VIEW4D: SYNCHRONIZED MULTI-VIEW VIDEO GENERATION IN THE OPEN WORLD

Anonymous authors
Paper under double-blind review

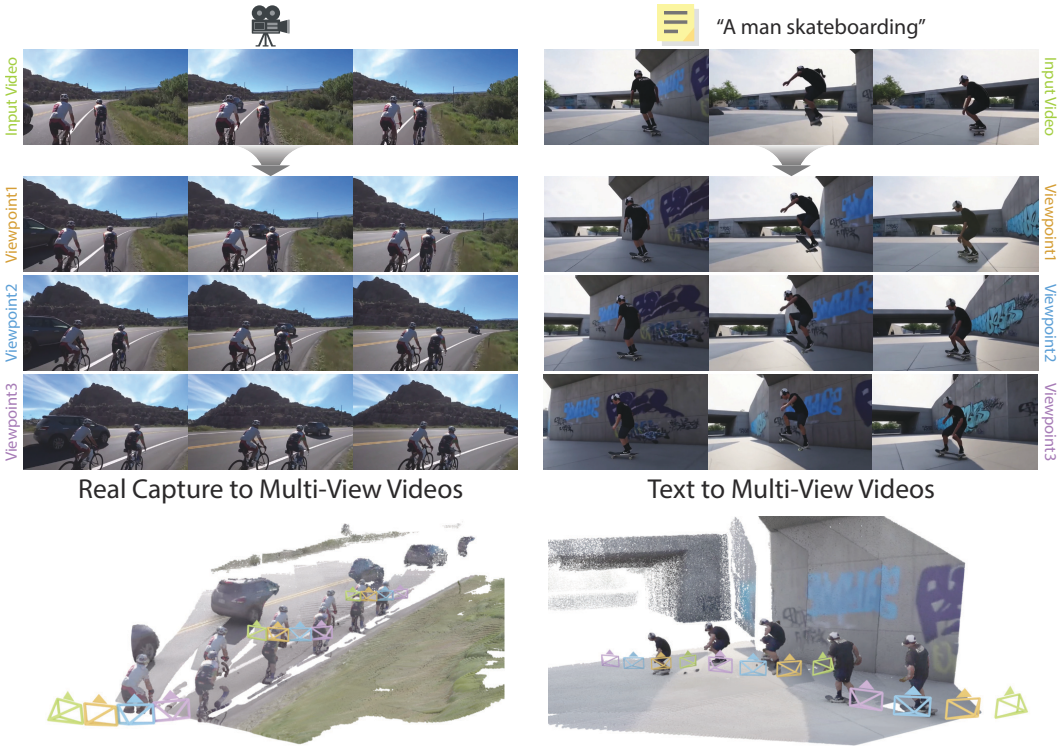

Figure 1: View4D enables spatially and temporally consistent multi-view video generation with camera motion from real (left) or generated (right) monocular videos.

ABSTRACT

We present View4D, a generative method that synthesizes synchronized multi-view videos with moving cameras from a single input video. Video generation models have recently evolved toward the new paradigm of world models, yet they typically generate scenes along a single camera trajectory. While recent methods explore multi-view video generation in the open world, they remain limited to static cameras. To achieve multi-view videos with camera movement, we propose a multi-view attention module that incorporates camera poses as relative positional encodings to ensure cross-view consistency. We further design an optional geometry-injection module to seamlessly leverage the 3D structures extracted from the input video. Finally, a hybrid training strategy is introduced to effectively integrate synthetic multi-view videos with real-world images and videos. Extensive experiments demonstrate that View4D significantly outperforms existing state-of-the-art methods, achieving superior geometric consistency and visual fidelity on complex in-the-wild videos. Overall, our work marks a crucial step in advancing generative world models—transitioning from single-camera trajectories to synchronized 4D observation under continuous camera motion.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

1 INTRODUCTION

A video captures one slice of a moment, inherently representing a 2D projection of a 4D world along a single camera trajectory. While current video generators (Brooks et al., 2024; Google DeepMind, 2025; Wan et al., 2025; Ball et al., 2025) produce highly realistic results, they are largely limited to generating this single-view slice. The actual moment — the scene as observed simultaneously from multiple moving viewpoints — remains unmodeled. We categorize the progression toward generating this complete moment into four stages, each adding a new capability over the previous one:

(S1) **A slice through time**: A controllable video generator redirects the input video given a single camera trajectory (Bai et al., 2025a; Yu et al., 2025; Yang et al., 2026), but can generate only one target trajectory in each inference run.
(S2) **The moment around an object**: Multi-view video generation is introduced, but the output videos are restricted to object-centric setups with static viewpoints (Yao et al., 2025).
(S3) **The moment across a scene**: Multi-view video generation is extended to open-domain scenes, but the cameras remain static (Wang et al., 2025a;b; Bai et al., 2025b).
(S4) **The moment in motion**: Synchronized multi-view video generation with moving cameras in arbitrary open-domain scenes, which remains unaddressed as shown in Figure 2.

An alternative paradigm, 4D reconstruction (Mildenhall et al., 2020; Kerbl et al., 2023; Wang et al., 2025d; Lei et al., 2025; Xu et al., 2025; Lin et al., 2026a), also targets S4 by lifting the input video into a 4D representation for novel view rendering. However, these methods strictly inherit the blind spots of the input camera, leaving severe artifacts or holes whenever the novel viewpoint extends beyond the original observation. To overcome this fundamental limitation, we propose generative framework **View4D** to achieve S4. Given a single monocular video or text input, View4D synthesizes synchronized multi-view videos with cameras moving together through arbitrary open-domain scenes as shown in Figure 1.

To recover the moment rather than a single-view slice, a model must reason across cameras. However, finetuning a pretrained backbone (Wan et al., 2025) overwrites the temporal prior learned from large-scale video pretraining. As a result, View4D freezes the pretrained spatio-temporal attention and instead adds a parallel multi-view attention module. To enable camera control, we inject the projection matrix of each view as a relative positional encoding (Li et al., 2025) inside this module. We find that jointly modeling the multi-view output improves cross-view consistency over predicting each view independently, and the projection-based encoding allows the model to generalize to varying camera configurations at inference. Moreover, when the input video is paired with per-frame depth estimated by geometry foundation models (Wang et al., 2026b; Lin et al., 2026b; Wang et al., 2025c; Zhang et al., 2025), we unproject each depth map into a 3D point cloud and reproject it into every target view. Then, a lightweight encoder injects the resulting pixel-level condition into the diffusion transformer, further improving alignment with the target cameras.

Training such a model is constrained by limited multi-view data. Currently, no public corpus provides open-domain multi-view videos captured with camera motion. To bridge this critical data gap, we construct a synchronized multi-view video dataset using synthetic assets rendered in game engines (Epic Games, 2025), covering diverse scenes and camera trajectories. We further augment training with multi-view images of static scenes (Ling et al., 2024) for real appearance and structure and multi-view videos (Sun et al., 2020) in the driving domain for real-world dynamics.

In summary, this work makes the following contributions:

- We introduce View4D, a novel generative model that synthesizes synchronized multi-view videos with camera motion from a monocular video or text input in open-domain scenes.
- We propose a relative-projection multi-view attention with an optional depth-reprojection module, achieving cross-view consistency and generalization across diverse camera configurations.
- We design a hybrid training strategy that couples synthetic multi-view videos with real-world multi-view images and videos to leverage synthetic multi-view priors while learning authentic scene dynamics and realistic viewpoint transformations from real data.
- Experiments demonstrate that our approach significantly outperforms prior state-of-the-art methods for synchronized multi-view video generation on challenging videos in the open world.

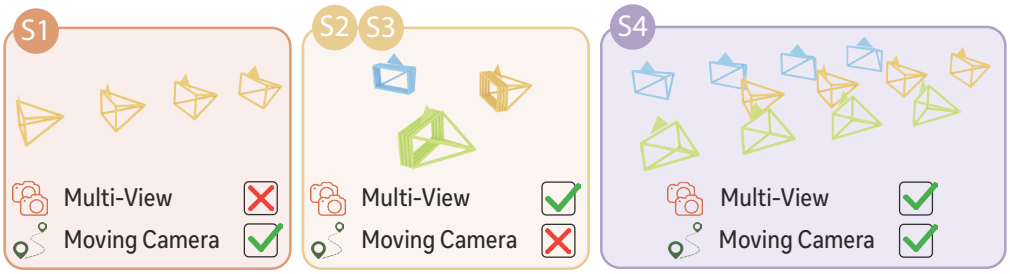

Figure 2: Illustration of video generation paradigms across stages defined in Section 1. While previous methods belong to S1-S3, View4D takes a step into S4, enabling synchronized multi-view video generation with camera motion in the open world.

2 Related Work

2.1 Camera-Controlled Video Generation

Early works in this direction primarily inject camera information into video models to achieve controllable generation from text or image inputs (Wang et al., 2024b; He et al., 2025; Bahmani et al., 2025a;b; Yu et al., 2024). Given an input video, prior methods generally fall into two main categories. The first category relies on game engines to construct synthetic datasets, training models to learn the target distribution conditioned on implicit camera poses (Van Hoorick et al., 2024; Bai et al., 2025a; Van Hoorick et al., 2026). Following this, later methods have progressively enabled temporally aware control (Wang et al., 2026c; Huang et al., 2026) while exploring alternative representations for camera trajectories (Luo et al., 2025). Fu et al. (2026) extends this paradigm to autoregressive multi-camera video generation based on a memory bank. However, these works produces only a single segment during each forward pass whereas View4D generates all views jointly in a single forward pass.

The second category constructs explicit geometric information, such as depth maps, from the input video to guide the target view generation (Ren et al., 2025b; Gu et al., 2025; Huang et al., 2025; Cao et al., 2025; Lin et al., 2026c). Current methods typically rely on depth double reprojection or degraded rendering to fine-tune video models on monocular videos (Yu et al., 2025; Yang et al., 2026; Lu et al., 2026; Hu et al., 2025). While effective for open-world videos, these methods fail to fully leverage the visual information of the input video during training, and they yield geometrically inconsistent results when extended to multi-view settings. Furthermore, naively extending them to generate multiple views iteratively suffers from severe error accumulation. In this work, we also leverage the explicit geometric information provided by recent geometry foundation models (Wang et al., 2025c; 2026b; Lin et al., 2026b; Jin et al., 2026; Cong et al., 2026) as an optional guidance, while directly conditioning on the input video during training and jointly predicting synchronized multi-view videos.

