

STABLE VIRTUAL CAMERA: Generative View Synthesis with Diffusion Models

Supplementary Material

In addition to this supplementary PDF, we highly recommend readers to **check our attached webpage and video** that presents different video results across different view synthesis settings.

Contents

A Broader Impact and Limitations	2
Broader Impact.	2
Limitations.	2
B Related Work	2
Novel view synthesis.	2
Feed-forward models.	2
Intermediate representation models.	2
Diffusion-based models.	2
C Implementation Details	2
Architecture.	2
Conditioning.	2
D Benchmark	3
Datasets, splits, and the number of input views.	3
Small-viewpoint <i>versus</i> large-viewpoint NVS.	3
Choice of scenes.	3
Choice of input and target views.	5
E Additional Experiments	5
E.1. Qualitative Results	5
E.2. Quantitative Results	5
PSNR on RE10K.	5
Ablation on two-pass procedural sampling.	5
LPIPS and SSIM.	6
E.3. Discussion	6
Zero-shot generalization of context window length T	6
Zero-shot generalization of image resolution.	6
Guidance scale on generation uncertainty.	7
Sampling diversity of unseen areas.	8
Samples <i>versus</i> 3DGS.	8
Padding T when $P + Q < T$	9
Artifacts on long-trajectory NVS.	9

A. Broader Impact and Limitations

Broader Impact. SEVA significantly advances immersive 3D experiences by synthesizing realistic and temporally consistent views from sparse camera inputs, addressing key limitations in NVS. Inspired by James Cameron’s pioneering Virtual Camera technology—which enabled filmmakers to intuitively navigate virtual environments and visualize precise camera trajectories—our generative AI-driven model similarly allows users to create intricate, controllable camera paths without the typical complexity of dense view captures or explicit 3D reconstructions. By generalizing across arbitrary viewpoint changes and enabling temporally smooth rendering without NeRF distillation, our approach simplifies the NVS pipeline, enhancing accessibility for content creators, developers, and researchers. This facilitates applications ranging from virtual cinematography and gaming to digital heritage preservation, substantially broadening the usability and scalability of NVS.

Limitations. The performance of SEVA is constrained by the scope of its training data, resulting in reduced quality for certain types of scenes. Specifically, input images featuring humans, animals, or dynamic textures (*e.g.*, water surfaces) typically lead to degraded outputs. Additionally, highly ambiguous scenes or complex camera trajectories pose challenges; for instance, trajectories that intersect with objects or surfaces may cause noticeable flickering artifacts. Similar issues arise for extremely irregularly shaped objects or when target viewpoints significantly diverge from the provided input viewpoints.

B. Related Work

Novel view synthesis. While traditional NVS has been studied for nearly several decades, it has recently achieved remarkable success with the help of techniques such as NeRF [1, 2] and diffusion models [3, 4]. Using these techniques, there are broadly two ways of generating novel views : 1) estimate a 3D representation using multiple sparse input views, then regress the novel views from this intermediate representation, 2) directly estimate the novel views from the sparse input views, either in a single shot in a feed-forward manner, or in multiple sampling steps using diffusion models.

Feed-forward models. Approaches like LFNR [5] and LVSM [6] directly generate target views and leverage data-driven learning to capture 3D inductive biases. While often efficient, these methods struggle with the inherent diversity of generative NVS, limiting their capacity to model multiple plausible solutions. In contrast, our approach frames generative NVS through a diffusion perspective, enabling us to sample diverse, plausible solutions during inference,

thereby addressing ambiguities and enhancing generation capacity.

Intermediate representation models. Techniques such as NeRF [1] and Gaussian Splatting [7] have made significant progress on per-scene optimization from input views by building 3D representations efficiently. Several works show that these representations can then be used to regress novel views. pixelNeRF [8] builds a NeRF from multiple input views; Splatter Image [9], pixelSplat [10], and MV-Splat [11] build a 3D representation using Gaussian Splatting; LRM [12] builds a triplane representation. However, these optimization-based methods cannot creatively synthesize missing regions, and rely on tens, if not hundreds, of posed input images which limits their practicality in real-world applications.

Diffusion-based models. Our work falls within this category, where target novel views are generated in multiple steps through a denoising diffusion process [3, 4]. As mentioned earlier, existing diffusion-based methods can be divided into two main types: *image models* and *video models*.

Image models are designed to synthesize distant viewpoints [13–15]. However, these early practices only generate one viewpoint at a time, and lack multi-view consistency, often resulting in jittery and inconsistent samples when generating along a camera trajectory. Works such as MVDream [16], SyncDreamer [17] and HexGen3D [18] generate multiple fixed views simultaneously. However, these models only generate specific views given a conditional image, not arbitrary viewpoints.

