Taming Camera-Controlled Video Generation with Verifiable Geometry Reward

Zhaoqing Wang[1], Xiaobo Xia[2⋆], Zhuolin Bie[1], Jinlin Liu[1], Dongdong Yu[1], Jia-Wang Bian[3], and Changhu Wang[1]

[1] PixVerse Team, AISphere
[2] School of Information Science and Technology, University of Science and Technology of China
[3] College of Computing and Data Science, Nanyang Technological University
derrickwang005@gmail.com, xiaoboxia@ustc.edu.cn

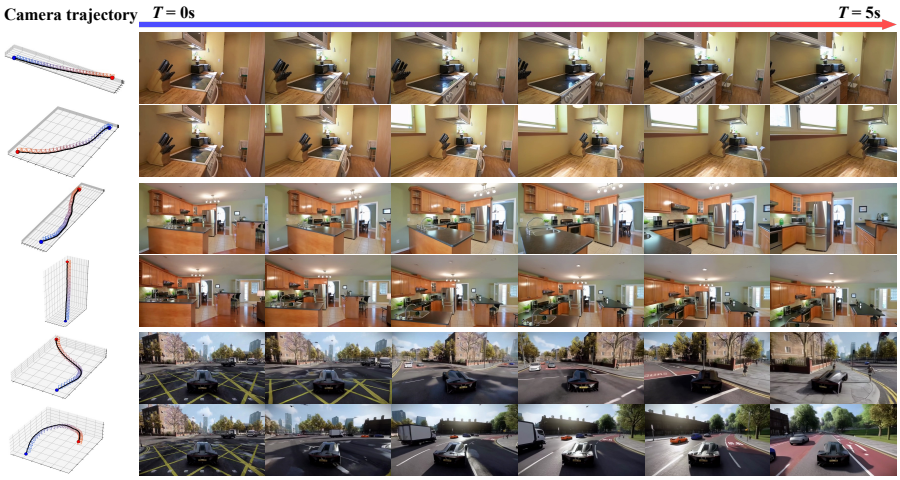

Fig. 1: Camera-controlled video generation with CamVerse. Given an initial frame, a text prompt, and a user-specified 3D camera trajectory, our model synthesizes a video that follows the trajectory while preserving scene layout and appearance. *Left*: input trajectories (start in blue, end in red). *Right*: frames sampled from $T = 0$ to $T = 5$s. Rows are grouped in pairs that share the same first frame and prompt; only the camera path differs within each pair. These examples span diverse camera movements (*e.g.*, planar curves and out-of-plane arcs) and scenarios (*e.g.*, indoor kitchens and outdoor driving), showcasing precise camera control.

Abstract. Recent advances in video diffusion models have remarkably improved camera-controlled video generation, but most methods rely solely on supervised fine-tuning (SFT), leaving online reinforcement learning (RL) post-training largely underexplored. In this work, we introduce an online RL post-training framework that optimizes a pretrained video generator for precise camera control. To make RL effective in this setting,

⋆ Corresponding author.

we design a verifiable geometry reward that delivers dense segment-level feedback to guide model optimization. Specifically, we estimate the 3D camera trajectories for both generated and reference videos, divide each trajectory into short segments, and compute segment-wise relative poses. The reward function then compares each generated-reference segment pair and assigns an alignment score as the reward signal, which helps alleviate reward sparsity and improve optimization efficiency. Moreover, we construct a comprehensive dataset featuring diverse large-amplitude camera motions and scenes with varied subject dynamics. Extensive experiments show that our online RL post-training clearly outperforms SFT baselines across multiple aspects, including camera-control accuracy, geometric consistency, and visual quality, demonstrating its superiority in advancing camera-controlled video generation.

Keywords: Camera-controlled video generation · Reinforcement learning fine-tuning

1 Introduction

Video diffusion models have advanced rapidly, delivering high-fidelity and temporally coherent frames directly from textual prompts [2, 7, 9, 30, 52]. Beyond raw quality, many methods increasingly accept user-specified controls and conditioning signals, enabling flexible manipulation over content and motion [14, 15, 18, 47, 53, 55, 58]. In parallel, action-conditioned models in interactive media show that policies can predict state transitions and future observations, pointing toward exploration of generated worlds rather than mere depiction [12, 43, 51, 67]. Within general video generation, camera control has emerged as a natural interface for such exploration by injecting camera parameters into pretrained diffusion models [3, 4, 31, 59, 61, 62]. These advances make it essential to reliably translate input camera trajectories into geometrically consistent videos, especially for film creation, environment simulation, and immersive experiences.

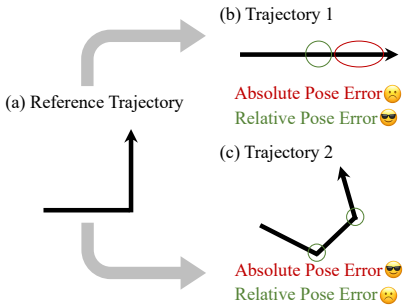

Fig. 2: Comparison between the absolute and relative pose error. Given a reference trajectory, the model randomly generates two videos with corresponding trajectories. The absolute pose error prefers global-similar but locally incorrect results, while the relative pose error emphasizes locally consistent and smooth results. The latter is more aligned with the task characteristics.

Despite their impressive progress, the predominant methods for camera-controlled video generation remain supervised fine-tuning (SFT), while online reinforcement learning (RL) is largely underexplored [36–38, 46, 64]. This gap mainly persists for two reasons. First, it is nontrivial to design verifiable and

geometry-aware reward functions for high-dimensional video outputs, and even estimating reliable camera trajectories from generated frames can be challenging. Second, camera trajectory is intrinsically continuous, yet most rewards used in AI-generated content (AIGC) are instance-level (*e.g.*, image/clip scores), resulting in sparse feedback and inefficient credit assignment during optimization. Nevertheless, online RL retains substantial untapped potential in this setting. Unlike SFT, which passively fits to fixed training data, online RL actively generates samples from the learned conditional distribution, reinforcing target camera motions through positive feedback while discouraging failure modes through negative feedback. Moreover, RL offers a principled mechanism to incorporate strong priors from large 3D models [29, 54, 57, 71] directly into video generators, thereby enhancing geometric consistency and improving camera-control accuracy of generated videos.

In this work, we introduce CamVerse, an online RL post-training framework that optimizes a video generative model for precise camera control. At its core lies a verifiable geometry reward that translates camera-control accuracy into dense segment-level feedback. Specifically, we first estimate 3D camera trajectories for both the generated and the reference videos using a large 3D model. Each trajectory is then divided into short non-overlapping segments, within which we compute relative camera motion instead of absolute poses. Each generated reference segment pair is subsequently scored with an alignment metric, forming a dense reward signal. As illustrated in Figure 2, the absolute pose error captures the global shape of trajectories and favors globally similar but locally inaccurate trajectories, whereas the relative pose error emphasizes local smoothness and continuity, which aligns with the behavior desired for camera-controlled video generation. Moreover, obtaining absolutely accurate videos typically requires lots of rollout, which is sample-inefficient for online RL.

Apart from that, most available datasets mainly focus on static video clips with camera parameters, which causes a negative effect on the motion strength of generated videos, and are challenging to encourage models to disentangle camera movement from subject dynamics. To overcome these limitations, we construct a large-scale dataset of video clips collected from both real-world footage and video games, emphasizing diverse and large-amplitude camera movement. We estimate camera trajectories with state-of-the-art 3D models [25, 57] and apply a multi-stage filtering pipeline to remove failure reconstruction cases and broaden the trajectory distribution. The resulting dataset contains 315k videos with annotated camera trajectories, of which 314k are used for training and 1k for testing. Before delving into details, our contributions are summarized as follows:

- We propose CamVerse, the first online RL post-training framework for camera-controlled video generation. The framework is equipped with a verifiable geometry reward that delivers dense segment-level feedback, effectively alleviating reward sparsity.
- We curate a new large-scale dataset spanning real-world footage and game environments, which is annotated with camera trajectory and covers diverse camera movements and varied subject dynamics.

