyas, Bowen Shi, Chih-Yao Ma, Ching-Yao Chuang, et al. Movie gen: A cast of media foundation models. *arXiv preprint arXiv:2410.13720*, 2024. 2

[41] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. 2

[42] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, 2016. 6, 14

[43] Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, et al. Make-a-video: Text-to-video generation without text-video data. *arXiv preprint arXiv:2209.14792*, 2022. 2

[44] Chester C Slama, Charles Theurer, and Soren W Henriksen. *Manual of photogrammetry*. American Society of Photogrammetry, 1980. 4

[45] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *ICML*, 2015. 2

[46] Yang Song and Stefano Ermon. Improved techniques for training score-based generative models. *NeurIPS*, 2020. 2

[47] Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. A benchmark for the evaluation of rgb-d slam systems. In *Proc. IEEE/RSJ Conf. on Intelligent Robots and Systems*, 2012. 6, 14

[48] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *ECCV*, 2020. 6, 17

[49] Thomas Unterthiner, Sjoerd Van Steenkiste, Karol Kurach, Raphael Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric & challenges. *arXiv preprint arXiv:1812.01717*, 2018. 6

[50] Aaron Van den Oord, Nal Kalchbrenner, Lasse Espeholt, Oriol Vinyals, Alex Graves, et al. Conditional image generation with pixelcnn decoders. *NeurIPS*, 2016. 3

[51] Jiuniu Wang, Hangjie Yuan, Dayou Chen, Yingya Zhang, Xiang Wang, and Shiwei Zhang. Modelscope text-to-video technical report. *arXiv preprint arXiv:2308.06571*, 2023. 2

[52] Xi Wang, Robin Courant, Jinglei Shi, Eric Marchand, and Marc Christie. Jaws: just a wild shot for cinematic transfer in neural radiance fields. In *CVPR*, 2023. 3

[53] Xiang Wang, Hangjie Yuan, Shiwei Zhang, Dayou Chen, Jiuniu Wang, Yingya Zhang, Yujun Shen, Deli Zhao, and Jingren Zhou. Videocomposer: Compositional video synthesis with motion controllability. *NeurIPS*, 2024. 3

[54] Yaohui Wang, Xinyuan Chen, Xin Ma, Shangchen Zhou, Ziqi Huang, Yi Wang, Ceyuan Yang, Yinan He, Jiashuo Yu, Peiqing Yang, et al. Lavie: High-quality video generation with cascaded latent diffusion models. *arXiv preprint arXiv:2309.15103*, 2023. 2

[55] Zhenzhi Wang, Yixuan Li, Yanhong Zeng, Youqing Fang, Yuwei Guo, Wenran Liu, Jing Tan, Kai Chen, Tianfan Xue, Bo Dai, et al. Humanvid: Demystifying training data for camera-controllable human image animation. *arXiv preprint arXiv:2407.17438*, 2024. 3

[56] 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. 2, 3, 6, 7, 8, 11, 12, 14

[57] 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. *CoRR*, 2024. 3

[58] 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. 3, 14

[59] Ge Ya, Gian Favero, Zhi Hao Luo, Alexia Jolicoeur-Martineau, Christopher Pal, et al. Beyond fvd: Enhanced evaluation metrics for video generation quality. *arXiv preprint arXiv:2410.05203*, 2024. 6

[60] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. *arXiv preprint arXiv:2406.09414*, 2024. 6

[61] 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 *SIGGRAPH*, 2024. 3

[62] Yang Yang, Haiting Lin, Zhan Yu, Sylvain Paris, and Jingyi Yu. Virtual dslr: High quality dynamic depth-of-field synthesis on mobile platforms. *Electronic Imaging*, 2016. 6

[63] Hu Ye, Jun Zhang, Sibo Liu, Xiao Han, and Wei Yang. Ip-adapter: Text compatible image prompt adapter for text-to-image diffusion models. *arXiv preprint arXiv:2308.06721*, 2023. 3

[64] Jason Y. Zhang, Amy Lin, Moneish Kumar, Tzu-Hsuan Yang, Deva Ramanan, and Shubham Tulsiani. Cameras as rays: Pose estimation via ray diffusion. In *ICLR*, 2024. 2, 4

[65] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *ICCV*, 2023. 3

[66] Tianwei Zhang, Huayan Zhang, Yang Li, Yoshihiko Nakamura, and Lei Zhang. Flowfusion: Dynamic dense rgb-d slam based on optical flow. In *ICRA*. IEEE, 2020. 16

[67] 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, 16

[68] Wenda Zhao, Fei Wei, Haipeng Wang, You He, and Huchuan Lu. Full-scene defocus blur detection with defbd+ via multilevel distillation learning. *IEEE Transactions on Multimedia*, 2023. 7

[69] 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. 3

[70] 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. 2

[71] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. *ACM TOG*, 2018. 3, 8, 11

AKiRa: Augmentation Kit on Rays for optical video generation
Supplementary Material

In this supplementary material, we include:

F. Additional quantitative results

In this section, we present additional quantitative results on the RealEstate10K dataset [71], comparing the performance of AKiRa with state-of-the-art camera control approaches for video generation: MotionCtrl [56] and CameraCtrl [21], incorporating the corresponding LoRA [26] module to ensure domain consistency.