To obtain consistent 3D objects, these models necessitate NeRF distillation, either through Score Distillation Sampling (SDS) [19, 20] or directly upon completely sampled images [21, 22].

Video models can produce smooth video sequences by maintaining certain constraints relative to the input views [23]. However, they are generally limited to smaller camera motions due to the natural frame rate in video training. Some works use video diffusion models to generate 4D scenes [24]. But in those works, the video diffusion models do not contribute to the consistency of the 3D object itself, that part is handled by image-based diffusion models such as MVDream.

C. Implementation Details

Architecture. We detail the architecture in Fig. 1. SEVA is trained with fixed sequence length as a “*M*-in *N*-out” multi-view diffusion model with standard architecture.

Conditioning. To fine-tune our base model into a multi-view diffusion model, we add camera conditioning as

Training: fixed seq. len (M -in N -out)

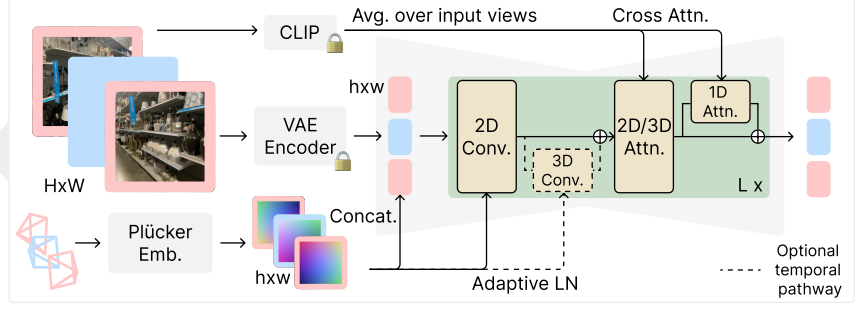
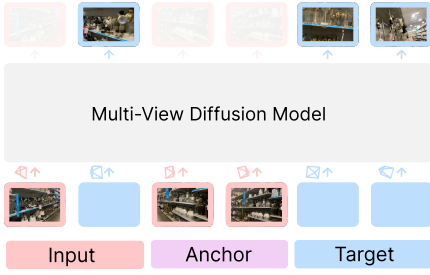


Figure 1. **Detailed Architecture.** SEVA conditions on CLIP embeddings, VAE latents of the input views, and their corresponding camera poses.

Plücker embedding [25] via concatenation [22] and adaptive layer normalization [26]. We normalize π^{inp} and π^{tgt} by first computing the relative pose with respect to the first input camera and then normalizing the scene scale such that all camera positions are within a $[-2, 2]^3$ cube. For each input frame, we first encode its latent then concatenate with its Plücker embedding and a binary mask [22, 27] differentiating between input and target views. For each target frame, we use the noisy state of its latent instead. Additionally, we find it helpful [23] to also inject high-level semantic information via CLIP [28] image embedding. We zero initialize new weights for additional channels in the first layer. In our experiment, we found that our model can quickly adapt to these conditioning changes and produce realistic images with as few as 5K iterations.

D. Benchmark

We collect 10 commonly used datasets to benchmark NVS, encompassing a diverse range of scene distributions and complexities, shown in Tab. 1.

Datasets, splits, and the number of input views. We consider (1) object datasets, e.g., OmniObject3D [29] (OO3D) and GSO [30]; (2) object-centric scene datasets, e.g., LLFF [34], DTU [35], CO3D [36], and WildRGBD [37] (WRGBD); and (3) scene datasets, e.g., RealEstate10K [31] (RE10K), Mip-NeRF 360 [38] (Mip360), DL3DV140 [39] (DL3DV), and Tanks and Temples [41] (T&T). We consider a wide range of the number of input views P , ranging from sparse-view regime to semi-dense-view regime, evaluating models’ input flexibility. To establish a comprehensive and rigorous comparison with baselines, we consider different dataset splits utilized in prior works with the same input-view configuration if not specified as our (O) split defined by ourselves. These include splits used in 4DiM [32] (D), ViewCrafter [33] (V), pixelSplat [10] (P), ReconFusion [21] (R), SV3D [23] (S), and Long-LRM [40] (L). For example, the 4DiM [32] (D) split on the RE10K dataset is 128 out of all 6711 test scenes

with $P = 1$.