2.2 Multi-View Video Generation

In the field of multi-view video generation, early works (Zhang et al., 2024; Jiang et al., 2024; Li et al., 2024; Yang et al., 2025; Yao et al., 2025) predominantly focus on object-centric domains. While effective for 4D object synthesis, these approaches fail to generalize to open-world scenes with complex backgrounds and multiple dynamic elements. Recent efforts have extended this task to open-world scenes with more diverse camera configurations by training on augmented monocular videos and synthetic data (Wang et al., 2025b;a; Wu et al., 2025; Bai et al., 2025b). A shared limitation of these methods is that they assume the output viewpoint to be static over time. Directly applying them to videos with camera motion results in geometric distortions and background artifacts. Although one recent attempt (Kuang et al., 2024) enables multiple camera trajectories, its formulation natively supports only two paired views. Consequently, when conditioned on an input video, it occupies one of the two slots, leaving the model to output only a single target view rather than actual multi-view videos. In parallel, domain-specific methods have achieved moving multi-view generation for driving scenes (Ren et al., 2025a; Wen et al., 2024; Hu et al., 2023; Gao et al., 2024; Wang et al., 2024a). However, they are inherently constrained by fixed camera configurations and the necessity of

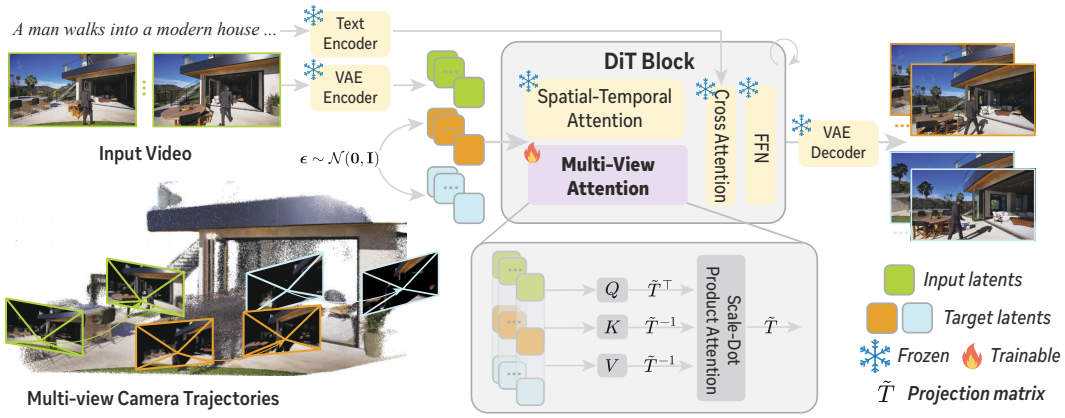

Figure 3: **Overview of View4D.** Given an input video and the multi-view camera trajectories, we combine the input latent with the multi-view target latents and feed them into the diffusion transformer. Within each DiT block, a multi-view attention module aggregates information across the latents at each frame, where the camera trajectory is injected as a relative positional encoding via projection matrices. The model jointly predicts the latents for multiple target views, which are finally decoded into synchronized multi-view videos.

additional conditions like HD maps and 3D boxes. In contrast, our method is designed to generate moving multi-view videos in the open world without additional constraints.

3 METHOD

Given an input video $V_0 \in \mathbb{R}^{F\times C\times H\times W}$ with corresponding text prompt P, camera poses $\{T_{0,i}\}_{i=1}^{F}$ and intrinsic matrix K, our goal is to generate n synchronized target videos $\{V_j\}_{j=1}^{n}$ with their associated camera poses $\big\{\{T_{j,i}\}_{i=1}^{F}\big\}_{j=1}^{n}$:

$$T_{j,i} = T_{0\to j}\cdot T_{0,i}, \qquad j = 1,\dots,n,\ i = 1,\dots,F, \tag{1}$$

where $\{T_{0\to j}\}_{j=1}^{n}$ is the pre-defined multi-view camera configuration shared across all frames. The camera parameters $\{T_{0,i}\}_{i=1}^{F}$ and K can be estimated from the input video by off-the-shelf geometry foundation models (Lin et al., 2026b; Wang et al., 2026b). We begin by briefly reviewing flow matching-based video generation models in Section 3.1. Next, we detail the architecture of View4D in Sections 3.2 and 3.3. Finally, we elaborate on the training data construction in Section 3.4.

3.1 PRELIMINARIES

Our method is built upon a pretrained text-to-video (T2V) latent diffusion model composed of a causal 3D Variational Autoencoder (VAE) and a Diffusion Transformer (DiT) (Peebles & Xie, 2023). The 3D VAE encoder compresses a video sequence $\mathbf{x}$ into a lower-dimensional latent $\mathbf{z} = \mathcal{E}(\mathbf{x})$ along both spatial and temporal axes, while the decoder reconstructs the video as $\hat{\mathbf{x}} = \mathcal{D}(\mathbf{z})$.

The DiT operates under the Rectified Flow framework (Liu et al., 2022; Lipman et al., 2022), where the forward process constructs a linear path between data and noise:

$$\mathbf{z}_t = (1-t)\,\mathbf{z} + t\,\boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\mathbf{I}),\ t\in[0,1]. \tag{2}$$

The model is trained to predict the velocity field $\mathbf{v}_\Theta$ via Conditional Flow Matching:

$$\mathcal{L} = \mathbb{E}_{t,\,\mathbf{z}_0,\,\boldsymbol{\epsilon}}\,\|\mathbf{v}_\Theta(\mathbf{z}_t,t,\mathbf{c}) - (\boldsymbol{\epsilon}-\mathbf{z})\|_2^2, \tag{3}$$

where $\mathbf{c}$ denotes the text condition encoded by a T5 text encoder and injected into each DiT block through cross-attention. Each Transformer block sequentially applies spatio-temporal self-attention, cross-attention, and FFN modules. The self-attention module employs 3D RoPE (Su et al., 2021), which partitions each head's channels into three groups and applies independent rotary embeddings along the temporal, height, and width axes respectively. During inference, clean latents are recovered from Gaussian noise by numerically integrating the learned velocity field using an ODE solver.

3.2 Multi-View Attention with Camera Conditioning

In this section, our goal is to extend the pretrained text-to-video model to synthesize synchronized multi-view videos. One direct approach concatenates multi-view tokens along the temporal axis and finetunes the self-attention module. However, this disrupts the pretrained temporal dynamics and incurs large memory cost as the number of views grows. Instead, we freeze the pretrained backbone and introduce a parallel attention module to perform cross-view interaction.

To be specific, we feed the diffusion transformer with the multi-view latent stack $[\mathbf{z}_0, \mathbf{z}_1, \ldots, \mathbf{z}_n]$, where $\mathbf{z}_0 = \mathcal{E}(V_0)$ is the clean latent of the input view and $\{\mathbf{z}_j\}_{j=1}^n$ are the noisy latents of the target views being denoised. For each transformer block, we add a multi-view attention layer in parallel with the frozen self-attention. Unlike the base self-attention, which attends within a single view's $f \cdot h \cdot w$ spatio-temporal tokens, our multi-view attention attends across the $(n+1) \cdot h \cdot w$ tokens of all views at each frame, enabling cross-view feature aggregation. Here f, h, w refer to the latent resolution. The camera poses are sub-sampled from F to f frames to match the latent temporal axis.

We choose to inject per-frame camera information into the multi-view attention as relative position encoding inspired by Li et al. (2025). Let $\tilde{T}_{j,i} \in \mathbb{R}^{4\times4}$ denote the lifted projection matrix combining the extrinsic $T_{j,i}$ and intrinsic K for view j at frame i. As shown in Figure 3, we apply block-diagonal transforms to the query, key, and value tokens of each token group (j, i):

$$\mathbf{Q}_{j,i} \leftarrow \mathbf{D}_{j,i}^{\top}\mathbf{Q}_{j,i} \qquad \mathbf{K}_{j,i} \leftarrow \mathbf{D}_{j,i}^{-1}\mathbf{K}_{j,i} \qquad \mathbf{V}_{j,i} \leftarrow \mathbf{D}_{j,i}^{-1}\mathbf{V}_{j,i} \tag{4}$$

and map the attention output back via $\mathbf{O}_{j,i} \leftarrow \mathbf{D}_{j,i}\mathbf{O}_{j,i}$. Here, $\mathbf{D}_{j,i}$ denotes the block-diagonal lifting of $\tilde{T}_{j,i}$ to the attention feature space. Details of its construction are provided in the supplementary.

Unlike previous approaches that add camera control through learnable encoders (Bai et al., 2025a; Yao et al., 2025), our design does not introduce additional learnable parameters. Moreover, for any two views $j, k \in \{0, \ldots, n\}$, the inner product between their tokens at the same frame i depends only on the relative projective relation $\tilde{T}_{j,i}\tilde{T}_{k,i}^{-1}$. According to Eq. (1), the attention computation remains unaffected by the global world coordinate and potential noise in the estimated poses. Therefore, our conditioning scheme guides the model to capture the underlying geometric logic of the multi-view camera layout, enabling accurate modeling of various camera configurations as shown in Figure 5.

3.3 Explicit Geometric Guidance

The multi-view attention module described in Section 3.2 enables multi-view video synthesis through an implicit geometric signal. Orthogonal to this, the geometry foundation models used at the beginning of Section 3 to estimate camera parameters also produce per-frame depth $D_0 \in \mathbb{R}^{F\times H\times W}$ with confidence $C_0 \in \mathbb{R}^{F\times H\times W}$, which serve as a readily available explicit geometric cue.

To apply this cue to each target view, we first lift D_0 into per-frame 3D point clouds and perform bilinear point splatting onto view j using the relative transformation $T_{0\to j}$, yielding per-view warped frames $\hat{V}_j$ and masks M_j. More details about the warping procedure can be found in the supplementary. We then encode $\hat{V}_j$ into the latent space via the frozen VAE encoder $\mathcal{E}$ to obtain $\hat{\mathbf{z}}_j$ and downsample M_j to the corresponding latent resolution to obtain $\bar{\mathbf{m}}_j$. For each target view $j = 1, \ldots, n$, we pass the concatenation $[\hat{\mathbf{z}}_j, \bar{\mathbf{m}}_j]$ through a trainable lightweight encoder $\mathcal{G}$ and add the result to the corresponding multi-view video latent $\mathbf{z}_j$ at the input of the diffusion transformer:

$$\mathbf{z}_j \leftarrow \mathbf{z}_j + \mathcal{G}([\hat{\mathbf{z}}_j, \bar{\mathbf{m}}_j]), \quad j = 1, \ldots, n. \tag{5}$$

In comparison with previous works that rely solely on geometric conditions as input (Yu et al., 2025; Yang et al., 2026; Hu et al., 2025; Lu et al., 2026) during training, our model accepts both the input video and geometric guidance. This encourages the model to better preserve appearance details and generate scene-consistent content in regions not covered by the guidance signal. In practice, we treat the explicit geometric guidance as an optional input to the multi-view attention module and randomly drop it during training, so that inference remains valid with or without depth. This ensures that the model learns to perform view synthesis from camera poses alone, while being able to further improve view transformation accuracy when the geometric guidance is available.

3.4 Training Data

Training dataset. The scarcity of multi-view videos with camera motion for open-world scenes is a key obstacle to training View4D. To address this, we construct a synchronized multi-view video dataset by rendering diverse dynamic scenes with various camera trajectories using real-time rendering engines. More details about the constructed multi-view video dataset can be found in the supplementary. To improve camera configuration diversity and reinforce multi-view geometric constraints, we incorporate multi-view images from DL3DV (Ling et al., 2024) as data augmentation by treating each image as a single-frame videos. Finally, we include a real multi-view video dataset Waymo (Sun et al., 2020) from the driving domain to further enhance authentic dynamics.

