OmniView: An All-Seeing Diffusion Model for 3D and 4D View Synthesis

Anonymous CVPR 2026 Submission

Paper ID #****

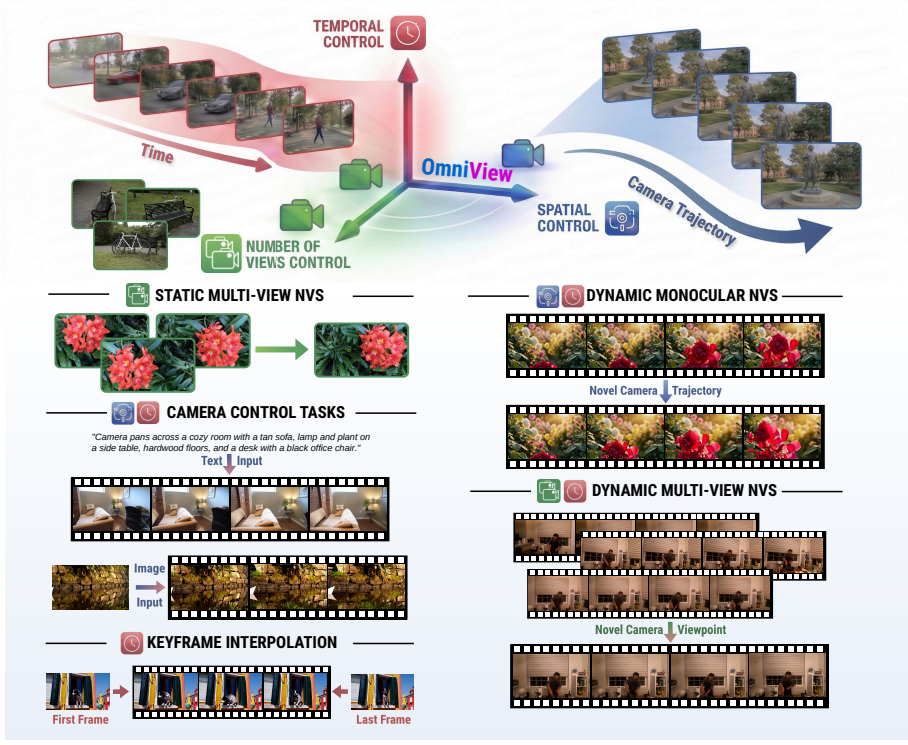

Fig. 1: OmniView: Common 4D consistent video generation tasks can be broken down into three control dimensions: controlling **time**, **space**, and the **number of conditioning views**. OmniView systematically models these three dimensions of control with one unified modeling and data sampling design, naturally enabling a variety of tasks such as static and dynamic novel view synthesis, camera control, key-frame interpolation, and so on.

Abstract. Prior approaches injecting camera control into diffusion models have focused on specific subsets of 4D consistency tasks: novel view synthesis, text-to-video with camera control, and image-to-video, amongst others. These fragmented approaches are trained on disjoint slices of available 3D/4D data. We introduce OmniView, a unified framework that generalizes across a wide range of such tasks. Our method separately represents space, time, and view conditions, enabling flexible combinations of these inputs. We introduce a RoPE mechanism in Diffusion Transformers (DiTs) for disentangling spatial and temporal information, enabling unified training with a wide variety of 3D/4D datasets.

OmniView can synthesize novel views from static, dynamic, and multiview inputs, extrapolate trajectories forward and backward in time, and create videos from text or image prompts with full camera control. OmniView is competitive or outperforms task-specific models across diverse benchmarks and metrics, improving image quality scores among camera-conditioned diffusion models by up to 33% in multiview NVS LLFF dataset, 60% in dynamic NVS Neural 3D Video benchmark, 20% in static camera control on RE-10K, and reducing camera trajectory errors by 4x in text-conditioned video generation. With strong generalizability in one model, OmniView demonstrates the feasibility of a generalist 4D video model.

1 Introduction

When trained on raw internet-scale data, video diffusion models internalize strong 3D priors [33, 43, 57]. They can synthesize long, coherent camera motions and maintain scene layout without any explicit geometry [47]. Yet, by default, they offer little explicit 3D control. To deploy them in applications such as virtual/augmented reality, film production, robotics, or autonomous driving, they need to be controllable. Many applications become possible if we can control how the generator's camera moves in space and in time. Naturally, this motivation has driven multiple sub-fields of computer vision to develop specialized solutions. Camera redirection now exists for multi-view static [18, 90] and dynamic Novel View Synthesis (NVS) [6, 60], camera control with text-to-video (T2V) [21, 76], image-to-video (I2V) [73] with camera control, or video-to-video (V2V) [5].

However, existing approaches are fragmented along task, architecture, and data. Methods designed for multi-view static NVS focus on reconstructing a single scene from sparse views [90], achieving strong 3D consistency but only at a fixed timestamp, and thus cannot handle dynamic videos. Camera-control T2V/I2V models convert text or a single image plus a camera trajectory and generate a moving video, but their architectures cannot ingest full input videos [21]. V2V redirection models accept a source video and re-render it from a new camera path at matched timestamps, yet they typically cannot operate with multi-view inputs [5]. Some works rely on explicit geometric representations (depth maps, point clouds, or other 3D/4D fields) for consistency [50, 69, 82, 83], rather than exploiting the implicit 3D priors already present in video models.

Because each family of methods is tailored to a narrow I/O configuration, they are trained on disjoint slices of available 3D/4D data, leaving most available geometric supervision unused. We argue that a single, flexible framework that can express all these tasks, trained on heterogeneous datasets, should improve generalization across 3D tasks and substantially reduce deployment overhead.

Our approach, OmniView, instantiates such a unified framework as a single video generative model for diverse view-synthesis tasks. We model each image $\mathbf{I}$ as a sample from a 4D world, parameterized by a camera pose $\mathbf{p}$ and a timestamp t. Under this view, static multi-view NVS corresponds to varying $\mathbf{p}_{1:N}$ and a target pose $\mathbf{p}^t$ while keeping t fixed; I2V with camera control corresponds to

predicting frames at future times $t_{1:N}$ along target poses $\mathbf{p}^t_{1:N}$ given an input $(\mathbf{I}_0, \mathbf{p}_0, t_0)$; and V2V camera redirection corresponds to re-rendering an input video from new poses $\mathbf{p}^t_{1:N}$ at the same timestamps $t_{1:N}$ as the source.

To realize this unified 4D formulation, we independently model space, time, and view conditions. First, we adopt a Diffusion Transformer (DiT) [48] backbone that naturally handles a variable number of input tokens. We tokenize each frame into a set of video tokens, concatenate tokens from all available inputs (images, views, or frames), and condition generation on this sequence. DiTs already support temporal reasoning via spatio-temporal Rotary Positional Embeddings (3D RoPE) [52], which encode (x, y, t) for each token. Prior works typically inject camera information by encoding poses directly or mapping them to Plücker ray embeddings and then applying 3D RoPE to both video and camera-conditioned tokens. This *entangles* camera pose $\mathbf{p}$ and time t in a single positional embedding space, making it difficult for the model to learn 3D structure independently of temporal dynamics and often leading to overfitting to seen trajectories.

To overcome this limitation, we explicitly disentangle space and time. We represent each token's camera pose as Plücker rays and apply *spatial* 2D RoPE only to these Plücker features, then concatenate them channel-wise with the corresponding video token. Time is encoded separately via temporal RoPE on the video tokens. This design cleanly separates camera geometry from temporal evolution, while still allowing the DiT to jointly attend over all tokens. Combined with the variable-token design, this lets OmniView flexibly ingest arbitrary combinations of frames, views, and timestamps and thereby support a wide range of 4D inputs under a single architecture. We then devise a joint training strategy that mixes heterogeneous 3D datasets, each corresponding to different task configurations (multi-view static, dynamic, T2V/I2V with camera control, V2V redirection), so that the model learns shared geometric priors across them.