Small-viewpoint versus large-viewpoint NVS. Sweeping across all datasets, splits, and input-view configurations reveals a diverse range of setups. To better evaluate models’ generation capacity and interpolation smoothness, we propose to categorize these setups into two groups—*large-viewpoint* NVS and *small-viewpoint* NVS—depending on the disparity between \mathbf{I}^{tgt} and \mathbf{I}^{inp} . Formally, for each target view, we consider the minimal distance between the CLIP [28] feature of that view and those of all input views. Averaging across all target views yields the CLIP distance, $\mathcal{D}_{\text{CLIP}}(\mathbf{I})$. Splits with $\mathcal{D}_{\text{CLIP}}(\mathbf{I}) \leq 0.11$ are grouped as small-view NVS, while those with $\mathcal{D}_{\text{CLIP}}(\mathbf{I}) > 0.11$ are grouped as large-view NVS. We concrete in Tab. 1 a detailed task setup including the choice of datasets and splits (depending on which scenes from each dataset and which views from each scene are used). Large-viewpoint NVS with larger disparities requires a model to generate prominent unseen areas from input observations, predominantly assessing models’ generation capacity, whereas small-viewpoint NVS with smaller disparities emphasizes interpolation smoothness and continuity with nearby input views.

Choice of scenes. We follow the choices of scenes for splits adopted from previous works. For our split, we use all scenes from the dataset without specification.

For Tanks and Temples, the 2 chosen scenes are `Train` and `Truck`.

For the DL3DV-140 dataset, the 10 test scenes we choose in O split are:

- 165F5AF8BFE32F70595A1C9393A6E442ACF7AF019998275144F605B89A306557
- 341B4FF3DFD3D377D7167BD81F443BEDAFBFF003BF04881B99760FC0AEB69510
- 3BB3BB4D3E871D79EB71946CBAB1E3AFC7A8E33A661153033F32DEB3E23D2E52
- 3BB894D1933F3081134AD2D40E54DE5F0636BD8B502B0A8561873BB63B0DCE85


















	type	split	#scene	$(\mathbf{I}^{\text{inp}}, \mathbf{I}^{\text{tgt}}) \sim \mathcal{V}$	P	$\mathcal{D}_{\text{CLIP}}(\mathbf{I})$
Small-viewpoint NVS						
OmniObject3D [29]		O (dynamic orbit)	308	✓	3	0.11
GSO [30]		O (dynamic orbit)	300	✓	3	0.11
RealEstate10K [31]		D [32]	128	✓	1	0.09
		R [21]	10	✓	1	0.08
		P [10]	6474	✓	3	0.03
		V [33]	10	✓	2	0.04
				✓	2	0.11
LLFF [34]		R [21]	8	✓	1 3	0.04 0.03
DTU [35]		R [21]	15	✓	1 3	0.07 0.06
CO3D [36]		R [21] V [33]	20 10	✓ ✓	3 2	0.09 0.09
WildRGB-D [37]		O_e (1/3 orbit) O_h (full orbit)	20	✓	3 6	0.07 0.11
Mip-NeRF360 [38]		R [21]	9	✗	6	0.11
DL3DV-140 [39]		O L [40]	10 140	✓ ✓	6 32	0.10 0.05
Tanks and Temples [41]		V [33] L [40]	22 2	✓ ✓	2 32	0.10 0.10
Large-viewpoint NVS						
OmniObject3D [29]		S [23] (dynamic orbit)	308	✓	1	0.16
GSO [30]		S [23] (dynamic orbit)	300	✓	1	0.18
CO3D [36]		R [21]	20	✓	1	0.15
WildRGB-D [37]		O_h (full orbit)	20	✓	1 3	0.19 0.14
Mip-NeRF360 [38]		R [21]	9	✗	1 3	0.19 0.13
DL3DV-140 [39]		O	10	✓	1 3	0.21 0.12
Tanks and Temples [41]		O	2	✓	1 3 6 9	0.21 0.18 0.16 0.14

Table 1. **Statistics for NVS benchmark.** We consider 10 publicly available datasets commonly used for evaluating NVS, encompassing both object-level and scene-level data. Views from Mip-NeRF360 [38] derive from several disjoint captures following different camera trajectories, thus all views $(\mathbf{I}^{\text{imp}}, \mathbf{I}^{\text{tgt}}) \sim \mathcal{I}$. P denotes the number of input views. Depending on the disparity between \mathbf{I}^{imp} and \mathbf{I}^{tgt} , we group NVS tasks into small-viewpoint NVS (top panel) where target views are similar to input views and large-viewpoint NVS (bottom panel) where target views are more different to input views.