Scale alignment. Consistent scene scale across different training datasets is important to avoid ambiguity in camera pose conditioning. Our rendered multi-view videos and Waymo dataset are at metric scale, whereas the camera poses of DL3DV estimated by COLMAP (Schönberger & Frahm, 2016) are scale-ambiguous. To resolve this, we run Depth Anything 3 (Lin et al., 2026b) on a sparse set of keyframes with the corresponding poses as conditions and recover a Sim(3) transformation, thereby aligning all DL3DV poses to the metric scale. We apply an analogous alignment to the depth used for geometric guidance in Section 3.3. On DL3DV, we apply the scale of the recovered Sim(3) transformation to the estimated depth. On the rendered multi-view videos, we run Pi3 (Wang et al., 2026b) with the ground-truth poses as input and recover a scale factor by aligning the predicted and ground-truth trajectories. For the Waymo dataset, geometric guidance is omitted during training.

4 Experiments

4.1 Experiment Setup

Implementation details. We build our model upon Wan2.1-T2V-14B (Wan et al., 2025). The multi-view attention is added to the first 30 of the 40 diffusion transformer blocks. The total number of trainable parameters is around 3B. During training, the model takes one view as input and jointly predicts n target views. We train our model for 20,000 steps with a batch size of 32. We use the AdamW optimizer (Loshchilov & Hutter, 2019) with a learning rate of 5×10^{-5} and a weight decay of 1×10^{-2}. The output layer of multi-view attention and guidance encoder are zero-initialized following Zhang et al. (2023). The text prompt and geometric guidance are independently dropped with probabilities of 20% and 50%, respectively. To stabilize training, we introduce a progressive view sampling strategy inspired by (Bai et al., 2025b). Specifically, we sort target views by their distance to the input view and linearly expand the sampling window from adjacent views to the full range as training progresses.

Evaluation setup. To evaluate the performance of View4D in open-domain scenarios, we randomly select 150 video clips with 49 frames from SpatialVid (Wang et al., 2026a). The input poses, intrinsics, and depth maps are estimated using Pi3 (Wang et al., 2026b). When geometric guidance is utilized, we recover the metric scale by aligning the input depth maps with the corresponding depth maps estimated by Depth Anything 3 (Lin et al., 2026b). We denote our models without and with geometric guidance as **Ours** and **Ours w/ guid.**, respectively. We create target views by iteratively rotating and translating the input camera poses, and infer with 3 target views per sequence, resulting in 450 videos in total. We evaluate our model across two principal dimensions: cross-time temporal quality and cross-view spatial consistency. Cross-time metrics include FVD and KVD for visual fidelity, and RotErr-F/TransErr-F for single-view camera accuracy (He et al., 2025). Cross-view metrics include RotErr-V/TransErr-V for multi-view camera accuracy, MET3R (Asim et al., 2025) for multi-view consistency. To assess pixel-level alignment, since no ground truth is available, we additionally predict per-frame depths and calculate the masked reprojection quality between each pair of views, denoted as PSNR-m and SSIM-m. The pose and depth for the above metrics are evaluated using Pi3 (Wang et al., 2026b). We provide more details about the metric calculation and results evaluated with different geometry estimators in the supplementary.

Baselines. We adapt several recent state-of-the-art camera-controlled video generation methods for our multi-view synthesis task: (1) **TrajectoryCrafter** (Yu et al., 2025) redirects camera trajectories using point cloud conditioning. We apply it to generate each target view independently. (2) **Neo-**

Table 1: **Quantitative comparison with state-of-the-art methods.** The metrics are grouped into cross-view and cross-time. The best performance is in **boldface**, while the second is underlined.

Method	Cross-Time				Cross-View				
	FVD↓	KVD↓	RotErr-F↓	TransErr-F↓	RotErr-V↓	TransErr-V↓	PSNR-m↑	SSIM-m↑	MET3R ↓
ReCamMaster	271.20	0.32	3.60	1410.34	0.54	2.28	13.48	0.317	0.197
TrajectoryCrafter	226.27	**0.20**	0.66	43.13	0.09	0.31	14.49	0.317	0.162
SynCamMaster	405.09	0.53	1.49	321.55	0.78	0.93	12.93	0.308	0.106
NeoVerse	236.85	0.25	0.71	224.09	0.13	0.69	14.33	0.342	0.179
NeoVerse w/ iter.	260.97	0.28	0.88	95.52	0.20	1.74	13.21	0.285	0.151
Ours	**215.27**	0.21	**0.32**	14.50	0.11	0.08	17.67	0.440	**0.092**
Ours w/ guid.	216.22	0.21	0.36	**8.67**	**0.05**	**0.06**	**18.90**	**0.487**	0.095

Table 2: **Human user study against state-of-the-art methods.** View4D is the most preferred method for both video quality and view consistency.

Criterion	Ours	Neoverse w/ iter.	Neoverse	SynCamMaster	ReCamMaster	TrajectoryCrafter
Video Quality	**71.0%**	6.0%	5.0%	13.0%	4.0%	1.0%
View Consistency	**72.0%**	5.0%	8.0%	11.0%	4.0%	0.0%

verse (Yang et al., 2026) synthesizes novel views based on degraded renderings from 4D Gaussian Splatting. To evaluate error accumulation in multi-view scenarios, we additionally implement an iterative generation variant, denoted as **Neoverse w/ iter.**. (3) **ReCamMaster** (Bai et al., 2025a) generates videos conditioned on target camera poses. Following Wang et al. (2025a), we design the trajectory to transition to the target view in the first part of the sequence before generating the target motion. (4) **SynCamMaster** (Bai et al., 2025b) is originally designed for text-driven multi-view generation. To support monocular video inputs, we reimplement a variant following their methodology, utilizing the first viewpoint as the reference condition. We do not make comparisons with CAT4D (Wu et al., 2025), 4Real-Video (Wang et al., 2025b) and 4Real-Video-V2 (Wang et al., 2025a) since they provide neither the source code nor the pretrained models.

4.2 Comparison Results

Qualitative comparison. We present qualitative comparisons in Figure 4. Our method synthesizes high-fidelity multi-view videos that faithfully preserve the dynamic trajectory and appearance details of the input video. Furthermore, it accurately hallucinates consistent content for disoccluded regions while strictly adhering to the specified multi-view camera configurations. In contrast, previous single-view video generation methods (Bai et al., 2025a; Yu et al., 2025; Yang et al., 2026) suffer from blurry results and spatial distortions when applied to real-world videos with large camera motion, often exhibiting severe cross-view inconsistencies in disoccluded regions as highlighted in red boxes. While naive iterative extensions can produce marginally more consistent backgrounds, the inherent error accumulation inevitably leads to severe condition degradation. The model progressively loses track of the original input video, culminating in the hallucination of non-existent large-scale artifacts and severe structural distortions of the foreground subjects as highlighted in orange boxes. Although SynCamMaster (Bai et al., 2025b) can produce multi-view consistent results, it struggles to handle input videos with camera motion, exhibiting pronounced temporal inconsistencies as highlighted in green boxes. We present more visualizations in the supplementary video.

Quantitative comparison Table 1 demonstrates that our method outperforms all baselines in both cross-time and cross-view consistencies. Specifically, ReCamMaster (Bai et al., 2025a) struggles to follow the target camera trajectory when it deviates substantially from the trajectory distribution of its training data, which is evidenced by its significantly higher cross-time camera pose errors. While TrajectoryCrafter (Yu et al., 2025) and Neoverse (Yang et al., 2026) achieve better camera alignment due to explicit conditioning signals, their generated results still lack multi-view consistency. Notably, the naive iterative generation (NeoVerse w/ iter.) fails to yield noticeable improvements in cross-view metrics due to severe error accumulation. Compared to these single-view baselines, SynCamMaster (Bai et al., 2025b) maintains much better multi-view consistency. However, its temporal coherence degrades dramatically when handling input videos with camera motion, as indicated by its inferior FVD and KVD scores.



Figure 4: **Qualitative comparisons with state-of-the-art methods.** We visualize two frames (t_0, t_1) of two target views (v_1, v_2) relative to the input video v_0 shown in the left column. While previous methods suffer from structural distortions and spatio-temporal inconsistencies, our approach generates high-fidelity target views with precise spatio-temporal alignment.

Table 3: **Ablation study on the design choice of View4D.** The best performance is in **boldface**, while the second is underlined.

Method	FVD↓	KVD↓	RotErr-V ↓	TransErr-V ↓	PSNR-m ↑	SSIM-m ↑	MET3R ↓
Ours	**215.27**	**0.21**	**0.11**	**0.08**	**17.67**	**0.440**	0.092
Ours w/o MV-V	222.39	0.23	0.12	**0.08**	17.29	0.423	0.094
Ours w/o MV-I	220.35	0.22	0.15	0.12	16.71	0.407	0.093
Ours w/ Cam Enc.	230.74	0.24	0.53	0.60	13.76	0.352	0.096
Single-View	218.36	0.22	0.16	0.11	15.79	0.392	0.122
Single-View w/ iter.	230.74	0.24	0.21	0.11	15.73	0.399	**0.087**

User study To further validate the effectiveness of View4D against the baselines, we conduct a human study assessing two dimensions: video quality and cross-view consistency. Specifically, we invite 10 participants to indicate their preferred method along each dimension across 10 videos randomly sampled from the evaluation set. As illustrated in Table 2, the majority of participants rated our results as the best in both video quality (over 71.0%) and cross-view consistency (over 72.0%), highlighting the clear advantage of View4D over the baselines.

4.3 Ablation and Analysis

We validate our framework design choices across the following aspects. For all variants in Table 3, the geometric guidance is dropped to ensure a fair comparison. The discussion of single-view variants and more detailed analysis of performance gains, camera pose conditioning and geometric guidance can be found in the supplementary.



Figure 5: **More multi-view videos generation results.** Our method supports generating videos under diverse multi-view camera configurations.

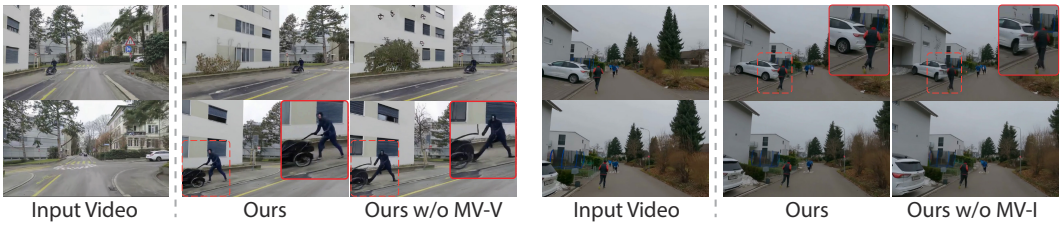

Figure 6: **Ablation study on dataset augmentation.** Training with real multi-view videos enhances the ability to generate realistic dynamic content and training with real multi-view images helps the model better preserve fine-grained geometric coherence.

Camera pose conditioning. We compare our relative positional encoding against the learnable camera encoder utilized in previous works (Bai et al., 2025a;b) (Ours w/ Cam Enc.). As indicated by the noticeable degradation in cross-view metrics in Table 3, the conventional camera encoder inherently struggles with multi-view consistency. We further provide more results in Figure 5. Thanks to the camera conditioning with relative positional encoding, our method can be applied to diverse multi-view camera configurations in open world video even **without** explicit geometric guidance, unlocking new potentials for a wide range of downstream applications.