We extensively evaluate OmniView on static and dynamic NVS benchmarks with monocular and multi-view inputs, as well as T2V/I2V camera-control tasks. OmniView consistently matches or outperforms specialized baselines, producing high-fidelity, 3D-consistent videos. With up to 33% increase in SSIM scores in multi-view static NVS (LLFF dataset), 60% in dynamic NVS (Neural 3D Video), 20% in I2V + camera control (RE-10K), and $\sim$4$\times$ reduction in camera errors in T2V + camera control (RE-10K), OmniView demonstrates strong 3D consistency and fidelity across tasks, and generalization to input configurations not seen during training. Ablations validate the benefit of our RoPE design.

2 Related work

Camera-controllable video generation. The emergence of powerful text-to-video diffusion models [2, 10, 13, 43, 79] has fueled extensive research on conditioning generated videos with additional controls, such as camera parameters [4, 20, 38, 51, 56, 65, 67, 77, 88, 89]. Early camera-control approaches integrate extrinsic camera information as part of the diffusion model's conditioning, either through tailored encoders or numeric input channels. Models like MotionCtrl [65]

and CameraCtrl [21] encode camera poses or trajectories to enable user-directed viewpoint changes throughout video generation, but often require specific paired training data and show limited generalization when camera motions deviate from training regimes. Other strategies bypass model retraining by employing 3D geometric cues, for example warping frames using estimated depth to match new camera placements and feeding these as priors during the denoising process [23, 80], though these methods face a trade-off between enforcing geometric consistency and visual fidelity.

Novel view synthesis and video-to-video generation. Generating unseen viewpoints from posed images or videos has advanced significantly in recent years [7, 8, 17, 24, 29, 34, 41, 45, 46, 54, 66, 75, 84]. Conventional novel view synthesis (NVS) frameworks leverage volumetric or Gaussian-based scene representations; meanwhile, feed-forward architectures [11, 12, 15, 26, 49, 55, 62, 81, 91] aim to directly predict target views from sparse or multi-view input, but usually struggle in generalization tasks or under challenging domain shifts. Some recent works attempt to harness image/video generative models to infuse prior knowledge and regularize deficits of view synthesis, as in ReconFusion [69] and CAT3D [18]. However, these strategies tend to be slow due to per-scene optimization depending on robust inter-view alignment, as seen in ReconX [37] and ViewCrafter [83], which become error-prone in the presence of thin or ambiguous structures. Relatedly, the video-to-video generation field [14, 39, 42, 61, 63, 64, 74, 78, 92] explores producing temporally consistent and controllable video outputs under various manipulation and conditioning tasks. Techniques such as GCD [56], Recapture [85], GS-DiT [9], DAS [19], and recent 4D-consistent pipelines [50, 71, 82] exploit geometric and dynamic scene information, such as tracked 3D points [28, 70], to condition or align the generative models, either via simulation or real-world sequences. DimensionX [53] and GenXD [87] further explore generating 3D and 4D scenes by disentangling camera and object movements, and spatial and temporal factors, respectively. These approaches enable synchronizing generated outputs across multiple cameras or time but are typically constrained by how accurately dynamic scene content can be retrieved or tracked. Some works tackle 4D and multi-view video generation by training generative models directly on synchronized video collections [6, 31, 59, 60, 68, 72], or by reconstructing an explicit scene representation first and then rendering views [35, 37, 86].

While the methods above have made significant progress, they typically target a narrow subset of 3D/4D tasks or are designed for a single setting—for example, camera-controlled text-to-video [3, 21, 65], single-image NVS [38, 51], or monocular video re-rendering [5, 82]. In contrast, OmniView unifies a broad range of tasks within a single model, including camera-controlled T2V/I2V generation, multi-view static NVS, monocular or multi-view dynamic NVS.

Consistency in video generation. Ensuring frame-to-frame and cross-view temporal or geometric consistency remains a critical challenge in video synthesis. Early efforts used 3D point clouds or height maps derived from input or generated content to guide learning and enforce consistency [16, 42]. Others [30] propagate consistency across parallel generated video streams but may lose coherence when

scene elements leave the views of all sequences. Latent feature histories have also been used to improve consistency for streaming or autoregressive video generation approaches [22], though explicit, interpretable 3D control remains an open research direction.

3 Method

3.1 Preliminary: Video Diffusion Models

Our framework builds on the popular architecture used in state-of-the-art text-to-video diffusion models [1, 2, 58, 79], which combine a 3D Variational Auto-Encoder (VAE) with a Diffusion Transformer (DiT) [48]. The VAE spatially and temporally compresses input videos into low-resolution latents that serve as compact representations for diffusion modeling. The diffusion process follows the rectified flow formulation [40], where the model learns to transform noise into coherent video latents through velocity prediction.

Within this framework, the DiT operates directly in latent space. It first patchifies the 3D latent tensor into spatio-temporal tokens $\mathbf{z}_{xyt} \in \mathbb{R}^d$, where x, y, and t denote the spatial and temporal coordinates of each d-dimensional token. These tokens are then processed through a stack of transformer blocks, each comprising a 3D spatio-temporal self-attention layer to capture motion and appearance consistency, a text cross-attention layer for semantic conditioning, and a feed-forward network (FFN) for feature transformation. This architecture enables efficient large-scale training and produces high-quality, temporally consistent video generations.

3.2 Network Architecture

As illustrated in Fig. 2, our model takes as input a set of images captured from different viewpoints and time steps, represented using Plücker ray maps. The objective is to denoise target tokens to generate video frames at any user-specified viewpoint and time. The configurations of context and target viewpoints and timestamps are flexible, enabling the model to adapt to a wide range of tasks.

To realize this capability, we next investigate the optimal designs for integrating context-frame conditioning, camera control, and the corresponding training strategies that best support these functionalities.

Context image conditioning. To facilitate flexibility and scalability in the number of inputs to our model, we propose a network that performs token concatenation as inputs to the DiT. The network takes as input a set of context tokens $\mathbf{z}^{\text{ctx}}_{xyt}$ (encoded from multiple input views), which are concatenated token-wise with a set of target tokens $\mathbf{z}^{\text{tgt}}_{xyt}$, where x, y, and t denote the spatial and temporal coordinates of the token vector. During the flow matching process, the target tokens are progressively denoised while attending to the clean context tokens, which serve as conditioning inputs to guide generation. The overall input to the DiT is therefore represented as a joint sequence with $\mathbf{z}^{\text{ctx}}$ and $\mathbf{z}^{\text{tgt}}$.

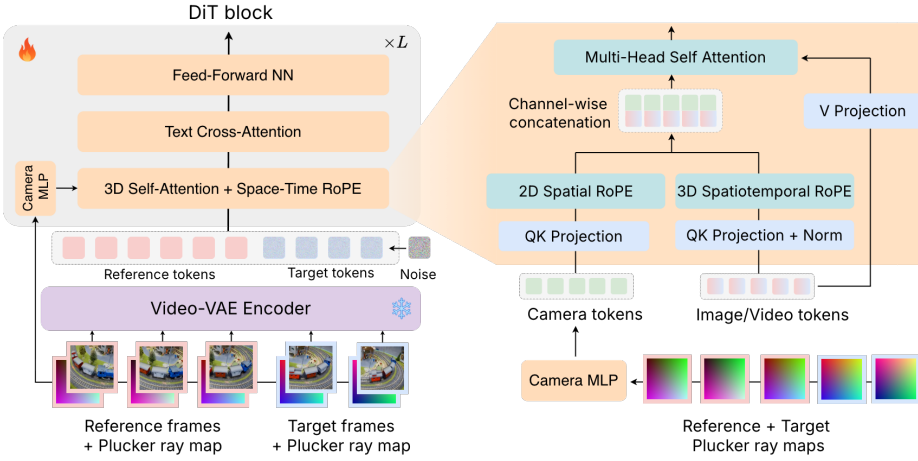

Fig. 2: Overview of the network architecture. We concatenate source input tokens and partly denoised target tokens as input to the DiT. Right: We apply an MLP to the camera embeddings for each view followed by separate 2D/3D RoPE mechanisms for camera and video tokens. The two sets of tokens have separate Query-Key projections and are channel-wise concatenated when input to the Self-Attention operation.