- 9E9A89AE6FED06D6E2F4749B4B0059F35CA97F848CEDC4A14345999E746F7884
- CD9C981EEB4A9091547AF19181B382698E9D9EE0A838C7C9783A8A268AF6AEE
- D4FBEB A0168AF8FDD B2FC695881787AEDCD62F477C7DCEC9EBCA7B8594BBD95B
- E78F8CEBD2BD93D960BFAEAC18FAC0BB2524F15C44288903CD20B73E599E8A81
- ED16328235C610F15405FF08711EAF15D88A05

03884F3A9CCB5A0EE69CB4ACB5

- F71AC346CD0FC4652A89AFB37044887EC3907D37D01D1CEB0AD28E1A780D8E03.

For the WildRGBD dataset, the 20 test scenes we choose in O split are:

- BALL/SCENE_563
- APPLE/SCENE_234
- MICROWAVE/SCENE_143
- SCISSOR/SCENE_489

- BUCKET/SCENE_294
- KEYBOARD/SCENE_092
- SHOE/SCENE_868
- KETTLE/SCENE_399
- CLOCK/SCENE_524
- HAT/SCENE_039
- BACKPACK/SCENE_264
- SCISSOR/SCENE_958
- TRUCK/SCENE_232
- HANDBAG/SCENE_575
- PINEAPPLE/SCENE_182
- TRAIN/SCENE_033
- REMOTE_CONTROL/SCENE_453
- BOWL/SCENE_673
- TV/SCENE_062

Full test scenes are chosen for the remaining datasets.

Choice of input and target views. We follow the same setup for splits adopted from previous works, by using the same set of input and target views. For split defined ourselves, we detail the choice of views as below. For the WildRGB-D [37] dataset, which consists of scenes captured while orbiting around an object, we define two splits with different difficulty levels. O_e represents the easy set, where each scene is trimmed to one-third of the original sequence (*i.e.*, approximately 120 degrees of rotation). In contrast, O_h corresponds to the hard set, using the full original sequence (*i.e.*, approximately 360 degrees of rotation). We first uniformly subsample 21 frames from the scene, and randomly choose P frames as input views with the remaining frames as target views. For each scene from DL3DV-140 [39] and Tanks and Temples [41] datasets, we selected target frames by using every 8th frame of the original sequence. For the remaining frames, we applied K -means clustering ($K = 32$) on a 6-dimensional vector formed by concatenating the camera translation and the unit vector of the camera direction.

E. Additional Experiments

E.1. Qualitative Results

We provide additional single-view conditioning sampling results with a diverse set of camera motions and effects on a variety of image prompts: a text-prompted object-centric scene (Fig. 8), a text-prompted scene (Fig. 9), a real-world object-centric scene (Fig. 10), and a real-world scene (Fig. 11). SEVA demonstrates strong generalization, adapting robustly across a wide range of scenarios.

E.2. Quantitative Results

PSNR on RE10K. As shown in the main text, on the RE10K dataset, SEVA underperforms when $P = 1$. This issue arises from scale ambiguity in the model due to two fac-

Method	samples		3DGS	video
	PSNR \uparrow	TSED \downarrow	PSNR \uparrow	MS \uparrow
SEVA (one-pass)	15.73	115.1	16.03	95.39
SEVA (two-pass: nearest)	13.74	120.9	14.21	94.71
SEVA (two-pass: gt + nearest)	15.58	116.2	15.96	95.22
SEVA (two-pass: gt + interp)	15.66	120.1	15.98	95.56
SEVA	15.76	116.7	16.11	95.76
SEVA (+ temp.)	15.78	109.0	16.17	95.77

Table 2. **3D consistency (TSED \downarrow and PSNR \uparrow) and temporal quality (MS \uparrow) on trajectory NVS.** SEVA uses interp procedural sampling by default. *temp.* denotes the optional temporal pathway. *MS* denotes motion smoothness from VBench [42]. Results are reported on our split of DL3DV with $P = 3$.

tors: (1) it always takes in unit-normalized cameras during training, and (2) it is trained on multiple datasets with diverse scales. This challenge is most pronounced on RE10K, where panning motion dominates. Additionally, the absence of a second input view negates any scale relativity. To address this, for all results with $P = 1$, we sweep the *unit length for camera normalization* from 0.1 to 2.0 (with 2.0 used during training), selecting the best scale for each scene. On P split with $P = 2$, we observe diffusion models lag behind regression-based models that are advantageous in small-viewpoint interpolation. SEVA bridges this gap by improving upon the state-of-the-art diffusion-based model by +4.2 dB. On R split with $P = 3$, the advantage of SEVA is pronounced exceeding the previously best result by +1.9 dB. Notably, ViewCrafter excels on V split due to capacity taking in wide-aspect-ratio images and thus more input pixels than others with square images. The advantage of ViewCrafter on V split diminishes on CO3D since the majority of informative pixels are centrally located.

