

Wonderland: Navigating 3D Scenes from a Single Image

Hanwen Liang^{1,2*}, Junli Cao^{2,3*}, Vudit Goel², Guocheng Qian², Sergei Korolev², Demetri Terzopoulos³, Konstantinos N. Plataniotis¹, Sergey Tulyakov², Jian Ren²

¹University of Toronto, ²Snap Inc., ³University of California, Los Angeles

<https://snap-research.github.io/wonderland/>

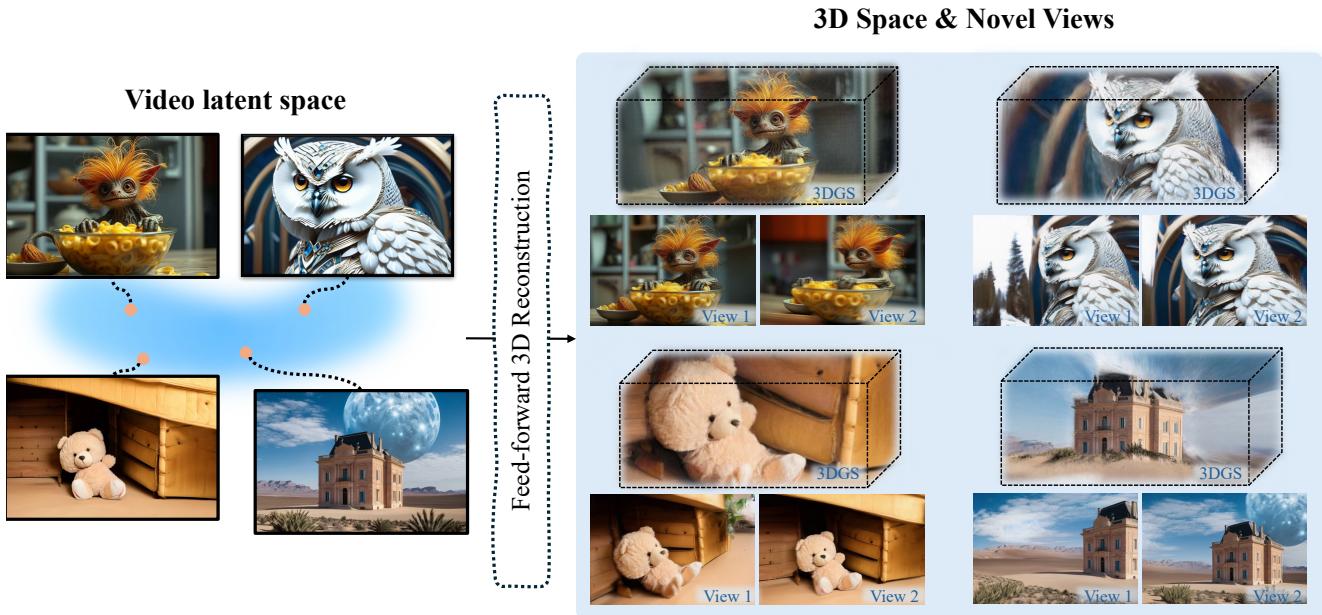


Figure 1. **Visual results generated by Wonderland.** Given a single image, Wonderland reconstructs 3D scenes from the latent space of a camera-guided video diffusion model in a feed-forward manner.

Abstract

This paper addresses a challenging question: How can we efficiently create high-quality, wide-scope 3D scenes from a single arbitrary image? Existing methods face several constraints, such as requiring multi-view data, time-consuming per-scene optimization, low visual quality in backgrounds, and distorted reconstructions in unseen areas. We propose a novel pipeline to overcome these limitations. Specifically, we introduce a large-scale reconstruction model that uses latents from a video diffusion model to predict 3D Gaussian Splattings for the scenes in a feed-forward manner. The video diffusion model is designed to create videos precisely following specified camera trajectories, allowing it to generate compressed video latents that contain multi-view information while maintaining 3D consistency. We train

the 3D reconstruction model to operate on the video latent space with a progressive training strategy, enabling the efficient generation of high-quality, wide-scope, and generic 3D scenes. Extensive evaluations across various datasets demonstrate that our model significantly outperforms existing methods for single-view 3D scene generation, particularly with out-of-domain images. For the first time, we demonstrate that a 3D reconstruction model can be effectively built upon the latent space of a diffusion model to realize efficient 3D scene generation.

1. Introduction

Humans have an innate ability to perceive and imagine three-dimensional information from a *single* image. We intuitively estimate distances, recognize shapes, and mentally infer occluded regions in an instant. This powerful and effective cognitive process allows us to interpret

*Equal contribution.

complex spatial arrangements, envision depth and relative object size, and hallucinate unseen regions of 3D scenes. However, replicating this cognitive process from a single image with learnable algorithms is extremely hard, since a single view provides limited information about object sizes and distances. And it is more challenging to estimate the geometry of unseen areas from a single image. Recent advancements in learnable scene representations (*e.g.*, Neural Radiance Fields (NeRF) [32] and 3D Gaussian Splatting (3DGS) [19]) show promising results for rendering photo-realistic 3D scenes. However, they have two major limitations that dramatically hinder their scalability and flexibility. First, they require dense *multi-view images* for training and, second, they use a time-consuming *per-scene optimization* strategy.

To alleviate the need for *multi-view data*, several studies [10, 25, 27, 38, 55] integrate generative priors from image diffusion models [15, 37, 43] for 3D synthesis from sparse views or a single image. Although these methods alleviate data requirements, they suffer from limited 3D consistency in novel view synthesis: *e.g.*, incorrect or distorted generation of the occluded regions [7, 38, 41], blurry background [10, 48], indicating image diffusion models have a limited ability to reason within complex 3D context.

To avoid a lengthy optimization process when building a 3D representation of a scene, recent efforts explore regression-based models that perform novel view synthesis in a feed-forward manner [4, 16, 44, 53, 56]. However, these methods face significant memory and computation challenges, as model training and complex scene rendering involve processing and predicting a vast number of pixels based on overlapping, high-resolution input views. Consequently, existing methods are mostly limited to object-level generation or scenes with narrow view angles and limited scope [16, 51, 56], where fewer input views and reduced computation are needed.

In this paper, we introduce **Wonderland**, which effectively addresses the above two challenges. From a single image, Wonderland can efficiently generate a high-quality point-based 3D representation (*i.e.*, 3DGS [19]) of a scene with a wide scope. We explore rich 3D scene understanding instilled in foundational video diffusion models and build a 3D representation directly from the video latents, dramatically reducing memory requirements. The *3DGS are regressed from video latents in a feed-forward manner, significantly accelerating the reconstruction process*. To achieve these features, we propose the following techniques:

- First, we introduce a representation for *controllable 3D generation by leveraging the generative priors from camera-guided video diffusion models*. Unlike image models, video diffusion models are trained on extensive video datasets, capturing comprehensive spatial relationships within scenes across multiple views and embedding

a form of “3D awareness” in their latent space, which allows us to maintain 3D consistency in novel view synthesis.

- Second, to achieve controllable novel view generation, *we empower the video models with precise control over specified camera motions*. Namely, we introduce a novel dual-branch conditioning mechanism that effectively incorporates the desired *diverse* camera trajectories into the video diffusion model, enabling it to expand a single image into multi-view consistent capturing of a 3D scene with precise pose control.
- Third, to achieve efficient 3D reconstruction, we directly *transform video latents into 3DGS*. We propose a novel *latent-based large reconstruction model (LaLRM)* that *lifts video latents to 3D in a feed-forward manner*. With such designs, during inference, our model directly predicts 3DGS from a single input image, effectively aligning the generation and reconstruction tasks—and bridging image space and 3D space—through the video latent space. Compared with reconstructing scenes from images, the video latent space offers a $256 \times$ spatial-temporal reduction, while retaining essential and consistent 3D structural details. Such a high degree of compression is crucial, as it allows the **LaLRM** to handle a wider range of 3D scenes within the reconstruction framework, with the same memory constraints.

We extensively evaluate **Wonderland** and verify that it achieves state-of-the-art performance in single-view conditioned 3D scene generation and favors several unique advantages, as follows:

- Leveraging the dual-branch camera conditioning strategy, our video diffusion model generates 3D consistent multi-view capturing of the scene with more precise pose control over existing works (Table 1 and Figure 3).
- Under the setting of zero-shot novel view synthesis, by using single-image as input for feed-forward 3D scene reconstruction, our approach outperforms existing works on various benchmark datasets; *i.e.*, RealEstate10K [59], DL3DV [23], and Tanks-and-Temples [20] (Table 2).
- By working on the latent space, our reconstruction pipeline harnesses the generative capabilities of the video diffusion model, enabling it to render *high-quality* images, *wide-scope* views, and more *generic* and *diverse* scenes (*e.g.*, out-of-domain) that go far beyond object-level reconstruction (Figure 4 and Figure 5).