Camera embeddings. To incorporate camera information, we utilize Plücker ray maps $\mathbf{P} \in \mathbb{R}^{6\times H\times W}$, which represent the camera ray direction and origin for each image pixel. Our camera encoder $\mathcal{E}_c$ divides the ray map into patch volumes with the same spatio-temporal compression rate as the video VAE + DiT patchifier. These patch volumes are flattened channel-wise and passed through an MLP to obtain camera tokens for both context and target frames, denoted as $\mathbf{c}^{\text{ctx}}_{xyt}$ and $\mathbf{c}^{\text{tgt}}_{xyt}$, with resolution and channels matching that of the video tokens. Separate camera encoders are used for each DiT layer, allowing the model to flexibly modulate the influence of camera conditions at different network stages.

A naive strategy for injecting camera tokens $\mathbf{c}_{xyt}$ is to simply concatenate or add them to the corresponding video tokens $\mathbf{z}_{xyt}$. A similar approach is adopted in [5], where a camera encoder produces a 12-dimensional pose embedding that is added to the video tokens in every DiT block. However, this formulation entangles the spatial location of the camera and the temporal position of the corresponding frame within the video, as discussed below.

Disentangling camera and temporal position embeddings. Video DiTs use 3D Rotary Positional Embeddings (RoPE) [52] to encode the spatial-temporal positions (x, y, t) of video tokens. Specifically, RoPE applies a frequency-based rotation on the queries and keys of a token:

$$\text{RoPE}(\mathbf{q}^z_{xyt}) = R(\mathbf{q}^z_{xyt}, \boldsymbol{\theta}_{xyt}), \tag{1}$$

where $\boldsymbol{\theta}_{xyt}$ denotes the sinusoidal phase parameters set by the position (x, y, t), see supplementary for details. $\mathbf{q}^z_{xyt}$ denotes the query vector corresponding to the video tokens $\mathbf{z}$, and Eq. 1 is similarly applied to the key vectors $\mathbf{k}^z_{xyt}$.

When camera embeddings $\mathbf{c}_{xyt}$, with corresponding query-key projections $\mathbf{q}^c_{xyt}, \mathbf{k}^c_{xyt}$, are directly added to the video tokens prior to applying 3D RoPE, the transformation becomes:

$$R(\mathbf{q}^z_{xyt} + \mathbf{q}^c_{xyt}, \boldsymbol{\theta}_{xyt}) = R(\mathbf{q}^z_{xyt}, \boldsymbol{\theta}_{xyt}) + R(\mathbf{q}^c_{xyt}, \boldsymbol{\theta}_{xyt}), \tag{2}$$

since RoPE is a linear projection. This formulation, however, entangles camera and temporal information: the camera embeddings are rotated according to their specific camera corresponding timestamps t, even though they should ideally remain temporally invariant. Consequently, the model tends to overfit to the specific camera trajectories seen during training, reducing generalization to unseen camera trajectories as it implicitly encodes temporal correlations into the camera embeddings. This is further discussed in Sec. 4.6.

To address this issue and disentangle camera and temporal representations, we propose the following approach:

(i) Setting t as a constant for camera tokens. To eliminate the temporal interference on the camera tokens, we propose fixing their temporal index to a constant value, i.e., $t = 0$, effectively reducing the 3D RoPE to a 2D form. Under this modification, separate RoPE transformations are applied to the video tokens and the camera tokens as follows:

$$\tilde{\mathbf{q}}^z_{xyt} = R(\mathbf{q}_{xyt}, \boldsymbol{\theta}_{xyt}), \quad \tilde{\mathbf{q}}^c_{xyt} = R(\mathbf{q}^c_{xyt}, \boldsymbol{\theta}_{xy0}). \tag{3}$$

For brevity, the corresponding formulation for key vectors $\tilde{\mathbf{k}}^z_{xyt}$,$\tilde{\mathbf{k}}^c_{xyt}$ is omitted.

(ii) Channel-wise concatenation of video and camera tokens. The RoPE-transformed camera queries and keys $\tilde{\mathbf{q}}^c$, $\tilde{\mathbf{k}}^c$ and the video queries and keys $\tilde{\mathbf{q}}^z$, $\tilde{\mathbf{k}}^z$ must be fused before being fed into the scaled dot-product attention. This fusion can be performed additively or via channel-wise concatenation. To analyze the design choices, we compare the resulting attention scores under both strategies.

Let $m = (x, y, t)$ and $n = (x', y', t')$ denote the spatial-temporal positions of the query and key tokens, respectively. Under additive fusion, the attention score is given by:

$$A^{\text{add}}_{n,m} = \langle \tilde{\mathbf{q}}^z_m + \tilde{\mathbf{q}}^c_m, \tilde{\mathbf{k}}^z_n + \tilde{\mathbf{k}}^c_n \rangle, \tag{4}$$

where $\langle \cdot, \cdot \rangle$ denotes the inner product. We observe that $A^{\text{add}}_{n,m}$ introduces entangled cross-terms between the camera and temporal positions, *i.e.* $\langle \tilde{\mathbf{q}}^z_m, \tilde{\mathbf{k}}^c_n \rangle$ and $\langle \tilde{\mathbf{k}}^z_n, \tilde{\mathbf{q}}^c_m \rangle$, which can lead to undesirable deviations in the attention map and unstable interactions between spatial-temporal and camera embeddings.

Instead, we advocate for *channel-wise concatenation* of the video and camera tokens, yielding an attention score of:

$$A^{\text{cat}}_{n,m} = \langle [\tilde{\mathbf{q}}^z_m; \tilde{\mathbf{q}}^c_m], [\tilde{\mathbf{k}}^z_n; \tilde{\mathbf{k}}^c_n] \rangle = \langle \tilde{\mathbf{q}}^z_m, \tilde{\mathbf{k}}^z_n \rangle + \langle \tilde{\mathbf{q}}^c_m, \tilde{\mathbf{k}}^c_n \rangle, \tag{5}$$

where the camera tokens and video tokens are fully disentangled. In the canonical case where two frames at t, t' share the same camera configuration ($\boldsymbol{c}_{xyt}$=$\boldsymbol{c}_{xyt'}$), regardless of their temporal index, the concatenation formulation yields a constant offset in the attention map, thereby maintaining behavior closely aligned

with the original temporal attention structure, as desired. We show ablations by exploring the two RoPE variants in Tab. 6 supporting our hypothesis.

(iii) Separate QK projections for camera tokens. Finally, we find it crucial to further enhance the representation capacity of camera tokens by introducing separate query and key (QK) projection layers that transform the camera embeddings $\mathbf{c}$ into $\mathbf{q}^c$ and $\mathbf{k}^c$. This modification allows the model to learn camera-specific attention patterns distinct from those of the video tokens.

The resulting transformer architecture is illustrated in Fig. 2, incorporating these design choices while computing the attention score as in Eq. (5).

3.3 Training Setup

We train on a diverse collection of 3D/4D datasets (Tab. 1). Stereo4D [27] provides stereo videos with poses, though we use only one view per video (treating it as monocular) since per-video stereo baselines are unavailable, but take advantage of its diverse pose annotations. RE10K [93] and DL3DV [36] serve as multi-view image NVS sources, while the synthetic SynCamMaster [6] and ReCamMaster [5] datasets provide temporally synchronized static and dynamic camera data, respectively. We target context and target configurations of 1, 3, 5, or 10 latent frames with up to 3 context views. By varying these, the model generalizes to longer sequences and more views at test time (Fig. 6), and even to untrained task compositions such as multi-view video NVS (3×3 → 1×3).

Table 1: Multitask training configurations for various datasets in terms of number of views V and number of latent frames F.

Task Type	Datasets	Context ($V \times F$)	Target ($V \times F$)
Monocular Image NVS, Multi-view Image NVS	RE10K, DL3DV, ReCamMaster, SynCamMaster	3×1, 2×1, 1×1	1×1
Monocular Video NVS	ReCamMaster, SynCamMaster	1×3	1×3
T2V+CamCtrl	RE10K, DL3DV, Stereo4D	–	1×3, 1×5, 1×10
I2V+CamCtrl	RE10K, DL3DV, Stereo4D	1×1	1×3, 1×5, 1×10
V2V+CamCtrl	RE10K, DL3DV, Stereo4D	1×3	1×3, 1×5, 1×10

During training, we randomly sample a task and a supporting dataset. For static and dynamic NVS, context and target timestamps are identical. For camera-control tasks, target frames are sampled such that $|t_0^{\text{tgt}} - t_0^{\text{ctx}}| \leq \Delta$, where Δ is a task-dependent offset, enabling I2V/V2V generation for both future and preceding frames relative to the context.