Ablation on two-pass procedural sampling. We conduct an ablation study comparing the default interp procedural sampling with one-pass sampling and alternative procedural sampling strategies. Beyond evaluating PSNR on individual views, we assess 3D consistency using the PSNR on 3D renderings of that same view and SED [32, 43] score. To compute the SED score, we first apply SIFT [44] to detect keypoints in two images. For each keypoint in the first image, we determine its corresponding epipolar line in the second image and measure the shortest distance to its match. Additionally, we report Motion Smoothness (MS) from VBench [42], a benchmark designed to evaluate temporal coherence in video generative models. As shown in Tab. 2, interp procedural sampling demonstrates a clear advantage over its alternatives, with the integration of the temporal pathway further reinforcing its superiority.

Method	dataset	OO3D GSO		RE10K				LLFF		DTU		CO3D		WRGBD		Mip360	DL3DV		T&T	
	split	O	O	D [32]	V [33]	P [10]	R [21]	R [21]	R [21]	R [21]	R [21]	V [33]	R [21]	O _e	O _h	R [21]	O	L [40]	V [33]	L [40]
	P	3	3	1	1	2	1	3	1	3	1	3	1	3	3	6	6	32	1	32
Regression-based models																				
Long-LRM [40]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0.262	-	0.375
MVSplat [11]	0.411	0.387	0.224	0.237	0.128	0.254	0.142	0.542	0.497	0.386	0.310	0.634	0.614	0.504	0.643	0.556	0.527	0.425	0.519	0.568
DepthSplat [45]	0.404	0.372	0.217	0.245	0.119	0.236	0.177	0.530	0.465	0.369	0.304	0.618	0.603	0.499	0.530	0.534	0.481	0.404	0.462	0.528
LVSM [6]	-	-	-	-	0.098	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Diffusion-based models																				
MotionCtrl [46]	-	-	0.500	0.386	-	-	-	-	-	-	-	0.443	-	-	-	-	-	-	0.473	-
4DiM [32]	-	-	0.302	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ViewCrafter [33]	0.427	0.379	0.220	0.178	0.203	0.287	0.164	0.620	0.435	0.485	0.272	0.324	0.513	0.324	0.639	0.464	0.558	-	0.283	-
SEVA	0.049	0.041	0.194	0.231	0.061	0.308	0.073	0.389	0.181	0.316	0.158	0.318	0.278	0.215	0.237	0.319	0.232	0.155	0.354	0.236
(a) LPIPS↓																				
Method	dataset	OO3D GSO		RE10K				LLFF		DTU		CO3D		WRGBD		Mip360	DL3DV		T&T	
	split	O	O	D [32]	V [33]	P [10]	R [21]	R [21]	R [21]	R [21]	R [21]	V [33]	R [21]	O _e	O _h	R [21]	O	L [40]	V [33]	L [40]
	P	3	3	1	1	2	1	3	1	3	1	3	1	3	3	6	6	32	1	32
Regression-based models																				
Long-LRM [40]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0.775	-	0.590
MVSplat [11]	0.554	0.621	0.788	0.769	0.869	0.812	0.857	0.283	0.358	0.576	0.624	0.403	0.370	0.405	0.368	0.312	0.487	0.512	0.394	0.314
DepthSplat [45]	0.636	0.689	0.801	0.745	0.887	0.820	0.824	0.299	0.396	0.601	0.638	0.429	0.402	0.436	0.417	0.324	0.513	0.564	0.413	0.359
LVSM [6]	-	-	-	-	0.906	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Diffusion-based models																				
MotionCtrl [46]	-	-	0.267	0.587	-	-	-	-	-	-	-	0.502	-	-	-	-	-	-	0.384	-
4DiM [32]	-	-	0.463	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ViewCrafter [33]	0.538	0.647	0.792	0.798	0.710	0.806	0.830	0.146	0.454	0.542	0.671	0.641	0.483	0.465	0.376	0.354	0.469	-	0.563	-
SEVA	0.935	0.942	0.615	0.693	0.847	0.700	0.892	0.384	0.602	0.652	0.750	0.585	0.647	0.670	0.646	0.395	0.546	0.661	0.437	0.505
(b) SSIM↑																				

Table 3. **LPIPS↓ (top) and SSIM↑ (bottom) on small-viewpoint set NVS.** For all results with $P = 1$, we sweep the unit length for camera normalization due to the model’s scale ambiguity. O_e and O_h denote the easy and hard split of our split. Underlined numbers are run by us using the officially released code.

LPIPS and SSIM. We provide additional quantitative evaluation results of our model against baselines on set NVS and trajectory NVS, measured using LPIPS [48] and SSIM [49], in Tabs. 3 to 5 and Appendix E.2.