Effect of geometric guidance. We compare Ours (without guidance) and Ours w/ guid. in Table 1. Injecting explicit geometric guidance improves multi-view consistency; the slight FVD/KVD increase reflects an expected trade-off of generative freedom for spatial precision (PSNR-m, SSIM-m). As illustrated in Figure 4, both variants generate consistent multi-view videos. When the explicit condition is available, our model anchors spatial locations using the injected guidance while maintaining cross-view consistency via the multi-view attention module.

Dataset augmentation. We study the effect of our joint training strategy by dropping individual real-world datasets. As shown in Figure 6, omitting real-world multi-view videos (Sun et al., 2020) (Ours w/o MV-V) impairs the generation of dynamic content, resulting in artifacts and unnatural motions. Meanwhile, removing real-world multi-view images (Ling et al., 2024) (Ours w/o MV-I) weakens the geometric coherence across synthesized views, resulting in prominent spatial artifacts and a corresponding drop in multi-view pose accuracy as shown in Table 3.

5 Conclusion

This paper introduced View4D, a novel generative model for synthesizing synchronized multi-view videos with moving cameras in the open world. Our approach is built upon implicit projection-matrix conditioning, coupled with optional explicit geometric guidance. Empowered by a hybrid training strategy that fuses synthetic geometric priors with real-world dynamics, View4D achieves unprecedented cross-view consistency and high-quality generation. Extensive experiments validate its effectiveness, unlocking new possibilities for various real-world applications. We hope that View4D inspires further exploration of generative world models that go beyond single-camera observation, toward a more complete spatiotemporal understanding of dynamic environments.

AI USE STATEMENT

We used generative AI tools to assist with LaTeX formatting, literature searches, and the verification of bibliographic information. The authors reviewed all AI-assisted changes and checked the final bibliographic information against the original sources. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work does not involve human subjects, sensitive personal data, or other issues covered by the ICLR Code of Ethics. Therefore, we do not anticipate any ethical concerns arising from this research.

REPRODUCIBILITY STATEMENT

To support reproducibility, we describe the main experimental settings and configurations required to reproduce our results in the main paper. We provide additional details regarding our model architecture and training dataset in the appendix.

REFERENCES

Mohammad Asim, Christopher Wewer, Thomas Wimmer, Bernt Schiele, and Jan Eric Lenssen. Met3r: Measuring multi-view consistency in generated images. In *CVPR*, 2025.

Sherwin Bahmani, Ivan Skorokhodov, Guocheng Qian, Aliaksandr Siarohin, Willi Menapace, Andrea Tagliasacchi, David B Lindell, and Sergey Tulyakov. Ac3d: Analyzing and improving 3d camera control in video diffusion transformers. In *CVPR*, 2025a.

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. In *ICLR*, 2025b.

Sherwin Bahmani, Tianchang Shen, Jiawei Ren, Jiahui Huang, Yifeng Jiang, Haithem Turki, Andrea Tagliasacchi, David B. Lindell, Zan Gojcic, Sanja Fidler, Huan Ling, Jun Gao, and Xuanchi Ren. Lyra: Generative 3d scene reconstruction via video diffusion model self-distillation. In *ICLR*, 2026.

Jianhong Bai, Menghan Xia, Xiao Fu, Xintao Wang, Lianrui Mu, Jinwen Cao, Zuozhu Liu, Haoji Hu, Xiang Bai, Pengfei Wan, et al. Recammaster: Camera-controlled generative rendering from a single video. In *ICCV*, 2025a.

Jianhong Bai, Menghan Xia, Xintao Wang, Ziyang Yuan, Zuozhu Liu, Haoji Hu, Pengfei Wan, and Di Zhang. Syncammaster: Synchronizing multi-camera video generation from diverse viewpoints. In *ICLR*, 2025b.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report, 2025c. URL https://arxiv.org/abs/2502.13923.

Philip J. Ball, Jakob Bauer, Frank Belletti, Bethanie Brownfield, Ariel Ephrat, Shlomi Fruchter, Agrim Gupta, Kristian Holsheimer, Aleksander Holynski, Jiri Hron, Christos Kaplanis, Marjorie Limont, Matt McGill, Yanko Oliveira, Jack Parker-Holder, Frank Perbet, Guy Scully, Jeremy Shar, Stephen Spencer, Omer Tov, Ruben Villegas, Emma Wang, Jessica Yung, Cip Baetu, Jordi Berbel, David Bridson, Jake Bruce, Gavin Buttimore, Sarah Chakera, Bilva Chandra, Paul Collins, Alex Cullum, Bogdan Damoc, Vibha Dasagi, Maxime Gazeau, Charles Gbadamosi, Woohyun Han, Ed Hirst, Ashyana Kachra, Lucie Kerley, Kristian Kjems, Eva Knoepfel, Vika Koriakin, Jessica Lo, Cong Lu, Zeb Mehring, Alex Moufarek, Henna Nandwani, Valeria Oliveira, Fabio Pardo, Jane Park, Andrew Pierson, Ben Poole, Helen Ran, Tim Salimans, Manuel Sanchez, Igor Saprykin, Amy Shen, Sailesh Sidhwani, Duncan Smith, Joe Stanton, Hamish Tomlinson, Dimple Vijaykumar, Luyu Wang, Piers Wingfield, Nat Wong, Keyang Xu, Christopher Yew, Nick Young, Vadim Zubov,

Douglas Eck, Dumitru Erhan, Koray Kavukcuoglu, Demis Hassabis, Zoubin Gharamani, Raia Hadsell, Aäron van den Oord, Inbar Mosseri, Adrian Bolton, Satinder Singh, and Tim Rocktäschel. Genie 3: A new frontier for world models. 2025.

Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, Clarence Ng, Ricky Wang, and Aditya Ramesh. Video generation models as world simulators. 2024. URL https://openai.com/research/video-generation-models-as-world-simulators.

Chenjie Cao, Jingkai Zhou, Shikai Li, Jingyun Liang, Chaohui Yu, Fan Wang, Xiangyang Xue, and Yanwei Fu. Uni3c: Unifying precisely 3d-enhanced camera and human motion controls for video generation. In *SIGGRAPH Asia*, 2025.

Zhongxiao Cong, Qitao Zhao, Minsik Jeon, and Shubham Tulsiani. Flow3r: Factored flow prediction for scalable visual geometry learning. In *CVPR*, 2026.

Epic Games. Unreal engine. https://www.unrealengine.com/, 2025.

Xiao Fu, Shitao Tang, Min Shi, Xian Liu, Jinwei Gu, Ming-Yu Liu, Dahua Lin, and Chen-Hsuan Lin. Plenoptic video generation. In *CVPR*, 2026.

Ruiyuan Gao, Kai Chen, Enze Xie, Lanqing Hong, Zhenguo Li, Dit-Yan Yeung, and Qiang Xu. MagicDrive: Street view generation with diverse 3d geometry control. In *ICLR*, 2024.

Google DeepMind. Veo 3. https://deepmind.google/models/veo/, 2025.

Zekai Gu, Rui Yan, Jiahao Lu, Peng Li, Zhiyang Dou, Chenyang Si, Zhen Dong, Qifeng Liu, Cheng Lin, Ziwei Liu, et al. Diffusion as shader: 3d-aware video diffusion for versatile video generation control. In *SIGGRAPH*, 2025.

Hao He, Yinghao Xu, Yuwei Guo, Gordon Wetzstein, Bo Dai, Hongsheng Li, and Ceyuan Yang. Cameractrl: Enabling camera control for video diffusion models. In *ICLR*, 2025.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

Anthony Hu, Lloyd Russell, Hudson Yeo, Zak Murez, George Fedoseev, Alex Kendall, Jamie Shotton, and Gianluca Corrado. Gaia-1: A generative world model for autonomous driving. *arXiv preprint arXiv:2309.17080*, 2023.

Tao Hu, Haoyang Peng, Xiao Liu, and Yuewen Ma. Ex-4d: Extreme viewpoint 4d video synthesis via depth watertight mesh, 2025. URL https://arxiv.org/abs/2506.05554.

Tianyu Huang, Wangguandong Zheng, Tengfei Wang, Yuhao Liu, Zhenwei Wang, Junta Wu, Jie Jiang, Hui Li, Rynson Lau, Wangmeng Zuo, et al. Voyager: Long-range and world-consistent video diffusion for explorable 3d scene generation. *ACM Transactions on Graphics (TOG)*, 44(6): 1–15, 2025.

Zhening Huang, Hyeonho Jeong, Xuelin Chen, Yulia Gryaditskaya, Tuanfeng Y. Wang, Joan Lasenby, and Chun-Hao Huang. Spacetimepilot: Generative rendering of dynamic scenes across space and time. In *CVPR*, 2026.

Sam Ade Jacobs, Masahiro Tanaka, Chengming Zhang, Minjia Zhang, Shuaiwen Leon Song, Samyam Rajbhandari, and Yuxiong He. Deepspeed ulysses: System optimizations for enabling training of extreme long sequence transformer models, 2023. URL https://arxiv.org/abs/2309.14509.

Yanqin Jiang, Chaohui Yu, Chenjie Cao, Fan Wang, Weiming Hu, and Jin Gao. Animate3d: Animating any 3d model with multi-view video diffusion. In *NeurIPS*, 2024.

Haian Jin, Rundi Wu, Tianyuan Zhang, Ruiqi Gao, Jonathan T. Barron, Noah Snavely, and Aleksander Holynski. ZipMap: Linear-time stateful 3d reconstruction via test-time training. In *CVPR*, 2026.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics (ToG)*, 42(4):1–14, 2023.

Zhengfei Kuang, Shengqu Cai, Hao He, Yinghao Xu, Hongsheng Li, Leonidas Guibas, and Gordon Wetzstein. Collaborative video diffusion: Consistent multi-video generation with camera control. In *NeurIPS*, 2024.

Jiahui Lei, Yijia Weng, Adam W Harley, Leonidas Guibas, and Kostas Daniilidis. Mosca: Dynamic gaussian fusion from casual videos via 4d motion scaffolds. In *CVPR*, 2025.

Bing Li, Cheng Zheng, Wenxuan Zhu, Jinjie Mai, Biao Zhang, Peter Wonka, and Bernard Ghanem. Vivid-zoo: Multi-view video generation with diffusion model. In *NeurIPS*, 2024.

Ruilong Li, Brent Yi, Junchen Liu, Hang Gao, Yi Ma, and Angjoo Kanazawa. Cameras as relative positional encoding. In *NeurIPS*, 2025.

Chenguo Lin, Yuchen Lin, Panwang Pan, Yifan Yu, Tao Hu, Honglei Yan, Katerina Fragkiadaki, and Yadong Mu. Movies: Motion-aware 4d dynamic view synthesis in one second. In *CVPR*, 2026a.

Haotong Lin, Sili Chen, Jun Hao Liew, Donny Y. Chen, Zhenyu Li, Yang Zhao, Sida Peng, Hengkai Guo, Xiaowei Zhou, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth anything 3: Recovering the visual space from any views. In *ICLR*, 2026b.