4 Experiments

The key takeaways of our experiments are: a) OmniView is capable of high-quality 4D consistency tasks across a wide variety of settings, including camera control and novel view synthesis for both static and dynamic scenes. b) Compared to specialized methods that focus on specific settings, OmniView effectively combines many types of camera and time conditions via our proposed RoPE, leading to generalization across a variety of tasks. c) As the number of input views increases, OmniView is able to effectively leverage the increased

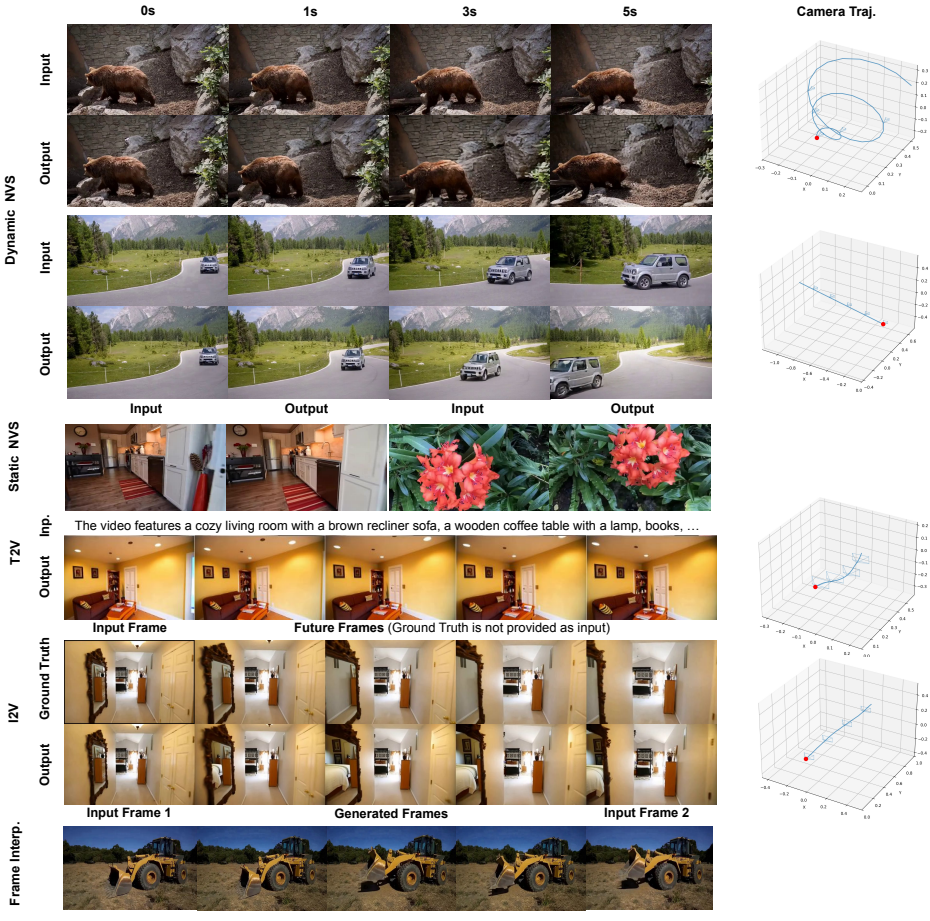

Fig. 3: Example generations from OmniView. We show results on (a) dynamic NVS; (b) static NVS; (c) text-to-video (T2V) with camera control; (d) image-to-video (I2V) with camera control; and (e) frame interpolation. In all cases, OmniView produces high-fidelity, 3D-consistent videos that adhere to the input conditioning(s).

signal in the input to improve reconstruction quality. d) Through ablations, we show that our proposed camera RoPE design is crucial for effective modeling of camera and time conditions, leading to improved performance across tasks.

4.1 Experimental Setup

Implementation Details. We train OmniView on the dataset mixture in Tab. 1 using the Wan 2.1 T2V 1.3B model [58] as the base architecture unless stated otherwise. Each iteration takes source views, target views, their cameras, and real-world timestamps as input. We train for 40K iterations on 32 H100 GPUs with a batch size of 64, linearly warming up the learning rate to 0.0001 over 3K iterations. During warmup, we train exclusively on the multiview static task to quickly adapt the Plücker ray map parameters and camera conditioning.

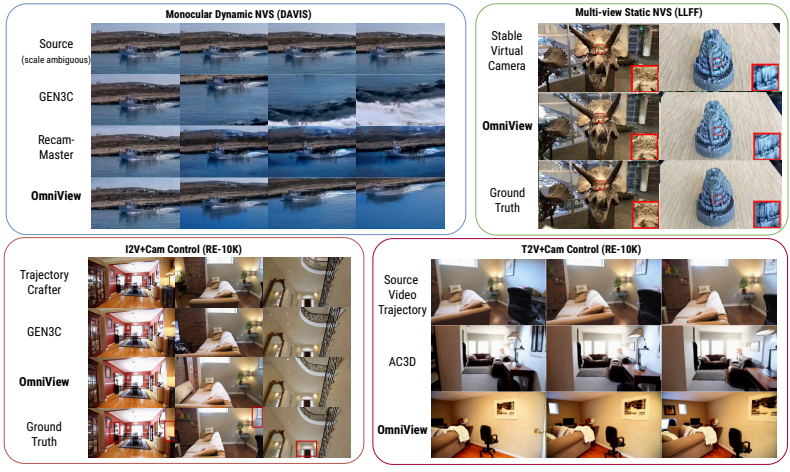

Fig. 4: Qualitative comparison of OmniView vs. baselines. OmniView consistently well-performs against high-quality baselines across a diverse set of tasks.

Evaluation and Baselines. We evaluate OmniView on three task groups: a) Monocular Video NVS (Tab. 5), b) Multi-view Image and Video NVS (Tab. 2), and c) T2V/I2V+Camera Control. Our primary baselines are RecamMaster [5], TrajectoryCrafter [82], and GEN3C [50] for monocular NVS and I2V camera control, while SEVA [90] serves as the multi-view baseline. Qualitative results are shown in Fig. 3. We report PSNR, SSIM, and LPIPS where GT views are available, and camera trajectory quality via Rotation Error (RotErr) and Translation Error (TrErr) [21], using MegaSaM [35] to estimate trajectories from generated videos. Following [3], we compute per-scene metric scale to provide a reasonable scale whenever scale ambiguity arises. A comparison overview is shown in Fig. 4.

4.2 Multi-view Static and Dynamic NVS

We evaluate OmniView on the LLFF dataset [44] consisting of multiple view captures of a static scene. We sparsely sample 3, 6, or 9 input views and choose one test view. We compare against SEVA [90], a high-quality model that allows for multi-view inputs with results shown in Tab. 2. Despite not being specifically trained on more than 3 static views, we see that OmniView still outperforms the baseline in all evaluated settings. Visually, in Fig. 4(b), we can see that OmniView preserves details to a higher degree compared to SEVA.

Table 2: Quantitative comparison on LLFF for static NVS with varying number of input views. We outperform the baseline [90] across all reconstruction metrics.

