Scaling4D: Pushing the Frontier of Video Novel View Synthesis through Large-Scale Monocular Videos

Hongrui Cai Junjie Luo Zhihong Fu Shengnan Zhu
Jiawei Wen Wanquan Feng* Songtao Zhao Qian He
Intelligent Creation Team, ByteDance

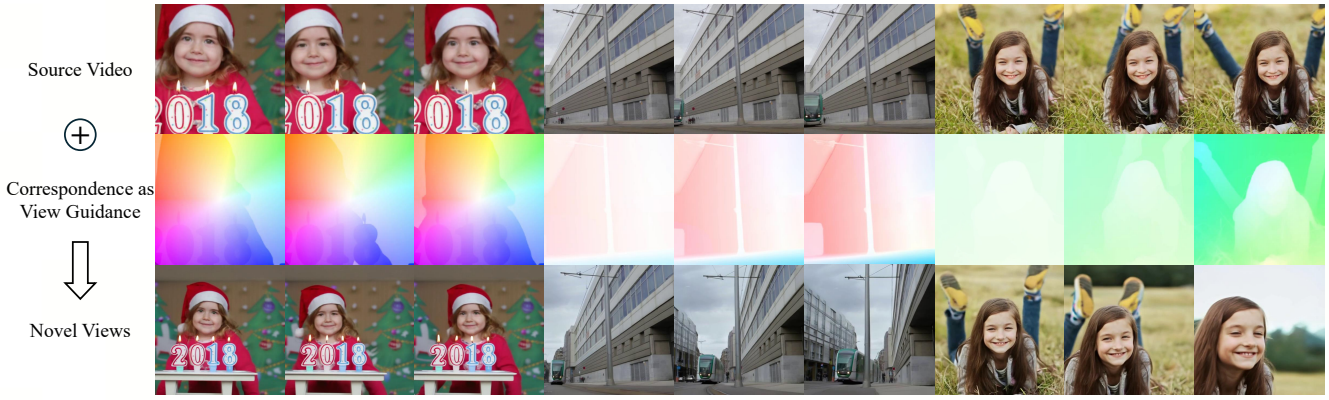


Figure 1. We introduce **Scaling4D**, a framework for Video Novel View Synthesis (VNVS). Given a monocular source video (first row) and novel poses (second row), Scaling4D generates the novel view video (third row). By reformulating VNVS as a correspondence-guided generation task, our approach bridges the training-inference gap present in previous methods and enables scalable training on large-scale monocular videos, substantially improving the visual quality and robustness of the synthesized results.

Abstract

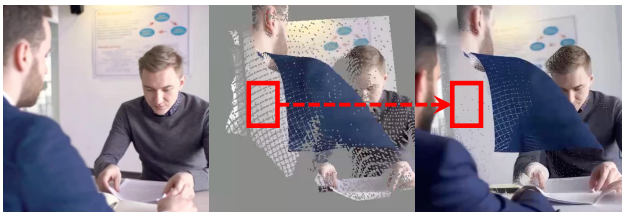
*Video Novel View Synthesis (VNVS) aims to render arbitrary novel viewpoints of dynamic scenes from a single-view video, but its algorithmic training faces a major challenge: the lack of large-scale multi-view video datasets. Prior methods often train on monocular data by framing it as an inpainting task, which typically leads to a training-inference gap and visual artifacts. While synthetic multi-view data can partially alleviate the data scarcity issue, its high acquisition costs and limited diversity restrict scalability. To address these problems, we propose **Scaling4D**, a novel strategy that theoretically bridges the training-inference gap while leveraging large-scale monocular videos for training. Specifically, we take a higher-level perspective on the problem, reformulating VNVS into a general correspondence-guided generation task. Furthermore, in conjunction with extensive real-world data, we establish a synthetic data pipeline integrated with our training strategy to enhance precision. Qualitative and quantitative results demonstrate a positive correlation between performance and training data volume, confirming the scalability.*

*Corresponding author

1. Introduction

In this work, we explore the task of Video Novel View Synthesis (VNVS) [2, 39, 47, 52], which aims to render arbitrary novel camera views of a 4D (dynamic 3D) scene given a single monocular video as input. Multi-view representations [23, 32, 37] have long been recognized as fundamental for understanding spatial environments. Consequently, the ability to synthesize multi-view videos from limited input unlocks vast potential for diverse domains including fundamental research (e.g., 4D perception tasks [7, 28, 49]) and entertainment (e.g., intelligent creation [2, 12, 13, 27, 38]).

Considering the sparsity of single-view input, video novel view synthesis from a single view is an ill-posed problem. Reconstruction and optimization based methods, such as 4D Gaussian Splatting [47] and Dynamic NeRF [37], are not suitable for this setting. Even some recent feed-forward Gaussian methods [9, 50] generally require sparse multi-view inputs rather than a single view, and their extrapolation ability is limited. Fortunately, recent advances in video generation models [15, 25, 45] enable us to leverage their prior knowledge to assist in building a robust VNVS framework.



Input View Point Map in Novel View Inpainting Result

Figure 2. This example demonstrates the essential difference between inpainting and novel view synthesis. In the novel view, the area inside the red box should show the back of the person, but the inpainting result incorrectly fills it with background pixels.

Despite the availability of priors from video generation models, building an effective training pipeline remains challenging due to the **lack of sufficient multi-view data**. Previous approaches have adopted two main strategies to address this issue. **(1) Some previous works [1, 2] train exclusively on synthetic multi-view data.** The advantage of synthetic data is that it is fully controllable and highly accurate; arbitrary viewpoints at any time can be freely obtained, which is nearly impossible with real-world data. However, the diversity of synthetic data is limited, as the available assets and scenarios (e.g., Fab [10]) are constrained. **(2) Some other previous works [39, 52] reformulate the task as a video inpainting task**. Specifically, this strategy predicts the depth of the video and converts it into a dynamic point map, which is rendered to the target view. Then, a video inpainting model inpaints the sparse rendering to produce the final video. This approach suffers from a training-inference gap, as illustrated by Fig. 2, which demonstrates the essential difference between inpainting and novel view synthesis.

The above analysis leads to a crucial question: can we leverage large-scale real monocular data for VNVS training while effectively bridging the traditional training-inference gap? To achieve this, we need to identify a bridge that fulfills two requirements:

- **It can serve as a control condition for VNVS.**
- **It naturally exists in abundant real monocular data.**

With this perspective, we identify an appropriate answer for the above question: **pixel correspondence**. By rendering the point map of the input video to a novel view, pixel correspondence between the source and target viewpoints can be established as shown in Fig. 1. Furthermore, in any monocular training video, optical flow can enable correspondence between frames. Therefore, using correspondence as the control condition ensures that all inference scenarios are well-covered subsets of the diverse correspondence situations seen during training. This naturally includes challenging cases like significant viewpoint changes (e.g., in Fig. 2), as such camera motions are abundant in the training data.