– Extensive experiments demonstrate consistent improvements over SFT baselines in camera-control accuracy, geometric consistency, and visual quality (*cf.*, Figure 1).

2 Related Work

2.1 Video Diffusion Models

Recent years have seen rapid progress in video generation, driven by advances in model architectures [20], the availability of large-scale datasets [6, 10, 63], the introduction of comprehensive evaluation benchmarks [26, 27], and improvements in training techniques [28, 39]. Two major research directions in video generation are text-to-video (T2V) and image+text-to-video (IT2V). The former directly synthesizes a video from a natural-language prompt. Early approaches [7, 8, 16, 17, 21] adapt UNet-based text-to-image (T2I) models by introducing temporal modules, which enable frame-to-frame dynamics while reusing strong image priors. Subsequent works further scale data and computation to enhance visual fidelity and diversity [7, 8, 21, 48]. Beyond pure text conditioning, IT2V methods [16, 24, 60] take a still image with a caption as input to guide video synthesis. More recent systems employ diffusion transformer architectures [11, 32, 40, 44, 69], large-scale training pipelines, and advanced post-training techniques [37, 38] to achieve long-range temporal coherence and high visual quality, represented by models such as [1,2,9,30,35,41,42,50,52,66,66]. Despite this progress, most existing models still operate purely in 2D pixel space without explicit 3D supervision, often leading to geometric distortions under complex viewpoint changes. Motivated by this limitation, we propose a post-training framework that explicitly rewards adherence to desired camera trajectories and multi-view geometric consistency, thereby improving both controllability and consistency in generated videos.

2.2 Camera-Controlled Video Generation

Controlling camera movement in video generation is commonly achieved by injecting camera information into pretrained models. For example, MotionCtrl [59], CameraCtrl [18], and I2VControl-Camera [13] condition on camera parameters (*e.g.*, extrinsics, point trajectories, and Plücker embeddings [49]). Afterwards, CamCo [62] incorporates epipolar constraints into attention, while CamTrol [22] leverages 3D point-cloud information. AC3D [3, 4] presents a deep analysis of camera-controlled video generation and designs an effective interface for injecting camera representations into pretrained backbones. Beyond single-camera settings, recent works [5,31,34,61] address multi-camera synchronization and cross-view consistency. CameraCtrl2 [19] further introduces a scalable data pipeline with camera-trajectory annotations for dynamic scenes, substantially improving the realism of dynamic content. Different from previous works, we propose a reinforcement learning post-training framework with verifiable geometry reward. This directly optimizes the model to follow user-specified camera trajectories while maintaining 3D-consistent content across frames.

2.3 Feed-Forward 3D Reconstruction

Feed-forward models that regress 3D scene structure from image sets offer an efficient alternative to optimization-heavy pipelines. In more detail, Dust3R [56] predicts pairwise point clouds in the first-view coordinate frame, but scaling to larger scenes typically requires a brittle global alignment. Fast3R [65] improves scalability by performing joint inference over thousands of images, mitigating the need for costly post-hoc alignment. Subsequent work simplifies the problem via decomposition. FLARE [68] first estimates camera poses and then infers geometry, while VGGT [54] jointly predicts camera parameters and dense geometry using multi-task learning and large-scale data. These feed-forward methods still anchor predictions to a reference frame. To address that, π^3 [57] addresses this with a fully permutation-equivariant architecture that removes reference-frame bias. Leveraging these advances, we propose a novel post-training framework that injects powerful 3D prior learned by feed-forward models into camera-controlled video generation. This sets new state-of-the-art results across diverse scenes.

3 Methodology

3.1 Preliminary

Camera-controlled video generation. Let $\mathbf{x}_{1:N} = \{\mathbf{x}_n\}_{n=1}^N$ denote a video and $\mathbf{c}$ denote a text prompt. Each frame n is accompanied by camera intrinsics $\mathbf{K}_n \in \mathbb{R}^{3\times3}$ and camera-to-world extrinsics $\mathbf{E}_n = [\mathbf{R}_n \mid \mathbf{t}_n] \in \mathrm{SE}(3)$ with $\mathbf{R}_n \in \mathrm{SO}(3)$ and $\mathbf{t}_n \in \mathbb{R}^3$. Directly feeding raw $(\mathbf{K}_n, \mathbf{R}_n, \mathbf{t}_n)$ to the model is poorly coupled with pixel-space features and leaves translations unconstrained. In contrast, we construct the per-frame Plücker embedding [49] as camera condition. Concretely, we compute the world-space ray direction $\mathbf{d}_{u,v} \propto \mathbf{R}_n \mathbf{K}_n^{-1}[u, v, 1]^\top$, and the camera center $\mathbf{o}_n$ for each pixel (u, v). The Plücker line coordinates for pixel (u, v) are $\mathbf{p}_{u,v} = \big(\mathbf{o}_n \times \mathbf{d}_{u,v},\ \mathbf{d}_{u,v}\big) \in \mathbb{R}^6$ and normalized to the unit direction. Stacking all pixels yields a per-frame embedding $\mathbf{P}_n \in \mathbb{R}^{6\times h\times w}$ and the condition of a camera trajectory is $\mathbf{P}_{1:N} = \{\mathbf{P}_n\}_{n=1}^N$. Given a noisy latent $\mathbf{z}_t$, a diffusion timestep t and these conditions, a pretrained video diffusion model π_θ is optimized with the standard flow-matching objective:

$$\mathcal{L}_{\mathrm{FM}}(\theta) = \mathbb{E}_{\mathbf{z}_t, t, \mathbf{c}, \mathbf{P}_{1:N}} \left[\left\| \boldsymbol{\epsilon} - \pi_\theta(\mathbf{z}_t \mid t, \mathbf{c}, \mathbf{P}_{1:N}) \right\|_2^2 \right], \tag{1}$$

where $\boldsymbol{\epsilon}$ denotes the target velocity.

Online RL. We treat the video diffusion model as a stochastic policy that samples videos, $\mathbf{x}_{1:N} \sim \pi_\theta(\cdot \mid \mathbf{c}, \mathbf{P}_{1:N})$. In each online iteration, given a pair of conditions $(\mathbf{c}, \mathbf{P}_{1:N})$, we draw G rollouts from the current policy and evaluate each with a reward function $r(\mathbf{x}_{1:N}, \mathbf{c}, \mathbf{P}_{1:N})$. Afterwards, by omitting subscripts of $\mathbf{x}_{1:N}$ and $\mathbf{P}_{1:N}$ below, the policy π_θ is updated to increase the expected reward:

$$\max_\theta \ \mathbb{E}_{\mathbf{x}\sim\pi_\theta}[r(\mathbf{x}, \mathbf{c}, \mathbf{P})], \tag{2}$$