Method	Views	PSNR↑	SSIM↑	LPIPS↓
SEVA [90]	3	14.84	0.30	0.46
OmniView		**15.43**	**0.38**	**0.41**
SEVA [90]	6	15.36	0.32	0.43
OmniView		**16.11**	**0.42**	**0.38**
SEVA [90]	9	15.60	0.33	0.42
OmniView		**16.49**	**0.45**	**0.34**

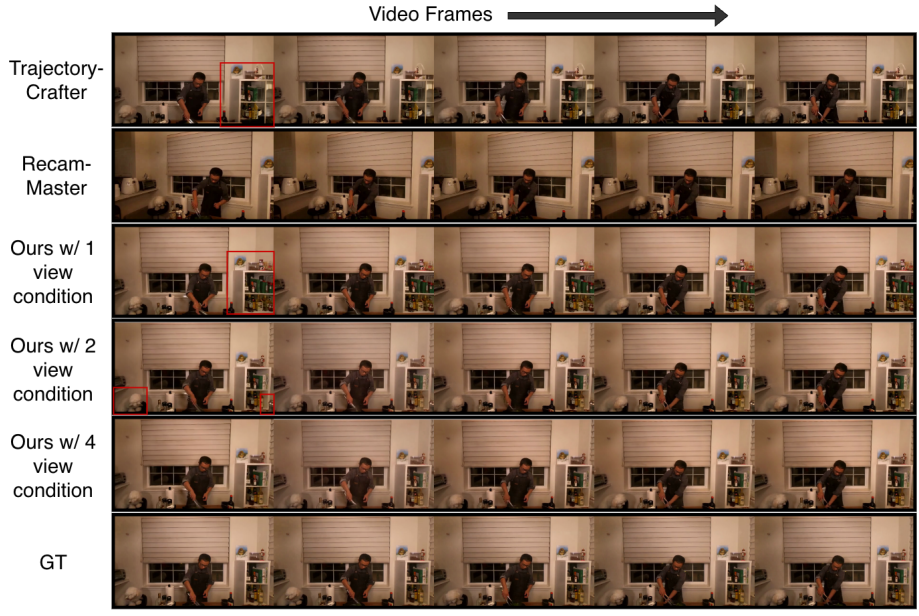

Fig. 5: Qualitative visualizations on a scene in N3DV. We outperform [5, 82] on single view input while improving reconstruction quality with increasing number of input conditions. [5] fails to generalize to the static camera trajectory unseen during training.

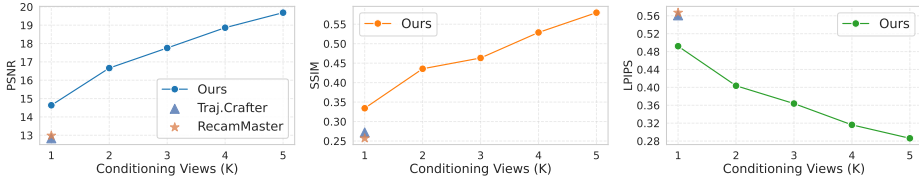

Fig. 6: Increasing number of conditional views improves reconstruction on N3DV [32] and is more aligned with the GT while outperforming [5, 82] with single view input.

This is similarly observed in the dynamic setting for N3DV, where we continue to improve reconstruction quality from 1 to 5 views (Fig. 6) in terms of PSNR, SSIM, and LPIPS. We visualize the reconstructions in Fig. 5. As the number of views increases, the model is able to better resolve the scale ambiguity with static cameras producing increasingly more aligned generations with the GT view. Notably, our training configuration (Tab. 1) does not include the multi-view dynamic NVS task but includes only multi-view static NVS and monocular dynamic NVS. This highlights the capability of our model to generalize to not only more views or more frames, but also to new 3D/4D tasks which are combinations of trained tasks. This opens the avenue to potentially include a variety of inputs such as multiple view combinations of images and videos.

4.3 I2V/T2V+Camera Control

While NVS targets generations with timestamps that are same as input conditions, I2V/T2V+Camera Control involves generating frames with target timestamps not present in the input.

For T2V, we pass no conditional views and only specify the target trajectories. We evaluate both I2V/T2V on RE10K on a subset of 1000/2000 samples respectively. For I2V, we compare with DimensionX [53], GenXD [87],TrajectoryCrafter [82], and Gen3C [50]. For T2V, we compare with AC3D [3] on both its original CogVideoX [79] backbone and the same Wan 2.1 1.3B backbone as OmniView. Results are summarized in Tabs. 3 and 4 with visualizations in Fig. 4.

Table 3: Quantitative comparison on RE10K for the I2V+Camera Control task. OmniView outperforms all baselines across reconstruction metrics.

Method	PSNR↑	SSIM↑	LPIPS↓
DimensionX [53]	14.30	0.47	0.46
GenXD [87]	15.18	0.55	0.56
TrajectoryCrafter [82]	16.94	0.53	0.36
Gen3C [50]	17.34	0.55	0.34
OmniView	**19.20**	**0.66**	**0.28**

Table 4: Camera-error metrics on RE10K for T2V+Camera Control. We compare AC3D on both its original CogVideoX backbone and the Wan 2.1 backbone.

Method	TransErr↓	RotErr↓
AC3D [3] (CogVideoX 5B)	5.170	1.365
AC3D [3] (Wan 2.1 1.3B)	14.034	2.292
OmniView (Wan 2.1 1.3B)	**1.412**	**0.572**

OmniView outperforms all I2V baselines across reconstruction metrics by a significant margin (Tab. 3). For T2V camera control (Tab. 4), we outperform AC3D on its original CogVideoX backbone. Fig. 4(c)(d) shows that OmniView consistently follows the source video camera trajectory, compared to increased trajectory error in TrajectoryCrafter and Gen3C and AC3D's inconsistency. In addition, we observe that AC3D's ControlNet-based camera injection struggles when trained on the Wan backbone (Tab. 4)—despite generating coherent videos, it fails to learn precise camera trajectories, suggesting that injecting camera control via ControlNet with a frozen main model does not generalize well across architectures. In contrast, OmniView yields strong camera controllability across backbones, despite AC3D being trained primarily on the camera control task while we train on a variety of different 3D/4D tasks.

4.4 Monocular Video NVS

For this task, we extract camera trajectories for 45 real-world videos from the DAVIS dataset [25]. For each video, we evaluate on 3 camera trajectory types: Arc Left, Arc Right, and Spiral. Results are summarized in Tab. 5. Averaged over all trajectories, we outperform prior SOTA approaches of [5, 50, 82] in translation error while remaining competitive in rotation error. Fig. 4(a) shows that OmniView maintains consistent content while Gen3C and ReCamMaster exhibits increased artifacts and altered background elements. Results from the N3DV dataset (Fig. 5) also demonstrate the effectiveness of OmniView. Compared to baselines, we obtain reconstructions well aligned with the ground truth.

Note that our conditioning is largely implicit via encoded latents and we do not use any form of explicit 3D supervision such as depth or point clouds apart from the metric scene scale. Despite being trained on 122K camera trajectories, ReCamMaster fails for this setting of static target camera and tends to implicitly induce camera motion. This highlights the generalizability issue of using 3D spatiotemporal RoPE on camera conditions as discussed in Sec. 3 where the model fails to disentangle camera and time, resulting in poor out-of-distribution trajectory performance.

Table 5: Quantitative comparison on different camera trajectory types on DAVIS. We perform competitively against prior SOTA Video NVS approaches while outperforming in certain categories. **TE** and **RE** refer to Translation Error and Rotation Error as defined in [21].

Method	Arc Left		Arc Right		Spiral		Overall	
	TE↓	RE↓	TE↓	RE↓	TE↓	RE↓	TE↓	RE↓
TrajCraft. [82]	14.37	4.00	24.92	2.64	23.11	2.05	20.80	2.90
ReCam. [5]	<u>8.85</u>	3.39	<u>16.49</u>	2.82	**13.49**	2.37	<u>12.94</u>	2.86
Gen3C [50]	129.17	**2.62**	21.80	**1.39**	25.72	**1.71**	58.90	**1.91**
OmniView	**5.77**	<u>3.27</u>	**8.88**	<u>2.50</u>	<u>14.53</u>	<u>2.03</u>	**9.73**	<u>2.60</u>

4.5 Key-frame interpolation

OmniView additionally provides the capability of conditioning generation based on past/future frames by changing the timestamps accordingly for the input frames. Fig. 3 shows qualitative results for keyframe interpolation with input first and last frames and static camera. We see that the intermediate frames produce realistic generations while also being consistent to the input frames highlighting the capability of our model to generalize to new tasks unseen during training.