To complement the extensive real-world data, we also develop a synthetic data pipeline that is seamlessly integrated into our training strategy to further enhance the precision and robustness of our framework. We utilize the cinematic-quality Procedural Dependency Graph (PDG) tool, Houdini [40], to construct the data pipeline. It includes scene layout creation, random composition of assets (human characters, objects and lighting), and the generation of randomized camera trajectories. Our synthetic data satisfies several key characteristics, including **high fidelity and diversity, accurate inter-frame correspondence, and a wide range of camera motions**.

For the network architecture, we design two core components: Correspondence Projector and VNVS Block. The former takes the source video and the correspondence as input, projecting them into control tokens. The latter is a multimodal module that facilitates cross-modal interaction between the video latent and the control tokens via spatial attention. Notably, our designed components are explicitly decoupled from text, thereby preserving the original prompt-following capability of the foundation model. Furthermore, we omit MLP layers in the block design, which ensures computational efficiency.

Overall, our contributions are summarized as follows.

Paradigm. We propose Scaling4D, which takes a high-level perspective on the VNVS problem, reformulating it into a general correspondence-guided generation task.

Technology. We enable scalable training on large-scale monocular real videos, develop a robust synthetic data pipeline as a supplement to real-world data, and design an appropriate network architecture to support our framework.

Significance. Scaling4D outperforms previous works in both quantitative and qualitative results.

Scalability. Our framework exhibits scalability as the data volume increases, indicating its promising potential.

2. Related Work

2.1. Video Generation

Foundation Model. Since the adoption of diffusion [19, 42] and flow matching [29] techniques in video generation, the quality of generated videos has been rapidly improving. Several commercial-grade models—including Kling [25], Dreamina [4], Veo3 [17], and Sora [33]—have demonstrated impressive performance. Early open-source models, such as AnimateDiff [18] and SVD [3], adopted UNet architectures with spatial attention and additional temporal modules. More recent models, such as Wan [45] and Hunyuan Video [24], have increasingly employed DiT [36] architectures, replacing convolutional structures with fully transformer-based designs. At present, the architectural design of video generation models has become increasingly unified, which has also inspired the design of our network architecture. The rich priors in video generation models play an important role in video novel view synthesis.

Controllable Generation. Controllability is an important

topic in video generation, allowing us to specify conditions beyond text and the initial frame. Using spatial motion as a condition [12, 13] enables the generated results to follow a specified motion trajectory. Employing a reference portrait image as a control [22, 26] allows for personalized character generation. In this work, we utilize the input video and correspondence guidance as control signals for generation.

2.2. 4D Scene Representation

Representing dynamic 4D scenes [5, 6, 11, 34, 37, 47, 48] is a long-standing challenge in computer vision and graphics. Traditional explicit representations, such as animated triangle meshes [48], often require significant manual effort. In recent years, implicit neural representations have gained significant traction. Dynamic Neural Radiance Fields (NeRF) [34, 35, 37] model a scene as a continuous function of space and time, enabling high-fidelity reconstruction of complex, non-rigid motion. Similarly, 4D Gaussian Splatting (4D-GS) [41, 47] extends its 3D counterpart by modeling the temporal evolution of 3D Gaussians, achieving real-time rendering of dynamic scenes. NeRF and Gaussian-based methods achieve excellent optimization results based on multi-view inputs. The reliance of these powerful methods on multi-view inputs underscores the importance of our task, which can generate multi-view videos from single-view videos.

2.3. Generative Video Novel View Synthesis

Rather than relying on reconstruction-based strategies [37, 47], some recent video novel view synthesis methods [2, 16, 39, 52] employ a generative approach, which can synthesize novel views from a limited single input view by leveraging the powerful priors embedded in foundation models [25, 45]. TrajectoryCrafter [52] and GEN3C [39] render point maps in novel views and train the video model on an inpainting task. ReCamMaster [2] employs the implicit geometry representation as control and trains with synthetic data. We rethink the task from a new perspective, reformulating it into a general correspondence guided task.

3. Method

3.1. Preliminary: Point Map Guided VNVS

Recently, several methods [39, 52] have achieved notable success in VNVS task by leveraging explicit 3D geometry to guide powerful pre-trained video diffusion models. This paradigm effectively decouples the problem into two stages: first, establishing a geometrically-grounded control signal, and second, using a conditional diffusion model to render a high-fidelity video based on this signal. The process typically begins with a source view video denoted as $\mathbf{I}^s = \{\mathbf{I}_i^s\}_{i=1}^n \in \mathbb{R}^{n\times 3\times h\times w}$, where n is the number of frames, h and w represent the height and width of each frame, and the superscript s refers to "`source`". A monocular depth estimator [20, 51] is employed to predict a corresponding depth map $\mathbf{D}^s = \{\mathbf{D}_i^s\}_{i=1}^n \in \mathbb{R}^{n\times h\times w}$. This RGB-D information is then unprojected into a 3D point map $\mathcal{P} = \{\mathcal{P}_i\}_{i=1}^n \in \mathbb{R}^{n\times 6\times h\times w}$ with 3D spatial coordinate and RGB color using camera intrinsics $\mathbf{K} \in \mathbb{R}^{3\times 3}$:

$$\mathcal{P}_i = \Phi^{-1}([\mathbf{I}_i^s, \mathbf{D}_i^s], \mathbf{K}), \forall i \in [1, n] \tag{1}$$

where Φ^{-1} denotes the inverse perspective projection. This point map serves as an explicit representation of the scene geometry. Given a target camera trajectory defined by a sequence of camera poses $\mathbf{T}^r = \{\mathbf{T}_i^r\}_{i=1}^n \in \mathbb{R}^{n\times 4\times 4}$, where the superscript r refers to "`rendered`", this point map $\mathcal{P}$ is then rendered into the target novel view:

$$\mathbf{I}_i^r = \Phi(\mathbf{T}_i^r \mathcal{P}_i, \mathbf{K}), \tag{2}$$

where Φ denotes the perspective projection. Further, there is a render mask video $\mathbf{M}^r = \{\mathbf{M}_i^r\}_{i=1}^n \in \mathbb{R}^{n\times h\times w}$ indicating which pixels are rendered. Finally, the pair $(\mathbf{I}^r, \mathbf{M}^r) \in \mathbb{R}^{n\times 4\times h\times w}$ acts as the control signal to generate the novel view video. Theoretically, this process can be regarded as an inpainting problem, where the regions indicated by $\mathbf{M}^r$ are preserved from $\mathbf{I}^r$, and the complementary regions need to be filled in by the video generation model.