We evaluate video quality, camera motion fidelity, and temporal dynamic consistency metrics (same as in Table 1 of the main manuscript) for the text-to-video (T2V) backbone Animatediff [18] and the image-to-video (I2V) backbone SVD [5].

All metrics are computed on 1,000 generated samples using text prompts from the RealEstate10K dataset or conditioning frames from the WebVid dataset [3]. For AKiRa, as it explicitly controls the bokeh, we set $\alpha = 0$ to align with the aperture behavior of the RealEstate10K dataset, where videos are recorded using small apertures and wide-angle cameras. The impact of aperture on performance is discussed in Section G and detailed in Table 6.

In Table 6, we present the performance of AKiRa in comparison with two state-of-the-art methods: MotionCtrl [56] and CameraCtrl [21]. In terms of video quality, AKiRa outperforms both SoTA methods on FVD and CD-FVD metrics for both the Animatediff and SVD backbones, achieving scores of 128.55 and 89.16 for Animatediff, and 54.83 and 41.55 for SVD. A similar trend is observed in dynamic consistency, where AKiRa leads across all metrics on both backbones. It achieves the highest scores for Consistency (0.9851), Smoothness (0.9933), and Flickering (0.9733), demonstrating superior temporal coherence. In terms of motion fidelity, AKiRa demonstrates significantly better motion controllability on the Animatediff backbone, achieving superior performance as evidenced by the highest FlowSim and lowest RPE errors. AKiRa competes closely with CameraCtrl on SVD, with only a narrow difference. We attribute this to the overfitting of CameraCtrl on the real-estate dataset, where intrinsic parameters and optical features remain unchanged during this experiment.

Method		Video quality		Camera motion fidelity			Dynamic consistency (VBench)		
Backbone	Camera control	FVD ↓	CD-FVD ↓	RPE-R (deg)↓	RPE-t (cm)↓	FlowSim ↑	Consistency ↑	Smoothness ↑	Flickering↑
AnimateDiff [18]	MotionCtrl [56]	237.22	543.24	0.387	1.536	67.83	0.9779	0.9834	0.9712
	CameraCtrl [21]	177.70	106.24	0.377	1.555	77.08	0.9779	0.9834	0.9712
	AKiRa (ours)	128.55	89.16	0.323	1.347	84.04	0.9809	0.9882	0.9745
SVD [5]	MotionCtrl [56]	122.67	330.75	1.030	1.326	24.14	0.9516	0.9814	0.9404
	CameraCtrl [21]	55.55	50.18	0.312	1.268	92.19	0.9836	0.9928	0.9695
	AKiRa (ours)	54.83	41.55	0.312	1.236	91.51	0.9851	0.9933	0.9733

Table 5. **Comparison with the state-of-the-art.** Comparison of AKiRa and concurrent methods with different backbones on RealEstate dataset, evaluating video quality, camera motion fidelity, and dynamic consistency. Best .

G. Additional analysis of bokeh map

Controllability of Bokeh - Aperture We propose a bokeh map with the same structure and dimensions as the direction and moment maps, assigning an aperture and focus (depth-of-field) parameter to each pixel in the frame. More specifically, we define the coordinates of the focus point (u_{in}, v_{in}) representing the sharpest point in the frame. We then define the per-pixel bokeh map as $\mathbf{a} \in \mathbb{R}^3$ for any point (u, v) on the frame as:

$$\mathbf{a} = \begin{bmatrix} u - u_{in} \\ v - v_{in} \\ \|(u, v) - (u_{in}, v_{in})\|^{\frac{1}{\sigma(\alpha)}} \end{bmatrix}, \tag{7}$$

We first present the visualization of the bokeh map and demonstrate how it influences the generated videos. We examine two groups of bokeh variations: the effect of varying the aperture α in Figure 7a, and the effect of adjusting the focus point f_{in} in Figure 7b (see the zoomed image in the second row highlighted the red rectangle).