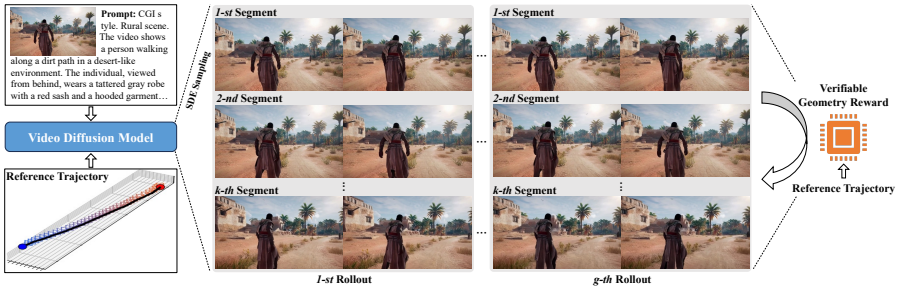

Fig. 3: Framework of CamVerse. Given the input first frame, a text prompt, and a reference camera trajectory, the video diffusion model π_θ samples G rollouts by a stochastic reverse process (SDE sampling). Each rollout is partitioned into non-overlapping segments. For each rollout, a large 3D model is used to estimate the camera trajectory. After aligning the generated trajectory to the reference one, the verifiable geometry reward compares each pair of generated-reference segments in a relative manner and assigns a dense alignment score. With this type of reward signal, we conduct GRPO fine-tuning, improving camera-control accuracy and geometric consistency.

where $r(\cdot)$ denotes a reward function. Interpreting the reward as an "optimality" probability $r(\mathbf{x}, \mathbf{c}, \mathbf{P}_{1:N}) \in [0, 1]$, a sample $\mathbf{x}$ drawn from the old policy is assigned to a positive subset with probability $r(\cdot)$ and to a negative subset with probability $1 - r(\cdot)$. This induces two reweighted distributions:

$$\begin{aligned} \pi^+(\mathbf{x} \mid \mathbf{c}, \mathbf{P}) &= \frac{r(\mathbf{x}, \mathbf{c}, \mathbf{P})\, \pi_{\theta_{\text{old}}}(\mathbf{x} \mid \mathbf{c}, \mathbf{P})}{\mathbb{E}_{\mathbf{x} \sim \pi_{\theta_{\text{old}}}}[r(\mathbf{x}, \mathbf{c}, \mathbf{P})]}, \\ \pi^-(\mathbf{x} \mid \mathbf{c}, \mathbf{P}) &= \frac{\big(1 - r(\mathbf{x}, \mathbf{c}, \mathbf{P})\big)\, \pi_{\theta_{\text{old}}}(\mathbf{x} \mid \mathbf{c}, \mathbf{P})}{\mathbb{E}_{\mathbf{x} \sim \pi_{\theta_{\text{old}}}}[1 - r(\mathbf{x}, \mathbf{c}, \mathbf{P})]}. \end{aligned} \tag{3}$$

Policy improvement moves probability mass from π^- toward π^+ while staying close to $\pi_{\theta_{\text{old}}}$, realizing via KL-regularized maximum likelihood. Denoting $J(\theta, \mathbf{c}, \mathbf{P}) := \mathbb{E}_{\mathbf{x} \sim \pi_\theta}[r(\mathbf{x}, \mathbf{c}, \mathbf{P})]$, the updated policy $\pi_{\theta_{\text{new}}}$ satisfies the improvement criterion:

$$\Delta J \;:=\; J(\theta_{\text{new}}, \mathbf{c}, \mathbf{P}) - J(\theta_{\text{old}}, \mathbf{c}, \mathbf{P}) \;>\; 0. \tag{4}$$

3.2 Verifiable Geometry Reward Design

For each generated video and its reference, a large 3D model is applied to estimate per-frame *camera-to-world* extrinsics and per-pixel confidence maps, allowing us to derive the camera trajectory for both sides:

$$\begin{aligned} &\mathbf{E}_n = [\mathbf{R}_n \mid \mathbf{t}_n], \quad \tilde{\mathbf{E}}_n = [\tilde{\mathbf{R}}_n \mid \tilde{\mathbf{t}}_n], \\ &\tilde{\omega}_n = \text{Agg}(\tilde{\Omega}_n) \in [0, 1], \quad \tilde{\boldsymbol{\Omega}}_n \in \mathbb{R}^{h \times w}. \end{aligned} \tag{5}$$

Here $\mathbf{E}$ and $\tilde{\mathbf{E}}$ denote extrinsics of the reference and generated video, $\tilde{\omega}$ denotes per-frame confidence of the generated video, and $\mathrm{Agg}(\cdot)$ is the operation of mean pooling. To address the scale ambiguity, we first estimate an Umeyama transform $(s^\star, \mathbf{R}^\star, \mathbf{t}^\star) = \mathrm{Umeyama}\big(\{\tilde{\mathbf{o}}_n\}_{n=1}^N, \{\hat{\mathbf{o}}_n\}_{n=1}^N\big)$ with two trajectories [33]. Then, we apply that to each camera pose of the generated videos:

$$\tilde{\mathbf{E}}'_n = \big[\, \mathbf{R}^\star \tilde{\mathbf{R}}_n \;\big|\; s^\star \mathbf{R}^\star \tilde{\mathbf{t}}_n + \mathbf{t}^\star \,\big]. \tag{6}$$

This normalization helps alleviate the scale issue caused by the randomness of generated content, making the following reward calculation more convincing.

Afterward, we partition both trajectories into non-overlapping segments of length L. Within the k-th segment, we compute end-to-start relative transforms, emphasizing local smoothness:

$$\tilde{\mathbf{T}}_k = (\tilde{\mathbf{E}}'_{n_k})^{-1} \tilde{\mathbf{E}}'_{n_k+L}, \quad \hat{\mathbf{T}}_k = (\hat{\mathbf{E}}_{n_k})^{-1} \hat{\mathbf{E}}_{n_k+L}. \tag{7}$$

We form the error transform $\mathbf{T}_k^{\mathrm{err}} = \hat{\mathbf{T}}_k^{-1} \tilde{\mathbf{T}}_k$ and extract translation component $\mathbf{R}_k^{\mathrm{err}} = \mathbf{R}(\mathbf{T}_k^{\mathrm{err}})$ and rotation component $\mathbf{t}_k^{\mathrm{err}} = \mathbf{t}(\mathbf{T}_k^{\mathrm{err}})$. Subsequently, the per-segment errors are calculated as:

$$\begin{aligned} e_t(k) &= \mathrm{clip}\left(\|\mathbf{t}_k^{\mathrm{err}}\|_2,\, -1,\, 1\right), \\ e_R(k) &= \arccos\Big(\mathrm{clip}\big(\tfrac{\mathrm{tr}(\mathbf{R}_k^{\mathrm{err}})-1}{2}, -1, 1\big)\Big). \end{aligned} \tag{8}$$

Errors are converted to alignment scores $\mathbf{s}$ by simple reversal,

$$\mathbf{s}_k \;=\; -\big(\lambda_t\, e_t(k) + \lambda_R\, e_R(k)\big), \tag{9}$$

where λ_t and λ_R are hyper-parameters to balance the importance of translation and rotation. In doing so, better-aligned segments (smaller errors) yield larger $\mathbf{s}_k$ (less negative). Given the per-frame confidence, we further mask out unreliable segments via $\mathbf{m}_k = \mathbf{1}[\boldsymbol{\omega}_k \geq \tau]$ and obtain the final reward $\tilde{\mathbf{s}}_k$. This construction yields dense feedback that is insensitive to the ambiguous scale, locally descriptive, and compatible with online RL.