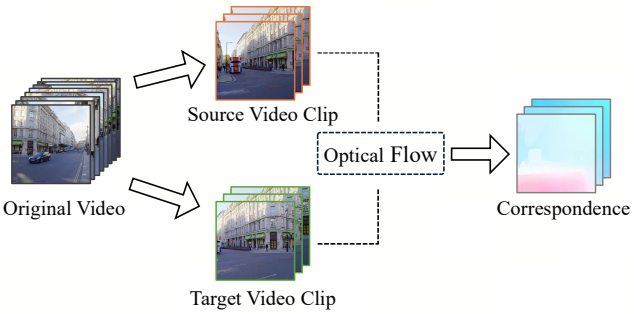


Figure 3. The process of extracting $\mathbf{I}^s$, $\mathbf{I}^t$ and $\mathbf{C}^r$ from raw video.

3.2. Paradigm Upgrade: Correspondence Control

As analyzed in Fig. 2, the method reviewed in Sec. 3.1 may result in a training-inference gap. How can we fundamentally overcome this gap? If multi-view videos are available, we can obtain ground truth video, thereby removing the gap. Let us use logical symbols to represent the entire framework:

$$\mathbf{I}^s \xrightarrow{\Phi^{-1}} \mathcal{P} \xrightarrow{\mathbf{T}^r} \mathbf{T}^r\mathcal{P} \xrightarrow{\Phi} \mathbf{I}^r \xrightarrow{G_\theta} \mathbf{I}^* \Leftarrow \mathbf{I}^t, \tag{3}$$

where G_θ denotes the video generation model, $\mathbf{I}^*$ is the generated result, $\mathbf{I}^t$ represents the ground truth video (the superscript t refers to "`target`"), $\Leftarrow$ means the process of training with ordinary flow matching loss.

However, the above training paradigm does not work for large-scale real-world videos, as ground truth $\mathbf{I}^t$ from target

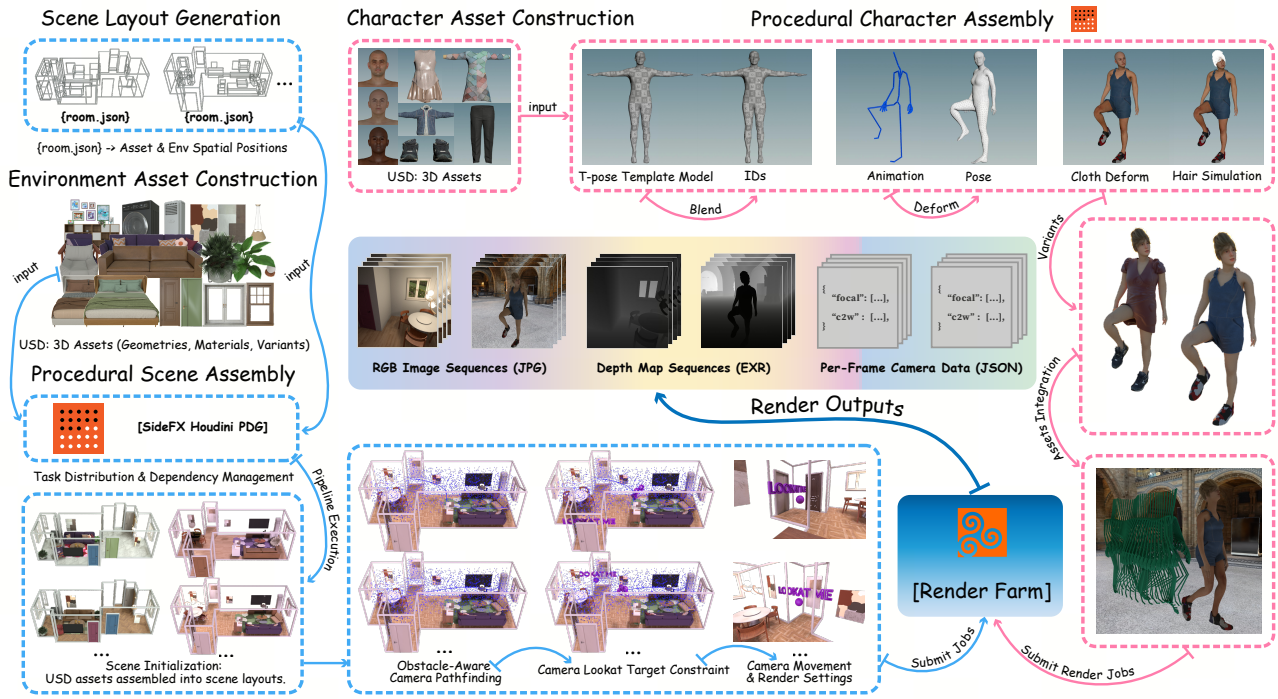


Figure 4. Overview of our synthetic data pipeline, including compositional scene layouts, procedural character generation, obstacle-aware camera trajectories, and storage of RGB images, depth maps, and camera matrices.

viewpoint is unavailable. Furthermore, as previously discussed, no matter how much synthetic data we generate, it cannot match the scale of real-world data. Therefore, it is necessary to upgrade the paradigm and develop a framework that differs from the one above, enabling scalable training on large-scale in-the-wild monocular video data.

Fortunately, we have identified a very simple solution. By revisiting Eq. 3, we observe that the composite operation of Φ^{-1}, $\mathbf{T}^r$, and Φ essentially defines a correspondence on the 2D image plane, which we denote as $\mathbf{C}^r = \{\mathbf{C}_i^r\}_{i=1}^n \in \mathbb{R}^{n\times 2\times h\times w}$. This correspondence is mathematically equivalent to the combination of the three operations:

$$\mathbf{C}^r \Longleftrightarrow \Phi \circ \mathbf{T}^r \circ \Phi^{-1}. \tag{4}$$

Substituting Eq. 4 into Eq. 3, we have:

$$\mathbf{I}^s \xrightarrow{\mathbf{C}^r} \mathbf{I}^r \xrightarrow{G_\theta} \mathbf{I}^* \Leftarrow \mathbf{I}^t. \tag{5}$$

Note that the process from $\mathbf{I}^s$ to $\mathbf{I}^r$ is actually a warping operation rather than rendering. However, for the sake of convenience, we still use the superscript r and denote the warped video as $\mathbf{I}^r$. At this point in the derivation, we make an exciting observation: for any in-the-wild monocular video, we can arbitrarily select two clips as $\mathbf{I}^s$ and $\mathbf{I}^t$, respectively. By applying an optical flow model (e.g., RAFT [43]), we can obtain the correspondence $\mathbf{C}^r$ between them, as shown in Fig. 3. In this way, the framework described in Eq. 5 forms a complete training loop, allowing us to leverage large-scale in-the-wild monocular video data for training. It is worth noting that when performing warping with $\mathbf{C}^r$, there might be cases where multiple source pixels correspond to the same target pixel. In the training phase, we randomly select one among these source pixels. In the inference phase, we employ a z-buffer (since we can obtain the point map) to select the pixel with the smallest depth.