In Figure 7a, we progressively increase the aperture level over time, with the focus fixed at the center of the image. As a result, the visualization of the bokeh map shrinks, and it shows that the blur area expands proportionally with increasing aperture α.

In Figure 7b, we shift the focus point f_{in} from the upper-left corner to the lower-right corner. This causes the center of the bokeh map to move accordingly, effectively shifting the blur area in generated videos. The results confirm our ability to dynamically control the blur area based on the focus point.

Both experiments validate the effectiveness and controllability of AKiRa in manipulating the depth of field.

Bokeh influence on T2V performance We analyze the influence of bokeh on video quality, flow similarity, and dynamic consistency using Animatediff, with the experimental settings identical to those in Table 1 of the main manuscript. By varying the aperture value α from 0 to 100, we measure its impact on the corresponding metrics.

Table 6 reports the results for different aperture settings. When the aperture is small and bokeh is weak (i.e.,

Apert.	Video quality		Motion fidelity	Dynamic consistency (VBench)		
	FVD ↓	CD-FVD ↓	FlowSim ↑	Consistency ↑	Smoothness ↑	Flickering↑
0	350.05	333.61	70.82	0.9697	0.9702	0.9525
5	353.09	337.48	70.94	0.9695	0.9705	0.9528
10	354.94	334.95	71.15	0.9692	0.9709	0.9534
30	341.31	327.02	71.23	0.9686	0.9728	0.9559
50	332.27	328.74	70.97	0.9686	0.9735	0.9572
100	342.77	328.13	70.82	0.9683	0.9737	0.9574

Table 6. **Influence of aperture.** Influence of aperture effect on AKiRa's performances with Animatediff backbone on RealEstate dataset. Best .

$\alpha = 0$), the generated videos exhibit better consistency. Conversely, when the aperture is large (i.e., $\alpha = 100$), the videos demonstrate greater smoothness and reduced flickering. Notably, the optimal video generation metrics are observed around $\alpha = 30$ and $\alpha = 50$. This can be attributed to **the intrinsic bokeh** present in the WebVid [3] dataset, where **adding an appropriate amount of bokeh enhances realism**, resulting in improved FVD and CD-FVD performance.

H. Ethical discussion

Our paper proposes a method to generate videos based on camera and optical control, enabling better alignment of the generation process with user intentions and geometric information. On the positive side, this approach can enhance the AIGC (AI-Generated Content) creative process by reducing biases introduced by training data, more accurately reflecting user intentions, and minimizing trial-and-error in content generation. This efficiency can also contribute to reducing the carbon footprint associated with the generation process. On the negative side, however, it may reduce the labour required for video production, potentially leading to job losses, and could stifle creativity if individuals become overly reliant on generative tools.

I. Algorithm of AKiRa

We present the complete AKiRa algorithm in Algorithm 1. As discussed in the main manuscript, random sampling is implemented using spline sampling. Augmentation dropout with a probability p is applied to all the optical features,

(a) Example of increasing apertures along video time

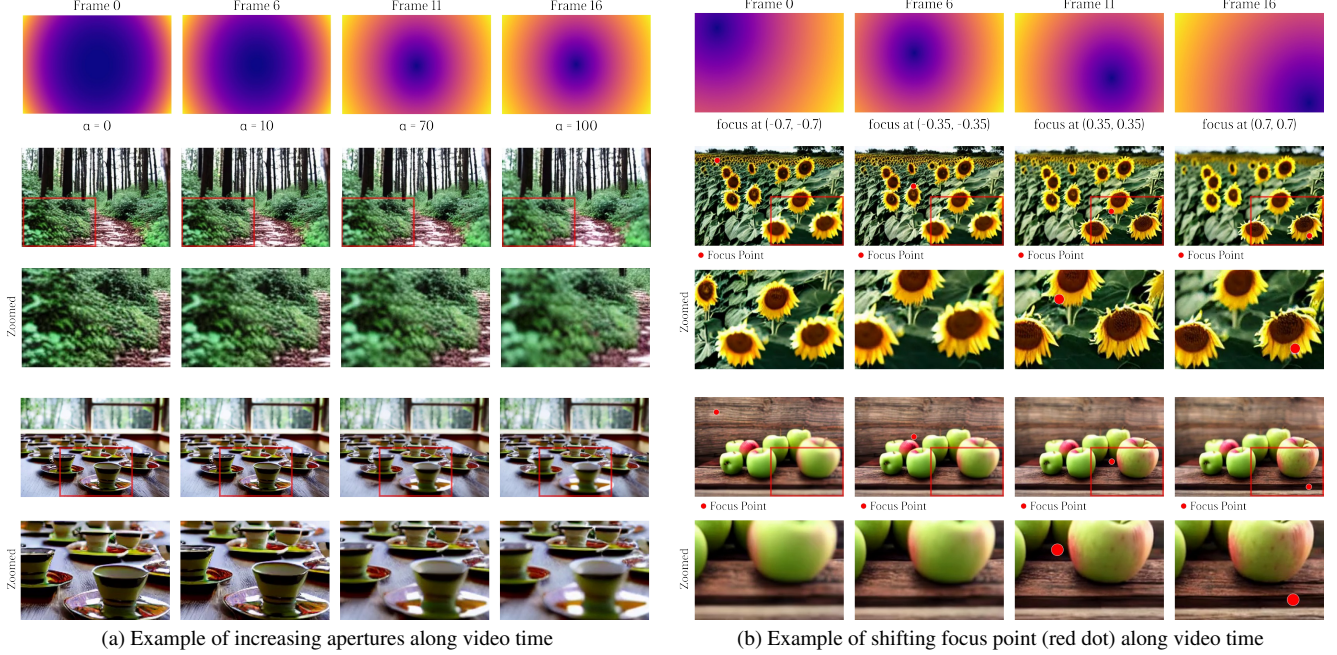(b) Example of shifting focus point (red dot) along video time