4.6 Ablation Study: Camera RoPE

We conduct an ablation study on the N3DV dataset to isolate the contribution of each design choice introduced in Sec. 3. Table 6 shows results by adding our components. Our full model incurs only a modest overhead of +3.7% parameters and +0.8% FLOPs relative to the baseline. We describe each variant below:

- **Baseline (Plücker+3D RoPE)**: Plücker embeddings are added directly to the video hidden states, as is typical in prior work [5]. The standard 3D RoPE is implicitly applied to both video and camera embeddings.
- **+Constant t for camera tokens (Sec. 3.2(i))**: We fix the temporal index of camera tokens to t=0, reducing their RoPE to 2D and removing temporal variation from camera embeddings. Video tokens retain full 3D RoPE. Camera and video tokens are summed before attention.
- **+Separate QK projections (Sec. 3.2(iii))**: We introduce independent projection layers for the camera queries and keys, allowing the model to learn camera-specific attention patterns. This accounts for 3.7% added parameters.
- **+Channel-wise concatenation (Sec. 3.2(ii))**: Instead of adding Plücker embeddings to video tokens, we concatenate them along the channel dimension. As derived in Sec. 3, this yields a fully disentangled attention score

Table 6: Ablation study of our camera RoPE design choices on N3DV. Each row incrementally adds one component from Sec. 3. Our full model adds only +3.7% parameters and +0.8% FLOPs over the baseline.

Variant	PSNR↑	SSIM↑	LPIPS↓	Params (M)	FLOPs (G)
Baseline (Plücker + 3D RoPE)	13.68	0.309	0.509	1914.78	54719.73
+ Constant t for camera (Sec. 3.2(i))	14.17	0.345	0.504	1914.78	54719.73
+ Separate QK projections (Sec. 3.2(ii))	14.62	0.358	0.487	1985.60	55159.66
+ Channel-wise concat (Sec. 3.2(ii))	**15.46**	**0.376**	**0.456**	1985.60	55160.81

$\langle \tilde{\mathbf{q}}_m^z, \tilde{\mathbf{k}}_n^z \rangle + \langle \tilde{\mathbf{q}}_m^c, \tilde{\mathbf{k}}_n^c \rangle$, avoiding the cross-terms that arise with additive fusion. The FLOPs overhead is negligible since attention is a small fraction of total compute relative to the QKV projection and MLP layers.

Each component yields consistent improvement, and the full approach achieves the best performance across all metrics. This confirms the importance of our proposed camera RoPE design, which effectively disentangles camera and time conditions, leading to improved performance across tasks.

Why time-invariant camera tokens? In all variants, the standard 3D RoPE is always applied to the video tokens exactly as in the base model, ensuring that temporal ordering and relative frame-to-frame distances are already fully encoded within the video representation. The camera tokens, by contrast, are introduced solely to convey spatial pose information and are not intended to carry any temporal signal. When 3D RoPE is additionally applied to camera tokens (as in the Baseline), the *same* input pose at different timesteps is mapped to *different* post-RoPE values—for instance, a static or looped camera trajectory would produce varying camera token representations across time despite corresponding to identical viewpoints. This unintended coupling between pose and time destabilizes camera conditioning and noticeably reduces controllability. Fixing t=0 for camera tokens avoids this issue entirely: the model reasons about camera pose exclusively via the camera tokens, while temporal progression is handled by the video tokens' 3D RoPE, thereby maintaining a clean and principled separation of concerns between spatial and temporal dimensions. The consistent improvement observed from the baseline to the constant-t variant in Tab. 6 empirically confirms the effectiveness of this design choice.

5 Conclusion

In this work, we introduced OmniView, a novel framework that effectively integrates a variety of 3D/4D tasks into a single unified model. We develop a novel camera RoPE mechanism for decoupling spatial and temporal conditions input to the model, enabling flexible and scalable training across a wide variety of 3D and 4D datasets. Our approach demonstrates superior performance across a range of diverse and challenging tasks, including accurate camera control and high-quality novel view synthesis for both static and dynamic scenes. In the future, we plan to explore further enhancements to the model architecture and training strategy, as well as investigate additional promising applications in the broader and rapidly growing domain of 4D content generation.

References

1. Agarwal, N., Ali, A., Bala, M., Balaji, Y., Barker, E., Cai, T., Chattopadhyay, P., Chen, Y., Cui, Y., Ding, Y., et al.: Cosmos world foundation model platform for physical ai. arXiv preprint arXiv:2501.03575 (2025)
2. et. al., W.K.: Hunyuanvideo: A systematic framework for large video generative models (2024), https://arxiv.org/abs/2412.03603
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., et al.: Vd3d: Taming large video diffusion transformers for 3d camera control. arXiv preprint arXiv:2407.12781 (2024)
5. Bai, J., Xia, M., Fu, X., Wang, X., Mu, L., Cao, J., Liu, Z., Hu, H., Bai, X., Wan, P., Zhang, D.: Recammaster: Camera-controlled generative rendering from a single video (2025), https://arxiv.org/abs/2503.11647
6. 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)
7. Barron, J.T., Mildenhall, B., Tancik, M., Hedman, P., Martin-Brualla, R., Srinivasan, P.P.: Mip-NeRF: a multiscale representation for anti-aliasing neural radiance fields. ICCV (2021)
8. Barron, J.T., Mildenhall, B., Verbin, D., Srinivasan, P.P., Hedman, P.: Mip-NeRF 360: unbounded anti-aliased neural radiance fields. CVPR (2022)
9. Bian, W., Huang, Z., Shi, X., Li, Y., Wang, F.Y., Li, H.: Gs-dit: Advancing video generation with pseudo 4d gaussian fields through efficient dense 3d point tracking. arXiv preprint arXiv:2501.02690 (2025)
10. 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)
11. Charatan, D., Li, S.L., Tagliasacchi, A., Sitzmann, V.: PixelSplat: 3D Gaussian splats from image pairs for scalable generalizable 3D reconstruction. In: CVPR (2024)
12. Chen, A., Xu, Z., Zhao, F., Zhang, X., Xiang, F., Yu, J., Su, H.: MVSNeRF: fast generalizable radiance field reconstruction from multi-view stereo. In: ICCV. pp. 14124–14133 (2021)
13. Chen, H., Zhang, Y., Cun, X., Xia, M., Wang, X., Weng, C., Shan, Y.: Videocrafter2: Overcoming data limitations for high-quality video diffusion models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7310–7320 (2024)
14. Chen, Q., Ma, Y., Wang, H., Yuan, J., Zhao, W., Tian, Q., Wang, H., Min, S., Chen, Q., Liu, W.: Follow-your-canvas: Higher-resolution video outpainting with extensive content generation. arXiv preprint arXiv:2409.01055 (2024)
15. Chen, Y., Xu, H., Zheng, C., Zhuang, B., Pollefeys, M., Geiger, A., Cham, T.J., Cai, J.: MVSplat: efficient 3D Gaussian splatting from sparse multi-view images. In: ECCV (2024)
16. Deng, B., Tucker, R., Li, Z., Guibas, L., Snavely, N., Wetzstein, G.: Streetscapes: Large-scale consistent street view generation using autoregressive video diffusion. In: ACM SIGGRAPH 2024 Conference Papers. pp. 1–11 (2024)