The control signal is set as $(\mathbf{I}^s, \mathbf{C}^r, \mathbf{I}^r, \mathbf{M}^r) \in \mathbb{R}^{n\times 9\times h\times w}$. With this design, we complete the paradigm upgrade. Our framework is now capable of scaling to large-scale real-world monocular data, enabling scalable training by unifying training and inference within a shared correspondence space.

3.3. Synthetic Data Pipeline

Although our framework can utilize real-world data, synthetic data remains important. Real videos often suffer from limited correspondence accuracy. To address this, we build a synthetic data pipeline for precise correspondence, while also working to mitigate diversity limitations.

For synthetic scenes, we can render $\mathbf{I}^s$ and $\mathbf{I}^t$, and record $\mathcal{P}$ and $\mathbf{T}^r$. This allows direct computation of $\mathbf{C}^r$ (as described in Eq. 4) and straightforward application of the training framework (as described in Eq. 5).

We utilize the cinematic-quality Procedural Dependency Graph (PDG, a procedural architecture designed to distribute tasks and manage dependencies, enabling better scalability, automation, and analysis for content pipelines) tool, Houdini [40], to construct the data pipeline (as shown in Fig. 4.). All of our assets are constructed in USD [44] format, which stores all 3D contents (geometries, materials, animations), leveraging its variant mechanism to equip each

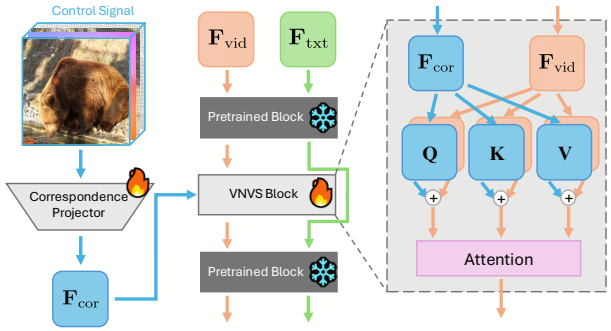


Figure 5. Network structure of our method. The trainable modules include the Correspondence Projector and the VNVS Block.

asset with multiple appearances—randomness via variant index selection, extensibility via custom variant additions.

Compositional scenes. We collect a diverse set of room layouts, each containing numerous 3D bounding boxes, following the format of SpatialLM [31]. For each asset category, we create multiple variants to increase scene diversity.

Procedural Characters. We collect a wide range of character models with diverse body shapes, clothing, hairstyles, and motion sequences. To enhance diversity, we also employ randomized combinations of these attributes.

Camera Trajectories. We generate diverse camera trajectories by first sampling points in the 3D space and then connecting selected points with smooth curves to form the camera paths. All trajectories are strictly constrained to avoid obstacles within the scene. Additionally, we support the specification of target points of interest, ensuring they remain within the camera's view throughout the trajectory, which further enhances the realism of the generated paths.

Data Storage. For each video, we store RGB image sequences, depth maps, and per-frame camera matrices.

3.4. Network Structure

For the network structure, we introduce two key components (see Fig. 5): Correspondence Projector and VNVS Block. Our proposed architecture is theoretically compatible with any foundation model, allowing for flexible integration across different generative frameworks. Specifically, we utilize an MMDiT model, which employs a 3D VAE to encode videos into the latent space and leverages a LLM to encode text inputs. These two modalities are jointly processed and interact within the MMDiT blocks, enabling effective cross-modal fusion over the generation process.

Correspondence Projector. As described in Sec. 3.2, the control signal is a 9-channel tensor $(\mathbf{I}^s, \mathbf{C}^r, \mathbf{I}^r, \mathbf{M}^r)$. We employ a series of convolutional layers followed by a patchifying layer to encode the input tensor into control tokens, denoted as $\mathbf{F}_{\text{cor}}$, whose shape matches the latent token shape used in the original DiT architecture.

VNVS Block. The original foundation model lacks explicit mechanisms for incorporating control tokens $\mathbf{F}_{\text{cor}}$; information exchange between blocks is restricted to text features $\mathbf{F}_{\text{txt}}$ and video features $\mathbf{F}_{\text{vid}}$. To address this limitation, we introduce VNVS Blocks between the pretrained blocks of the foundation model. Each VNVS Block accepts both $\mathbf{F}_{\text{cor}}$ and $\mathbf{F}_{\text{vid}}$ as inputs, and outputs the updated video features $\mathbf{F}_{\text{vid}}$ only, while keeping the text features unchanged (explicitly decoupled from text tokens, preserving the prompt-following capability). Specifically, the attention mechanism within the VNVS Block is formulated as:

$$\mathbf{F}_{\text{vid}} \leftarrow \mathbf{F}_{\text{vid}} + \text{Attn}(\mathbf{Q}_{\text{vid}} + \mathbf{Q}_{\text{cor}},\ \mathbf{K}_{\text{vid}} + \mathbf{K}_{\text{cor}},\ \mathbf{V}_{\text{vid}} + \mathbf{V}_{\text{cor}}) \tag{6}$$

where $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$ denote the query, key, and value projections of the video and correspondence features, respectively. The updated video features $\mathbf{F}_{\text{vid}}$ are then seamlessly propagated to the subsequent pretrained blocks, thereby maintaining compatibility with the original model design.

We apply summation over the query, key, and value projections of the video and control tokens to enable joint attention across both modalities. This is feasible because the control tokens are spatially aligned with the video tokens.

Unlike usual transformer blocks, we omit the feed-forward network (FFN) layers in our design. This choice not only reduces computational overhead, but is also supported by prior works [53], which have demonstrated that FFN layers primarily capture long-term memory, whereas transient contextual memory is mainly handled by attention layers. We argue that controllability is inherently a form of transient contextual memory, and our experiments further verify the effectiveness of this architectural design.

4. Experiments

4.1. Experiment Settings

Implementation Details. We train with 49 frames, resolution 480×480 for each video. During training, we utilize the pre-trained optical flow estimator RAFT [43] to compute dense correspondences from sequential video frames. During inference, we estimate depth maps of source videos using GeometryCrafter [51]. The training process is performed on 64 A100 GPUs, with a world batch size of 256. We use AdamW [30] with a learning rate of 4×10^{-5}.

Datasets. For **real-world training data**, we use the SpatialVID [46] dataset, which contains approximately 3M monocular videos with rich camera movement. For **synthetic training data**, we generate 10,000 training samples using our pipeline. We evaluate the model performance on both single-view and multi-view video benchmarks. For the **single-view testing set**, we randomly extract 400 videos from Panda-70M [8] dataset. For **multi-view testing set**, we follow previous works [52] and use the iPhone dataset [14], pixel-wisely evaluating the quality of the synthesized videos with the ground truth target video.