3.3 Training Paradigm

By resorting to our verifiable geometry reward function, we conduct online RL training on the video diffusion model π_θ with group-relative policy optimization (GRPO) [36, 46, 64]. As shown in Figure 3, for each pair of conditions $(\mathbf{c}, \mathbf{P})$, we sample a group of G rollouts under a stochastic reverse process so that each per-step policy is Gaussian, $\pi_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{c}, \mathbf{P}) = \mathcal{N}\big(\boldsymbol{\mu}_\theta(\mathbf{x}_t, t, \mathbf{c}, \mathbf{P}),\, \sigma_t^2 \mathbf{I}\big)$, thereby injecting exploration noise while keeping the original marginals. The reward function assigns a list of reward values to the g-th rollout $\tilde{\mathbf{s}}_{1:K}^g = \{\tilde{\mathrm{s}}_k^g\}_{k=1}^K$. With a group of reward values, we adopt a "z-score" normalization to obtain group-relative advantages below:

$$\mathbf{A}_k^g = \big(\mathrm{s}_k^g - \mu\big) \big/ \big(\delta + \gamma\big), \tag{10}$$

Algorithm 1 GRPO Training for Diffusion Models

Require: Dataset $\mathcal{D}$, Diffusion Policy π_θ (initialized from π_{ref}), Reward Model $R(\cdot)$, Group size G, KL coefficient β, Clip parameter ϵ, Inner epochs K.
Ensure: Optimized Diffusion Policy π_θ.

Initialize $\theta \leftarrow \theta_{\text{ref}}$
while not converged **do**
  Sample a prompt $c \sim \mathcal{D}$
  // **1. Sampling (Expensive Step)**
  Generate G outputs $\{x_1, \ldots, x_G\}$ using $\pi_{\theta_{\text{old}}}(\cdot|c)$
  Store trajectory log-probabilities: $\pi_{\text{old}}(x_i) = \log p_{\theta_{\text{old}}}(x_i|c)$
  // **2. Advantage Estimation (Group Relative)**
  Compute rewards: $r_i = R(x_i, c)$ for $i \in \{1, \ldots, G\}$
  Compute group stats: $\bar{r} = \frac{1}{G}\sum r_i, \quad \sigma_r = \text{Std}(r)$
  for $i = 1$ to G **do**
    $A_i = \frac{r_i - \bar{r}}{\sigma_r + 1e^{-8}}$ ▷ Group Normalized Advantage
  end for
  // **3. Optimization (Inner Loop)**
  for epoch $k = 1$ to K **do**
    Compute current log-probabilities $\pi_{\text{new}}(x_i) = \log p_\theta(x_i|c)$
    Compute ratio: $\rho_i = \exp(\pi_{\text{new}}(x_i) - \pi_{\text{old}}(x_i))$
    Compute KL approximation: $D_{\text{KL}} \approx \pi_{\text{new}}(x_i) - \log p_{\text{ref}}(x_i|c)$
    // **GRPO/PPO Objective with Clipping**
    $\mathcal{L}_{\text{surr}} = \frac{1}{G}\sum_{i=1}^{G} \min\left(\rho_i A_i, \text{clip}(\rho_i, 1-\epsilon, 1+\epsilon)A_i\right)$
    $\mathcal{L}_{\text{total}} = -\mathcal{L}_{\text{surr}} + \beta D_{\text{KL}}$ ▷ Minimize Loss
    Update $\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}_{\text{total}}$
  end for
  Update $\theta_{\text{old}} \leftarrow \theta$
end while

where μ and δ denote the mean and standard deviation of all reward values in a group. Here γ is used to improve numerical stability. Afterward, we optimize the video diffusion model π_θ by maximizing the following objective:

$$\max_{\theta} \frac{1}{G\,T\,K} \sum_{g=1}^{G} \sum_{t=1}^{T} \sum_{k=1}^{K} \Big(f\big(\mathbf{r}_t^g, \mathbf{A}_k^g, \Delta\big) - \beta\, \mathrm{KL}(\pi_\theta \,\|\, \pi_{\text{ref}}) \Big), \tag{11}$$

where the policy ratio $\mathbf{r}_t^g$ and the clipping function $f(\cdot)$ are defined as follows:

$$\mathbf{r}_t^g = \frac{p_\theta(\mathbf{x}_{t-1}^g \mid \mathbf{x}_t^g, \mathbf{c}, \mathbf{P})}{p_{\theta_{\text{old}}}(\mathbf{x}_{t-1}^g \mid \mathbf{x}_t^g, \mathbf{c}, \mathbf{P})}, \tag{12}$$

$$f(\cdot) = \min\big(\mathbf{r}_t^g\, \mathbf{A}_k^g,\ \text{clip}(\mathbf{r}_t^g, 1-\Delta, 1+\Delta)\, \mathbf{A}_k^g\big). \tag{13}$$

We initialize π_{ref} by the model after supervised fine-tuning, which can mitigate reward hacking. In practice, we use a few-step scheduler to collect rollouts efficiently. For better understanding, we provide the detailed algorithm flow for the training pipeline of our framework, shown in Algorithm 1.

Table 1: Quantitative comparison with state-of-the-arts. We evaluate each model using metrics of camera control accuracy (*i.e.*, "*Trans. Err.*" and "*Rot. Err.*"), geometric consistency (*i.e.*, "*Geo. Con.*") and visual quality (*i.e.*, "*VQ*") on the mixed test set.

Method	*Trans. Err.*↓	*Rot. Err.*↓	*Geo. Con.*↑	*VQ*↑
Text-to-Video				
CameraCtrl [18]	0.0887	1.4586	0.7567	2.93
AC3D-2B [3]	0.0476	1.0451	0.8156	3.82
AC3D-5B [3]	0.0428	0.9120	0.8820	4.69
CamVerse*	0.0395	0.6506	0.9081	5.30
CamVerse	**0.0293**	**0.5140**	**0.9173**	**5.91**
Image-to-Video				
CameraCtrl [18]	0.1696	3.2809	0.6087	2.63
CamVerse*	0.0337	0.5613	0.9174	**5.02**
CamVerse	**0.0286**	**0.4685**	**0.9226**	4.87

4 Experiments

4.1 Implementation Details

We adopt the latent diffusion architecture [45], including a spatial-temporal variation autoencoder (ST-VAE) and a diffusion transformer. The input video is downsampled by 4 for temporal and 8 for spatial via ST-VAE. The diffusion transformer is initialized by an internal pre-trained model, with about 2B parameters. We interpolate input camera poses to the same number of frames in the video data, because only keyframes are annotated with camera information. These interpolated camera poses are further embedded by a light-weight camera network, as the input camera condition.

Supervised fine-tuning. We conduct supervised fine-tuning on a curated dataset consisting of 315k text-video-camera samples, with 314k used for training and 1k reserved for evaluation. Each video is preprocessed by resizing the longer side to 512 pixels while preserving the aspect ratio. Video durations range from 2 to 10 seconds. Our infrastructure supports variable-length sequences. We use a packing dataloader to maintain load balance during training. The training is performed with a batch size of approximately 128, and all model parameters are fine-tuned for 10k iterations. We apply a text condition dropout ratio of 0.3 during training, while the camera condition is always retained to learn camera movement. Training is conducted in a multi-task setting, incorporating both text-to-video and image-to-video generation. Task sampling is imbalanced, with 30% text-to-video and 70% image-to-video. We employ the AdamW optimizer with a learning rate of 5×10^{-5}, weight decay of 0.01, and ϵ set to 1×10^{-15}. To effectively learning camera movement, the timestep scheduler is shifted by 5. Note that, for efficiency, we pre-extract both the VAE latent representations and the caption embeddings before training begins.