2. Related Work

3D Scene Generation. The object-level 3D generation [26, 35, 36, 60] has made significant progress in terms of quality and speed, whereas 3D scene generation still lags behind. Most 3D scene generation approaches follow a two-stage process, where novel views are generated from a single image and then used to per-scene optimize 3D rep-

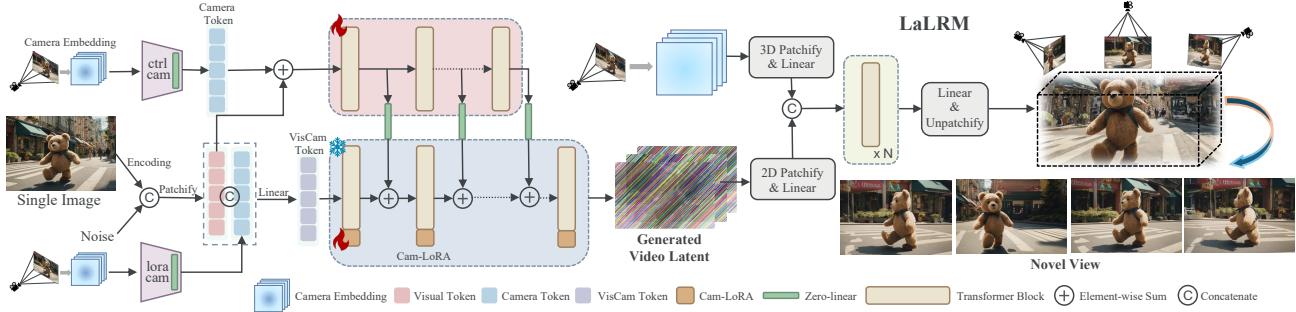


Figure 2. Overview of Wonderland. Given a single image, a camera-guided video diffusion model follows the camera trajectory and generates a 3D-aware video latent, which is leveraged by the latent-based large reconstruction model to construct the 3D scene in a feed-forward manner. The video diffusion model involves dual-branch camera conditioning to fulfill precise pose control. The LaLRM operates in latent space and efficiently reconstructs a wide-scope and high-fidelity 3D scene.

resentations. Early methods [7, 41, 54] combine depth-based warping with diffusion-based inpainting for novel view synthesis. However, reliance on monocular depth estimation and per-view inpainting can introduce severe distortions and artifacts, which limits the 3D consistency. Recent works [10, 48] incorporate camera conditioning into image diffusion models to control the poses in novel view generation, yet the 3D inconsistency issue persists due to the limitations with image-based models and they struggle with blurry backgrounds. More recent approaches utilize video diffusion models and global point clouds to improve multi-view consistency [25, 55] and guide the novel view generation process. However, this is sensitive to the point clouds construction quality and is constrained to scenes with narrow scope. Importantly, all previous methods depend on time-consuming per-scene optimization with NeRF [32] or 3DGS [19]. By contrast, our approach integrates explicit camera control into a video diffusion model to fulfill precise-pose control and multi-view consistency for expansive 3D scene generation. A large 3D reconstruction model is developed to efficiently construct 3D scenes directly from video latents in a feed-forward manner.

Camera-Conditioned Video Diffusion Models have attracted much attention [1, 3, 47]. Prior efforts explore grouping camera movements [3, 12] and apply LoRA [17] modules in video diffusion models for specific-type camera control. Later research explores injecting the camera matrices into the video diffusion models [13, 47, 50], yet these approaches can degrade visual generation quality. Recently, VD3D [1] adopts controlnet [57]-like conditioning with cross-attention mechanism. Due to the heavy computation cost, they integrate pose control only to a low-resolution video generator in a cascaded generation pipeline, resulting in imprecise pose control and unsatisfactory visual quality. Comparatively, we devise a dual-branch conditioning applied to the video diffusion foundations that obtain precise pose control and high generation quality. The sum-up operations in controlnet-branch make the integration simpler.

Sparse-View Reconstruction. Using vanilla NeRF or 3DGS for reconstruction has strict requirements for dense multi-view consistent images. Some studies lift the requirement for dense capture [11, 18, 29, 32]. Yet, they are slow to optimize. The recent efforts on feed-forward 3D reconstruction have gained growing interests [6, 16, 53, 56]. Trained on large-scale 3D datasets, these methods use transformers to regress 3D representations directly from sparse input images. Though effective, the computational overhead and number of views (*e.g.*, 2 – 6 views) limit these methods to narrower tasks, *e.g.*, object-level reconstructions. We alleviate these limitations by operating in latent space of video diffusion models, allowing us to work with a large number of views while keeping computational efficiency in check.

3. Method

This section presents our unified framework for feed-forward 3D scene generation conditioned on a single image. We first develop a camera-guided video diffusion transformer to generate video latents covering a wide scope of the scene. Precise pose control is achieved with our novel dual-branch camera conditioning module. The generated video latents are compact and 3D-aware, since they encapsulate multi-view capturing of the scene consistent in both structure and appearance, making them ideal to be lifted to 3D. Subsequently, we propose a novel Latent Large Reconstruction Model (LaLRM) to directly decode the video latents to 3D Gaussian Splatting (3DGS) [19] for scene construction in a feed-forward and memory-efficient manner.

3.1. Preliminaries

Latent Video Diffusion Transformers. Recent video diffusion models work in latent space [3, 28, 52] for efficiency and use transformer-based architectures for scalability and quality [30, 33]. Specifically, given a source video $x \in R^{T \times H \times W \times 3}$ with spatial dimension $H \times W$ and T frames, an encoder \mathcal{E} from 3D-VAE [28, 52] first com-

presses it to a latent $z \in R^{t \times h \times w \times c}$, where t , h , w , and c are temporal length, height, width, and channel dimension. The spatial and temporal compression rates are defined as $r_s = \frac{H}{h} = \frac{W}{w}$, $r_t = \frac{T}{t}$. During training, a noisy latent is created in the forward diffusion process as $z_\tau = \alpha_\tau z + \sigma_\tau \epsilon$, where $\epsilon \sim \mathcal{N}(0, I)$, and α and σ depend on the noise scheduler parameterized via a diffusion-time τ . Then, z_τ is passed to the transformer model D_θ , parameterized by θ , that first patchifies z_τ into visual tokens as a long sequence $o_v \in R^{N_v \times d_v}$ of length N_v and dimensionality d_v . Later the o_v is passed through a series of transformer blocks. The transformer output is unpatchified and projected to restore the original latent shape for the loss computation, formulated as $\mathcal{L}_\theta \triangleq E_{x \sim p_{\text{data}}, \tau \sim p_\tau} [\|D_\theta(z_\tau; y, \tau) - \epsilon\|_2^2]$. y is the conditional signal. The exact objective may vary depending on the model’s parameterization [3, 30, 52].

Gaussian Splatting. We use 3DGS as 3D representation for its fast rendering speed and high quality results [19]. 3DGS represents a scene using a collection of Gaussian points (parameterized by position, 3D covariance matrix, color, and opacity) to model appearance and geometry.

3.2. Camera-Guided Video Latent Generation

Video diffusion models have made significant progress in generating high-quality, physically plausible visualizations of the scenes. However, it is non-trivial to use them to synthesize 3D-aware latents, since they lack explicit control over pose trajectory and may produce dynamic scenes that are unsuitable for downstream 3D reconstruction. To address this, we enhance the video generation models with precise camera control and adapt them to generate static scenes. This enables the model to do a controllable and comprehensive exploration of the scenes and generate 3D-aware latents that are suitable for 3D reconstruction.

Camera Representation. To achieve precise pose control, instead of using frame-level camera parameters, we enrich the condition information with more fine-grained pixel-level positional representations; *i.e.*, the Plücker embedding. Given camera parameters of frame f consisting of rotation \mathbf{R}_f , translation \mathbf{t}_f and intrinsic \mathbf{K}_f , the Plücker coordinate of pixel at position (u_f, v_f) in frame f is formulated as $\dot{p}_{u_f, v_f} = (\mathbf{t}_f \times d'_{u_f, v_f}, d'_{u_f, v_f}) \in R^6$, where \times denotes cross-product and $d'_{(\cdot)} = \frac{d_{(\cdot)}}{\|d_{(\cdot)}\|}$ is the normalized ray direction defined as $d_{u_f, v_f} = \mathbf{R}_f \mathbf{K}_f^{-1} [u_f, v_f, 1]^T + \mathbf{t}_f$. For each video x , computing Plücker coordinates at pixel-level results in Plücker embedding $p \in R^{T \times H \times W \times 6}$, encompassing spatial-temporal camera pose information.

Dual Branch Camera Guidance. It is challenging to adapt a pretrained video diffusion transformer to generate static scenes that precisely follow the camera trajectory while preserving high visual quality. The reason is that such models have entangled attention operations over

spatial-temporal visual tokens. Even minor modifications to the original architecture can lead to degradation in visual quality [1]. Also, directly fine-tuning the model on small-scale static scene datasets without architectural changes can result in overfitting and losing the generalized knowledge acquired from large-scale pre-training. To address these challenges, we propose a dual-branch conditioning mechanism inspired by the principles of ControlNet [57] and LoRA [17], known for their excellent compatibility and extensions to pretrained models without alternating original weights. ControlNet is effective in integrating conditions and LoRA is cost-effective to finetune diffusion models with less risk of overfitting to customized training datasets.

As illustrated in Figure 2, given the Plücker embedding p of the camera, we first feed it into two lightweight camera encoders and get two sets of camera tokens: o_{ctrl} and o_{lora} . Both camera encoders contain 3D convolution (3DConv) layers to spatial-temporally compress p , and unfolding operation, resulting in camera tokens of the same dimension as video tokens o_v ($R^{N_v \times d_v}$). At the end of the camera encoders, motivated by prior work [5], we design a zero-linear layer (\mathcal{F}_{d_v, d_v}) with weight and bias initialized as zero.

To build the ControlNet branch, we create a trainable copy of the first N base transformer blocks of the foundation video model, for a trade-off between controllability and memory cost. The o_{ctrl} is element-wise added with o_v and fed into the first trainable block. The output of i th trainable block is connected to a zero-liner layer and then element-wisely added to the output of the corresponding i th frozen block of the pre-trained video diffusion model. Such design facilitates an in-depth and effective integration of camera conditional information. To further enhance camera control, we inject camera information into the main branch and add LoRA to the pre-trained video transformer. We do channel-wise concatenation of o_{lora} and o_v , and feed the concatenated tokens into a tailored linear layer ($\mathcal{F}_{2 \times d_v, d_v}$). The weights $W_{\mathcal{F}} \in R^{2d_v, d_v}$ of this layer are initialized with identity matrix in o_v part and zero matrix in o_{lora} part, so that we can get the output identical to o_v at the initial training stage. The output is forwarded into frozen transformer blocks in the main branch, where the trainable camera-LoRA module is learned [52]. This design allows us to fine-tune the foundation model with minimal computation cost. Also, the camera-LoRA module helps the model better adapt to the static scene training set, enhancing the overall static nature of the generated scenes. The effects of ControlNet and LoRA branches are analyzed in Table 1.