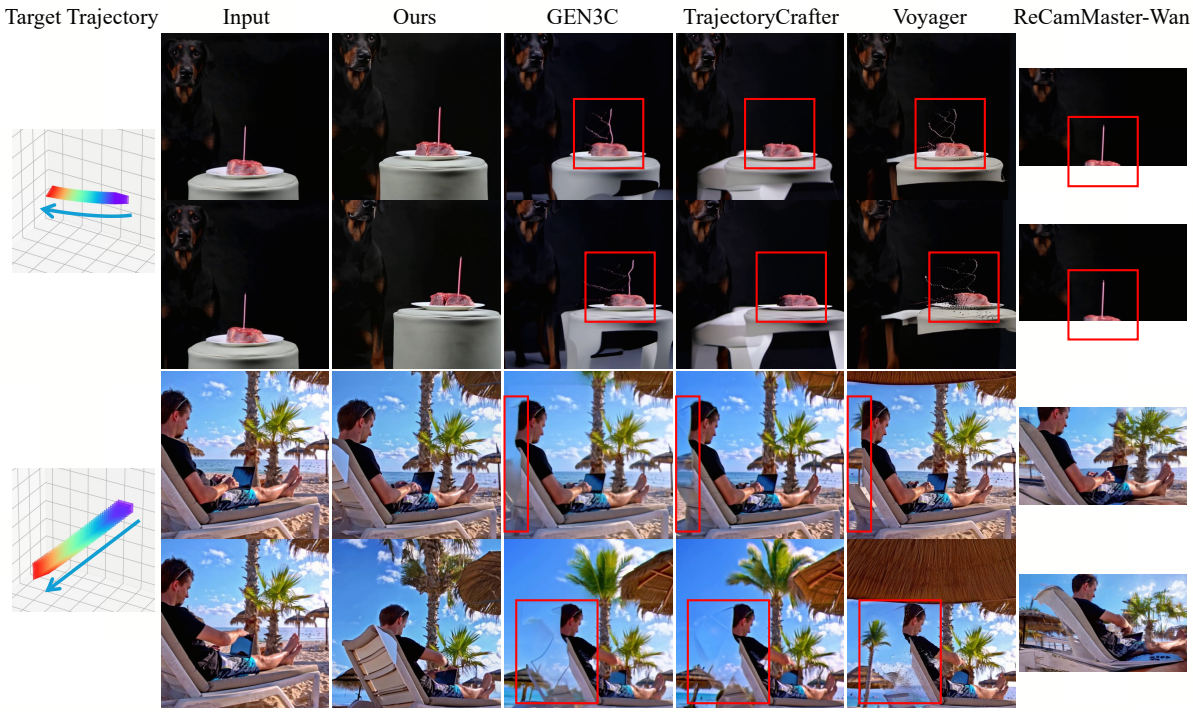


Figure 6. The qualitative results on the single-view dataset. From left to right, we display target camera trajectory, input images, results of ours, GEN3C [39], TrajectoryCrafter [52], Voyager [21] and ReCamMaster-Wan [2], respectively.

Method	FID ↓	FVD ↓	CLIP-T ↑	CLIP-F ↑	CLIP-V ↑	RotErr ↓	TransErr ↓
GEN3C [39]	69.35	442.70	27.43	98.49	90.14	6.98	299.68
TrajectoryCrafter [52]	68.94	425.20	27.35	98.37	90.41	6.65	320.38
Voyager [21]	68.41	414.89	27.38	98.11	91.07	7.04	347.57
ReCamMaster-Wan [2]	83.71	635.86	27.14	98.37	86.21	12.29	737.71
Ours	**62.83**	**411.17**	**27.79**	**98.55**	**91.81**	**6.48**	**286.77**

Table 1. The quantitative results on the single-view dataset. The best results are highlighted in **bold**.

Evaluation Metrics. For rigorous and comprehensive evaluation, we adopt a suite of metrics from prior works to assess various aspects of generation quality. Following [2], we report **FID**, **FVD**, **CLIP-T**, **CLIP-F**, **CLIP-V**, **RotErr**, and **TransErr**. Additionally, following [52], we include **PSNR**, **SSIM**, and **LPIPS**. These metrics enable effective measurement of both control accuracy and visual quality.

Comparison baselines. For a comprehensive evaluation, we compare our method against several state-of-the-art approaches: **GEN3C** [39], **TrajectoryCrafter** [52], **Voyager** [21], and **ReCamMaster** [2]. To ensure a fair comparison, we re-implement GEN3C [39], TrajectoryCrafter [52], and Voyager [21] within our framework, using the same foundation model for all methods. For ReCamMaster, we failed to successfully train it on our foundation model, which might be due to incompatibilities between the model architectures. Therefore, we report results using their official pre-trained model built upon Wan [45], referred to as ReCamMaster-Wan. To accommodate its requirements, we adjust the input videos to a resolution of 480×832.

TrajectoryCrafter [52] employs a double reprojection strategy that reframes novel view synthesis as a strict inpainting problem. As a result, it struggles to handle occlusions and often fails to accurately inpaint foreground objects over the background. GEN3C [39] and Voyager [21] construct training data based on inter-frame camera alignment; however, real-world videos typically contain non-rigid motions, which leads to even more pronounced artifacts. ReCamMaster-Wan [2] utilizes implicit geometry representation (camera matrix) as control signals, but it is relatively coarse and the generalization across diverse camera trajectories remains questionable. We show the detailed results and comparisons in the following sections.

4.2. Evaluation on Single-view Video Dataset

For fair comparison, all methods are evaluated using identical, randomly sampled camera trajectories. Quantitatively (Tab. 1), our method achieves state-of-the-art performance across all metrics, validating its superior visual quality and control accuracy. In the first case of Fig. 6, GEN3C, Voy-