Kuan Heng Lin, Zhizheng Liu, Pablo Salamanca, Yash Kant, Ryan Burgert, Yuancheng Xu, Koichi Namekata, Yiwei Zhao, Bolei Zhou, Micah Goldblum, Paul Debevec, and Ning Yu. Vista4D: Video reshooting with 4d point clouds. In *CVPR*, 2026c.

Lu Ling, Yichen Sheng, Zhi Tu, Wentian Zhao, Cheng Xin, Kun Wan, Lantao Yu, Qianyu Guo, Zixun Yu, Yawen Lu, et al. Dl3dv-10k: A large-scale scene dataset for deep learning-based 3d vision. In *CVPR*, 2024.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019.

Dongyue Lu, Ao Liang, Tianxin Huang, Xiao Fu, Yuyang Zhao, Baorui Ma, Liang Pan, Wei Yin, Lingdong Kong, Wei Tsang Ooi, et al. See4d: Pose-free 4d generation via auto-regressive video inpainting. In *Computer Graphics Forum*, pp. e70345. Wiley Online Library, 2026.

Yawen Luo, Xiaoyu Shi, Jianhong Bai, Menghan Xia, Tianfan Xue, Xintao Wang, Pengfei Wan, Di Zhang, and Kun Gai. Camclonemaster: Enabling reference-based camera control for video generation. In *SIGGRAPH Asia*, 2025.

Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *ICCV*, 2023.

Xuanchi Ren, Yifan Lu, Tianshi Cao, Ruiyuan Gao, Shengyu Huang, Amirmojtaba Sabour, Tianchang Shen, Tobias Pfaff, Jay Zhangjie Wu, Runjian Chen, Seung Wook Kim, Jun Gao, Laura Leal-Taixe, Mike Chen, Sanja Fidler, and Huan Ling. Cosmos-drive-dreams: Scalable synthetic driving data generation with world foundation models, 2025a. URL https://arxiv.org/abs/2506.09042.

Xuanchi Ren, Tianchang Shen, Jiahui Huang, Huan Ling, Yifan Lu, Merlin Nimier-David, Thomas Müller, Alexander Keller, Sanja Fidler, and Jun Gao. Gen3c: 3d-informed world-consistent video generation with precise camera control. In *CVPR*, 2025b.

Johannes L. Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, 2016.

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *arXiv preprint arXiv:2104.09864*, 2021.

Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *CVPR*, 2020.

Basile Van Hoorick, Rundi Wu, Ege Ozguroglu, Kyle Sargent, Ruoshi Liu, Pavel Tokmakov, Achal Dave, Changxi Zheng, and Carl Vondrick. Generative camera dolly: Extreme monocular dynamic novel view synthesis. In *ECCV*, 2024.

Basile Van Hoorick, Dian Chen, Shun Iwase, Pavel Tokmakov, Muhammad Zubair Irshad, Igor Vasiljevic, Swati Gupta, Fangzhou Cheng, Sergey Zakharov, and Vitor Campagnolo Guizilini. Anyview: Synthesizing any novel view in dynamic scenes. *arXiv preprint arXiv:2601.16982*, 2026.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, Jianyuan Zeng, Jiayu Wang, Jingfeng Zhang, Jingren Zhou, Jinkai Wang, Jixuan Chen, Kai Zhu, Kang Zhao, Keyu Yan, Lianghua Huang, Mengyang Feng, Ningyi Zhang, Pandeng Li, Pingyu Wu, Ruihang Chu, Ruili Feng, Shiwei Zhang, Siyang Sun, Tao Fang, Tianxing Wang, Tianyi Gui, Tingyu Weng, Tong Shen, Wei Lin, Wei Wang, Wei Wang, Wenmeng Zhou, Wente Wang, Wenting Shen, Wenyuan Yu, Xianzhong Shi, Xiaoming Huang, Xin Xu, Yan Kou, Yangyu Lv, Yifei Li, Yijing Liu, Yiming Wang, Yingya Zhang, Yitong Huang, Yong Li, You Wu, Yu Liu, Yulin Pan, Yun Zheng, Yuntao Hong, Yupeng Shi, Yutong Feng, Zeyinzi Jiang, Zhen Han, Zhi-Fan Wu, and Ziyu Liu. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

Chaoyang Wang, Ashkan Mirzaei, Vidit Goel, Willi Menapace, Aliaksandr Siarohin, Michael Vasilkovsky, Ivan Skorokhodov, Vladislav Shakhrai, Sergei Korolev, Sergey Tulyakov, and Peter Wonka. Fused view-time attention and feedforward reconstruction for 4d scene generation. In *NeurIPS*, 2025a.

Chaoyang Wang, Peiye Zhuang, Tuan Duc Ngo, Willi Menapace, Aliaksandr Siarohin, Michael Vasilkovsky, Ivan Skorokhodov, Sergey Tulyakov, Peter Wonka, and Hsin-Ying Lee. 4real-video: Learning generalizable photo-realistic 4d video diffusion. In *CVPR*, 2025b.

Jiahao Wang, Yufeng Yuan, Rujie Zheng, Youtian Lin, Jian Gao, Lin-Zhuo Chen, Yajie Bao, Chang Zeng, Yanxi Zhou, Xiao-Xiao Long, et al. Spatialvid: A large-scale video dataset with spatial annotations. In *CVPR*, 2026a.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *CVPR*, 2025c.

Qianqian Wang, Vickie Ye, Hang Gao, Weijia Zeng, Jake Austin, Zhengqi Li, and Angjoo Kanazawa. Shape of motion: 4d reconstruction from a single video. In *ICCV*, 2025d.

Xiaofeng Wang, Zheng Zhu, Guan Huang, Xinze Chen, Jiagang Zhu, and Jiwen Lu. Drivedreamer: Towards real-world-drive world models for autonomous driving. In *ECCV*, 2024a.

Yifan Wang, Jianjun Zhou, Haoyi Zhu, Wenzheng Chang, Yang Zhou, Zizun Li, Junyi Chen, Jiangmiao Pang, Chunhua Shen, and Tong He. π^3: Permutation-equivariant visual geometry learning. In *ICLR*, 2026b.

Yiming Wang, Qihang Zhang, Shengqu Cai, Tong Wu, Jan Ackermann, Zhengfei Kuang, Yang Zheng, Frano Rajič, Siyu Tang, and Gordon Wetzstein. Bullettime: Decoupled control of time and camera pose for video generation. In *CVPR*, 2026c.

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*, 2024b.

Yuqing Wen, Yucheng Zhao, Yingfei Liu, Fan Jia, Yanhui Wang, Chong Luo, Chi Zhang, Tiancai Wang, Xiaoyan Sun, and Xiangyu Zhang. Panacea: Panoramic and controllable video generation for autonomous driving. In *CVPR*, 2024.

Rundi Wu, Ruiqi Gao, Ben Poole, Alex Trevithick, Changxi Zheng, Jonathan T Barron, and Aleksander Holynski. Cat4d: Create anything in 4d with multi-view video diffusion models. In *CVPR*, 2025.

Zhen Xu, Zhengqin Li, Zhao Dong, Xiaowei Zhou, Richard Newcombe, and Zhaoyang Lv. 4dgt: Learning a 4d gaussian transformer using real-world monocular videos. In *NeurIPS*, 2025.

Yuxue Yang, Lue Fan, Ziqi Shi, Junran Peng, Feng Wang, and Zhaoxiang Zhang. Neoverse: Enhancing 4d world model with in-the-wild monocular videos. In *CVPR*, 2026.

Zeyu Yang, Zijie Pan, Chun Gu, and Li Zhang. Diffusion²: Dynamic 3d content generation via score composition of video and multi-view diffusion models. In *ICLR*, 2025.

Chun-Han Yao, Yiming Xie, Vikram Voleti, Huaizu Jiang, and Varun Jampani. Sv4d 2.0: Enhancing spatio-temporal consistency in multi-view video diffusion for high-quality 4d generation. In *ICCV*, 2025.

Mark Yu, Wenbo Hu, Jinbo Xing, and Ying Shan. Trajectorycrafter: Redirecting camera trajectory for monocular videos via diffusion models. In *ICCV*, 2025.

Wangbo Yu, Jinbo Xing, Li Yuan, Wenbo Hu, Xiaoyu Li, Zhipeng Huang, Xiangjun Gao, Tien-Tsin Wong, Ying Shan, and Yonghong Tian. Viewcrafter: Taming video diffusion models for high-fidelity novel view synthesis. *arXiv preprint arXiv:2409.02048*, 2024.

Haiyu Zhang, Xinyuan Chen, Yaohui Wang, Xihui Liu, Yunhong Wang, and Yu Qiao. 4diffusion: Multi-view video diffusion model for 4d generation. In *NeurIPS*, 2024.

Junyi Zhang, Charles Herrmann, Junhwa Hur, Varun Jampani, Trevor Darrell, Forrester Cole, Deqing Sun, and Ming-Hsuan Yang. Monst3r: A simple approach for estimating geometry in the presence of motion. In *ICLR*, 2025.

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

A More Implementation Details

A.1 Block-diagonal Camera Pose Transform

In this section, we provide implementation details for the camera-conditioning scheme introduced in Section 3.2. Let d_l and d_t denote the dimensionalities of the spatial and temporal channel subspaces, respectively, within each multi-view attention head. The transformation matrix $\mathbf{D}_{j,i}$ associated with token group (j, i) and projection matrix $\tilde{T}_{j,i}$ is defined as:

$$\mathbf{D}_{j,i} = \begin{bmatrix} \mathbf{D}_{j,i}^{\mathrm{RoPE}} & \mathbf{0} \\ \mathbf{0} & \mathbf{D}_{j,i}^{\mathrm{Proj}} \end{bmatrix}, \tag{6}$$

$$\mathbf{D}_{j,i}^{\mathrm{Proj}} = \mathbf{I}_{d_t/4} \otimes \tilde{T}_{j,i} \in \mathbb{R}^{d_t \times d_t}, \tag{7}$$

where $\otimes$ denotes the Kronecker product, and $\mathbf{D}_{j,i}^{\mathrm{RoPE}} \in \mathbb{R}^{d_l \times d_l}$ denotes the 2D RoPE transformation applied to the spatial channel subspace. In practice, we initialize the multi-view attention module from the pretrained spatio-temporal attention module (Wan et al., 2025). We retain its spatial RoPE components and replace the temporal RoPE component with the camera projection transformation.

A.2 Geometric Construction

We provide additional details on how the explicit geometric guidance $(\hat{V}_j, M_j)$ introduced in Section 3.3 is constructed for each target view j given the estimated depth $D_{0,i}$ with confidence $C_{0,i}$.

Before performing depth warping, we propose two strategies to filter out unreliable regions of the depth map. First, we employ confidence filtering to discard unreliable depth estimates in low-texture or out-of-distribution regions, retaining only pixels with $C_{0,i}(p) > \tau$. Second, we introduce a mesh-based occlusion masking strategy inspired by Lyra (Bahmani et al., 2026). Specifically, we build a triangle mesh on the image grid, with each patch contributing two triangles. For each target view, we ray-trace from the target camera through this mesh to determine the closest visible surface at each target pixel. Splatted points whose depth d_p lies behind the traced mesh depth d_m by more than a tolerance d are masked out. We set $\tau = 0.2$ and $d = 0.02$ meters in all experiments.