3.3. Latent Large Reconstruction Model

Given a single image, our camera-guided video model generates compact video latents, that capture multi-view consistent appearance and geometry information of a 3D scene. At this stage, we lift the video latents to explicit 3D repre-

sentations (*i.e.*, 3DGS) in a feed-forward manner with our Latent Large Reconstruction Model (LaLRM), largely reducing the memory and time cost compared with image-level per-scene optimized strategy. Different from previous feed-forward 3D reconstruction methods that work in image space [44, 56], LaLRM operates on latent space with **three main advantages**. First, the video latent provides diverse and high-fidelity visual information contained in the video frame sequence. Second, the latent space is highly compressed so that the model can cover large scenes in a memory-efficient way. Third, latent space is a robust representation since video diffusion models are trained on web-scale datasets, which helps our model to generalize well in out-of-domain scenes. As a result, developed upon latent space, our model can reconstruct *high-quality, wide-scope, and generic* 3D scenes efficiently.

Latent-based 3D Reconstruction Model. To perform large-scale reconstruction, we draw inspiration from prior work [56] and employ a transformer-based architecture that regresses Gaussian attributes in a pixel-aligned manner. Given the video latent $z \in R^{t \times h \times w \times c}$ and the corresponding camera poses in Plücker embedding format $p \in R^{T \times H \times W \times 6}$, we first transform them into latent tokens and pose tokens. For the video latent, we patchify along spatial dimensions with a patch size of p_l to obtain $o_l \in R^{N_l \times d_l}$, where $N_l = t \cdot \frac{h}{p_l} \cdot \frac{w}{p_l}$. For the Plücker embedding, we do 3D-patchify along spatial-temporal dimensions with 3DConv. We intentionally set the spatial patch size as $p_l \cdot r_s$ and temporal patch size as r_t (defined in Section 3.1), producing pose tokens o_p that match the length with o_l . These two sets of tokens are channel-wise concatenated, linearly projected for a lower channel dimension, and fed into a chain of base transformer blocks. We devise a lightweight latent decoding module that uses the output tokens to regress Gaussian attributes. We fulfill pixel-level correspondence between the Gaussians and source video x in RGB space via the latent decoding module, which involves 3D-DeConv layer with upsampling strides $(r_t, p_l \cdot r_s, p_l \cdot r_s)$ and output channel 12, giving us a Gaussian feature map $G \in R^{(T \cdot H \cdot W) \times 12}$. Following [56], the output 12-channel features correspond to 3D Gaussian features (3-channel RGB, 3-channel scale, 4-channel rotation quaternion, 1-channel opacity, and 1-channel ray distance). Please refer to the Supp. for details about our designed modules. During training, we render the images from the predicted Gaussians by randomly choosing V supervision views, and minimizing the image reconstruction loss with a combination of mean squared error loss (\mathcal{L}_{mse}) and perceptual loss (\mathcal{L}_{perc}) based on VGG-19 [42] network. The total loss is formulated as $\mathcal{L}_{recon} = \lambda_1 \mathcal{L}_{mse} + \lambda_2 \mathcal{L}_{perc}$, where λ_1 and λ_2 are losses weights.

Progressive Training Strategy. Due to the large domain gap between the video latent and Gaussian Splats, we face several challenges to train our large reconstruction model:

- To realize 3D geometry construction and avoid overfitting to the seen views (the views that can be directly decoded from video latent by 3DVAE), we need to involve more unseen views to guarantee 3D consistency.
- To facilitate LaLRM for generic 3D scene generation, the reconstruction model should adapt to in-the-wild video latent generated from the video diffusion model.
- We target at reconstructing the scene at high resolution for superior visual quality.

Therefore, we adopt a progressive training strategy in terms of the data source and image resolution. We initiate the model training at a lower video resolution using benchmark video datasets with known camera poses. Each video has abundant frames sufficient for seen views and unseen views sampling. We use a large stride of s to sample a video clip with T frames, covering a scene range $s \times T$ frames. The sampled frames are considered *seen views*. The rest frames in the scene range, and the frames in the same video out of the scene range are considered *unseen views*. The VAE encoder \mathcal{E} projects the video clip to the video latent, which is forwarded into our reconstruction model for GS prediction. We use V' seen views and $V - V'$ unseen views for supervision during training. Then, we scale up the training to higher resolutions. At this stage, we incorporate both benchmark datasets and a substantial volume of *out-of-domain* videos generated by our own camera-guided video diffusion model. For these generated samples, we use the video latents along with their conditioned camera poses as inputs, while the decoded video frames from these latents provide supervision views. The progressive training enables our model to better generalize to out-of-domain videos, enhancing robustness and fidelity in 3D reconstruction.

4. Experiments

Implementation Details. We use an image-conditioned transformer-based video diffusion model (*i.e.*, CogVideoX-5B-I2V [52] that generates 49 frames with resolutions 480×720) as a base model to build our camera-guided video generation model. A 3DVAE is used to compress video clips with temporal and spatial ratios of $r_t = 4$ and $r_s = 8$, producing latents of dimensions $13 \times 60 \times 90$. To build the ControlNet branch, we use the first $N = 21$ base transformer blocks from the video model to initialize the weights. And the camera-LoRA has a low rank of dimension 256. To build the transformer architecture for LaLRM, we use 24 base transformer blocks with a hidden dimension of 1024. The model is initially pretrained on low-resolution video clips($49 \times 240 \times 360$) and then finetuned with higher resolution clips($49 \times 480 \times 720$). A total of $V = 48$ supervision views are used, for which we randomly select $V' = 24$ frames from the source video clip as seen views, and an additional 24 frames disjoint from the video clip as unseen views.

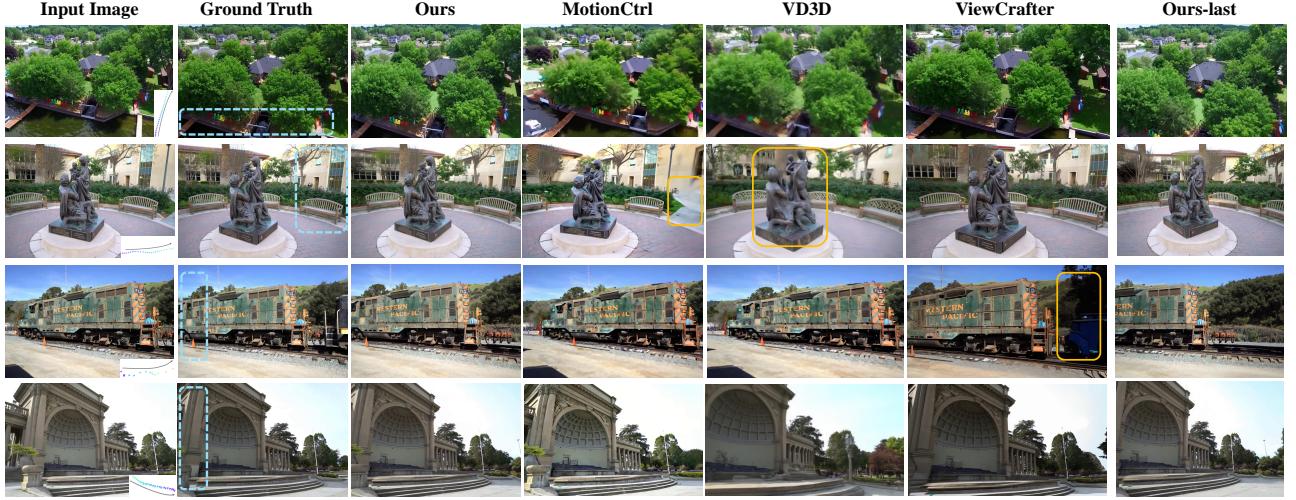


Figure 3. **Qualitative comparison to prior arts in camera-guided video generation.** The 14_{th} frame in each sample is shown for comparison, with the first column displaying the conditional image and camera trajectory (bottom-right). Blue bounding boxes denote reference areas to assist comparison and orange bounding boxes highlight low-quality generations. We also show our last frames in the rightmost column. Our method outperforms the priors in both precise camera control and high-quality and wide-scope video generation.

Table 1. **Quantitative comparison to the prior arts in camera-guided video generation** on RealEstate10K, DL3DV, and Tanks and Temples dataset. We report the performance for visual quality (FID and FVD), camera-guidance precision (R_{err} and T_{err}), and visual similarity (LPIPS, PSNR, and SSIM).

Method	Metrics							
	<i>Dataset</i>	FID ↓	FVD ↓	$R_{\text{err}} \downarrow$	$T_{\text{err}} \downarrow$	LPIPS ↓	PSNR ↑	SSIM ↑
<i>RealEstate10K</i>								
MotionCtrl [47]		22.58	229.34	0.231	0.794	0.296	14.68	0.402
VD3D [1]		21.40	187.55	0.053	0.126	0.227	17.26	0.514
ViewCrafter [55]		20.89	203.71	0.054	0.152	0.212	18.91	0.501
Ours		16.16	153.48	0.046	0.093	0.206	19.71	0.557
<i>DL3DV</i>								
MotionCtrl [47]		25.58	248.77	0.467	1.114	0.309	14.35	0.385
VD3D [1]		22.70	232.97	0.094	0.237	0.259	16.28	0.487
ViewCrafter [55]		20.55	210.62	0.092	0.243	0.237	17.10	0.519
Ours		17.74	169.34	0.061	0.130	0.218	17.56	0.543
<i>Tanks and Temples</i>								
MotionCtrl [47]		30.17	289.62	0.834	1.501	0.312	14.58	0.386
VD3D [1]		24.33	244.18	0.117	0.292	0.284	15.35	0.467
ViewCrafter [55]		22.41	230.56	0.125	0.306	0.245	16.20	0.506
Ours		19.46	189.32	0.094	0.172	0.221	16.87	0.529
<i>Ablations on RE10K</i>								
Lora-branch		19.02	212.74	0.102	0.157	-	-	-
Ctrl-branch		18.75	205.45	0.058	0.104	-	-	-
Dual-branch		17.22	183.54	0.052	0.095	-	-	-

Training Datasets. We utilize three benchmark datasets with camera pose annotations to train our models, including RealEstate10K (RE10K) [59], ACID [24], and DL3DV [23]. RE10K consists of around 80K videos, primarily capturing static real estate environments. We use a dynamic stride $s \in \{3, 4, 5\}$ to sample video clips, covering wide scene ranges of approximately 150 ~ 250 frames. ACID contains videos of natural landscapes, with 11K scenes in the training set and 20K in the test set. Since most

videos contain fewer than 100 frames with annotated camera poses, we set $s \in \{1, 2\}$ for this dataset. DL3DV, comprising training split DL3DV-10K and test split DL3DV-140, includes diverse real-world indoor and outdoor scenes. Each video contains 200 to 300 keyframes with camera pose annotations. Due to the dramatic view changes between consecutive keyframes, we sample clips using a stride $s \in \{1, 2\}$. For all three datasets, we utilize only the standard training splits during model deployment. For out-of-domain data used in training LaLRM, we generate 20K videos with image prompts from Flux.1 [21] and camera poses sampled from RE10K.