Figure 7. **Qualitative Results.** We demonstrate the qualitative performance of AKiRa on bokeh variations for both (a) aperture levels and (b) focus points. The results are generated using Animatediff [18]. In (a), we observe that the blur area and intensity increase proportionally with the aperture parameter α. In (b), the blur area shifts dynamically following changes in the focus point, showing the effectiveness of AKiRa in handling variations in depth of field.

both collectively and individually.

Since augmentation is performed on-the-fly during the training process, the augmentation order is carefully designed to optimize computational efficiency. Specifically, we first augment the bokeh aperture to leverage the pre-computed depth map derived from the original frames. Next, distortion augmentation is applied, which may implicitly alter the focal length due to a necessary cropping operation to avoid undefined borders during image warping. Finally, we augment the zoom aspect, incorporating the results of the distortion augmentation.

During training, the augmentation parameters are sampled as follows: the dropout probability p is set to 0.2; the bokeh aperture is sampled uniformly between 0 and 100; the distortion parameters are sampled uniformly within the range $[-0.1, 0.1]$ for all three parameters in $\mathbf{D}$; and the zoom factor is sampled between 1.0 and 3.0, as zoom factors below 1.0 are ill-defined due to the difficulty to generate outpainting content through augmentation.

J. Details about user study

In this section, we elaborate on the specifics of our user study setup corresponding to Section 4.2 in our main manuscript.

For the evaluation, each participant was presented with a total of 10 video sets, 5 with Animatediff backbone (T2V)

Algorithm 1 AKiRa augmentation algorithm

Require: $\mathbf{I}$: frames, $\mathbf{Z}$: depth maps, p: aug. dropout.
  if True with probability p **then**
    if True with probability p **then**
      $\{\alpha\} = \text{RANDOMAPERTURESPLINE}()$
      $\{(u,v)\} = \text{RANDOMINFOCUSSPLINE}()$
      $\mathbf{I} \leftarrow \text{BOKEHAUGMENTER}(\mathbf{I}, \mathbf{Z}, \{\alpha\}, \{(u,v)\})$
    end if
    if True with probability p **then**
      $\{\mathbf{D}\} = \text{RANDOMDISTORTIONSPLINE}()$
      $\mathbf{I} \leftarrow \text{DISTORTIONAUGMENTER}(\mathbf{I}, \{\mathbf{D}\})$
    end if
    if True with probability p **then**
      $\{f\} = \text{RANDOMFOCALSPLINE}()$
      $\mathbf{I} \leftarrow \text{ZOOMAUGMENTER}(\mathbf{I}, \{f\})$
    end if
      return $\mathbf{I}$
  else return $\mathbf{I}$
  end if