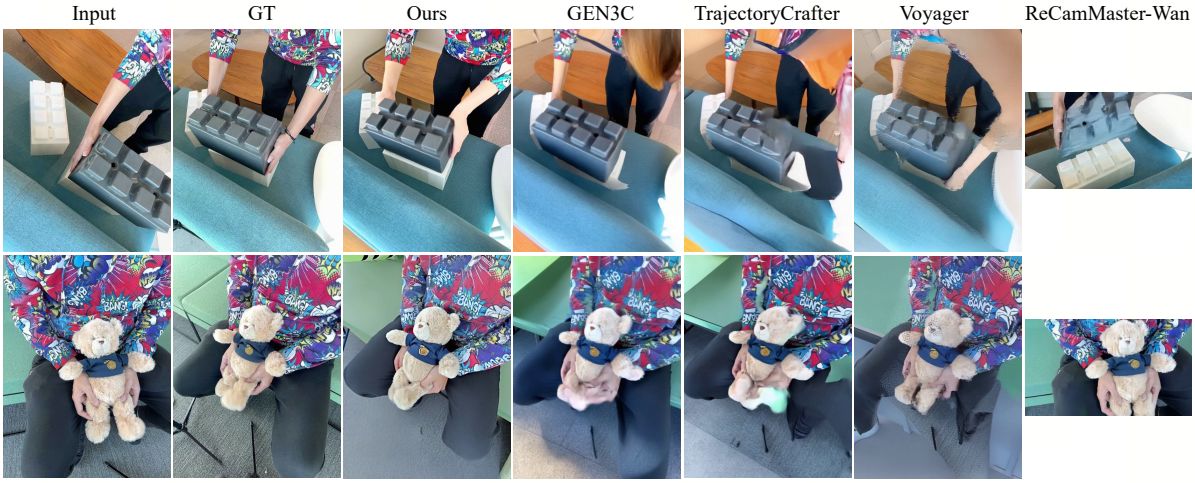


Figure 7. The qualitative results on the multi-view dataset. From left to right, we display ground truth images, results of ours, GEN3C [39], TrajectoryCrafter [52], Voyager [21] and ReCamMaster-Wan [2], respectively.

Method	PSNR ↑	SSIM ↑	LPIPS ↓
GEN3C [39]	14.09	0.304	0.531
TrajectoryCrafter [52]	14.13	0.309	0.539
Voyager [21]	14.03	0.303	0.513
ReCamMaster-Wan [2]	10.25	0.318	0.709
Ours	**14.85**	**0.336**	**0.468**

Table 2. The quantitative results on the multi-view dataset.

ager, and TrajectoryCrafter all exhibit significant visual artifacts with the candle and the cake in the scene. Specifically, the candle either disappears entirely or is corrupted by severe noise and artifacts, while noticeable holes appear beneath the cake. These failures are primarily due to their point cloud-based inpainting paradigms, which struggle to handle occlusions and reconstruct fine foreground details. ReCamMaster-Wan does not exhibit obvious visual artifacts such as missing candles or holes beneath the cake. However, its generated camera trajectory is almost static, indicating poor generalization to novel camera motions and limited control over viewpoint changes. In the second case of Fig. 6, GEN3C, Voyager, and TrajectoryCrafter all fail to complete the back of the chair. In contrast, our method not only generates stable and complete view transformations that are temporally coherent with the input dynamic video but also demonstrates highly precise camera control. This shows that our approach effectively satisfies the spatio-temporal consistency crucial for 4D generation tasks.

4.3. Evaluation on Multi-view Video Dataset

On the multi-view iPhone [14] dataset, our method establishes clear quantitative superiority (Tab. 2). Visually (Fig. 7), GEN3C and Voyager produce noticeable visual artifacts, while TrajectoryCrafter fails to inpaint occluded backgrounds. Moreover, ReCamMaster-Wan fails to handle camera cuts and requires the initial camera pose of the target trajectory to be aligned with the source frame. Since the

Figure 8. Visual comparison between flow-based and depth-based correspondence on a static scene. The strong alignment effectively bridges the training-inference gap.

first frame of target trajectory in this test set is not aligned with the source frame, ReCamMaster-Wan is unable to process these cases well, resulting in complete failure on this dataset. In contrast, our method accurately follows the camera trajectory and delivers high visual quality in the results.

4.4. Analysis of the Training-Inference Gap

To validate that Scaling4D effectively bridges the training-inference gap, we analyze the alignment between the flow-based training correspondence C^r_{flow} and the depth-based inference correspondence C^r_{depth} across 100 static videos. As shown in Fig. 8, they align remarkably well (EPE $\downarrow = 1.18$ px; vec-corr $\uparrow = 0.986$). Although highly correlated, C^r_{flow} exhibits slightly rougher spatiotemporal statistics than C^r_{depth}: $\text{TV}_1(C) = \mathbb{E}[\|\nabla u_t\|_1 + \|\nabla v_t\|_1]$: 0.151 vs. 0.146; $\text{LapE}(C) = \mathbb{E}[\|\Delta u_t\|_2^2 + \|\Delta v_t\|_2^2]$: 0.213 vs. 0.082; $\text{Temp}(C) = \mathbb{E}[\|C_t - C_{t-1}\|_2]$: 2.47 vs. 2.43. Consequently, training with the inherently rougher C^r_{flow} forces the model to learn noise-resistant mappings, enhancing its robustness when evaluated on C^r_{depth} during inference.

4.5. Ablation Study

We conduct ablation studies on several variants against our full model in Tab. 3. In the first row (Ours + DoubleProj), we train our model with additional double reprojection data. In the second row (Ours w/o RealData), the model is trained solely on synthetic data, excluding any real data. In the third row (Ours w/o SynData), the model is trained only on real

Method	FID ↓	FVD ↓	CLIP-T ↑	CLIP-F ↑	CLIP-V ↑	RotErr ↓	TransErr ↓
Ours + DoubleProj	65.17	439.77	27.64	98.51	91.66	**6.27**	**282.85**
Ours w/o RealData	64.26	472.62	27.61	98.48	91.20	6.79	318.61
Ours w/o SynData	63.61	**409.16**	**27.84**	**98.59**	**91.84**	6.58	308.92
Ours	**62.83**	411.17	27.79	98.55	91.81	6.48	286.77

Table 3. Ablation study on the single-view dataset. Results on different variants of our method: training with additional double reprojection data (Ours + DoubleProj), training solely on synthetic data (Ours w/o RealData) and training exclusively on real data (Ours w/o SynData).

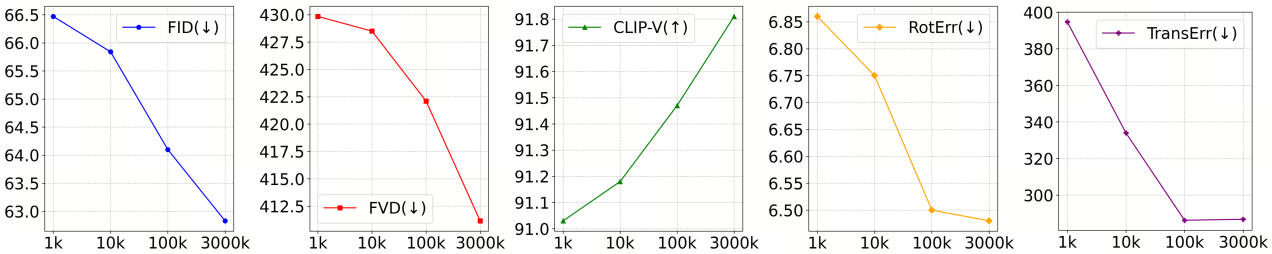


Figure 9. Scalability analysis of different metrics with respect to the training data volume. The x-axis represents the amount of training data, while the y-axis shows the corresponding metric scores on the testing set.

data, without any synthetic data. Based on the results in Tab. 3, we highlight some key findings here.