4.1. Comparison of Camera-Guided Video Generation

We evaluate our camera-guided latent generation model (Section 3.2) by comparing the visual generation quality and camera-guidance precision against three baselines; *i.e.*, MotionCtrl [47], VD3D [1], and ViewCrafter [55].

Evaluation Datasets includes three benchmark datasets:

- RE10K [59]. We randomly select 300 videos from the RE10K test set. For each video, we sample a starting frame as the image condition and the subsequent n camera poses at a stride of 3 for pose guidance, with n determined by the video temporal length.
- DL3DV-140 [23]. We randomly sample 300 video clips from DL3DV-140 dataset with a sampling stride of 2. Similar to RE10K, a starting frame is considered as the image condition and the subsequent n camera poses provide pose guidance.
- Tanks-and-Temples (Tanks) [20] is used to test out-of-

domain generalization. We randomly sample 100 video sequences from all 14 scenes with a sampling stride 4. We also use COLMAP to annotate poses for source videos as they lack dense pose annotations for all frames.

Evaluation Metrics. We compare generated videos with ground truth video clips using various metrics.

- **Visual Quality and Temporal Coherence** of the generated videos are evaluated using Fréchet Inception Distance (FID) [14] and Fréchet Video Distance (FVD) [45].
- **Camera-guidance Precision** is measured by following prior works [1, 47, 55] to use rotation error (R_{err}) and translation error (T_{err}) metrics. Camera poses of generated videos are obtained via COLMAP [39, 40], followed by converting the camera systems to be relative to the first frame and normalizing all cameras into a unified scale [1]. Since different methods vary in video length (*e.g.*, from 14 to 49 frames), we report mean errors across all frames.
- **Visual Similarity** is assessed by calculating PSNR, SSIM [46], and LPIPS [58] between the generated images and ground-truth views. To facilitate a fair comparison, we evaluate different methods over the first generated 14 frames. The reason is that the generated videos tend to deviate from the conditional single-view and present diverse appearances as the scene progresses. It becomes less reliable to use similarity metrics to evaluate generation quality and understand the difference between the generated frames and ground-truth views.

Qualitative Comparisons are presented in Figure 3, where the conditioned images and the camera trajectories are displayed in the left-most columns. As can be seen, although MotionCtrl produces images with good quality, it fails in precise alignment with the given camera conditions. This is because MotionCtrl adopts a high-level camera embedding to control camera poses, which lacks fine-grained pose control. The frames generated by VD3D show limited quality due to the low resolution. ViewCrafter has artifacts across frames, mainly because it uses incomplete point clouds to render images as conditions, which have irregular missing regions. In contrast, our method demonstrates superior quality for precise pose control and higher visual quality.

Quantitative Comparisons are shown in Table 1. Our approach consistently *outperforms* the baselines in *all* the metrics. The lower FID and FVD suggest our model better follows the data distribution of the ground-truth data. The lower LPIPS and higher PSNR and SSIM indicate better visual quality and higher similarity to the groundtruth. Moreover, our method is capable of generating more precise camera control than baselines (*i.e.*, lower R_{err} and T_{err}).

4.2. Comparison of 3D Scene Generation

We evaluate our method and two baseline approaches (*i.e.*, ZeroNVS [38] and ViewCrafter [55]) for 3D scene generation on real-world and in-the-wild synthetic datasets.

Table 2. **Quantitative comparison** on various benchmark datasets for 3D scene novel view synthesis with *single view* condition.

Method	RealEstate10K			DL3DV			Tanks-and-Temples		
	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑
ZeroNVS [38]	0.448	13.01	0.378	0.465	13.35	0.339	0.470	12.94	0.325
ViewCrafter [55]	0.341	16.84	0.514	0.352	15.53	0.525	0.384	14.93	0.483
Ours	0.292	17.15	0.550	0.325	16.64	0.574	0.344	15.90	0.510
Ablation-LaLRM									
RGB-14	0.137	21.39	0.751	0.205	18.76	0.696	0.221	19.70	0.605
RGB-49	0.126	25.06	0.830	0.196	20.94	0.733	0.192	20.54	0.687
Latent-based	0.122	27.10	0.864	0.159	23.25	0.786	0.170	22.66	0.743

ViewCrafter generates 3D scenes conditioned on a single image and a pose trajectory, and ZeroNVS reconstructs 360-degree scenes without any pose conditioning.

Comparison on Benchmark Datasets. To evaluate 3D scene generation on the benchmark datasets, we sample 100, 100, and 50 images along with camera trajectories from RE10K, DL3DV, and Tanks test sets, respectively, using the same sampling strategy detailed in Section 4.1. For quantitative comparison, we measure the LPIPS, SSIM, and PSNR by comparing the renderings against ground-truth frames from the source video. Evaluating in this under-constrained setting is challenging, as multiple 3D scenes can be considered as consistent generations for a given view [10]. Therefore, we follow Section 4.1 and measure the metrics using frames that are *temporally close* to conditional image, *i.e.*, a total of 14 sampled frames and poses that right after the conditional image.

The qualitative comparison, illustrated in Figure 4, reveals the superior 3D generation capabilities of our model. ZeroNVS produces renderings that are noticeably blurry and lack details. ViewCrafter shows improved results in visible regions from the conditional image while failing to handle occluded areas properly. In contrast, our model excels at preserving intricate details and accurately reconstructing the visible regions in the condition image. By leveraging priors from the video diffusion model, our approach generates high-fidelity and visually satisfactory novel views, even in unseen regions. We further report the quantitative comparison in Table 2, indicating our method *outperforms* all baselines by *a large margin* across multiple datasets. These results demonstrate our model can generate high-fidelity and geometrically consistent 3D scenes from single views.

Comparison on In-the-wild Scene Generation. We generate synthetic images using SDXL [34], and randomly sample trajectories from RE10K dataset for novel view synthesis. At this point, we focus on wide scene generation, with qualitative comparisons against ViewCrafter [55] and Wonderjourney [54], due to their notable performance in expansive 3D scene synthesis. Since Wonderjourney generates scenes by extending the conditional image only in a zooming-out manner, thus, our comparisons with Wonderjourney are limited to this direction. For ViewCrafter, we perform comparisons on more complex pose trajectories to evaluate the varied and wide-scope scene generation.

In Figure 5, we observe that ViewCrafter can only gener-

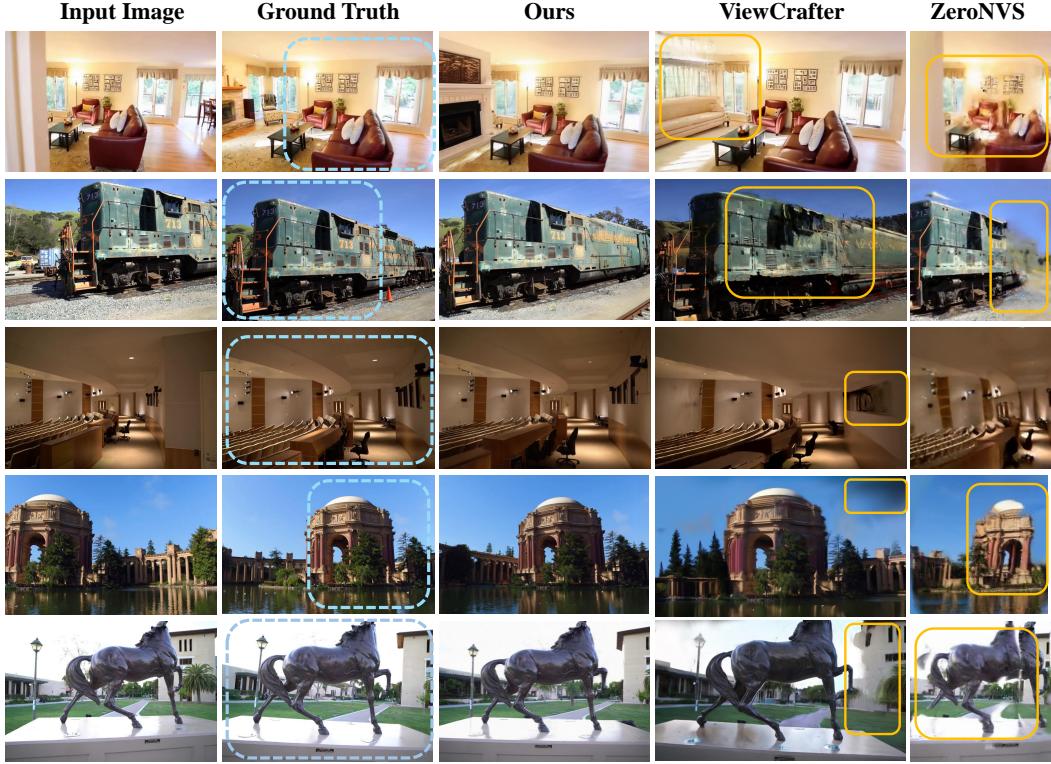


Figure 4. **Qualitative comparison for 3D scene generation.** Blue bounding boxes show visible regions from conditional images and yellow bounding boxes show low-quality regions. Our approach generates much higher quality novel views from *one* conditional image.