---

and 5 with SVD backbone (I2V). Each set comprised 4 generated videos from (i) the baseline (backbone without camera control), (ii) MotionCtrl, (iii) CameraCtrl, and (iv) AKiRa (ours), we shuffled the results and displayed them in random order.
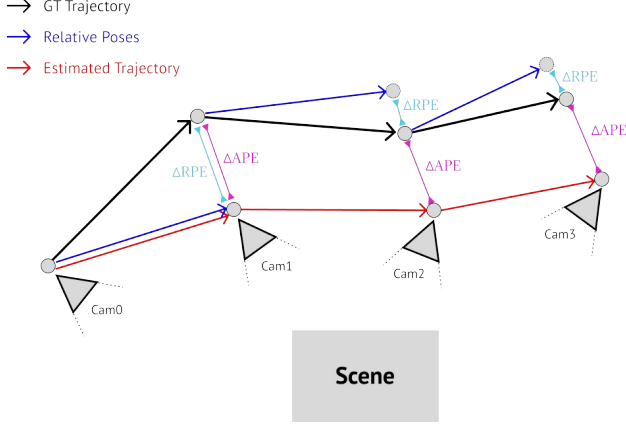
Figure 8. **Illustrates the instability of Absolute Pose Error (APE), purple** when measuring trajectory accuracy, compared to the robustness of Relative Pose Error (RPE), cyan. APE tends to accumulate errors, especially at later frames. In contrast, RPE calculates errors based on relative transformations between consecutive frames, making it less sensitive to single-frame errors and more robust for trajectory evaluation.

Subsequently, participants were prompted with 6 questions for each comparison:

1. *Rank the consistency of the video with the text prompt* (Only for Animatediff backbone).
2. *Rank the video quality (i.e. temporal consistency).*
3. *Rank the camera motion consistency with the reference.*
4. *Rank the best zoom-in or -out effect.*
5. *Rank the best distortion effect (e.g. fisheye).*
6. *Rank the best bokeh (in- or out-of-focus effect).*

In total, we recorded 25 participants with each participant responding to 55 questions. We analyzed the results by examining responses to each question individually, summarizing the collective feedback.

K. Discussion about metrics

K.1. Drawbacks of SfM-based metrics

Absolute Pose Error Many recent works [21, 56, 58] on camera motion-controlled video generation rely on Structure-from-Motion (SfM) or SLAM-based metrics to evaluate the effectiveness of camera control. The primary metric utilized in these works is similar to the Absolute Pose Error (APE) used in SLAM and SfM applications [42, 47], which computes the average trajectory error for translation and rotation separately. These errors are defined as follows:

For translation error in $\mathbb{R}^3$:

$$\text{APE}_{\text{trans}} = \frac{1}{N} \sum_{i=1}^{N} \|\hat{\mathbf{t}}_i - \mathbf{t}_i^*\|,$$

where N is the total number of frames, $\hat{\mathbf{t}}_i \in \mathbb{R}^3$ is the estimated camera translation vector for frame i, and $\mathbf{t}_i^* \in \mathbb{R}^3$ is

the ground truth translation vector for the same frame.

For rotation error in $SO(3)$, the error is often computed as the angle of the relative rotation:

$$\text{APE}_{\text{rot}} = \frac{1}{N} \sum_{i=1}^{N} \arccos\left(\frac{\text{trace}\left(\hat{\mathbf{R}}_i(\mathbf{R}_i^*)^\top\right) - 1}{2}\right),$$

where $\hat{\mathbf{R}}_i \in SO(3)$ is the estimated rotation matrix for frame i, and $\mathbf{R}_i^* \in SO(3)$ is the ground truth rotation matrix for the same frame. However, APE is highly sensitive to errors in individual frames, which is a common issue in generated videos due to **flickers or sudden object movements**. These artifacts, often **irrelevant to the quality of motion control**, can disproportionately affect the evaluation.

Relative Pose Error To address the APE limitation, related domains often rely on the Relative Pose Error (RPE), which reduces the impact of **accumulated errors** caused by single-frame inaccuracies, especially when such errors occur at the early stages of the trajectory.

RPE is computed by comparing the relative transformations between consecutive frames, rather than the absolute poses. It is defined separately for translation and rotation as follows:

For translation error in $\mathbb{R}^3$:

$$\text{RPE}_{\text{trans}} = \frac{1}{N-1} \sum_{i=1}^{N-1} \|\Delta\hat{\mathbf{t}}_i - \Delta\mathbf{t}_i^*\|,$$

where $\Delta\hat{\mathbf{t}}_i = \hat{\mathbf{t}}_{i+1} - \hat{\mathbf{t}}_i$ is the estimated relative translation, and $\Delta\mathbf{t}_i^* = \mathbf{t}_{i+1}^* - \mathbf{t}_i^*$ is the ground truth relative translation.

For rotation error in $SO(3)$, the relative error is defined as:

$$\text{RPE}_{\text{rot}} = \frac{1}{N-1} \sum_{i=1}^{N-1} \arccos\left(\frac{\text{trace}\left(\Delta\hat{\mathbf{R}}_i \Delta\mathbf{R}_i^*\right) - 1}{2}\right),$$

where $\Delta\hat{\mathbf{R}}_i = \hat{\mathbf{R}}_{i+1}\hat{\mathbf{R}}_i^\top$ is the estimated relative rotation, and $\Delta\mathbf{R}_i^* = \mathbf{R}_{i+1}^*\mathbf{R}_i^{*\top}$ is the ground truth relative rotation.