Impact of Double Reprojection Data. The inclusion of additional double reprojection data (Ours + DoubleProj) leads to improved pose precision (lower RotError and TransError), as this data inherently lacks correspondence errors. However, the visual quality is degraded, which is consistent with the analysis of inpainting-based methods.

Impact of Real Data. When the model is trained without real data (Ours w/o RealData), there is a significant degradation across most metrics, particularly in visual quality. Synthetic data alone, despite its feasibility for training (as in ReCamMaster [2]), cannot fully compensate for the richness and diversity provided by real-world observations.

Impact of Synthetic Data. When we remove synthetic data (Ours w/o SynData), we observe a decline in camera pose precision, indicating that synthetic data has the potential to enhance control precision. Moreover, while the absence of synthetic data leads to a slight improvement in visual quality metrics, the numerical differences are not substantial, which further underscores the robust capability and diversity of our synthetic data generation pipeline.

4.6. Scalability with Data Volume

We further explore the scalability of data volume, as illustrated in Fig. 9. All experiments share a consistent total number of training iterations, with the number of epochs adjusted inversely to the data volume. Specifically, the settings are: 1k data with 3000 epochs, 10k data with 300 epochs, 100k data with 30 epochs, and 3000k data with 1 epoch. Our observations from Fig. 9 reveal several key insights:

Positive Correlation with Data Volume. When the training data volume is small, the overall quality of the trained model is generally suboptimal across all evaluated metrics. As the training data volume increases, we consistently observe an improvement in both visual quality metrics (FID and FVD decrease, CLIP-V increases) and pose accuracy metrics (RotError and TransError decrease). While this improvement is not strictly linear, it demonstrates a clear positive correlation between data volume and model performance. This trend confirms that our method effectively leverages larger datasets to enhance its capabilities.

Approaching Pose Accuracy Ceiling. For pose accuracy metrics (RotError and TransError), improvements tend to plateau at high data volumes (e.g., from 100k to 3000k). This indicates that the control precision approaches an upper limit given the current architecture and task complexity.

Potential for Further Visual Quality Improvement. For visual quality metrics (FID, FVD, CLIP-V), while resource constraints prevent us from reaching a saturation point, their upward trend indicates significant potential for further improvement with more data, inspiring future study.

5. Conclusion

In this work, we introduced Scaling4D, a novel framework that pushes the frontier of video novel view synthesis by reformulating it as a correspondence-guided generation task. This key insight allowed our model to be trained in a self-supervised manner on large-scale video data, using optical flow as a dense correspondence signal. By doing so, Scaling4D directly addressed the critical data scarcity issue and bridged the training-inference gap inherent in previous inpainting-based approaches. Our extensive experiments demonstrated that Scaling4D not only achieved state-of-the-art results in generating high-fidelity, consistent novel views but also, crucially, exhibited strong scalability with increasing data volume, confirming the effectiveness and potential.

References

[1] Jianhong Bai, Menghan Xia, Xintao Wang, Ziyang Yuan, Xiao Fu, Zuozhu Liu, Haoji Hu, Pengfei Wan, and Di Zhang. Syncammaster: Synchronizing multi-camera video generation from diverse viewpoints. *arXiv preprint arXiv:2412.07760*, 2024. 2

[2] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14834–14844, 2025. 1, 2, 3, 6, 7, 8

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

[4] BtyeDance. Dreamina, 2024. 2

[5] Hongrui Cai, Wanquan Feng, Xuetao Feng, Yan Wang, and Juyong Zhang. Neural surface reconstruction of dynamic scenes with monocular rgb-d camera. *Advances in Neural Information Processing Systems*, 35:967–981, 2022. 3

[6] Hongrui Cai, Yuting Xiao, Xuan Wang, Jiafei Li, Yudong Guo, Yanbo Fan, Shenghua Gao, and Juyong Zhang. Hera: hybrid explicit representation for ultra-realistic head avatars. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 260–270, 2025. 3

[7] Sili Chen, Hengkai Guo, Shengnan Zhu, Feihu Zhang, Zilong Huang, Jiashi Feng, and Bingyi Kang. Video depth anything: Consistent depth estimation for super-long videos. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 22831–22840, 2025. 1

[8] Tsai-Shien Chen, Aliaksandr Siarohin, Willi Menapace, Ekaterina Deyneka, Hsiang-wei Chao, Byung Eun Jeon, Yuwei Fang, Hsin-Ying Lee, Jian Ren, Ming-Hsuan Yang, and Sergey Tulyakov. Panda-70m: Captioning 70m videos with multiple cross-modality teachers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. 5

[9] Yuedong Chen, Haofei Xu, Chuanxia Zheng, Bohan Zhuang, Marc Pollefeys, Andreas Geiger, Tat-Jen Cham, and Jianfei Cai. Mvsplat: Efficient 3d gaussian splatting from sparse multi-view images. In *European Conference on Computer Vision*, pages 370–386. Springer, 2024. 1

[10] EPIC. Fab, 2025. 2

[11] Wanquan Feng, Juyong Zhang, Hongrui Cai, Haofei Xu, Junhui Hou, and Hujun Bao. Recurrent multi-view alignment network for unsupervised surface registration. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10297–10307, 2021. 3

[12] Wanquan Feng, Jiawei Liu, Pengqi Tu, Tianhao Qi, Mingzhen Sun, Tianxiang Ma, Songtao Zhao, Siyu Zhou, and Qian He. I2vcontrol-camera: Precise video camera control with adjustable motion strength. *arXiv preprint arXiv:2411.06525*, 2024. 1, 3

[13] Wanquan Feng, Tianhao Qi, Jiawei Liu, Mingzhen Sun, Pengqi Tu, Tianxiang Ma, Fei Dai, Songtao Zhao, Siyu Zhou, and Qian He. I2vcontrol: Disentangled and unified video motion synthesis control. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14051–14060, 2025. 1, 3

[14] Hang Gao, Ruilong Li, Shubham Tulsiani, Bryan Russell, and Angjoo Kanazawa. Monocular dynamic view synthesis: A reality check. *Advances in Neural Information Processing Systems*, 35:33768–33780, 2022. 5, 7

[15] Yu Gao, Haoyuan Guo, Tuyen Hoang, Weilin Huang, Lu Jiang, Fangyuan Kong, Huixia Li, Jiashi Li, Liang Li, Xiaojie Li, et al. Seedance 1.0: Exploring the boundaries of video generation models. *arXiv preprint arXiv:2506.09113*, 2025. 1

[16] Daniel Geng, Charles Herrmann, Junhwa Hur, Forrester Cole, Serena Zhang, Tobias Pfaff, Tatiana Lopez-Guevara, Yusuf Aytar, Michael Rubinstein, Chen Sun, et al. Motion prompting: Controlling video generation with motion trajectories. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 1–12, 2025. 3