Figure 5. Comparison with ViewCrafter (*left*) and WonderJourney (*right*) for in-the-wild 3D scene generation from *single* input images.

ate 3D scenes within a very limited area and the quality deteriorates significantly when the view range becomes large. While Wonderjourney can produce scenes with a broader field of view, the generated views tend to be blurry and contain many artifacts. Comparatively, our method generates expansive scenes that maintain high realism and are consistent in both appearance and 3D geometry.

Comparison on Mip-NeRF. In this part, we compare our method with prior work Cat3D [10] on more complex scenes from Mip-NeRF [2]. Due to the lack of open-source code for Cat3D, we retrieve the demonstration results directly from the source webpage of [10]. The images in Fig-

ure 6 are rendered from 3DGS generated with orbital camera trajectories. For each scene, we show renderings from two viewpoints: one at the conditional image (starting) view and another at around 120° rotation from the starting view. We observe that for views close to the conditional image, our method achieves rendering quality similar to Cat3D and noticeably better than ZeroNVS. However, as the view deviates from the conditional image, Cat3D suffers from severe blurring, particularly in the background. In contrast, our method generates scenes with clearer textures, sharper details, and greater consistency with the conditional images.

Comparison on Latency. To highlight the efficiency of

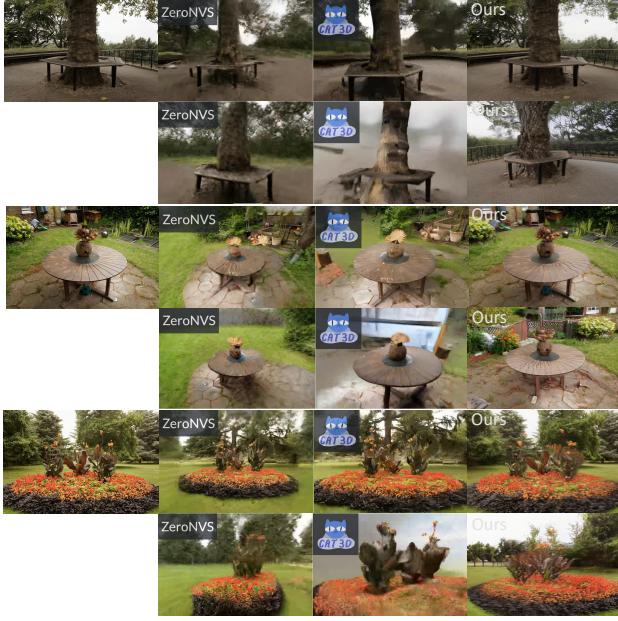


Figure 6. Comparison to ZeroNVS and Cat3D with Mip-Nerf dataset on 3D scene generation from *single* input images. For each scene, the conditional image is shown in the left-most column. We show renderings from two viewpoints, one at the conditional image (starting) view (upper) and another at around 120° -rotation from the starting view(lower).

our pipeline, we report the end-to-end latency on a single NVIDIA A100 GPU and compare it with recent works, including Cat3D [10], ViewCrafter [55], and ZeroNVS [38]. Cat3D requires approximately 16 minutes(1min with 16A100), ViewCrafter takes around 6 minutes (25-frame video), and ZeroNVS demands an extensive 3 hours. In contrast, our method completes scene generation with 5 minutes, achieving a $3.2\times$ speedup over Cat3D, $1.2\times$ over ViewCrafter, and an impressive $36\times$ improvement over ZeroNVS. This demonstrates the superior efficiency of our pipeline in single-image 3D scene generation.

4.3. Analysis of Architecture Design

Dual-branch Camera-pose Guidance. We analyze the effect of each conditioning branch in the camera-pose guidance video diffusion model. We develop two single-branch camera-pose guidance video diffusion models and provide the comparison in the second section of Table 1:

- *Ctrl-branch* that has the ControlNet-based conditioning.
- *Lora-branch* that has feature concatenation and lora-based finetuning.

We sample 100 video clips from RE10K test set and measure the metrics FID, FVD, R_{dist} and T_{dist} . Results show that compared with *Lora-branch*, *Ctrl-branch* contributes more for precise pose control. Adding *Lora-branch* on top of *Ctrl-branch*, which gives *Dual-branch*, further improves

the controllability and generation quality of the scene. More ablations are provided in the Supp. file.

Latents vs. RGB for 3D Reconstruction. We compare the methods for using video latents and RGB-frames for 3D reconstruction. We develop two RGB-based reconstruction models that adopt a similar architecture as LaLRM:

- *RGB-49*: we sample 49-frame clips from the training sets by using the same sampling strategy as LaLRM.
- *RGB-14*: we sample 14-frame clips from the training sets and increase the sampling stride by a factor of four to cover the same scene range as LaLRM.

To ensure comparable computational and memory cost with LaLRM, we introduce a 3DConv layer at the entrance of the *RGB-49* model and a 2DConv layer in the *RGB-14* model. This setup embeds RGB inputs into a lower-dimensional space, matching the input latents in LaLRM. All model architectures and training strategies are the same as LaLRM.

We evaluated the three reconstruction models on the test sets of RE10K and DL3DV, as well as the Tanks dataset. From RE10K, we sample 100 testing clips covering the scene range of 49×3 frames in source videos. From DL3DV, we sample 100 testing clips covering the scene range of 49×2 frames. For Tanks, we sample 50 testing clips covering the scene range of 49×4 frames. The sampled clips or the embedded latents are fed into the models to construct the scenes. The remaining frames within each scene range are treated as unseen views for evaluation, using LPIPS, PSNR, and SSIM metrics. As we observe in the second section of Table 2, the LaLRM (denoted as Latent-based), taking latents as input, performs the best among the others.

5. Conclusions

We have introduced **Wonderland**, a novel framework for high-fidelity 3D scene generation from a *single* image in a *feed-forward* manner. Unlike traditional methods that operate in pixel space or rely on per-scene optimization, our approach leverages the rich generative priors embedded in pose-conditioned video diffusion models. By operating within the compact latent space, **Wonderland** achieves efficient and scalable 3D scene synthesis while addressing the challenges of temporal-spatial consistency and pose controllability. We introduced a *dual-branch* camera conditioning mechanism, enabling precise pose control and diverse trajectory generation for novel view synthesis. Furthermore, our latent-based large reconstruction model (**LaLRM**) seamlessly integrates the generative capabilities of video diffusion models with 3D Gaussian Splatting, ensuring computational efficiency and scalability to wide-view coverage. Our extensive evaluations across diverse datasets have demonstrated the superior performance of our approach in generating visually consistent and high-fidelity 3D scenes. It outperforms existing state-of-the-art methods in both video generalization and 3D rendering quality.

References

- [1] Sherwin Bahmani, Ivan Skorokhodov, Aliaksandr Siarohin, Willi Menapace, Guocheng Qian, Michael Vasilkovsky, Hsin-Ying Lee, Chaoyang Wang, Jiaxu Zou, Andrea Tagliasacchi, et al. Vd3d: Taming large video diffusion transformers for 3d camera control. *arXiv preprint arXiv:2407.12781*, 2024.
- [2] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields supplemental materials.
- [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.
- [4] David Charatan, Sizhe Lester Li, Andrea Tagliasacchi, and Vincent Sitzmann. Pixelsplat: 3d gaussian splats from image pairs for scalable generalizable 3d reconstruction. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19457–19467, 2023.
- [5] Junsong Chen, Jincheng Yu, Chongjian Ge, Lewei Yao, Enze Xie, Yue Wu, Zhongdao Wang, James Kwok, Ping Luo, Huchuan Lu, et al. Pixart-alpha: Fast training of diffusion transformer for photorealistic text-to-image synthesis. *arXiv preprint arXiv:2310.00426*, 2023.
- [6] 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. 2024.
- [7] Jaeyoung Chung, Suyoung Lee, Hyeongjin Nam, Jaerin Lee, and Kyoung Mu Lee. Luciddreamer: Domain-free generation of 3d gaussian splatting scenes, 2023.
- [8] Tri Dao. Flashattention-2: Faster attention with better parallelism and work partitioning. *arXiv preprint arXiv:2307.08691*, 2023.
- [9] P Kingma Diederik. Adam: A method for stochastic optimization. (*No Title*), 2014.
- [10] Ruiqi Gao, Aleksander Holynski, Philipp Henzler, Arthur Brussee, Ricardo Martin-Brualla, Pratul Srinivasan, Jonathan T. Barron, and Ben Poole. Cat3d: Create anything in 3d with multi-view diffusion models, 2024.
- [11] Guangcong, Zhaoxi Chen, Chen Change Loy, and Ziwei Liu. Sparsenerf: Distilling depth ranking for few-shot novel view synthesis. *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.
- [12] Yuwei Guo, Ceyuan Yang, Anyi Rao, Yaohui Wang, Yu Qiao, Dahua Lin, and Bo Dai. Animatediff: Animate your personalized text-to-image diffusion models without specific tuning. *Proc. ICLR*, 2024.
- [13] Hao He, Yinghao Xu, Yuwei Guo, Gordon Wetzstein, Bo Dai, Hongsheng Li, and Ceyuan Yang. Cameractrl: Enabling camera control for text-to-video generation. *arXiv preprint arXiv:2404.02101*, 2024.
- [14] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.
- [15] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.
- [16] Yicong Hong, Kai Zhang, Jiuxiang Gu, Sai Bi, Yang Zhou, Difan Liu, Feng Liu, Kalyan Sunkavalli, Trung Bui, and Hao Tan. Lrm: Large reconstruction model for single image to 3d. *arXiv preprint arXiv:2311.04400*, 2023.
- [17] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.
- [18] Ajay Jain, Matthew Tancik, and Pieter Abbeel. Putting nerf on a diet: Semantically consistent few-shot view synthesis. In *CVPR*, pages 5885–5894, 2021.
- [19] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4), 2023.
- [20] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics*, 36(4), 2017.
- [21] Black Forest Labs. Flux: Decentralized computation framework, 2023. Accessed: 2024-11-14.
- [22] Hanwen Liang, Yuyang Yin, Dejia Xu, Hanxue Liang, Zhangyang Wang, Konstantinos N Plataniotis, Yao Zhao, and Yunchao Wei. Diffusion4d: Fast spatial-temporal consistent 4d generation via video diffusion models. *arXiv preprint arXiv:2405.16645*, 2024.
- [23] Lu Ling, Yichen Sheng, Zhi Tu, Wentian Zhao, Cheng Xin, Kun Wan, Lantao Yu, Qianyu Guo, Zixun Yu, Yawen Lu, et al. Dl3dv-10k: A large-scale scene dataset for deep learning-based 3d vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22160–22169, 2024.
- [24] Andrew Liu, Richard Tucker, Varun Jampani, Ameesh Makadia, Noah Snavely, and Angjoo Kanazawa. Infinite nature: Perpetual view generation of natural scenes from a single image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14458–14467, 2021.
- [25] Fangfu Liu, Wenqiang Sun, Hanyang Wang, Yikai Wang, Haowen Sun, Junliang Ye, Jun Zhang, and Yueqi Duan. Reconx: Reconstruct any scene from sparse views with video diffusion model, 2024.
- [26] Minghua Liu, Ruoxi Shi, Linghao Chen, Zhuoyang Zhang, Chao Xu, Xinyue Wei, Hansheng Chen, Chong Zeng, Jiayuan Gu, and Hao Su. One-2-3-45++: Fast single image to 3d objects with consistent multi-view generation and 3d diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10072–10083, 2024.
- [27] Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3d object, 2023.
- [28] Yixin Liu, Kai Zhang, Yuan Li, Zhiling Yan, Chujie Gao, Ruoxi Chen, Zhengqing Yuan, Yue Huang, Hanchi Sun, Jianfeng Gao, et al. Sora: A review on background, technology,