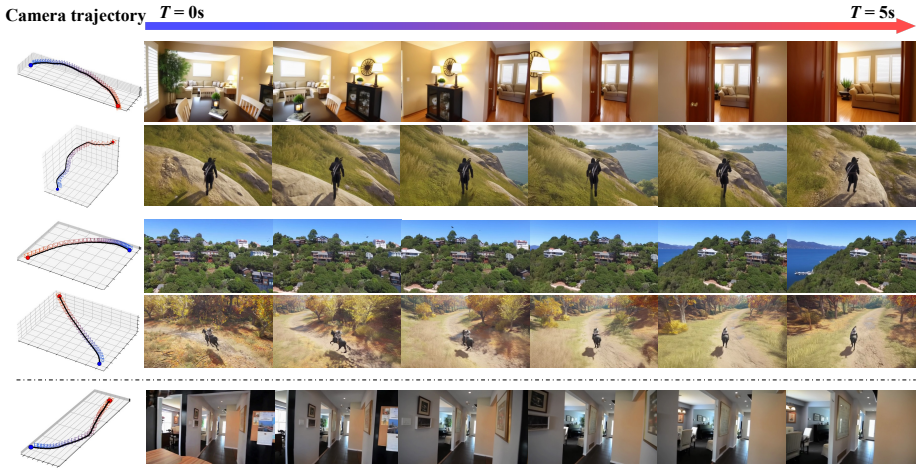

Fig. 4: Illustrations of qualitative results. For each example, we condition on a reference camera trajectory (start in blue, end in red) and show uniformly sampled frames from $T = 0$ to $T = 5$s. The first and second rows are text-to-video (T2V) cases. The third and fourth rows are image-to-video (I2V) cases. The last row is a failure case. In the first half of the clip, the camera does not execute right-forward translation and instead moves straight forward, resulting in early trajectory misalignment.

GRPO fine-tuning. We perform GRPO-based fine-tuning on the image-to-video task, using identical hyperparameters for both. A subset of approximately 3.2k samples is selected from the training set for this stage. We apply LoRA [23] to all linear layers within each transformer block, with a rank of $r = 64$ and $\alpha = 128$. Fine-tuning is conducted for 200 iterations. Each training group consists of 16 samples. From each group, we select the top 4 and bottom 4 samples based on reward values for GRPO optimization. We use 14 timesteps for the denoising process. The first three steps adopt a stochastic differential equation (SDE) sampler with a noise level of 0.7, while the remaining steps utilize a first-order ordinary differential equation (ODE)-based sampler. The timestep scheduler is shifted by 6. We set the classifier-free guidance (CFG) scale to 3.5 and apply a KL divergence loss with a weight of 1×10^{-4} to avoid reward hacking. All experiments are conducted on 32×NVIDIA H200 GPUs.

4.2 Evaluation

Datasets. The evaluation is conducted on a test set consisting of 1k video clips. This set includes 342 clips from the RealEstate10K dataset [70] and 658 clips from our curated dataset. These clips are sampled to cover a diverse range of camera trajectories, ensuring a comprehensive assessment of generation quality and robustness. All video durations are constrained between 4 and 8 seconds. For the ablation study, we randomly select a subset of 100 clips from the 342

Table 2: Quantitative results of supervised fine-tuning. We evaluate each model using metrics of camera control accuracy (*i.e.*, "*Trans. Err.*" and "*Rot. Err.*") and visual quality (*i.e.*, "*VQ*") on the RealEstate10K subset. "CamVerse*" adopts a training recipe consisting of 3:7 T2V-I2V tasks by default. Results are reported for both text-to-video and image-to-video settings. The best result in each case is marked in bold.

Method	*Trans. Err.*↓	*Rot. Err.*↓	*VQ*↑
Text-to-Video			
CamVerse*	**0.0395**	**0.6506**	5.30
w/ 7:3 T2V-I2V	0.0411	0.7112	5.42
w/o dynamic content	0.0464	0.7306	**5.45**
w/o shifted timesteps	0.0420	0.7023	5.12
w/o pose normalization	0.0416	0.7458	5.18
Image-to-Video			
CamVerse*	**0.0337**	**0.5613**	5.02
w/ 7:3 T2V-I2V	0.0363	0.6378	5.05
w/o dynamic content	0.0394	0.6555	**5.35**
w/o shifted timesteps	0.0342	0.6055	4.96
w/o pose normalization	0.0356	0.7135	4.99

RealEstate10K samples to accelerate the evaluation while preserving effectiveness.

Metrics. Our evaluation framework assesses model performance across three key aspects: visual quality, camera control accuracy, and geometric consistency. Specifically,

- Camera control accuracy: We assess camera trajectory alignment by measuring both translation and rotation errors between the generated and reference videos. For each input video, we use π^3 [57] to estimate camera trajectories. To account for scale ambiguity, we align the estimated trajectory of the generated video to that of the reference video using a similarity transformation. The translation error is computed as the average RMSE between corresponding camera positions, while the rotation error is defined as the mean angular difference between corresponding orientations.
- Geometric consistency: In addition to trajectory estimation, π^3 [57] outputs a per-pixel confidence map. We threshold this confidence map to obtain a binary mask, and define the geometric consistency score as the ratio of valid pixels to the total number of pixels. Here, the threshold is set to 0.1.
- Visual quality: To evaluate visual fidelity, we uniformly sample 8 frames from each generated video clip and compute the HPSv3 score using the prompt "`A high-quality, clear video frame.`" This metric quantitatively reflects perceptual quality as judged by a powerful vision-language model.

Table 3: Effects on different combinations of error functions. We evaluate each setting with metrics of camera control accuracy (*i.e.*, "*Trans. Err.*" and "*Rot. Err.*") on the RealEstate10K subset. $e_t(\cdot)$ and $e_R(\cdot)$ indicate the error function of translation and rotation, respectively. We report results on the image-to-video setting. † denotes the clip-level reward. The best result in each case is marked in bold.

$e_t(\cdot)$	$e_R(\cdot)$	*Trans. Err.*↓	*Rot. Err.*↓
SFT baseline		0.0280	0.4444
rel.	*abs.*	0.0292	0.4537
abs.	*rel.*	0.0293	0.4281
rel.	*rel.*	**0.0238**	**0.3286**
rel.†	*rel.*†	0.0277	0.3739

Table 4: Effects on different numbers of timesteps. Given the fixed number of sampling steps T_{total}, we evaluate effects on different numbers of SDE steps T_{sde} used in optimization. "*Time.*" denotes the consumed time per optimization step. We report results on the image-to-video setting. The best result in each case is marked in bold.

T_{sde} (T_{total})	*Trans. Err.*↓	*Rot. Err.*↓	*Time.*↓
1 (14)	0.0272	0.4829	**274**s
3 (14)	**0.0238**	**0.3286**	343s
5 (14)	0.0278	0.4440	412s

4.3 Main Results

Quantitative comparison. As demonstrated in Table 1, we compare our method with other state-of-the-art methods on the RealEstate10K test set, using camera-control accuracy, geometric consistency, and visual quality (VQ) in both text-to-video (T2V) and image-to-video (I2V) settings. Benefiting from the advanced training recipe and our curated dataset, CamVerse* clearly surpasses existing methods [3, 18] by large margins. Despite our strong SFT baselines, our online RL post-training (see CamVerse) further improves camera control and geometry. In T2V generation, the translation error drops **25.8%** from 0.0395 to 0.0293, while the rotation error drops **21.0%** from 0.6506 to 0.5140. Geometric consistency rises from 0.9081 to 0.9173 after post-training. Similar trends also show in I2V generation. Concretely, translation and rotation errors decrease by **15.1%** and **16.5%**, respectively. In the I2V setting, VQ slightly decreases after post-training (from 5.02 to 4.87), consistent with our reward not directly optimizing perceptual quality. Incorporating a multi-objective reward is an effective way to alleviate this issue.