E.3. Discussion

Zero-shot generalization of context window length T . We surprisingly find our model, though only trained on $T = 21$ frames, can generalize reasonably to larger T during sampling in the *semi-dense-view regime*. On our split of T&T for set NVS, we evaluate the predictions against ground truth in both sparse-view (*i.e.*, $1 \leq P \leq 8$) and semi-dense-view regime (*i.e.*, $9 \leq P$) using PSNR↑ and Image Quality↑ [42]. Image Quality refers to the distortion (*e.g.*, over-exposure, noise, blur) presented in the generated image. We experiment with different sampling strategies: one-pass sampling zero-shot extending the context window length T ; two-pass procedural sampling by first generating anchor views using nearest- K ($K < T$) input views and then interpolating anchor views into target views.

Our results are shown in Fig. 2. Procedural sampling with the nearest- K anchor views plateau after taking K views as input, indicating inefficiencies in procedural sampling and an inability to effectively utilize all available input views when $P > T$. Conversely, the metrics steadily improve with respect to the number of input frames for one-pass sampling with T extending to $P + Q$ in a zero-shot manner. However, we observe that this generalization fails in the sparse-view regime, resulting in blurry samples, as indicated by the low Image Quality when $P < 9$ and qualitative samples when $P = 3$. In the semi-dense-view setting, although quantitative metrics show minimal differences between one-pass and procedural sampling, we consistently observe that one-pass produces more 3D-consistent samples, as illustrated in the bottom-right figure.

Zero-shot generalization of image resolution. Surprisingly, we find our model, despite being trained only on square images with $H = W = 576$, generalizes well to different image resolution during sampling, similar to [50].

Method	dataset	OO3D		GSO	CO3D		WRGBD	Mip360		DL3DV	T&T				
	split	S [23]	S [23]	R [21]	O _h		O _h	R [21]		O		O			
	P	1	1	1	1	3		1	3	1	3	1	3	6	9
SV3D [23]		0.158	0.140	-	-	-	-	-	-	-	-	-	-	-	
DepthSplat [45]		0.610	0.543	0.756	0.732	0.588	0.691	0.491	0.580	0.405	0.774	0.706	0.611	0.487	
CAT3D [22]		-	-	-	-	-	-	0.488	-	-	-	-	-	-	
ViewCrafter [33]		0.634	0.559	0.789	0.775	0.603	0.723	0.540	0.616	0.576	0.755	0.671	0.604	0.546	
SEVA		0.160	0.137	0.445	0.423	0.289	0.573	0.364	0.484	0.316	0.571	0.463	0.387	0.328	

(a) LPIPS↓														
Method	dataset	OO3D	GSO	CO3D	WRGBD		Mip360		DL3DV		T&T			
	split	S [23]	S [23]	R [21]	O _h		R [21]		O		O			
	P	1	1	1	1	3	1	3	1	3	1	3	6	9
SV3D [23]		0.850	0.880	-	-	-	-	-	-	-	-	-	-	-
DepthSplat [45]		0.549	0.612	0.385	0.234	0.335	0.206	0.291	0.349	0.452	0.304	0.315	0.326	0.367
CAT3D [22]		-	-	-	-	-	-	0.294	-	-	-	-	-	-
ViewCrafter [33]		0.463	0.575	0.277	0.225	0.321	0.199	0.264	0.323	0.400	0.312	0.328	0.337	0.343
SEVA		0.857	0.873	0.536	0.505	0.603	0.282	0.377	0.360	0.480	0.342	0.385	0.427	0.452

(a) LPIPS↓

Table 4. **LPIPS↓ (top) and SSIM↑ (bottom) on large-viewpoint set NVS.** For all results with $P = 1$, we sweep the unit length for camera normalization due to the model’s scale ambiguity. Underlined numbers are run by us using the officially released code.

Method	split	small-viewpoint				large-viewpoint			
	dataset	V [33]				O			
	dataset	RE	CO3D	T&T	RE	DTU	WR	DL	T&T

(a) LPIPS↓									
MotionCtrl [46]	0.386	0.443	0.473	-	-	-	-	-	-
DepthSplat [45]	0.224	0.532	0.415	0.134	0.253	0.452	0.572	0.685	-
ViewCrafter [33]	0.178	0.283	0.324	<u>0.120</u>	0.187	<u>0.346</u>	<u>0.566</u>	<u>0.674</u>	-
SEVA	0.231	0.318	0.353	0.079	0.159	0.284	0.329	0.514	-
SEVA (+ temp.)	<u>0.228</u>	<u>0.312</u>	<u>0.356</u>	<u>0.078</u>	<u>0.156</u>	<u>0.280</u>	<u>0.328</u>	<u>0.510</u>	-

(a) LPIPS↓

Table 6. **LPIPS↓ (top) and SSIM↑ (bottom) on trajectory NVS.** For the V [33] split, $P = 1$ with unit length sweep; for the O split, $P = 3$. RE, WR, and DL denote RE10K, WRGBD, and DL3DV, respectively. Underlined numbers are run by us using the officially released code.