- limitations, and opportunities of large vision models. *arXiv preprint arXiv:2402.17177*, 2024.
- [29] Jinjie Mai, Wenxuan Zhu, Sara Rojas, Jesus Zarzar, Abdullah Hamdi, Guocheng Qian, Bing Li, Silvio Giancola, and Bernard Ghanem. Tracknerf: Bundle adjusting nerf from sparse and noisy views via feature tracks. 2024.
- [30] Willi Menapace, Aliaksandr Siarohin, Ivan Skorokhodov, Ekaterina Deyneka, Tsai-Shien Chen, Anil Kag, Yuwei Fang, Aleksei Stolar, Elisa Ricci, Jian Ren, 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*, pages 7038–7048, 2024.
- [31] Paulius Micikevicius, Sharan Narang, Jonah Alben, Gregory Diamos, Erich Elsen, David Garcia, Boris Ginsburg, Michael Houston, Oleksii Kuchaiev, Ganesh Venkatesh, et al. Mixed precision training. *arXiv preprint arXiv:1710.03740*, 2017.
- [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.
- [33] 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.
- [34] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023.
- [35] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *ICLR*, 2022.
- [36] Guocheng Qian, Jinjie Mai, Abdullah Hamdi, Jian Ren, Aliaksandr Siarohin, Bing Li, Hsin-Ying Lee, Ivan Skorokhodov, Peter Wonka, Sergey Tulyakov, and Bernard Ghanem. Magic123: One image to high-quality 3d object generation using both 2d and 3d diffusion priors. In *ICLR*. OpenReview.net, 2024.
- [37] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.
- [38] Kyle Sargent, Zizhang Li, Tanmay Shah, Charles Herrmann, Hong-Xing Yu, Yunzhi Zhang, Eric Ryan Chan, Dmitry Lagon, Li Fei-Fei, Deqing Sun, and Jiajun Wu. Zeronvs: Zero-shot 360-degree view synthesis from a single image, 2024.
- [39] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [40] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision (ECCV)*, 2016.
- [41] Jaidev Shriram, Alex Trevithick, Lingjie Liu, and Ravi Ramamoorthi. Realdmdreamer: Text-driven 3d scene generation with inpainting and depth diffusion, 2024.
- [42] Karen Simonyan. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
- [43] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.
- [44] Jiaxiang Tang, Zhaoxi Chen, Xiaokang Chen, Tengfei Wang, Gang Zeng, and Ziwei Liu. Lgm: Large multi-view gaussian model for high-resolution 3d content creation, 2024.
- [45] Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphaël Marinier, Marcin Michalski, and Sylvain Gelly. Fvd: A new metric for video generation, 2019.
- [46] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.
- [47] Zhouxia Wang, Ziyang Yuan, Xintao Wang, Yaowei Li, Tianshui Chen, Menghan Xia, Ping Luo, and Ying Shan. Motionctrl: A unified and flexible motion controller for video generation. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–11, 2024.
- [48] Rundi Wu, Ben Mildenhall, Philipp Henzler, Keunhong Park, Ruiqi Gao, Daniel Watson, Pratul P. Srinivasan, Dor Verbin, Jonathan T. Barron, Ben Poole, and Aleksander Holynski. Reconfusion: 3d reconstruction with diffusion priors, 2023.
- [49] Dejia Xu, Hanwen Liang, Neel P Bhatt, Hezhen Hu, Hanxue Liang, Konstantinos N Plataniotis, and Zhangyang Wang. Comp4d: Llm-guided compositional 4d scene generation. *arXiv preprint arXiv:2403.16993*, 2024.
- [50] Dejia Xu, Weili Nie, Chao Liu, Sifei Liu, Jan Kautz, Zhangyang Wang, and Arash Vahdat. Camco: Camera-controllable 3d-consistent image-to-video generation. *arXiv preprint arXiv:2406.02509*, 2024.
- [51] Hongbin Xu, Weitao Chen, Zhipeng Zhou, Feng Xiao, Baigui Sun, Mike Zheng Shou, and Wenxiong Kang. Controlrm: Fast and controllable 3d generation via large reconstruction model. *arXiv preprint arXiv:2410.09592*, 2024.
- [52] Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazhen Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, Da Yin, Xiaotao Gu, Yuxuan Zhang, Weihang Wang, Yean Cheng, Ting Liu, Bin Xu, Yuxiao Dong, and Jie Tang. Cogvideox: Text-to-video diffusion models with an expert transformer, 2024.
- [53] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images, 2021.
- [54] Hong-Xing Yu, Haoyi Duan, Junhwa Hur, Kyle Sargent, Michael Rubinstein, William T Freeman, Forrester Cole, Deqing Sun, Noah Snavely, Jiajun Wu, et al. Wonderjourney: Going from anywhere to everywhere. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6658–6667, 2024.
- [55] Wangbo Yu, Jinbo Xing, Li Yuan, Wenbo Hu, Xiaoyu Li, Zhipeng Huang, Xiangjun Gao, Tien-Tsin Wong, Ying Shan, and Yonghong Tian. Viewcrafter: Taming video diffusion models for high-fidelity novel view synthesis, 2024.

- [56] Kai Zhang, Sai Bi, Hao Tan, Yuanbo Xiangli, Nanxuan Zhao, Kalyan Sunkavalli, and Zexiang Xu. Gs-lrm: Large reconstruction model for 3d gaussian splatting, 2024.
- [57] Lvmmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3836–3847, 2023.
- [58] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018.
- [59] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. *arXiv preprint arXiv:1805.09817*, 2018.
- [60] Peiyi Zhuang, Songfang Han, Chaoyang Wang, Aliaksandr Siarohin, Jiaxu Zou, Michael Vasilkovsky, Vladislav Shakhrai, Sergey Korolev, Sergey Tulyakov, and Hsin-Ying Lee. Gtr: Improving large 3d reconstruction models through geometry and texture refinement. *arXiv preprint arXiv:2406.05649*, 2024.

Wonderland: Navigating 3D Scenes from a Single Image

Supplementary Material

Table of Contents

A. More Analysis on Controllable Video Generation	1
A.1. Effect of LoRA on Static Scene Generation and Camera Controllability	1
A.2. Analysis of the Design for ControlNet-branch	1
B. More Analysis on 3D Reconstruction	3
B.1. Fine-tuning with In-the-wild Dataset	3
C. More discussion with Related Works	3
D. Implementation Details	5
D.1. More Details for Model Architectures	5
D.2. Training Details	6
E. Limitations and Future Work	6

A. More Analysis on Controllable Video Generation

Here we provide more ablation analysis to better understand our design for camera-conditioned video generation.

A.1. Effect of LoRA on Static Scene Generation and Camera Controllability

In our camera-guided video generation model, we employ LoRA fine-tuning in the main branch. LoRA is particularly advantageous for its compatibility with pre-trained models, as it introduces extensions without altering the original weights. The lightweight module offers a cost-effective approach to fine-tune large-scale models, requiring minimal computational resources and reducing the risk of overfitting to customized training datasets. In our framework, we take advantage of LoRA to enhance both static scene generation and camera controllability.

To evaluate the impact on **static scene generation**, we fine-tune the I2V source model using LoRA on customized datasets dominated by static scenes, including RealEstate10K (RE10K), ACID, and DL3DV. In this setup, no pose control is applied. We compare the performance of the fine-tuned model against the source model on 20 in-the-wild image prompts (along with text descriptions). The results show that the fine-tuned model generates significantly

more static scenes compared to the source model, especially for cases with humans and animals. Visualizations provided in Fig. A1 illustrate that LoRA enables the generation of more static scenes without compromising visual quality.

To assess the role of LoRA in **enhancing camera controllability**, we train our full model without incorporating LoRA modules. Under such settings, in the main branch, the camera embeddings are fed into the network in a channel-concatenation manner without LoRA tuning in the main backbone. Only the LoRA-camera encoder and linear processing layers at the top are learned. Following the experimental setup outlined in Sec. 4.3, we evaluate the models on 100 video clips sampled from RealEstate10K. As shown in the middle section of Tab. A2, a comparison between the “*Dual w/o LoraModule*” and “*Dual-branch*” configurations reveals that LoRA plays a critical role in fine-tuning the main branch. Excluding LoRA results in a noticeable performance drop in both visual quality and camera-guidance precision.