We demonstrate this phenomenon in Figure 8. When the first frame exhibits a high APE (purple), even if the subsequent trajectory is relatively accurate, the error is accumulated throughout the trajectory. In contrast, RPE is computed between relative poses (cyan), **making it less biased by errors in previous estimations**, therefore providing a more robust assessment of motion control quality.

Scaling Ambiguity Another challenge when using 3D metric errors to evaluate video camera control quality arises from unknown intrinsic parameters. Similar camera motions in image frames can have different interpretations in
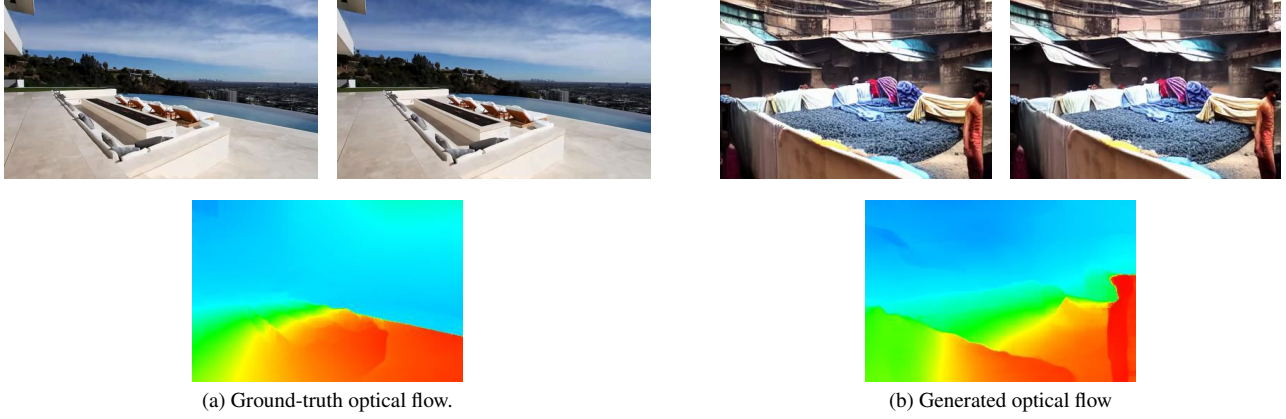
(a) Ground-truth optical flow.

(b) Generated optical flow

Figure 9. **Comparison of optical flow** for different content and similar camera motion.
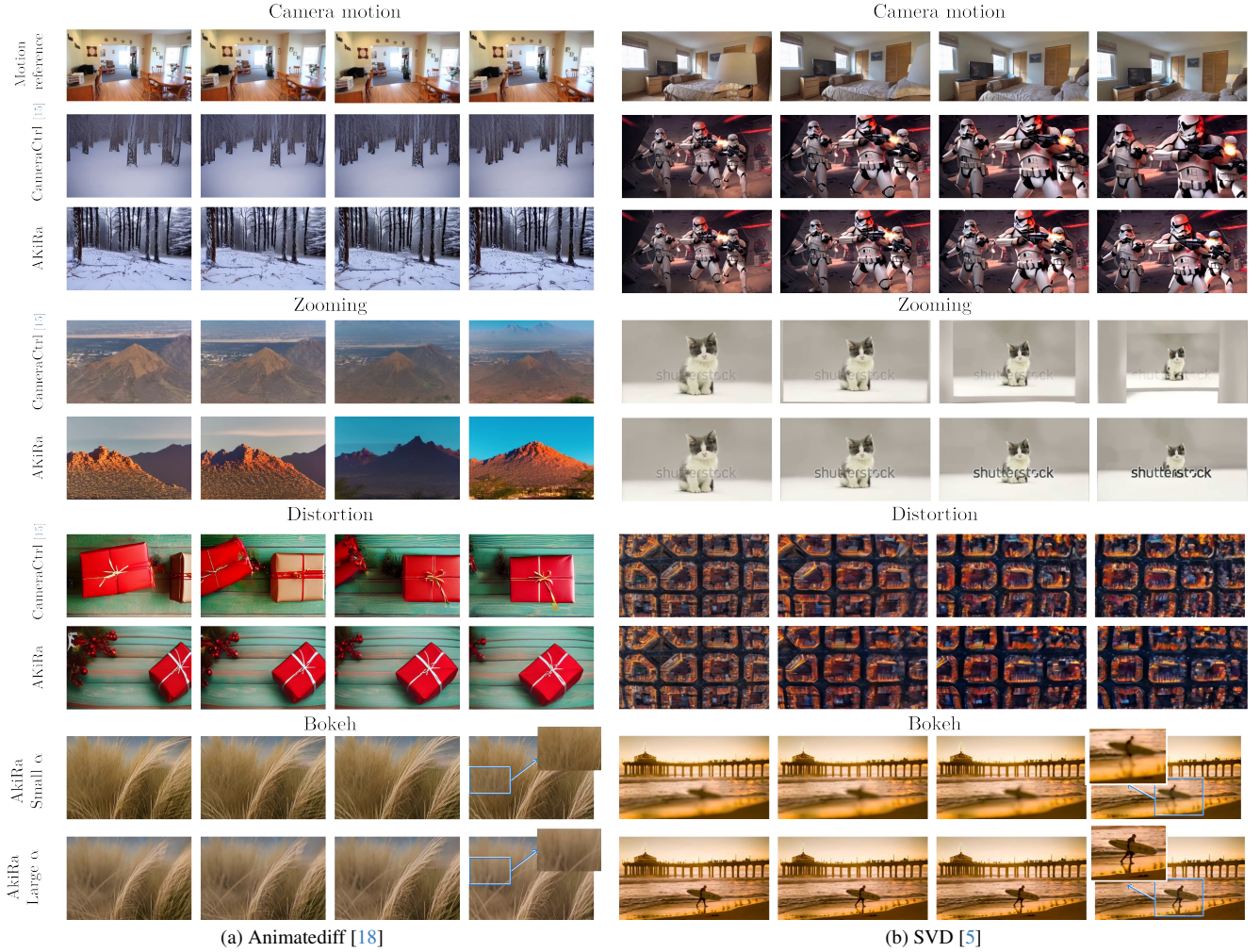