As shown in Fig. 3, SEVA can produce high-quality results

Method	small-viewpoint				large-viewpoint
	RE10K	LLFF	DTU	CO3D	Mip360
	1	3	6	9	1

(a) LPIPS↓					
ZipNeRF [47]	0.332	0.373	0.383	0.652	0.705
ZeroNVS [20]	0.422	0.512	0.223	0.566	0.680
ReconFusion [21]	0.144	0.203	0.124	0.398	0.585
CAT3D [22]	0.132	0.181	0.121	0.351	0.515
SEVA	0.078	0.164	0.107	0.256	0.435

(a) LPIPS↓

Table 5. **LPIPS↓ (top) and SSIM↑ (bottom) on 3DGS renderings for set NVS.** Results are reported on the ReconFusion [21] split with $P = 3$.

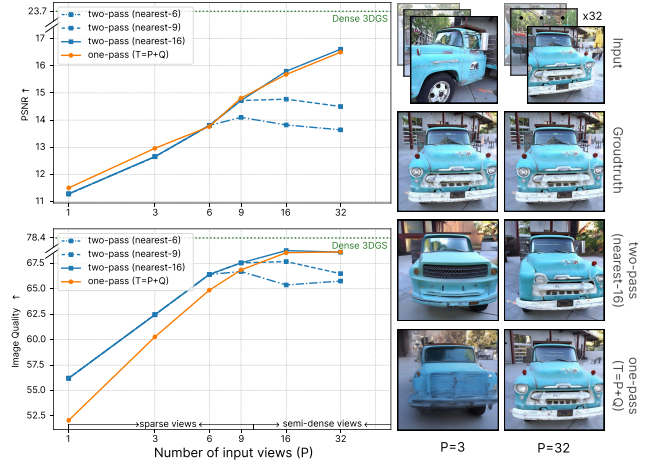


Figure 2. **Generation quality on the number of input views.** PSNR↑ (top) and Image Quality↑ (bottom) on set NVS. Results are reported on our split of T&T. Extending T to more input views in a zero-shot manner produces more consistent samples in the semi-dense-view regime. Dense 3DGS denotes results of [7] with full views.

in both portrait (16 : 9) and landscape (9 : 16) orientations of different image resolutions.

Guidance scale on generation uncertainty. We employ classifier-free guidance [51] (CFG) to enhance sampling quality. Empirically, we find that the CFG scale, a hyperparameter at test time, has a significant impact on the final

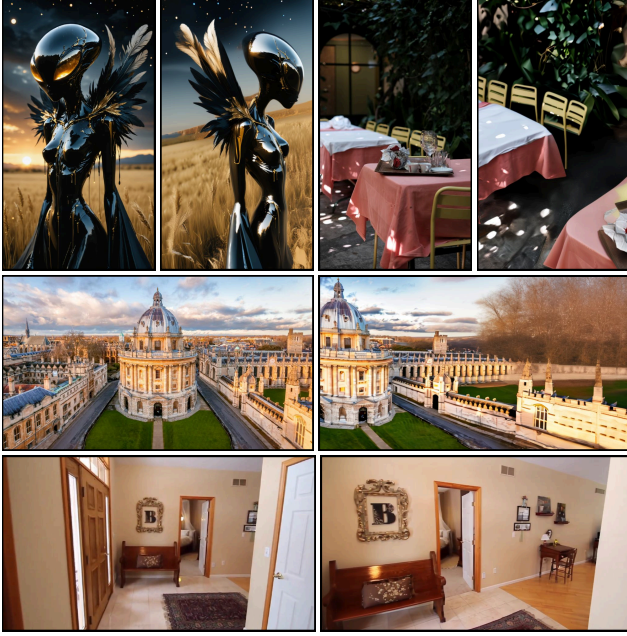


Figure 3. **Generation quality on different image resolutions.** Our model generalizes to different image resolution of varying aspect ratios, including both portrait (top) and landscape orientations (bottom). Results are presented as a pair of the input view and the target views.