Qualitative analysis. We visualize results for both T2V and I2V settings in Figure 4. Across a wide range of diverse, large-amplitude trajectories, our model closely follows the reference camera trajectories and maintains smooth local viewpoint changes at turning points. Meanwhile, subjects exhibit substantial dynamics (*e.g.*, running and riding a horse), and the rendered scenes remain temporally coherent, indicating that the model can disentangle camera movement from content motion while preserving visual fidelity. For the shown failure,

Table 5: Effects on best-of-N sampling. We evaluate effects on different numbers of positive and negative samples. "α-β" indicates α positive and β negative samples. "*Time.*" denotes the consumed time per optimization step. We report results on the image-to-video setting. The best result in each case is marked in bold.

Best-of-N	*Trans. Err.*↓	*Rot. Err.*↓	*Time.*↓
4-4	**0.0238**	**0.3286**	343s
2-2	0.0240	0.3760	294s
1-1	0.0249	0.4330	**269**s
1-4	0.0303	0.4948	305s
4-1	0.0262	0.3487	305s

Table 6: Effects on different ranks of LoRA. We evaluate each setting with metrics of camera control accuracy (*i.e.*, "*Trans. Err.*" and "*Rot. Err.*") on the RealEstate10K subset. We report results on the image-to-video setting. The best result in each case is marked in bold.

Rank r	*Trans. Err.*↓	*Rot. Err.*↓
32	**0.0233**	0.3550
64	0.0238	**0.3286**
128	0.0249	0.3778

Table 7: Effects on different timestep shift. We evaluate each setting with metrics of camera control accuracy (*i.e.*, "*Trans. Err.*" and "*Rot. Err.*") on the RealEstate10K subset. We report results on the image-to-video setting. The best result in each case is marked in bold.

Timestep shift	*Trans. Err.*↓	*Rot. Err.*↓
4	0.0264	0.3878
6	**0.0238**	**0.3286**
8	0.0297	0.4165

straightforward translation is still the dominant pattern in the dataset, even diversifies the trajectory distribution. The model therefore drifts toward straightforward motion rather than exactly executing the reference trajectory, resulting in early misalignment. Mitigating the dataset bias is an important direction for the research community.

4.4 Ablation Study

We analyze 6 design choices that influence GRPO fine-tuning. Unless otherwise noted, all of the results are reported on the RealEstate10K test set [70], containing 100 clips, in the I2V setting. We mainly evaluate camera control accuracy using translation error (*Trans. Err.*) and rotation error (*Rot. Err.*). For training efficiency, we report the time spent on the optimization step.

SFT training recipe. As illustrated in Table 2, we first compare different sample ratios for T2V and I2V, demonstrating that animation of the first frame encourages the model to learn camera movement. Besides, training without dynamic videos clearly degrades camera control accuracy. Afterwards, the shifted

timestep scheduler and pose normalization each contribute to advancing camera-control video generation. We therefore adopt this recipe for all RL experiments, excluding the sample ratio for T2V and I2V.

Error function. It is crucial to design an appropriate error function for the reward model. As reported in Table 3, using the relative error for both translation and rotation yields the lowest overall errors, outperforming the other mixed configurations. We further replace the dense, segment-level reward with a clip-level reward and apply GRPO fine-tuning. This variant results in noticeably worse camera-control accuracy, demonstrating the importance of dense feedback.

Number of SDE timesteps. With total sampling steps fixed, we vary the number of stochastic steps T_{sde} used during online rollouts, shown in Table 4. $T_{\text{sde}} = 3$ shows the most accurate control with 0.0238 translation error and 0.3286 rotation error at a reasonable cost. Too few steps are under-explored, while too many steps slow training and slightly decrease accuracy.

Best-of-N sampling. We evaluate the groups with α positive and β negative samples. A balanced larger group (4−4) achieves the best performance of 0.0238 translation error and 0.3286 rotation error. Smaller or asymmetric groups (*e.g.*, 1−1 and 1−4) increase both translation and rotation errors, which indicate that both adequate positive and negative samples are important.

Rank in LoRA. In Table 6, we evaluate the effects of capacity on different ranks in LoRA. As shown, $r = 64$ offers the best overall trade-off between camera control accuracy and trainable parameters. $r = 32$ slightly decreases translation error but harms rotation accuracy, while $r = 128$ degrades control accuracy and increases training costs.

Timestep shift. In the GRPO fine-tuning stage, we offset the diffusion timesteps by different constants when sampling and policy optimization, shown in Table 7. Camera control accuracy is highly sensitive to the model's behavior near the start of denoising. If the shift is too small, these noise-proximal steps are under-trained, and errors accumulate along the denoising process. Conversely, excessive shift overpowers extremely noisy states, producing blurrier rollouts. This causes challenges to camera pose estimation, resulting in inferior camera control accuracy.

5 Conclusion

In this paper, we presented CamVerse, an online RL framework that post-trains video diffusion models for precise camera control. Through a verifiable geometry reward that converts 3D alignment into dense segment-level feedback, our method effectively mitigates reward sparsity and leads to efficient model optimization. Coupled with a large-scale dataset covering diverse and high-amplitude camera trajectories, CamVerse achieves significant improvements in camera-control accuracy, geometric consistency, and visual quality over SFT competitors. These results highlight the promise of RL as a powerful post-training paradigm for refining generative video models beyond static supervision. In future work,

we will integrate our method into the world model for precise action control and explore it as a scalable data engine for embodied AI.

References

1. Cosmos World Foundation Model Platform for Physical AI
2. AISphere: Pixverse (2025), https://app.pixverse.ai, accessed: 2025-10-30
3. Bahmani, S., Skorokhodov, I., Qian, G., Siarohin, A., Menapace, W., Tagliasacchi, A., Lindell, D.B., Tulyakov, S.: Ac3d: Analyzing and improving 3d camera control in video diffusion transformers. arXiv preprint arXiv:2411.18673 (2024)
4. Bahmani, S., Skorokhodov, I., Siarohin, A., Menapace, W., Qian, G., Vasilkovsky, M., Lee, H.Y., Wang, C., Zou, J., Tagliasacchi, A., Lindell, D.B., Tulyakov, S.: Vd3d: Taming large video diffusion transformers for 3d camera control. arXiv preprint arXiv:2407.12781 (2024)
5. Bai, J., Xia, M., Wang, X., Yuan, Z., Fu, X., Liu, Z., Hu, H., Wan, P., Zhang, D.: Syncammaster: Synchronizing multi-camera video generation from diverse viewpoints. arXiv preprint arXiv:2412.07760 (2024)
6. Bain, M., Nagrani, A., Varol, G., Zisserman, A.: Frozen in time: A joint video and image encoder for end-to-end retrieval. In: ICCV. pp. 1728–1738 (2021)
7. Blattmann, A., Dockhorn, T., Kulal, S., Mendelevitch, D., Kilian, M., Lorenz, D., Levi, Y., English, Z., Voleti, V., Letts, A., et al.: Stable video diffusion: Scaling latent video diffusion models to large datasets. arXiv preprint arXiv:2311.15127 (2023)
8. Blattmann, A., Rombach, R., Ling, H., Dockhorn, T., Kim, S.W., Fidler, S., Kreis, K.: Align your latents: High-resolution video synthesis with latent diffusion models. In: CVPR. pp. 22563–22575 (2023)
9. Brooks, T., Peebles, B., Holmes, C., DePue, W., Guo, Y., Jing, L., Schnurr, D., Taylor, J., Luhman, T., Luhman, E., Ng, C., Wang, R., Ramesh, A.: Video generation models as world simulators (2024), https://openai.com/research/video-generation-models-as-world-simulators
10. Chen, T.S., Siarohin, A., Menapace, W., Deyneka, E., Chao, H.w., Jeon, B.E., Fang, Y., Lee, H.Y., Ren, J., Yang, M.H., et al.: Panda-70m: Captioning 70m videos with multiple cross-modality teachers. In: CVPR. pp. 13320–13331 (2024)
11. Esser, P., Kulal, S., Blattmann, A., Entezari, R., Müller, J., Saini, H., Levi, Y., Lorenz, D., Sauer, A., Boesel, F., et al.: Scaling rectified flow transformers for high-resolution image synthesis. In: ICML (2024)
12. Feng, R., Zhang, H., Yang, Z., Xiao, J., Shu, Z., Liu, Z., Zheng, A., Huang, Y., Liu, Y., Zhang, H.: The matrix: Infinite-horizon world generation with real-time moving control. arXiv preprint arXiv:2412.03568 (2024)
13. Feng, W., Liu, J., Tu, P., Qi, T., Sun, M., Ma, T., Zhao, S., Zhou, S., He, Q.: I2vcontrol-camera: Precise video camera control with adjustable motion strength. In: ICLR (2025)
14. Fu, X., Liu, X., Wang, X., Peng, S., Xia, M., Shi, X., Yuan, Z., Wan, P., Zhang, D., Lin, D.: 3dtrajmaster: Mastering 3d trajectory for multi-entity motion in video generation. arXiv preprint arXiv:2412.07759 (2024)
15. Guo, Y., Yang, C., Rao, A., Agrawala, M., Lin, D., Dai, B.: Sparsectrl: Adding sparse controls to text-to-video diffusion models. In: ECCV. pp. 330–348 (2024)
16. Guo, Y., Yang, C., Rao, A., Liang, Z., Wang, Y., Qiao, Y., Agrawala, M., Lin, D., Dai, B.: Animatediff: Animate your personalized text-to-image diffusion models without specific tuning. arXiv preprint arXiv:2307.04725 (2023)