(a) Animatediff [18]

(b) SVD [5]

Figure 10. **Qualitative results** of AKiRa on Animatediff [18] and SVD [5] backbones. We recommend viewing the supplementary video.

3D metrics; for example, a leftward motion with similar visual displacement can correspond to varying metric dis-tances depending on the scale of the scene [20]. While some Structure-from-Motion (SfM) methods estimate cam-

era intrinsics, these estimates are often unreliable due to limited frame numbers (often around 15) and the typically smooth nature of camera motion, with a short stereo baseline needed for accurate intrinsic estimation.

To address this issue, in our paper, we report the **scale-corrected** camera trajectory by normalizing the trajectory length to match the ground truth. Formally, this is done by rescaling the estimated trajectory $\hat{\mathbf{T}}$ such that:

$$\hat{\mathbf{T}}_{\text{scaled}} = \frac{\|\mathbf{T}^*\|}{\|\hat{\mathbf{T}}\|} \cdot \hat{\mathbf{T}},$$

where $\|\mathbf{T}^*\|$ is the length of the ground truth trajectory and $\|\hat{\mathbf{T}}\|$ is the length of the estimated trajectory.

In our paper, all computations are performed using `evo`[1], a standard trajectory evaluation toolbox widely used for SLAM and visual odometry evaluations.

Computational Efficiency   Unfortunately, the computation time for SfM is relatively long and, most importantly, difficult to parallelize on GPU due to the sequential nature of the optimization problem. For instance, processing a single video with 16 frames using ParticleSfM [67] can take up to 4 minutes on average, including feature extraction and the optimization pipeline required for convergence. In our study, computing SfM for 1000 generated videos required an average of 66 hours on a single CPU.

K.2. FlowSim metric

Flow Similarity.   As detailed in Section 4.1 and Equation 6 of the main manuscript, we introduce the flow similarity metric. Similar to RPE, the concept of optical flow involves computing the relative on-image pixel motion between frames, which is widely used as an intermediate feature in SLAM and dense SfM processes [9, 66].

Compared to other on-image features, optical flow is less content-dependent and can serve as a robust metric for comparing video similarity. Unlike SfM trajectory features, optical flow does not rely on prior knowledge or precise estimation of the ground truth scaling of the scene. This makes it inherently robust to scaling ambiguities.

In our implementation, we measure the alignment of two optical flow directions while ignoring magnitudes to avoid content bias, as magnitudes are often influenced by disparity (i.e., differences in depth). Additionally, we exclude low-magnitude components because their directions are typically unreliable.

Figure 9 highlights the robustness of optical flow in comparing camera motion between videos. Despite differing content, similar flows indicate similar camera motion.