17. Duan, Y., Wei, F., Dai, Q., He, Y., Chen, W., Chen, B.: 4D-rotor Gaussian splatting: Towards efficient novel view synthesis for dynamic scenes. In: Proc. SIGGRAPH (July 2024)
18. Gao, R., Holynski, A., Henzler, P., Brussee, A., Martin-Brualla, R., Srinivasan, P., Barron, J.T., Poole, B.: CAT3D: create anything in 3D with multi-view diffusion models. arXiv preprint arXiv:2405.10314 (2024)
19. Gu, Z., Yan, R., Lu, J., Li, P., Dou, Z., Si, C., Dong, Z., Liu, Q., Lin, C., Liu, Z., et al.: Diffusion as shader: 3d-aware video diffusion for versatile video generation control. arXiv preprint arXiv:2501.03847 (2025)
20. 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)
21. 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)
22. Henschel, R., Khachatryan, L., Hayrapetyan, D., Poghosyan, H., Tadevosyan, V., Wang, Z., Navasardyan, S., Shi, H.: Streamingt2v: Consistent, dynamic, and extendable long video generation from text. arXiv preprint arXiv:2403.14773 (2024)
23. Hou, C., Wei, G., Zeng, Y., Chen, Z.: Training-free camera control for video generation. arXiv preprint arXiv:2406.10126 (2024)
24. Huang, B., Yu, Z., Chen, A., Geiger, A., Gao, S.: 2D Gaussian splatting for geometrically accurate radiance fields. In: SIGGRAPH 2024 Conference Papers. ACM (2024). https://doi.org/10.1145/3641519.3657428
25. Huang, J.B., Kang, S.B., Ahuja, N., Kopf, J.: Temporally coherent completion of dynamic video. In: ACM (2016)
26. Jin, H., Jiang, H., Tan, H., Zhang, K., Bi, S., Zhang, T., Luan, F., Snavely, N., Xu, Z.: LVSM: a large view synthesis model with minimal 3D inductive bias (2024), https://arxiv.org/abs/2410.17242
27. Jin, L., Tucker, R., Li, Z., Fouhey, D., Snavely, N., Holynski, A.: Stereo4D: Learning How Things Move in 3D from Internet Stereo Videos. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2025)
28. Karaev, N., Rocco, I., Graham, B., Neverova, N., Vedaldi, A., Rupprecht, C.: Cotracker: It is better to track together. In: European Conference on Computer Vision. pp. 18–35. Springer (2024)
29. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G.: 3D Gaussian splatting for real-time radiance field rendering. ACM Transactions on Graphics **42**(4) (July 2023), https://repo-sam.inria.fr/fungraph/3d-gaussian-splatting/
30. Kuang, Z., Cai, S., He, H., Xu, Y., Li, H., Guibas, L., Wetzstein, G.: Collaborative video diffusion: Consistent multi-video generation with camera control. arXiv preprint arXiv:2405.17414 (2024)
31. Li, B., Zheng, C., Zhu, W., Mai, J., Zhang, B., Wonka, P., Ghanem, B.: Vivid-zoo: Multi-view video generation with diffusion model. Advances in Neural Information Processing Systems **37**, 62189–62222 (2024)
32. Li, T., Slavcheva, M., Zollhoefer, M., Green, S., Lassner, C., Kim, C., Schmidt, T., Lovegrove, S., Goesele, M., Newcombe, R., et al.: Neural 3d video synthesis from multi-view video. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 5521–5531 (2022)
33. Li, X., Lai, Z., Xu, L., Qu, Y., Cao, L., Zhang, S., Dai, B., Ji, R.: Director3d: Real-world camera trajectory and 3d scene generation from text. arXiv:2406.17601 (2024)

34. Li, Z., Müller, T., Evans, A., Taylor, R.H., Unberath, M., Liu, M.Y., Lin, C.H.: Neuralangelo: High-fidelity neural surface reconstruction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 8456–8465 (June 2023)
35. Li, Z., Tucker, R., Cole, F., Wang, Q., Jin, L., Ye, V., Kanazawa, A., Holynski, A., Snavely, N.: Megasam: Accurate, fast, and robust structure and motion from casual dynamic videos. arXiv preprint arXiv:2412.04463 (2024)
36. Ling, L., Sheng, Y., Tu, Z., Zhao, W., Xin, C., Wan, K., Yu, L., Guo, Q., Yu, Z., Lu, Y., et al.: DL3DV-10K: a large-scale scene dataset for deep learning-based 3D vision. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 22160–22169 (2024)
37. Liu, F., Sun, W., Wang, H., Wang, Y., Sun, H., Ye, J., Zhang, J., Duan, Y.: ReconX: reconstruct any scene from sparse views with video diffusion model. arXiv preprint arXiv:2408.16767 (2024)
38. Liu, R., Wu, R., Hoorick, B.V., Tokmakov, P., Zakharov, S., Vondrick, C.: Zero-1-to-3: Zero-shot one image to 3d object (2023)
39. Liu, S., Zhang, Y., Li, W., Lin, Z., Jia, J.: Video-p2p: Video editing with cross-attention control. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8599–8608 (2024)
40. 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)
41. Luiten, J., Kopanas, G., Leibe, B., Ramanan, D.: Dynamic 3D Gaussians: tracking by persistent dynamic view synthesis. In: 3DV (2024)
42. Mallya, A., Wang, T.C., Sapra, K., Liu, M.Y.: World-consistent video-to-video synthesis. In: Proceedings of the European Conference on Computer Vision (2020)
43. Menapace, W., Siarohin, A., Skorokhodov, I., Deyneka, E., Chen, T.S., Kag, A., Fang, Y., Stoliar, A., Ricci, E., Ren, J., et al.: Snap video: Scaled spatiotemporal transformers for text-to-video synthesis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7038–7048 (2024)
44. Mildenhall, B., Srinivasan, P.P., Ortiz-Cayon, R., Kalantari, N.K., Ramamoorthi, R., Ng, R., Kar, A.: Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. ACM Transactions on Graphics (ToG) **38**(4), 1–14 (2019)
45. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: NeRF: representing scenes as neural radiance fields for view synthesis. Communications of the ACM **65**(1), 99–106 (2021)
46. Müller, T., Evans, A., Schied, C., Keller, A.: Instant neural graphics primitives with a multiresolution hash encoding. ACM Trans. Graph. **41**(4), 102:1–102:15 (Jul 2022). `https://doi.org/10.1145/3528223.3530127`, `https://doi.org/10.1145/3528223.3530127`
47. Parker-Holder, J., Fruchter, S.: Genie 3: A new frontier for world models. Blog post, Google DeepMind (Aug 2025), `https://deepmind.google/blog/genie-3-a-new-frontier-for-world-models/`, accessed: YYYY-MM-DD
48. Peebles, W., Xie, S.: Scalable diffusion models with transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 4195–4205 (2023)
49. Ren, X., Lu, Y., Liang, H., Wu, Z., Ling, H., Chen, M., Fidler, S., Williams, F., Huang, J.: Scube: Instant large-scale scene reconstruction using voxsplats. arXiv preprint arXiv:2410.20030 (2024)

50. Ren, X., Shen, T., Huang, J., Ling, H., Lu, Y., Nimier-David, M., Müller, T., Keller, A., Fidler, S., Gao, J.: Gen3c: 3d-informed world-consistent video generation with precise camera control. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2025)
51. Shi, R., Chen, H., Zhang, Z., Liu, M., Xu, C., Wei, X., Chen, L., Zeng, C., Su, H.: Zero123++: a single image to consistent multi-view diffusion base model (2023)
52. Su, J., Ahmed, M., Lu, Y., Pan, S., Bo, W., Liu, Y.: Roformer: Enhanced transformer with rotary position embedding. Neurocomputing **568**, 127063 (2024)
53. Sun, W., Chen, S., Liu, F., Chen, Z., Duan, Y., Zhang, J., Wang, Y.: Dimensionx: Create any 3d and 4d scenes from a single image with controllable video diffusion. arXiv preprint arXiv:2411.04928 (2024)
54. Tancik, M., Weber, E., Ng, E., Li, R., Yi, B., Kerr, J., Wang, T., Kristoffersen, A., Austin, J., Salahi, K., Ahuja, A., McAllister, D., Kanazawa, A.: Nerfstudio: A modular framework for neural radiance field development. In: ACM SIGGRAPH 2023 Conference Proceedings. SIGGRAPH '23 (2023)
55. Tang, J., Chen, Z., Chen, X., Wang, T., Zeng, G., Liu, Z.: Lgm: Large multi-view gaussian model for high-resolution 3d content creation. arXiv preprint arXiv:2402.05054 (2024)
56. Van Hoorick, B., Wu, R., Ozguroglu, E., Sargent, K., Liu, R., Tokmakov, P., Dave, A., Zheng, C., Vondrick, C.: Generative camera dolly: Extreme monocular dynamic novel view synthesis. European Conference on Computer Vision (ECCV) (2024)
57. Voleti, V., Yao, C.H., Boss, M., Letts, A., Pankratz, D., Tochilkin, D., Laforte, C., Rombach, R., Jampani, V.: Sv3d: Novel multi-view synthesis and 3d generation from a single image using latent video diffusion. In: European Conference on Computer Vision. pp. 439–457. Springer (2024)
58. 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)
59. Wang, C., Mirzaei, A., Goel, V., Menapace, W., Siarohin, A., Vinella, A., Vasilkovsky, M., Skorokhodov, I., Shakhrai, V., Korolev, S., et al.: 4real-video-v2: Fused view-time attention and feedforward reconstruction for 4d scene generation. NeurIPs (2025)
60. Wang, C., Zhuang, P., Ngo, T.D., Menapace, W., Siarohin, A., Vasilkovsky, M., Skorokhodov, I., Tulyakov, S., Wonka, P., Lee, H.Y.: 4real-video: Learning generalizable photo-realistic 4d video diffusion. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 17723–17732 (2025)
61. Wang, F.Y., Wu, X., Huang, Z., Shi, X., Shen, D., Song, G., Liu, Y., Li, H.: Be-your-outpainter: Mastering video outpainting through input-specific adaptation. In: European Conference on Computer Vision. pp. 153–168. Springer (2024)
62. Wang, Q., Wang, Z., Genova, K., Srinivasan, P., Zhou, H., Barron, J.T., Martin-Brualla, R., Snavely, N., Funkhouser, T.: Ibrnet: Learning multi-view image-based rendering. In: CVPR (2021)
63. Wang, T.C., Liu, M.Y., Tao, A., Liu, G., Kautz, J., Catanzaro, B.: Few-shot video-to-video synthesis. arXiv preprint arXiv:1910.12713 (2019)
64. Wang, T.C., Liu, M.Y., Zhu, J.Y., Liu, G., Tao, A., Kautz, J., Catanzaro, B.: Video-to-video synthesis. arXiv preprint arXiv:1808.06601 (2018)
65. 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: Burbano, A., Zorin, D., Jarosz, W. (eds.) SIGGRAPH (2024)