17. Gupta, A., Yu, L., Sohn, K., Gu, X., Hahn, M., Li, F.F., Essa, I., Jiang, L., Lezama, J.: Photorealistic video generation with diffusion models. In: ECCV. pp. 393–411 (2024)
18. He, H., Xu, Y., Guo, Y., Wetzstein, G., Dai, B., Li, H., Yang, C.: Cameractrl: Enabling camera control for text-to-video generation. arXiv preprint arXiv:2404.02101 (2024)
19. He, H., Yang, C., Lin, S., Xu, Y., Wei, M., Gui, L., Zhao, Q., Wetzstein, G., Jiang, L., Li, H.: Cameractrl ii: Dynamic scene exploration via camera-controlled video diffusion models. arXiv preprint arXiv:2503.10592 (2025)
20. Ho, J., Salimans, T., Gritsenko, A., Chan, W., Norouzi, M., Fleet, D.J.: Video diffusion models. In: NeurIPS. pp. 8633–8646 (2022)
21. Hong, W., Ding, M., Zheng, W., Liu, X., Tang, J.: Cogvideo: Large-scale pretraining for text-to-video generation via transformers. arXiv preprint arXiv:2205.15868 (2022)
22. Hou, C., Wei, G., Zeng, Y., Chen, Z.: Training-free camera control for video generation. arXiv preprint arXiv:2406.10126 (2024)
23. Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W., et al.: Lora: Low-rank adaptation of large language models. In: ICLR (2022)
24. Hu, Y., Luo, C., Chen, Z.: Make it move: controllable image-to-video generation with text descriptions. In: CVPR. pp. 18219–18228 (2022)
25. Huang, J., Zhou, Q., Rabeti, H., Korovko, A., Ling, H., Ren, X., Shen, T., Gao, J., Slepichev, D., Lin, C.H., et al.: Vipe: Video pose engine for 3d geometric perception. arXiv preprint arXiv:2508.10934 (2025)
26. Huang, Z., He, Y., Yu, J., Zhang, F., Si, C., Jiang, Y., Zhang, Y., Wu, T., Jin, Q., Chanpaisit, N., Wang, Y., Chen, X., Wang, L., Lin, D., Qiao, Y., Liu, Z.: VBench: Comprehensive benchmark suite for video generative models. In: CVPR (2024)
27. Huang, Z., Zhang, F., Xu, X., He, Y., Yu, J., Dong, Z., Ma, Q., Chanpaisit, N., Si, C., Jiang, Y., Wang, Y., Chen, X., Chen, Y.C., Wang, L., Lin, D., Qiao, Y., Liu, Z.: Vbench++: Comprehensive and versatile benchmark suite for video generative models. arXiv preprint arXiv:2411.13503 (2024)
28. Karras, T., Aittala, M., Aila, T., Laine, S.: Elucidating the design space of diffusion-based generative models. In: NeurIPS. pp. 26565–26577 (2022)
29. Keetha, N., Müller, N., Schönberger, J., Porzi, L., Zhang, Y., Fischer, T., Knapitsch, A., Zauss, D., Weber, E., Antunes, N., et al.: Mapanything: Universal feed-forward metric 3d reconstruction. arXiv preprint arXiv:2509.13414 (2025)
30. KuaiShou: Kling (2025), `https://klingai.com/`, accessed: 2025-02-23
31. Kuang, Z., Cai, S., He, H., Xu, Y., Li, H., Guibas, L.J., Wetzstein, G.: Collaborative video diffusion: Consistent multi-video generation with camera control. In: NeurIPS. pp. 16240–16271 (2025)
32. Labs, B.F., Batifol, S., Blattmann, A., Boesel, F., Consul, S., Diagne, C., Dockhorn, T., English, J., English, Z., Esser, P., et al.: Flux. 1 kontext: Flow matching for in-context image generation and editing in latent space. arXiv preprint arXiv:2506.15742 (2025)
33. Lawrence, J., Bernal, J., Witzgall, C.: A purely algebraic justification of the kabsch-umeyama algorithm. Journal of research of the National Institute of Standards and Technology **124**, 1 (2019)
34. Li, B., Zheng, C., Zhu, W., Mai, J., Zhang, B., Wonka, P., Ghanem, B.: Vivid-zoo: Multi-view video generation with diffusion model. In: NeurIPS. pp. 62189–62222 (2025)