After the filtering process, the depth is forward-warped to each target view j via bilinear point splatting, using the relative camera transformation $T_{0\to j}$ and the recovered intrinsic K. This produces a warped RGB frame $\hat{V}_{j,i}$ together with a binary mask $M_{j,i}$ that combines the aforementioned confidence and occlusion filters.

A.3 Guidance Encoder

The guidance encoder contains four 3D convolutional layers. The input has 16 channels, the three intermediate layers have a hidden dimension of 1024, and the output has 5120 channels to match the backbone's feature dimension. The first three layers use a 1×1×1 kernel, while the last layer uses a 1×2×2 kernel to downsample the guidance to the same spatial resolution as the video latents.

A.4 Training and Inference

We jointly train our model on three distinct multi-view data sources: our rendered multi-view videos, real multi-view images (Ling et al., 2024), and real multi-view videos (Sun et al., 2020), with sampling weights of 1.0, 0.2, and 0.2, respectively. The number of target views n is set to 3, 7, and 2, respectively. Across all datasets, the spatial resolution is set to 480×832, and the text prompt for each clip is obtained using Qwen2.5-VL (Bai et al., 2025c). During training, we adopt the Fully Sharded Data Parallel (FSDP) strategy for the DiT and text encoder to reduce memory cost. The model is trained for 20,000 steps on 32 Hopper-architecture GPUs with 80 GB of memory.

During inference, we set the number of inference steps to 50 and the classifier-free guidance (Ho & Salimans, 2022) scale to 5.0. In fact, We adopt Ulysses-style sequence parallelism (Jacobs et al., 2023) along the attention-head dimension across multiple GPUs to accelerate inference. Since we keep the pretrained spatio-temporal attention frozen, View4D can generate results with an arbitrary number

of frames supported by the base model. Thanks to the multi-view attention with the relative camera projection matrix as condition, View4D can generate multi-view videos with **various numbers of target views**. We provide deeper analysis on cross-view accuracy and computation cost in Section D. More visual results with different number of views can be found in the supplementary video.

B TRAINING DATASET

We provide details about the composition and construction of our rendered multi-view videos below. Visual example of the dataset is shown in Figure 7 for better understanding.

Composition. We characterize the composition of the training data from three aspects:

1. 3D Environment. The scenes cover diverse indoor and outdoor settings, including urban and residential areas, highways, industrial zones, parks, beaches, rooms, corridors, stairways, and garages. They range from open outdoor areas to confined indoor spaces and contain large depth variations, occlusions, reflective surfaces, and indoor–outdoor transitions. They also vary in illumination, weather conditions, object density, and environmental content such as buildings, vegetation, furniture, and street facilities.
2. Character. The scenes contain human characters with varied appearances, clothing, and spatial distributions. Depending on the scene and viewpoint, characters may appear as primary foreground subjects or background actors.
3. Animation. The scenes contain various character motion patterns, including idle behavior, walking, running, and turning.

Construction. The training data are rendered and collected using Unreal Engine (UE). For each sequence, the multi-view cameras are uniformly placed along a horizontal circle and oriented either inward (third-person viewpoint) or outward (first-person viewpoint), with the radius randomly sampled from 2.00 to 9.00 m or from 0.50 to 1.00 m, respectively, and the field of view sampled from 50° to 100°. Over time, the multi-view cameras move together in the horizontal plane, following trajectories with varying speeds, turns, pauses, and direction changes. For the raw data, we capture 25 views uniformly covering a full 360°.

During rendering, temporal anti-aliasing (TAA) effects are disabled to avoid temporal smearing and ghosting artifacts, and RGB frames are streamed directly to FFmpeg for video encoding, with the corresponding camera parameters recorded synchronously for each frame. For training of View4D, we select 9 views spanning a 120° range, with the middle view chosen as the input view and the remaining eight views serving as candidate target views. In total, the constructed dataset contains 3,410 source sequences and 64,787 training clips. Each source sequence consists of 500 frames captured at 25 fps with a resolution of 720 × 1280, and each training clip contains 25 frames.

C EVALUATION DETAILS

We evaluate View4D at the same resolution used during training. Details of metrics, baselines and the evaluation setup, are provided below.

C.1 CAMERA POSE ERROR

Given a set of ground-truth camera poses $\{\mathbf{R}^k_{gt}, \mathbf{T}^k_{gt}\}_{k=1}^K$ and predicted poses $\{\mathbf{R}^k_{pred}, \mathbf{T}^k_{pred}\}_{k=1}^K$, the rotation and translation errors are calculated as:

$$\text{RotErr} = \sum_{k=1}^{K} \arccos \frac{\text{tr}\big(\mathbf{R}^k_{gt}\mathbf{R}^{k\top}_{pred}\big) - 1}{2}, \quad \text{TransErr} = \sum_{k=1}^{K} \big\|\mathbf{T}^k_{gt} - \mathbf{T}^k_{pred}\big\|_2^2, \tag{8}$$

where $\text{tr}(\cdot)$ denotes the matrix trace. We report the rotation error in radians. In our experiments, we first align the translation scale of cameras estimated by the pose estimator with the input cameras before computing the translation error.

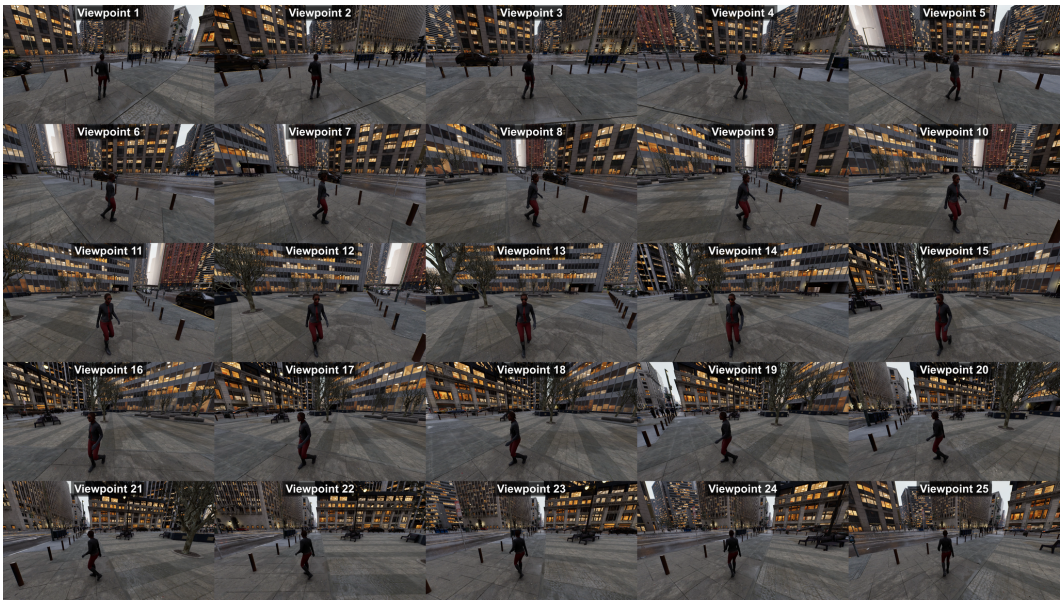

Figure 7: **Visual example of our training data.** We visualize synchronized multi-view observations at a single time step from one training sequence.



Figure 8: **Visualization of the depth-based warping process for pixel-level evaluation.** From left to right: (a) the source view, (b) the target view, (c) the forward-warped image from the source to the target camera via bilinear point splatting, (d) the binary validity mask combining confidence and mesh-based occlusion filtering, and (e) the absolute error map.

For **cross-time** accuracy RotErr-F/TransErr-F, we run Pi3 (Wang et al., 2026b) on each of the $n{+}1$ generated single-view videos to recover its per-frame trajectory ($K{=}F$). For **cross-view** accuracy RotErr-V/TransErr-V, we feed the $n{+}1$ views at each frame jointly to Pi3 and recover the per-frame multi-view camera configuration ($K{=}n{+}1$).

C.2 Pixel-level Fidelity

For each evaluation frame, we jointly run Pi3 (Wang et al., 2026b) on the $n{+}1$ generated views to obtain the depth and confidence per-pixel. We employ the same depth-based warping algorithm used for our geometric guidance construction. Specifically, for each pair (j_1, j_2), we forward-warp the view j_1 into the camera of view j_2 using bilinear point splatting, applying the same confidence and mesh-based occlusion filters as described in Section A. PSNR and SSIM are then computed between the warped frame and the target view within the masked region. We show some examples with error map visualization in Figure 8.

C.3 Baselines

For TrajectoryCrafter (Yu et al., 2025) and Neoverse (Yang et al., 2026), we align their estimated explicit point clouds or Gaussian primitives to the scale of the input camera poses. Since Neoverse generates novel views by completing degraded renderings from 4D Gaussian Splatting with a video diffusion model, it can be viewed as an enhanced version of 4D reconstruction methods.



Figure 9: **Ablation study on camera pose conditioning.** We shift the camera trajectory to the left to evaluate the ability for accurate spatial control. The camera encoder fails to follow the control signal and generates results similar to the input video. In contrast, our approach effectively anchors the extreme viewpoint shift, generating geometrically accurate results.

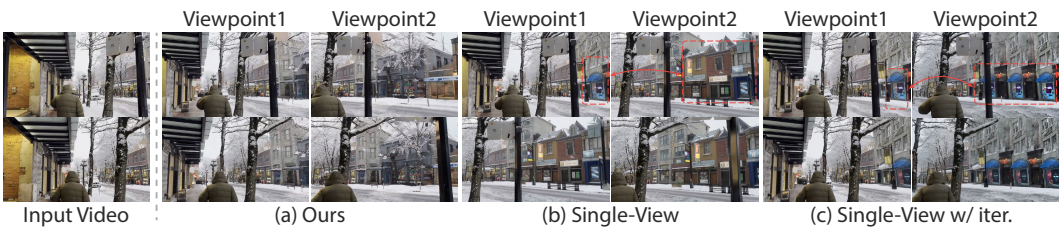

Figure 10: **Ablation study on multi-view joint prediction.** Independent single-view generation results in multi-view inconsistency. The iterative version merely replicates existing patterns in unseen regions, failing to synthesize plausible novel structures due to the absence of cross-view interaction.

As demonstrated in their paper, the 4D Gaussians recovered by Neoverse already surpass current state-of-the-art reconstruction approaches (Xu et al., 2025; Zhang et al., 2025). We therefore treat the comparison with Neoverse as an upper-bound proxy for comparisons against the broader family of 4D reconstruction methods.

For ReCamMaster (Bai et al., 2025a), which only supports novel-view video generation with a shared start frame and a fixed length of 81 frames, we adapt its inference as follows to obtain reasonable results for each target view j following Wang et al. (2025a). We first construct a smooth trajectory from $T_{0,0}$ to $T_{j,0}$ over $81-F$ frames, followed by the target camera trajectory $\{T_{j,i}\}_{i=1}^{F}$. Accordingly, the first frame of the source video is repeated $81-F$ times to match the required video length. Finally, we extract the last F frames as the result.