---

[1] https://github.com/MichaelGrupp/evo

Zoom flow similarity.   To evaluate the quality of the zooming effect, we compute the flow similarity between the theoretical zooming flow and the generated one. The theoretical zooming flow is derived from Equation 5 in the main manuscript as:

$$\begin{bmatrix} u_{\text{fl}} \\ v_{\text{fl}} \end{bmatrix} - \begin{bmatrix} u_{\text{fl}'} \\ v_{\text{fl}'} \end{bmatrix} = (s - s') \begin{bmatrix} u - c_x \\ v - c_y \end{bmatrix}, \tag{8}$$

where s and s' denote the zooming scales, and (c_x, c_y) are the principal point coordinates (i.e. the screen center). As illustrated in Figure 11a, the flow direction for a zoom-in effect converges toward the principal point (or diverges outward for a zoom-out effect), aligning with Equation 8. The generated flow, shown in Figure 11b, closely matches the theoretical flow, with minor deviations caused by variations in frame content.
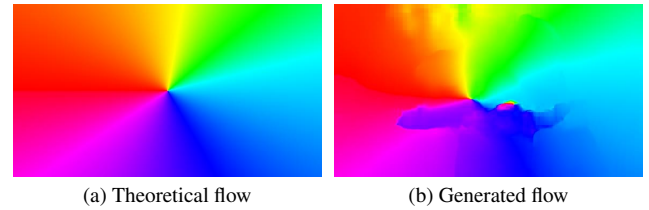

(a) Theoretical flow   (b) Generated flow

Figure 11. **Theoretical vs. generated zooming flows.**

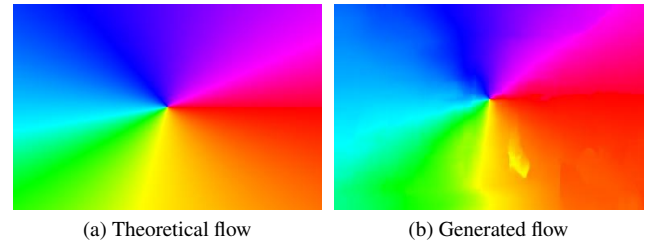(a) Theoretical flow   (b) Generated flow

Figure 12. **Theoretical vs. generated distorted flows.**

Distortion flow similarity.   To evaluate the quality of the distortion effect, we compute the flow similarity between the theoretical distortion flow and the generated one. The theoretical distortion flow is derived from Equation 2 in the main manuscript as:

$$\begin{bmatrix} u_{\mathbf{D}} \\ v_{\mathbf{D}} \end{bmatrix} - \begin{bmatrix} u_{\mathbf{D}'} \\ v_{\mathbf{D}'} \end{bmatrix} = \begin{bmatrix} u \\ v \end{bmatrix} (\mathbf{D} - \mathbf{D}') \begin{bmatrix} r^2 \\ r^4 \\ r^6 \end{bmatrix}, \tag{9}$$

where $\mathbf{D}$ and $\mathbf{D}'$ denote the distortion parameters and $r = \sqrt{(u - c_x)^2 + (v - c_y)^2}$ distance of each pixel towards image center. As illustrated in Figure 12a, the flow direction for a distortion effect also converges toward the principal point —or diverges outward depending on the sign

of $\mathbf{D} - \mathbf{D}'$—, aligning with Equation 9. The generated flow, shown in Figure 12b, closely matches the theoretical flow, again, with minor deviations caused by variations in frame content.

Computational Efficiency During implementation, we use RAFT [48], a fast deep optical flow estimator with GPU-based implementation, which significantly speeds up the flow estimation process compared to SfM and can be easily parallelized on GPU. For example, computing optical flows for 1000 generated videos takes approximately 10 minutes on a GPU, compared to 66 hours required by SfM on a single CPU.

As a result, our key messages concerning the evaluation metrics are:
1. Pose metrics for assessing camera control in generated videos are intrinsically less accurate, particularly when directly computing APE.
2. Using RPE with scale correction techniques improves robustness; however, the computational cost is prohibitively high for scaling to AI-generated videos.
3. We propose optical flow similarity (FlowSim), based on flow direction, offers a viable alternative. It serves as a good approximation of RPE while being computationally efficient, fast to compute, and scalable for large-scale AI-generated videos.
4. The flow similarity metric can also be used to confirm other optic features than motion, such as focal length change (zoom) and distortions.

L. Additional qualitative results

We provide additional qualitative results in Figure 10, demonstrating that AKiRa method performs well in several key aspects: accurately capturing camera motion (trees) with high video quality (the middle stormtrooper's head cf. CameraCtrl); maintaining consistency during zooming (mountain peak, cat's frame); effectively reflecting distortion effects (gift, bird's-eye view of Barcelona city); and rendering various aperture and bokeh effects (surfing on the beach, grass) with controllability.