35. Lin, B., Ge, Y., Cheng, X., Li, Z., Zhu, B., Wang, S., He, X., Ye, Y., Yuan, S., Chen, L., et al.: Open-sora plan: Open-source large video generation model. arXiv preprint arXiv:2412.00131 (2024)
36. Liu, J., Liu, G., Liang, J., Li, Y., Liu, J., Wang, X., Wan, P., Zhang, D., Ouyang, W.: Flow-grpo: Training flow matching models via online rl. arXiv preprint arXiv:2505.05470 (2025)
37. Liu, J., Liu, G., Liang, J., Yuan, Z., Liu, X., Zheng, M., Wu, X., Wang, Q., Xia, M., Wang, X., et al.: Improving video generation with human feedback. arXiv preprint arXiv:2501.13918 (2025)
38. Liu, R., Wu, H., Zheng, Z., Wei, C., He, Y., Pi, R., Chen, Q.: Videodpo: Omni-preference alignment for video diffusion generation. In: CVPR. pp. 8009–8019 (2025)
39. Liu, X., Gong, C., Liu, Q.: Flow straight and fast: Learning to generate and transfer data with rectified flow. arXiv preprint arXiv:2209.03003 (2022)
40. Luo, R., Xia, X., Wang, L., Chen, L., Shan, R., Luo, J., Yang, M., Chua, T.S.: Next-omni: Towards any-to-any omnimodal foundation models with discrete flow matching. arXiv preprint arXiv:2510.13721 (2025)
41. Ma, G., Huang, H., Yan, K., Chen, L., Duan, N., Yin, S., Wan, C., Ming, R., Song, X., Chen, X., et al.: Step-video-t2v technical report: The practice, challenges, and future of video foundation model. arXiv preprint arXiv:2502.10248 (2025)
42. Ma, X., Wang, Y., Jia, G., Chen, X., Liu, Z., Li, Y.F., Chen, C., Qiao, Y.: Latte: Latent diffusion transformer for video generation. arXiv preprint arXiv:2401.03048 (2024)
43. Parker-Holder, J., Ball, P., Bruce, J., Dasagi, V., Holsheimer, K., Kaplanis, C., Moufarek, A., Scully, G., Shar, J., Shi, J., Spencer, S., Yung, J., Dennis, M., Kenjeyev, S., Long, S., Mnih, V., Chan, H., Gazeau, M., Li, B., Pardo, F., Wang, L., Zhang, L., Besse, F., Harley, T., Mitenkova, A., Wang, J., Clune, J., Hassabis, D., Hadsell, R., Bolton, A., Singh, S., Rocktäschel, T.: Genie 2: A large-scale foundation world model (2024), https://deepmind.google/discover/blog/genie-2-a-large-scale-foundation-world-model/
44. Peebles, W., Xie, S.: Scalable diffusion models with transformers. In: ICCV. pp. 4195–4205 (2023)
45. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: CVPR. pp. 10684–10695 (2022)
46. Shao, Z., Wang, P., Zhu, Q., Xu, R., Song, J., Bi, X., Zhang, H., Zhang, M., Li, Y., Wu, Y., et al.: Deepseekmath: Pushing the limits of mathematical reasoning in open language models. arXiv preprint arXiv:2402.03300 (2024)
47. Shi, X., Huang, Z., Wang, F.Y., Bian, W., Li, D., Zhang, Y., Zhang, M., Cheung, K.C., See, S., Qin, H., et al.: Motion-i2v: Consistent and controllable image-to-video generation with explicit motion modeling. In: ACM SIGGRAPH. pp. 1–11 (2024)
48. Singer, U., Polyak, A., Hayes, T., Yin, X., An, J., Zhang, S., Hu, Q., Yang, H., Ashual, O., Gafni, O., et al.: Make-a-video: Text-to-video generation without text-video data. arXiv preprint arXiv:2209.14792 (2022)
49. Sitzmann, V., Rezchikov, S., Freeman, B., Tenenbaum, J., Durand, F.: Light field networks: Neural scene representations with single-evaluation rendering. In: NeurIPS. pp. 19313–19325 (2021)
50. Team, H.F.M.: Hunyuanvideo: A systematic framework for large video generative models (2024), https://arxiv.org/abs/2412.03603
51. Valevski, D., Leviathan, Y., Arar, M., Fruchter, S.: Diffusion models are real-time game engines. arXiv preprint arXiv:2408.14837 (2024)

52. Wan, T., Wang, A., Ai, B., Wen, B., Mao, C., Xie, C.W., Chen, D., Yu, F., Zhao, H., Yang, J., et al.: Wan: Open and advanced large-scale video generative models. arXiv preprint arXiv:2503.20314 (2025)
53. Wan, Z., Xu, Y., Wang, Z., Liu, F., Liu, T., Gong, M.: Ted-viton: Transformer-empowered diffusion models for virtual try-on. arXiv preprint arXiv:2411.17017 (2024)
54. Wang, J., Chen, M., Karaev, N., Vedaldi, A., Rupprecht, C., Novotny, D.: Vggt: Visual geometry grounded transformer. In: CVPR. pp. 5294–5306 (2025)
55. Wang, J., Wang, Z., Pan, H., Liu, Y., Yu, D., Wang, C., Wang, W.: Mmgen: Unified multi-modal image generation and understanding in one go. arXiv preprint arXiv:2503.20644 (2025)
56. Wang, S., Leroy, V., Cabon, Y., Chidlovskii, B., Revaud, J.: Dust3r: Geometric 3d vision made easy. In: CVPR. pp. 20697–20709 (2024)
57. Wang, Y., Zhou, J., Zhu, H., Chang, W., Zhou, Y., Li, Z., Chen, J., Pang, J., Shen, C., He, T.: π^3: Permutation-equivariant visual geometry learning. arXiv preprint arXiv:2507.13347 (2025)
58. Wang, Z., Xia, X., Chen, R., Yu, D., Wang, C., Gong, M., Liu, T.: Lavin-dit: Large vision diffusion transformer. In: CVPR. pp. 20060–20070 (2025)
59. Wang, Z., Yuan, Z., Wang, X., Li, Y., Chen, T., Xia, M., Luo, P., Shan, Y.: Motionctrl: A unified and flexible motion controller for video generation. In: ACM SIGGRAPH. pp. 1–11 (2024)
60. Xing, J., Xia, M., Zhang, Y., Chen, H., Yu, W., Liu, H., Liu, G., Wang, X., Shan, Y., Wong, T.T.: Dynamicrafter: Animating open-domain images with video diffusion priors. In: ECCV. pp. 399–417 (2024)
61. Xu, D., Jiang, Y., Huang, C., Song, L., Gernoth, T., Cao, L., Wang, Z., Tang, H.: Cavia: Camera-controllable multi-view video diffusion with view-integrated attention. arXiv preprint arXiv:2410.10774 (2024)
62. Xu, D., Nie, W., Liu, C., Liu, S., Kautz, J., Wang, Z., Vahdat, A.: Camco: Camera-controllable 3d-consistent image-to-video generation. arXiv preprint arXiv:2406.02509 (2024)
63. Xue, H., Hang, T., Zeng, Y., Sun, Y., Liu, B., Yang, H., Fu, J., Guo, B.: Advancing high-resolution video-language representation with large-scale video transcriptions. In: CVPR. pp. 5036–5045 (2022)
64. Xue, Z., Wu, J., Gao, Y., Kong, F., Zhu, L., Chen, M., Liu, Z., Liu, W., Guo, Q., Huang, W., et al.: Dancegrpo: Unleashing grpo on visual generation. arXiv preprint arXiv:2505.07818 (2025)
65. Yang, J., Sax, A., Liang, K.J., Henaff, M., Tang, H., Cao, A., Chai, J., Meier, F., Feiszli, M.: Fast3r: Towards 3d reconstruction of 1000+ images in one forward pass. In: CVPR. pp. 21924–21935 (2025)
66. Yang, Z., Teng, J., Zheng, W., Ding, M., Huang, S., Xu, J., Yang, Y., Hong, W., Zhang, X., Feng, G., et al.: Cogvideox: Text-to-video diffusion models with an expert transformer. arXiv preprint arXiv:2408.06072 (2024)
67. Yu, J., Qin, Y., Wang, X., Wan, P., Zhang, D., Liu, X.: Gamefactory: Creating new games with generative interactive videos. arXiv preprint arXiv:2501.08325 (2025)
68. Zhang, S., Wang, J., Xu, Y., Xue, N., Rupprecht, C., Zhou, X., Shen, Y., Wetzstein, G.: Flare: Feed-forward geometry, appearance and camera estimation from uncalibrated sparse views. arXiv preprint arXiv:2502.12138 (2025)
69. Zheng, J., Pan, S., Yao, Y., Wang, Z., Wang, D., Liu, T.: Aligning what matters: Masked latent adaptation for text-to-audio-video generation. In: NeurIPS. pp. 173244–173272 (2026)

70. Zhou, T., Tucker, R., Flynn, J., Fyffe, G., Snavely, N.: Stereo magnification: Learning view synthesis using multiplane images. arXiv preprint arXiv:1805.09817 (2018)
71. Zhou, Z., Xia, X., Ma, F., Fan, H., Yang, Y., Chua, T.S.: Dreamdpo: Aligning text-to-3d generation with human preferences via direct preference optimization. In: ICML (2025)