A.2. Analysis of the Design for ControlNet-branch

In our full model, the ControlNet branch is designed by utilizing a trainable copy of the first 21 base transformer blocks of the foundational video model, which consists of 42 blocks in total. Here we extensively evaluate the designs by using the ControlNet branch only. Specifically, we train

Table A1. Overview of the notations used in the paper.

Expression	Specification	Explanation
<i>commonly used</i>		
x	$x \in R^{T \times H \times W \times 3}$	source video clip
s	-	stride to sample clip x from source video
z	$z \in R^{t \times h \times w \times c}$	video latent embedded from x
\mathcal{E}	-	encoder from 3D-VAE
r_s	$r_s = \frac{H}{h} = \frac{W}{w}$	spatial compression rate
r_t	$r_t = \frac{T}{t}$	temporal compression rate
p	$p \in R^{T \times H \times W \times 6}$	Plücker embedding of cameras of video clip x
<i>diffusion used</i>		
τ	-	diffusion time step
α_τ, σ_τ	-	diffusion noise scheduler parameters
z_τ	-	noisy video latent
D_θ	-	diffusion model parameterized by θ
o_v	$o_v \in R^{N_v \times d_v}$	visual tokens as a sequence in diffusion model
o_{ctrl}, o_{lora}	$o_{ctrl}, o_{lora} \in R^{N_v \times d_v}$	camera tokens as a sequence in diffusion model
N	-	number of transformer blocks in ControlNet branch
<i>reconstruction used</i>		
p_l	-	spatial patch size applied to z in LaLRM
o_l	$o_l \in R^{N_l \times d_l}$	visual latent tokens as a sequence in LaLRM
N_l	$N_l = t \cdot \frac{h}{p_l} \cdot \frac{w}{p_l}$	number of visual latent tokens in LaLRM
o_p	$o_p \in R^{N_l \times d_l}$	camera tokens as a sequence in LaLRM
V	-	number of supervision views in LaLRM
G	$G \in R^{(T \cdot H \cdot W) \times 12}$	Gaussian feature map in LaLRM



Figure A1. Comparison of video generations between the source model (*upper row*) and the model fine-tuned on static-scene datasets with LoRA modules (*lower row*). The results demonstrate that fine-tuning the model on static-scene datasets equipped with LoRA produces significantly more static scenes.

the model with ControlNet conditioning under various configurations: using the first 21 blocks without weight copying (denoted as *w/o weight copy*), and using the first 1, 10, or 30 blocks with weight copying, denoted as block-1, blocks-10, and blocks-30, respectively. As shown in the third section of Tab. A2, comparisons among different architecture configurations and the “*Ctrl-branch*” reveal that weight copying improves all metrics a lot, particularly visual quality. Using

only one block results in weak camera controllability, while increasing the number of blocks strengthens the ability of the model to guide camera poses. Notably, using 21 blocks (“*Ctrl-branch*”) achieves similar levels of pose controllability as using 30 blocks, while maintaining high visual quality. Based on these observations, we select the trainable copy of the first 21 base transformer blocks as it provides an optimal balance between pose controllability and com-



Figure A2. Comparison of 3D rendering performance between latent reconstruction models fine-tuned *without* in-the-wild dataset (*upper row*) and *with* in-the-wild dataset (*lower row*). Involving in-the-wild datasets during fine-tuning improves the generalization capability.

Table A2. Analysis on architecture designs in camera-guided video generation model. We report the performance for visual quality (FID and FVD) and pose control precision (R_{err} and T_{err}) from models trained on RealEstate10K dataset. The first section of the table is adopted from Tab.1 in the main paper.

Architecture	Metrics			
	FID ↓	FVD ↓	$R_{\text{err}} \downarrow$	$T_{\text{err}} \downarrow$
Lora-branch	19.02	212.74	0.102	0.157
Ctrl-branch	18.75	205.45	0.058	0.104
Dual-branch	17.22	183.54	0.052	0.095
Dual w/o LoraModule	17.84	195.07	0.062	0.101
<i>Ctrl-branch</i> only				
w/o weight copy	18.92	206.75	0.065	0.108
block-1	19.90	214.66	0.114	0.162
blocks-10	19.15	210.74	0.075	0.126
blocks-30	20.15	221.61	0.056	0.105

putational efficiency.

B. More Analysis on 3D Reconstruction

In the following section, we provide more analysis to understand our choices for building the large-scale 3D reconstruction model.

B.1. Fine-tuning with In-the-wild Dataset

In the deployment of Latent Large Reconstruction Model (LaLRM), we adopt a progressive training strategy. During the second stage, we fine-tune the model by involving a self-generated in-the-wild dataset. To assess the impact of this dataset, we further fine-tune a separate reconstruction model, LaLRM-, without incorporating in-the-wild dataset. We quantitatively compare the performance of LaLRM- and LaLRM on benchmark datasets and qualitatively evaluate them on 20 disjoint in-the-wild image prompts. The results shown in Tab. A3 indicate that incorporating in-the-wild dataset during fine-tuning enhances the generaliza-

Table A3. Analysis on involving in-the-wild dataset to fine-tune LaLRM. We report the performance on various benchmark datasets for novel view synthesis of 3D scenes, which are built from *single view* condition.

Method	RealEstate10K			DL3DV			Tanks-and-Temples			
	Metrics	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑
LaLRM-		0.295	17.06	0.538	0.343	16.62	0.570	0.359	15.85	0.502
LaLRM		0.292	17.15	0.550	0.325	16.64	0.574	0.344	15.90	0.510

tion capabilities of our model. Furthermore, as shown in Fig. A2, LaLRM demonstrates noticeably better rendering quality compared to LaLRM-, further validating the benefits of using in-the-wild data in the fine-tuning process.

C. More discussion with Related Works

We provide a detailed discussion between our work and the related studies that use generative priors for 3D rendering. Significant advancements have been achieved in 3D object generation from text or single images [22, 26, 35, 36, 49, 60], with notable improvements in quality and efficiency. However, progress in 3D scene generation has remained relatively limited. Most approaches to 3D scene generation follow a **two-stage process**: *First*, generating novel views from a single image; *Second*, using these views to train a 3D representation with a per-scene optimization strategy.

Early methods, such as LucidDreamer [7] and RealmDreamer [41], explore scene-level 3D generation conditioned on text descriptions or single images. They rely on the 3D priors from incomplete point clouds constructed via depth prediction from single images. Then, they combine depth-based warping with diffusion-based image inpainting to complete the scenes in an autoregressive manner. However, these methods often struggle with inconsistencies in occluded regions, as the per-view inpainting process can introduce severe artifacts and discontinuities, particularly in unseen areas. WonderJourney [54], which targets wide-scene generation, also employs image inpainting diffusion models to fill unseen regions rendered from limited point clouds. However, as shown in our main comparisons, this

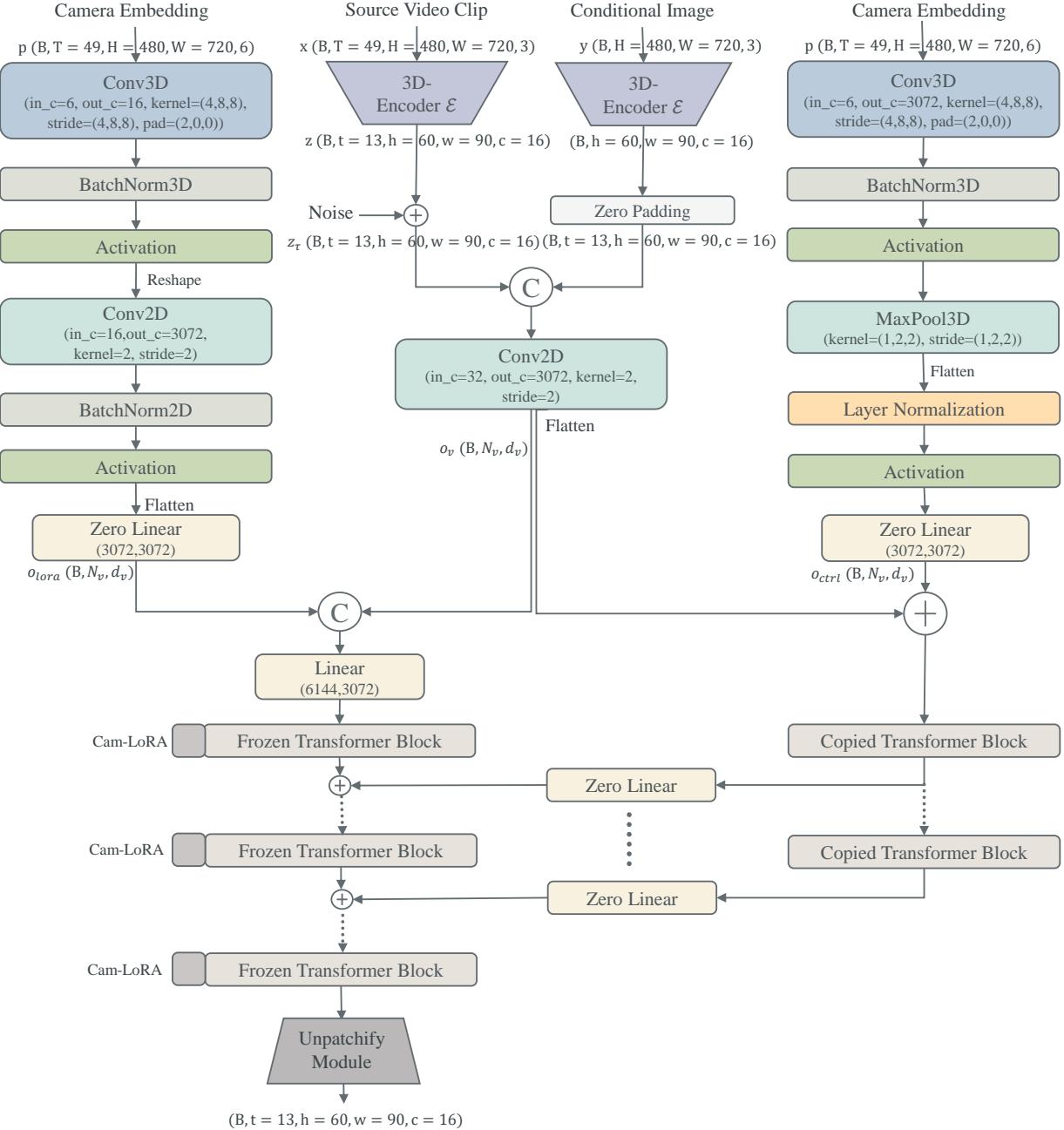


Figure A3. Structure of Dual-branch Camera-guided Video Diffusion Model. We show the skeletons of the training pipeline, where random noise is added to the video latents. The conditional image is merged to the noisy latents via feature concatenation. The camera guidance is integrated with LoRA-branch (left) and ControlNet-branch (right). We ignore the text tokens, the diffusion time embeddings, the positional embeddings, and some reshaping operations for simplicity in the figure. In the foundation diffusion transformer, the text tokens are concatenated along *number-of-token* dimension with visual tokens. Thus we apply zero-padding to camera tokens to guarantee the same length before concatenation or element-wise sum. By default, we use SiLu as our activation function.

method similarly suffers from 3D incoherence in occluded areas. Also, all these works do not have automatic and explicit control over camera poses during the generation process.