66. Wang, Z., Shen, T., Nimier-David, M., Sharp, N., Gao, J., Keller, A., Fidler, S., Müller, T., Gojcic, Z.: Adaptive shells for efficient neural radiance field rendering. ACM Trans. Graph. **42**(6) (2023). https://doi.org/10.1145/3618390, https://doi.org/10.1145/3618390
67. Watson, D., Saxena, S., Li, L., Tagliasacchi, A., Fleet, D.J.: Controlling space and time with diffusion models. arXiv preprint arXiv:2407.07860 (2024)
68. Wu, R., Gao, R., Poole, B., Trevithick, A., Zheng, C., Barron, J.T., Holynski, A.: Cat4d: Create anything in 4d with multi-view video diffusion models. arXiv preprint arXiv:2411.18613 (2024)
69. Wu, R., Mildenhall, B., Henzler, P., Park, K., Gao, R., Watson, D., Srinivasan, P.P., Verbin, D., Barron, J.T., Poole, B., Holynski, A.: ReconFusion: 3D reconstruction with diffusion priors. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 21551–21561 (2024)
70. Xiao, Y., Wang, Q., Zhang, S., Xue, N., Peng, S., Shen, Y., Zhou, X.: Spatialtracker: Tracking any 2d pixels in 3d space. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 20406–20417 (2024)
71. Xiao, Z., Ouyang, W., Zhou, Y., Yang, S., Yang, L., Si, J., Pan, X.: Trajectory attention for fine-grained video motion control. In: The Thirteenth International Conference on Learning Representations (2025), https://openreview.net/forum?id=2z1HT5lw5M
72. Xie, Y., Yao, C.H., Voleti, V., Jiang, H., Jampani, V.: Sv4d: Dynamic 3d content generation with multi-frame and multi-view consistency. arXiv preprint arXiv:2407.17470 (2024)
73. 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)
74. Xu, Y., Park, T., Zhang, R., Zhou, Y., Shechtman, E., Liu, F., Huang, J.B., Liu, D.: Videogigagan: Towards detail-rich video super-resolution. arXiv preprint arXiv:2404.12388 (2024)
75. Yang, J., Ivanovic, B., Litany, O., Weng, X., Kim, S.W., Li, B., Che, T., Xu, D., Fidler, S., Pavone, M., Wang, Y.: Emernerf: Emergent spatial-temporal scene decomposition via self-supervision. arXiv preprint arXiv:2311.02077 (2023)
76. Yang, S., Hou, L., Huang, H., Ma, C., Wan, P., Zhang, D., Chen, X., Liao, J.: Direct-a-video: Customized video generation with user-directed camera movement and object motion. In: Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers '24 (SIGGRAPH Conference Papers '24). p. 12. ACM, New York, NY, USA (2024). https://doi.org/10.1145/3641519.3657481
77. Yang, S., Hou, L., Huang, H., Ma, C., Wan, P., Zhang, D., Chen, X., Liao, J.: Direct-a-video: Customized video generation with user-directed camera movement and object motion. In: ACM SIGGRAPH 2024 Conference Papers. pp. 1–12 (2024)
78. Yang, S., Zhou, Y., Liu, Z., Loy, C.C.: Rerender a video: Zero-shot text-guided video-to-video translation. In: SIGGRAPH Asia 2023 Conference Papers. pp. 1–11 (2023)
79. 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)
80. You, M., Zhu, Z., Liu, H., Hou, J.: NVS-Solver: video diffusion model as zero-shot novel view synthesizer. CoRR **abs/2405.15364** (2024)
81. Yu, A., Ye, V., Tancik, M., Kanazawa, A.: pixelNeRF: Neural radiance fields from one or few images. In: CVPR (2021)

82. YU, M., Hu, W., Xing, J., Shan, Y.: Trajectorycrafter: Redirecting camera trajectory for monocular videos via diffusion models (2025), https://arxiv.org/abs/2503.05638
83. Yu, W., Xing, J., Yuan, L., Hu, W., Li, X., Huang, Z., Gao, X., Wong, T.T., Shan, Y., Tian, Y.: ViewCrafter: taming video diffusion models for high-fidelity novel view synthesis. arXiv preprint arXiv:2409.02048 (2024)
84. Yu, Z., Sattler, T., Geiger, A.: Gaussian opacity fields: Efficient adaptive surface reconstruction in unbounded scenes. ACM Transactions on Graphics (2024)
85. Zhang, D.J., Paiss, R., Zada, S., Karnad, N., Jacobs, D.E., Pritch, Y., Mosseri, I., Shou, M.Z., Wadhwa, N., Ruiz, N.: Recapture: Generative video camera controls for user-provided videos using masked video fine-tuning. arXiv preprint arXiv:2411.05003 (2024)
86. Zhang, J., Herrmann, C., Hur, J., Jampani, V., Darrell, T., Cole, F., Sun, D., Yang, M.H.: Monst3r: A simple approach for estimating geometry in the presence of motion. arXiv preprint arXiv:2410.03825 (2024)
87. Zhao, Y., Lin, C.C., Lin, K., Yan, Z., Li, L., Yang, Z., Wang, J., Lee, G.H., Wang, L.: Genxd: Generating any 3d and 4d scenes. arXiv preprint arXiv:2411.02319 (2024)
88. Zheng, G., Li, T., Jiang, R., Lu, Y., Wu, T., Li, X.: Cami2v: Camera-controlled image-to-video diffusion model. arXiv preprint arXiv:2410.15957 (2024)
89. Zheng, S., Peng, Z., Zhou, Y., Zhu, Y., Xu, H., Huang, X., Fu, Y.: Vidcraft3: Camera, object, and lighting control for image-to-video generation. arXiv preprint arXiv:2502.07531 (2025)
90. Zhou, J., Gao, H., Voleti, V., Vasishta, A., Yao, C.H., Boss, M., Torr, P., Rupprecht, C., Jampani, V.: Stable virtual camera: Generative view synthesis with diffusion models. arXiv preprint arXiv:2503.14489 (2025)
91. Zhou, K., Li, W., Wang, Y., Hu, T., Jiang, N., Han, X., Lu, J.: NeRFLix: high-quality neural view synthesis by learning a degradation-driven inter-viewpoint mixer. In: CVPR. pp. 12363–12374 (2023)
92. Zhou, S., Yang, P., Wang, J., Luo, Y., Loy, C.C.: Upscale-a-video: Temporal-consistent diffusion model for real-world video super-resolution. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2535–2545 (2024)
93. Zhou, T., Tucker, R., Flynn, J., Fyffe, G., Snavely, N.: Stereo magnification: Learning view synthesis using multiplane images. arXiv preprint arXiv:1805.09817 (2018)