For SynCamMaster (Bai et al., 2025b), we reimplement a reference-video-conditioned variant following the method described in their paper, using the same base model as View4D. The model architecture and training dataset follow its official implementation.

Another related work, Plenoptic Video Generation (Fu et al., 2026), employs a 3D FOV-based video retrieval mechanism and generates each view autoregressively. Since its open-source code lacks several critical modules and the corresponding checkpoint, we are currently unable to compare against it directly. [1] Nevertheless, the Single-View w/ iter. variant in our ablations can be regarded as a method analogous to Plenoptic Video Generation, where the context video size k is set to 1. As shown in Table 3, jointly predicting multiple views better preserves cross-view consistency.

C.4 Multi-view configuration

Evaluating cross-view structural alignment requires target views that offer both substantial spatial baselines and sufficient field-of-view overlap. We satisfy these requirements by configuring multiple target views through iterative rotation and translation of the input camera poses. We derive target views by rotating around the world's up-axis, where the sign of the rotation radius explicitly dictates an inward-facing or outward-facing layout. Specifically, the inter-view rotation angle is randomly sampled from $[15°, 20°]$ and the rotation radius from $[-1, 3]$ meters. This orbit-style configuration ensures that adjacent views share enough overlap for pixel-level alignment evaluation while exposing hallucinated regions that are mutually visible across views, enabling direct measurement of cross-view hallucination consistency. Additionally, View4D natively supports pure translational movements without rotation, as illustrated in Figure 5.

[1] Related reproducibility issues have also been reported in Issue #191 and Issue #212 of the official repository.



Figure 11: **Analysis on explicit geometric guidance.** From left to right: (a) generation using only implicit conditioning (Ours); (b) unaligned guidance with its corresponding outputs (Ours w/ guid. (no align)); and (c) aligned guidance with final outputs (Ours w/ guid.). In (b) and (c), the left sub-columns show the warped frames, and the right show the generated views.

Table 4: **Ablation study on the source of performance gain.** The best performance is in **boldface**, while the second best is underlined.

Method	FVD ↓	KVD ↓	RotErr-V ↓	TransErr-V ↓	PSNR-m ↑	SSIM-m ↑	MET3R ↓
Ours w/o data	381.81	0.51	0.35	0.53	14.76	0.355	0.118
Ours w/o arch	230.74	0.24	0.53	0.60	13.76	0.352	0.096
Ours	**215.27**	**0.21**	**0.11**	**0.08**	**17.67**	**0.440**	**0.092**

D Additional Ablations and Analysis

Analysis on performance gain. We analysis the sources of the performance gain from two perspectives: the custom rendered multi-view data and the proposed architecture. To be specific, we implemented two variants:

- Ours w/o data: We replace our constructed dataset with the multi-view video dataset with static cameras (Bai et al., 2025b), isolating the effect of the additional training data.
- Ours w/o arch: We replace our camera-conditioning scheme with the learnable camera encoder used in prior works (Bai et al., 2025b;a), isolating the effect of our multi-view attention with camera conditioning. This experiment is identical to Ours w/ Cam Enc. in Table 3.

As shown in Table 3, both the constructed dataset and the proposed architecture are essential to our method. The constructed dataset improves video realism and overall consistency, whereas the proposed architecture plays a critical role in cross-view geometric alignment and fine-grained appearance preservation.

Analysis on camera pose conditioning. To further investigate the effectiveness of our camera conditioning scheme, we explore the results in out-of-distribution (OOD) test cases, as illustrated in Figure 9. The conventional encoder tends to overfit the training camera configuration, failing to learn the complex mapping from raw camera poses to high-dimensional video tokens. In contrast, our conditioning scheme directly leverages relative camera poses to modulate the attention weights. This inherently geometric approach prevents overfitting, ensuring accurate spatial transformations and robust generalization under unseen test scenarios.

Analysis on multi-view joint prediction. To study the effect of multi-view prediction on cross-view consistency, we train a variant that predicts only one target view at a time (Single-View). As illustrated in Figure 10(b), the generated multi-view videos lack geometric coherence, which is quantitatively reflected by the low cross-view scores in Table 3. Additionally, we implement an iterative version (Single-View w/ iter.). We observe that this variant tends to hallucinate repeated contents in unseen regions, as shown in Figure 10(c). While this leads to better semantic consistency, as evidenced by the MET3R score (which calculates the similarity of DINO features), the method fails to accurately capture the cross-view camera layout, resulting in even poorer camera pose accuracy.

Table 5: **Ablation study on the number of generated views.** We evaluate the camera pose error across different numbers of target views n under fixed multi-view range.

n	2	3	4	5
RotErr-V↓	0.08	0.11	0.13	0.15
TransErr-V↓	0.09	0.08	0.08	0.10

Table 6: **Ablation study on the choice of estimator.** The metrics below are calculated using Depth Anything 3 (Lin et al., 2026b).

Method	RotErr-F ↓	TransErr-F ↓	RotErr-V ↓	TransErr-V ↓	PSNR-m ↑	SSIM-m ↑
ReCamMaster (Bai et al., 2025a)	3.52	1805.38	0.62	3.60	12.41	0.397
TrajectoryCrafter (Yu et al., 2025)	0.62	74.10	0.23	0.87	13.42	0.393
SynCamMaster (Bai et al., 2025b)	1.56	323.70	0.78	0.93	12.39	0.370
NeoVerse (Yang et al., 2026)	0.68	367.49	0.26	1.73	12.94	0.383
NeoVerse w/ iter. (Yang et al., 2026)	0.82	149.35	0.32	2.16	12.42	0.366
Ours	**0.43**	48.73	0.25	0.29	15.77	0.473
Ours w/ guid.	0.45	**43.69**	**0.20**	**0.26**	**16.12**	**0.483**

Analysis on robustness to noisy geometric guidance. We add more analysis about the explicit geometric guidance presented in Section 3. As shown in Figure 11(c), when the scale of the depth is aligned with the camera pose (Ours w/ guid.), our model achors each target view with the warped frame and maintains consistency with the multi-view attention module. We show the results of directly applying the depth estimated by Pi3 (Wang et al., 2026b) as guidance without scaling (Ours w/ guid. (no align)). In this situation our model learns to ignore the incorrect conditioning signal and fall back to the case with implicit camera pose conditioning (Ours) as shown in Figure 11(b). We attribute this to the inherent noise in the depth estimation and alignment process during training data construction. By observing these imperfect condition-target pairs during training, the network naturally learns when to discard them during inference.

Analysis on the number of target views. Thanks to the multi-view attention module with the projection matrix as relative positional encoding, View4D can generate different numbers of target views during inference. As shown in Table 5, we hold the total multi-view range fixed and vary the number of target views n. Since RotErr-V is summed over the n target views (Eq. 8), its near-linear growth with n indicates that the per-view rotation error remains stable. TransErr-V, in contrast, is computed after a Sim(3) scale alignment that normalizes the total trajectory length to the ground truth. Therefore, it captures scene-dependent shape distortion rather than per-view error accumulation, and remains approximately constant across different N. These results consistently demonstrate that our framework can accurately and robustly handle varying numbers of target views without degradation in geometric fidelity.

Analysis on the choice of estimator. We adopt Pi3 (Wang et al., 2026b) as our default estimator for both camera poses and depth, as we find it most robust on dynamic monocular videos and sparse multi-view images. To verify that our conclusions are not estimator-dependent, we re-evaluate both the camera pose errors and the pixel-level fidelity metrics (PSNR-m and SSIM-m) using Depth Anything 3 (Lin et al., 2026b). As shown in Table 6, our method consistently outperforms all baselines regardless of the estimator choice.

Analysis on computation cost. We measure the runtime and memory cost of View4D on a single Hopper-architecture GPU with 80 GB of memory, and report how they scale as the number of generated views increases. The number of generated frames and the resolution are set to 49 and 480 × 832, respectively. As shown in Table 7, both runtime and memory scale approximately linearly with the number of views. Notice that both the runtime and memory cost can be further reduced by running on multiple GPUS.

E Additional Visualizations

Point cloud visualization. To inspect cross-view consistency, we extract multi-view images across multiple frames, estimate their depth using Pi3, and unproject them into 3D space. As shown in Figure 12, the fused point clouds preserve plausible 3D structures without severe structural

Table 7: **Ablation study on computation cost.** We evaluate the runtime and memory consumption of View4D under different numbers of target views n.

n	2	3	4	5	6	7	8
Runtime (s) ↓	793	1070	1329	1602	1885	2216	2473
Memory (GB) ↓	45.87	50.27	54.67	59.07	63.47	67.86	72.26



Figure 12: **Visualization of 3D point clouds.** Given the multi-view videos generated by View4D, we sample a few frames (e.g., 5) and estimate their per-frame depth as in Section C.2. The multi-view frames are then unprojected into point clouds, which are fused into a unified 3D point cloud using estimated camera poses. The bottom row shows the corresponding sample frames.

collapse. Notably, planar regions such as the ground align reasonably well across different frames and viewpoints, qualitatively demonstrating the spatio-temporal coherence of View4D.

Additional visual comparisons. We present additional qualitative comparisons with baseline methods in Figure 13. The results demonstrate that View4D consistently produces high-fidelity novel views while maintaining strong cross-view consistency across diverse scenarios.

Additional visual results We present additional visual results in Figure 14. Given challenging in-the-wild input videos, View4D generates consistent, high-fidelity multi-view videos, demonstrating its potential for a wide range of applications such as world model, autonomous driving, robotics and virtual reality. It is also worth noting that View4D produces reasonable results for static-camera input videos, which can be regarded as a special case where the camera remains fixed over time.

F User Study Interface

Figure 15 illustrates the interface used for the human study described in Section 4.2. For each test case, we present the results of View4D and all baseline methods in a randomized order and ask participants to independently select the best result in terms of video quality and cross-view consistency. The interface provides three playback controls—*First frame*, *Pause all*, and *Restart all*—to facilitate careful comparison. Before the study, participants are also provided with instructions and representative positive and negative examples to familiarize them with the evaluation criteria.

G Limitations

Constrained by available computational resources and training data, our model is trained at a moderate scale, limiting its generative upper bound. However, our proposed explicit geometric guidance and relative projective positional encoding are fundamentally model-agnostic. We expect that simply scaling up the training corpus and deploying more powerful foundation models will naturally unlock higher generation quality and stronger open-world generalization, establishing a highly scalable baseline for future research.