Other works, such as Cat3D [10] and ReconFusion [48],

address multi-view consistency by incorporating camera conditioning into image diffusion models. Nonetheless, a noticeable issue is their tendency to produce blurry or distorted background regions, particularly when conditioned on a single image. This arises from their usage of image

diffusion models to obtain dense views auto-regressively, which are then used for 3D reconstruction via per-scene optimization. Image diffusion models lack built-in mechanisms to guarantee cross-view consistency and such a multiple-shots generation strategy often introduces inconsistencies, especially for wide-view scenarios.

More recent works, like ReconX [25] and ViewCrafter [55], leverage video diffusion models and global point clouds to enhance multi-view consistency. However, as demonstrated in our main comparisons, these methods are sensitive to the initialization of point clouds and are restricted to generating scenes with a narrow scope. Additionally, they lack explicit pose control during the generation process.

Importantly, all the previous methods depend on time-consuming per-scene optimization like NeRF [32] or 3DGS [19]. In contrast, our approach integrates explicit camera control into a video diffusion model to enable precise and expansive scene generation. We develop a large-scale 3D reconstruction model capable of efficiently constructing 3D scenes directly from video latents. The design effectively aligns the generation and reconstruction tasks and bridges the image space and 3D space through the video latent space, fulfilling single-image to 3D scene generation in a feed-forward manner, as well as eliminating the need for time-consuming per-scene optimization.

D. Implementation Details

We provide a list of notations in Tab. A1 to facilitate the presentation for our neural network architectures and formulas in the paper.

D.1. More Details for Model Architectures

In our framework, we develop a camera-guided video diffusion model and a latent large reconstruction model to directly generate 3D scenes, which are conditioned on single images. Such designs effectively align the generation and reconstruction tasks and bridge the image space and 3D space through the video latent space. In this section, we provide detailed illustrations of the proposed model architectures in Fig. A3 and Fig. A4.

In Fig. A3, we show the details of integrating camera embeddings to the pre-trained video diffusion transformer during the training stage, with a ControlNet-branch and a LoRA-branch. Each branch involves a lightweight camera encoder composed of Convolutional layers and zero-linear layers. The camera encoders project the camera embedding into camera tokens of the same dimension as the visual tokens. The visual tokens and camera tokens are concatenated or element-wisely added before feeding into the main branch and the ControlNet branch. The visual-camera tokens are further processed with a sequence of transformer blocks and mapped to the same dimension of added noise

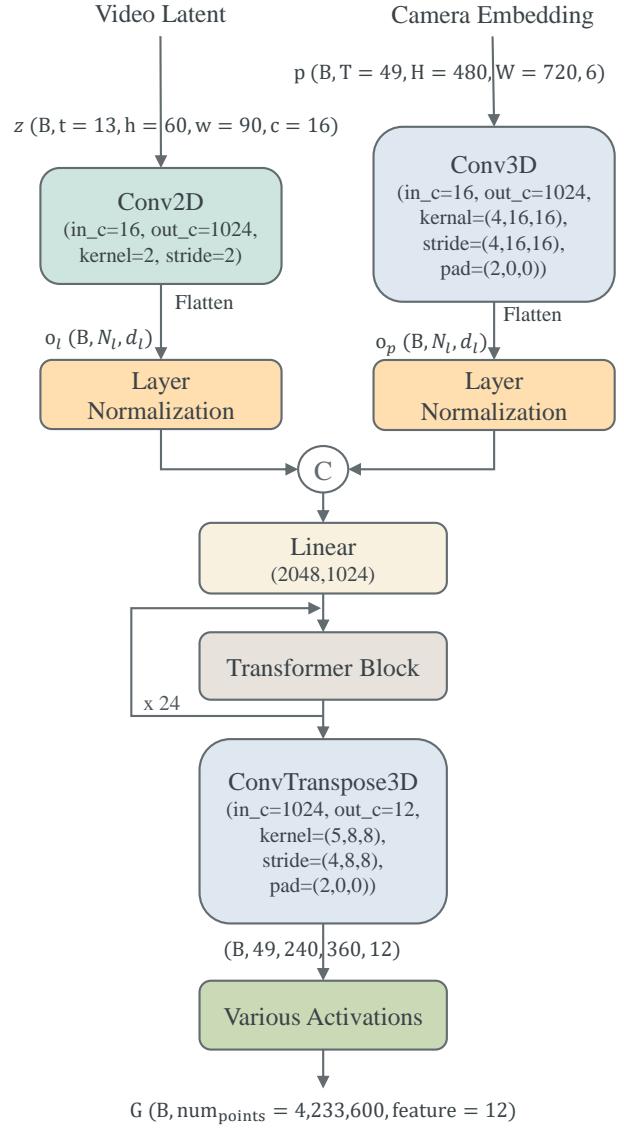


Figure A4. Structure of Latent Large Reconstruction Model (LaLRM). Given a video latent z and the camera embeddings p , the LaLRM directly regresses the 3DGS features in a feed-forward manner.

by unpatchify module. Note that in Fig. A3, we ignore the text tokens, diffusion time embeddings, and positional embeddings for simplicity. In Fig. A4, we show the details of the Latent Large Reconstruction Model (LaLRM). Given the video latents which can be generated from the video diffusion model (during inference time) or can be embedded from source video clips (during training time), a lightweight tokenization module projects video latents to visual tokens. The camera embeddings are also projected to camera tokens with 3D-patchify operations. The two sets of tokens are merged with channel-wise concatenation and fed into

a sequence of transformer blocks to regress the Gaussian points features.

D.2. Training Details

In the **camera-guided video diffusion model**, we use a transformer-based video diffusion model (*i.e.*, CogVideoX-5B-I2V [52] that generates 49 frames with resolution as 480×720). An encoder from 3DVAE is used to compress video clips with ratios of $r_t = 4$ and $r_s = 8$, giving latents with dimension as $13 \times 60 \times 90$. To build ControlNet branch, we use the first $N = 21$ base transformer blocks from the video model to initialize the weights. The camera-LoRA has a low rank of dimension 256. The model is trained with a batch size of 24 for 40K steps, using Adam optimizer [9] with a learning rate of 2×10^{-5} , $\beta_1 = 0.9$, $\beta_2 = 0.95$, and weight decay = 1×10^{-4} . For the **latent-based large reconstruction model**, we use a patch size $p_l = 2$ for visual latent and use temporal patch size 4 and spatial patch size 16 for camera Plücker embedding. We follow [56] and use the same architecture for transformer blocks. we use 24 base transformer blocks with a hidden dimension of 1024. The latent decoding module has 3D-DeConv layer with upsampling strides (4, 16, 16). The backbone transformer network is implemented efficiently with FlashAttentionV2 [8] and optimized with mixed-precision training [31] with BF16 datatype. We first pre-train the model with low-resolution video clips with dimensions $49 \times 240 \times 360$ and their corresponding latents with dimensions $13 \times 30 \times 45$. Then, we fine-tune the model with high-resolution video clips with dimensions as $49 \times 480 \times 720$ and the corresponding latents with the dimension as $13 \times 60 \times 90$. At this stage, we modify 3D-DeConv layer in latent decoding module with upsampling strides (4, 8, 8). Even with smaller upsampling rate, for each 3D scene, our Gaussian prediction brings us an enormous quality of Gaussians points, *i.e.*, $T \times \frac{H}{2} \times \frac{W}{2}$ (4,233,600), to construct each scene. A total of $V' = 48$ supervision views are used, for which we randomly select $V' = 24$ frames from each sampled video clip as seen views, and an additional 24 frames disjoint from the video clip as unseen views. The model is pre-trained and fine-tuned respectively for 200K and 100K iterations with a cosine annealing schedule at a peak learning rate of 4×10^{-4} and 1×10^{-5} . We use a batch size of 24 with Adam optimizer $\beta_1 = 0.9$, $\beta_2 = 0.95$ and weight decay = 1×10^{-4} .

E. Limitations and Future Work

While our method achieves superior generation performance and higher efficiency compared to prior works, there are still some limitations. Although the overall process is efficient, the inference speed of the video generation model remains a bottleneck. Most of the computational time in our pipeline is consumed during the video generation phase. This limitation could be improved through parallel comput-

ing, *e.g.*, using xDiT¹ for parallel inference, or utilizing more efficient denoising strategy. Also, our approach primarily focuses on static scenes. In the future, we aim to extend this pipeline to dynamic scenes, exploring its potential for generating 4D content that incorporates temporal dynamics. By addressing these limitations, our framework could be further optimized for broader applications and enhanced performance.

¹<https://github.com/xdit-project/xDiT>