[17] Google. Veo 3, 2025. 2

[18] Yuwei Guo, Ceyuan Yang, Anyi Rao, Zhengyang Liang, Yaohui Wang, Yu Qiao, Maneesh Agrawala, Dahua Lin, and Bo Dai. Animatediff: Animate your personalized text-to-image diffusion models without specific tuning. *International Conference on Learning Representations*, 2024. 2

[19] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 2

[20] Wenbo Hu, Xiangjun Gao, Xiaoyu Li, Sijie Zhao, Xiaodong Cun, Yong Zhang, Long Quan, and Ying Shan. Depthcrafter: Generating consistent long depth sequences for open-world videos. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 2005–2015, 2025. 3

[21] 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. 6, 7

[22] Yuzhou Huang, Ziyang Yuan, Quande Liu, Qiulin Wang, Xintao Wang, Ruimao Zhang, Pengfei Wan, Di Zhang, and Kun Gai. Conceptmaster: Multi-concept video customization on diffusion transformer models without test-time tuning. *arXiv preprint arXiv:2501.04698*, 2025. 3

[23] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 2023. 1

[24] Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuozhuo Dai, Jin Zhou, Jiangfeng Xiong, Xin Li, Bo Wu, Jianwei Zhang, et al. Hunyuanvideo: A systematic framework for large video generative models. *arXiv preprint arXiv:2412.03603*, 2024. 2

[25] KuaiShou. Kling, 2024. 1, 2, 3

[26] Mengtian Li, Jinshu Chen, Wanquan Feng, Bingchuan Li, Fei Dai, Songtao Zhao, and Qian He. Hyperlora: Parameter-efficient adaptive generation for portrait synthesis. In *Pro-*

ceedings of the Computer Vision and Pattern Recognition Conference, pages 13114–13123, 2025. 3

[27] Xinghui Li, Qichao Sun, Pengze Zhang, Fulong Ye, Zhichao Liao, Wanquan Feng, Songtao Zhao, and Qian He. Anydressing: Customizable multi-garment virtual dressing via latent diffusion models. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 23723–23733. IEEE, 2025. 1

[28] Zhengqi Li, Richard Tucker, Forrester Cole, Qianqian Wang, Linyi Jin, Vickie Ye, Angjoo Kanazawa, Aleksander Holynski, and Noah Snavely. Megasam: Accurate, fast and robust structure and motion from casual dynamic videos. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 10486–10496, 2025. 1

[29] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022. 2

[30] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 5

[31] Yongsen Mao, Junhao Zhong, Chuan Fang, Jia Zheng, Rui Tang, Hao Zhu, Ping Tan, and Zihan Zhou. Spatiallm: Training large language models for structured indoor modeling. In *Advances in Neural Information Processing Systems*, 2025. 5

[32] 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. *Communications of the ACM*, 65(1):99–106, 2021. 1

[33] OpenAI. Sora, 2025. 2

[34] Keunhong Park, Utkarsh Sinha, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Steven M Seitz, and Ricardo Martin-Brualla. Nerfies: Deformable neural radiance fields. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5865–5874, 2021. 3

[35] Keunhong Park, Utkarsh Sinha, Peter Hedman, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Ricardo Martin-Brualla, and Steven M Seitz. Hypernerf: A higher-dimensional representation for topologically varying neural radiance fields. *arXiv preprint arXiv:2106.13228*, 2021. 3

[36] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4195–4205, 2023. 2

[37] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-nerf: Neural radiance fields for dynamic scenes. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10318–10327, 2021. 1, 3

[38] Tianhao Qi, Jianlong Yuan, Wanquan Feng, Shancheng Fang, Jiawei Liu, SiYu Zhou, Qian He, Hongtao Xie, and Yongdong Zhang. Maskˆ 2dit: Dual mask-based diffusion transformer for multi-scene long video generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 18837–18846, 2025. 1

[39] 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 *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025. 1, 2, 3, 6, 7

[40] SideFX. Houdini, 2025. 2, 4

[41] Colton Stearns, Adam Harley, Mikaela Uy, Florian Dubost, Federico Tombari, Gordon Wetzstein, and Leonidas Guibas. Dynamic gaussian marbles for novel view synthesis of casual monocular videos. In *SIGGRAPH Asia 2024 Conference Papers*, pages 1–11, 2024. 3

[42] Mingzhen Sun, Weining Wang, Gen Li, Jiawei Liu, Jiahui Sun, Wanquan Feng, Shanshan Lao, SiYu Zhou, Qian He, and Jing Liu. Ar-diffusion: Asynchronous video generation with auto-regressive diffusion. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 7364–7373, 2025. 2

[43] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *European conference on computer vision (ECCV)*, pages 402–419. Springer, 2020. 4, 5

[44] USD. Usd, 2025. 4

[45] 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. 1, 2, 3, 6

[46] Jiahao Wang, Yufeng Yuan, Rujie Zheng, Youtian Lin, Jian Gao, Lin-Zhuo Chen, Yajie Bao, Yi Zhang, Chang Zeng, Yanxi Zhou, et al. Spatialvid: A large-scale video dataset with spatial annotations. *arXiv preprint arXiv:2509.09676*, 2025. 5

[47] Guanjun Wu, Taoran Yi, Jiemin Fang, Lingxi Xie, Xiaopeng Zhang, Wei Wei, Wenyu Liu, Qi Tian, and Xinggang Wang. 4d gaussian splatting for real-time dynamic scene rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 1, 3

[48] Zijie Wu, Chaohui Yu, Fan Wang, and Xiang Bai. Animateanymesh: A feed-forward 4d foundation model for text-driven universal mesh animation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13557–13568, 2025. 3

[49] Yuxi Xiao, Qianqian Wang, Shangzhan Zhang, Nan Xue, Sida Peng, Yujun Shen, and Xiaowei Zhou. Spatialtracker: Tracking any 2d pixels in 3d space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 1

[50] Haofei Xu, Songyou Peng, Fangjinhua Wang, Hermann Blum, Daniel Barath, Andreas Geiger, and Marc Pollefeys.

Depthsplat: Connecting gaussian splatting and depth. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 16453–16463, 2025. 1

[51] Tian-Xing Xu, Xiangjun Gao, Wenbo Hu, Xiaoyu Li, Song-Hai Zhang, and Ying Shan. Geometrycrafter: Consistent geometry estimation for open-world videos with diffusion priors. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6632–6644, 2025. 3, 5

[52] Mark Yu, Wenbo Hu, Jinbo Xing, and Ying Shan. Trajectorycrafter: Redirecting camera trajectory for monocular videos via diffusion models. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 100–111, 2025. 1, 2, 3, 5, 6, 7

[53] Shu Zhong, Mingyu Xu, Tenglong Ao, and Guang Shi. Understanding transformer from the perspective of associative memory. *arXiv preprint arXiv:2505.19488*, 2025. 5