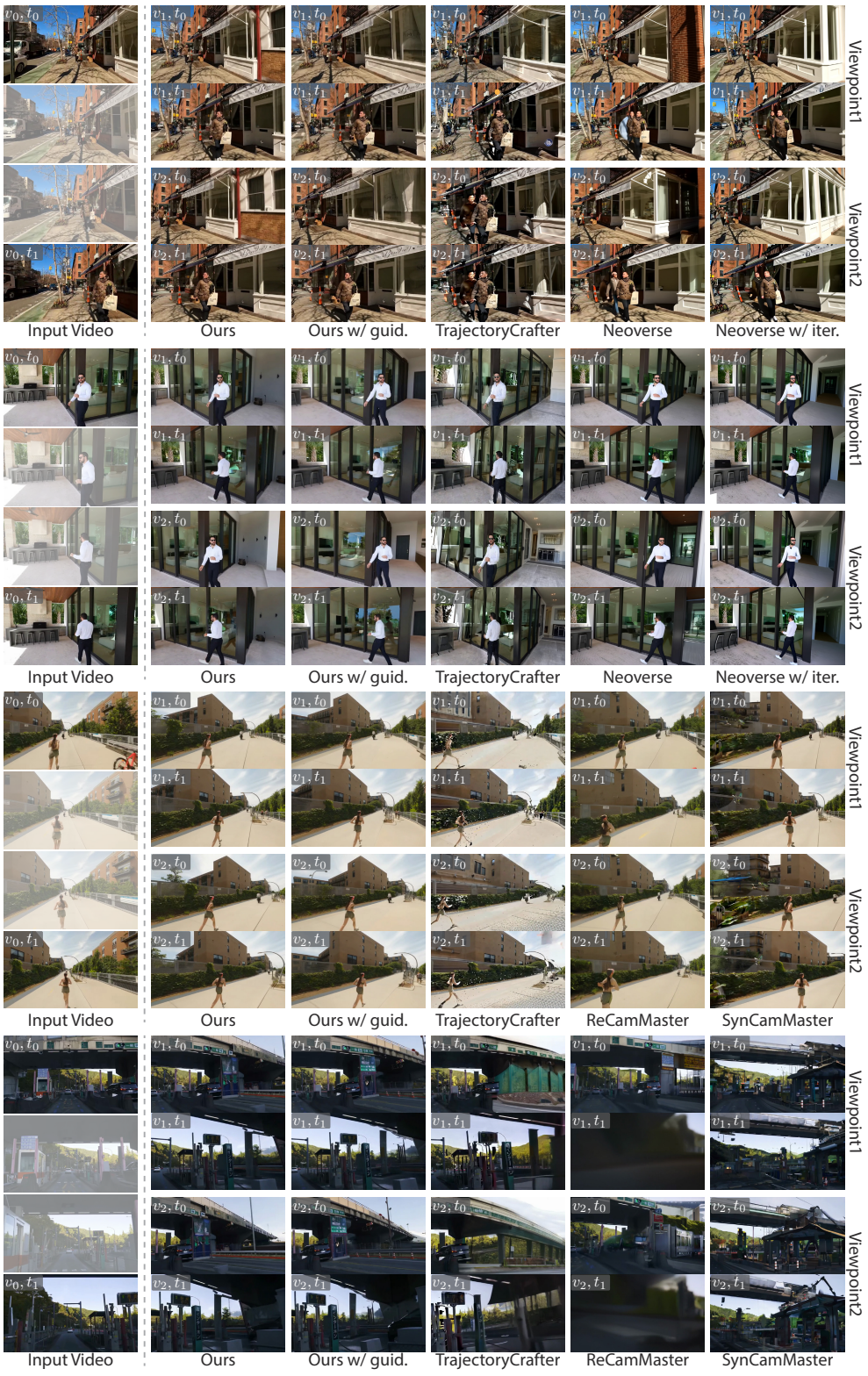

Figure 13: **Additional qualitative comparisons with state-of-the-art methods.** The visualization follows the same layout and notation as Fig. 4.

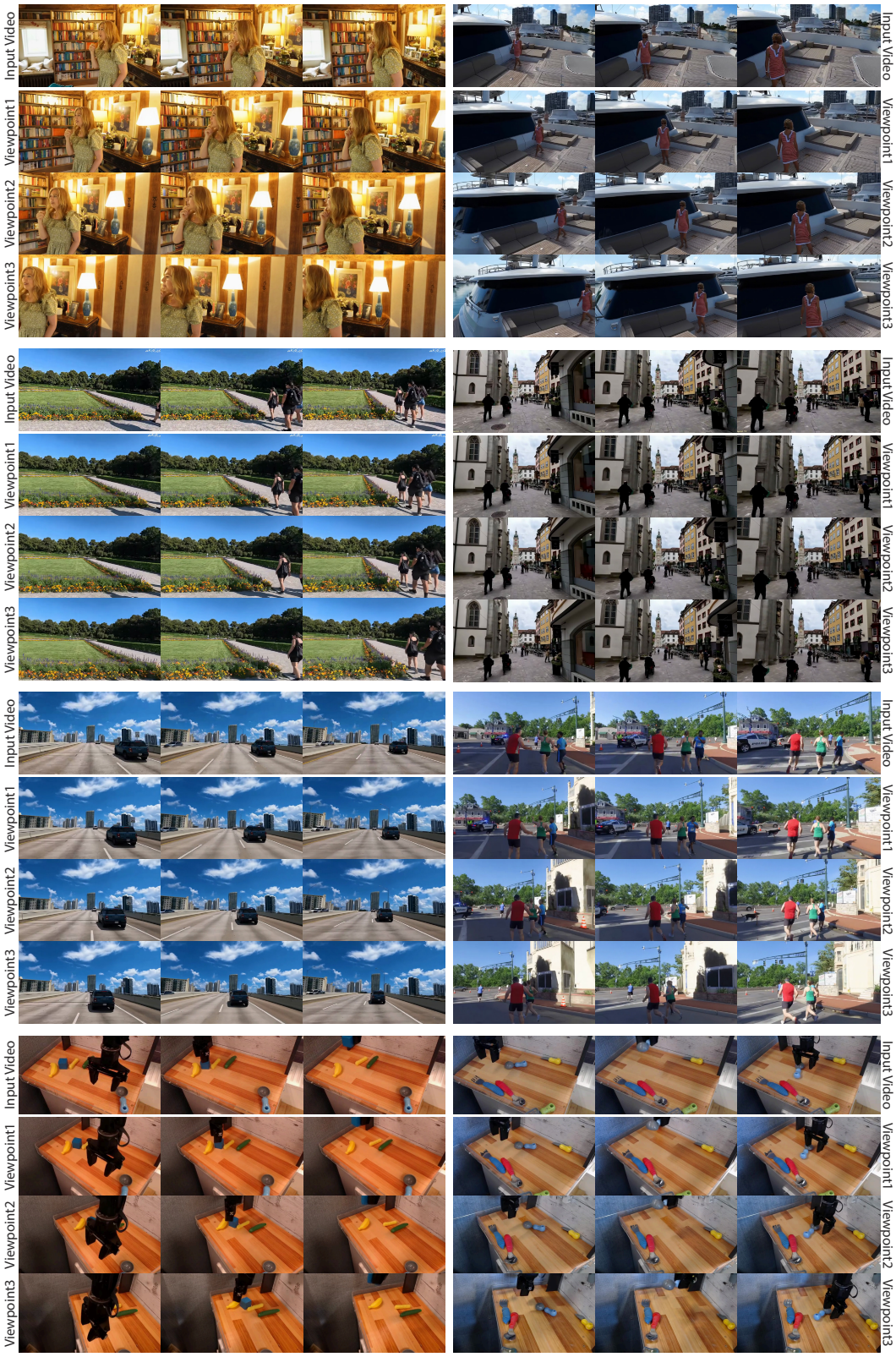

Figure 14: **Additional generation results of View4D**

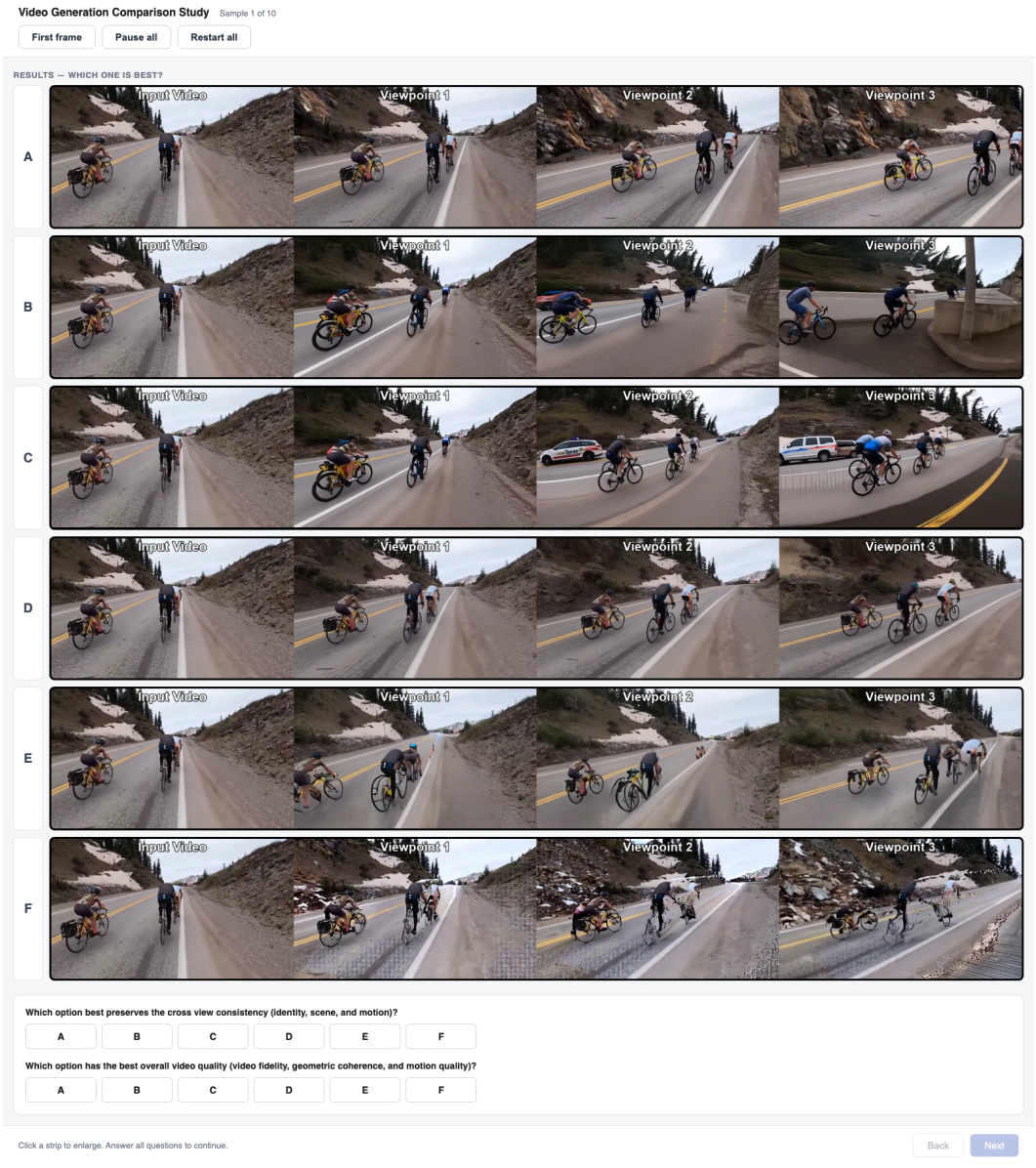

Figure 15: **Human user study interface.**