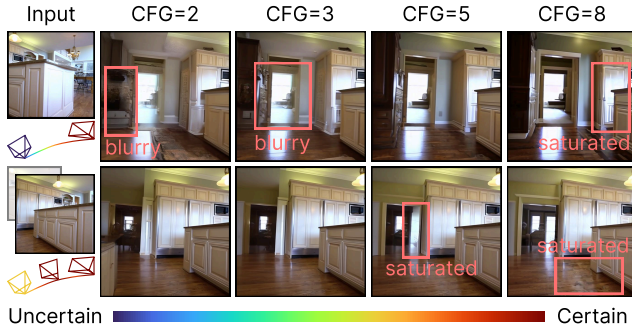


Figure 4. **Generation uncertainty on CFG.** The CFG scale should be increased as generation uncertainty rises. For single-view conditioning (top), a higher CFG scale is typically required, whereas few-view conditioning (bottom) benefits from a lower scale.

result [27], as shown in Fig. 4. Specifically, the optimal CFG scale is strongly correlated with the inherent uncertainty of the generation. When uncertainty is high (top row), a higher CFG scale (e.g., 5) is preferable to prevent excessive blurriness in the generated samples. Conversely, when uncertainty is low (bottom row), a lower CFG scale (e.g., 3) helps avoid oversaturation. In practice, setting the CFG scale between 2 and 5 consistently produces high-quality results across all our samples.



Figure 5. **Generation diversity in unseen regions.** Our model generates diverse samples by varying randomization seeds during the sampling process.

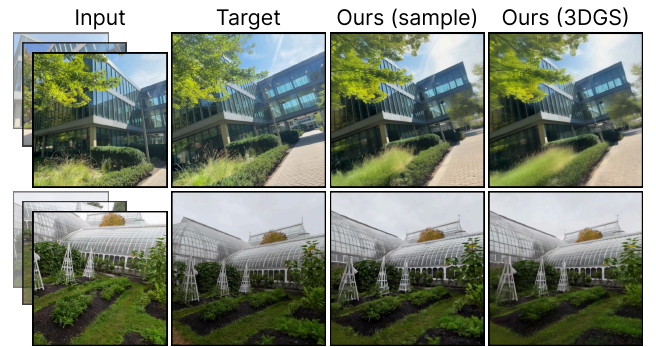


Figure 6. **3DGS versus samples.** The model generates consistent renderings that closely resemble those from 3DGS [7], with minimal perceptual differences.

Sampling diversity of unseen areas. Fig. 5 demonstrates the capability of the model to generate diverse and plausible predictions for unseen regions of input observations. In the first row, the input view depicts a frontal view of a classical statue. We sample multiple back views by varying the random seeds, producing distinct yet coherent interpretations of the unseen geometry and texture while preserving fidelity to the input. Similarly, in the second row, the model generates multiple plausible continuations of the scene given an input view of a scenic road, each reflecting unique variations in environmental and structural details. These results highlight the model’s ability to synthesize realistic and diverse outputs for occluded or ambiguous regions.

Samples versus 3DGS. We compare our samples to their 3DGS distillation on the O split of DL3DV, shown in Fig. 6. First, we note that our samples contain plausible hallucinations when uncertainty is high (first row, building on the right). Second, we note that our 3DGS renderings remain sharp and are close to the samples. These results suggest that our samples are 3D consistent enough.

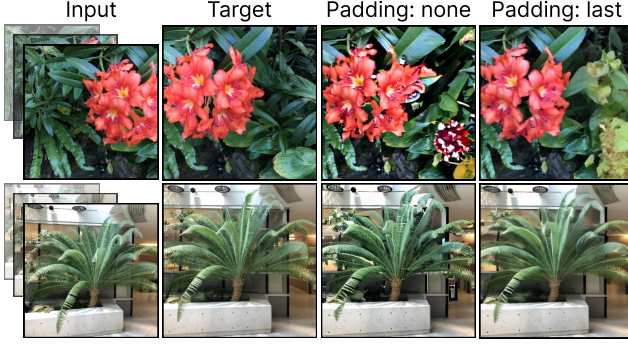


Figure 7. **Padding.** Padding the last elements within one forward reduces artifacts compared to changing T .

Padding T when $P + Q < T$. We analyze the effect of different padding strategies when $P + Q < T$ in Fig. 7. We observe that zero-shot generalization of T to $P + Q$ without padding leads to abnormal color overflows. This is in stark contrast to the excessive blurriness observed when generalizing T when $P + Q \gg T$ in sparse-view regime. Hypothetically, sampling with a T unseen during training induces a distribution shift in the attention scores [52]. Specifically, a smaller T sharpens the attention distribution, whereas a larger T disperses it. This shift may explain the contrasting behavior observed when using the model for sampling. Training the model with a dynamically varying T during training could mitigate this issue by exposing the model to a broader range of attention score distributions, improving generalization across different T .

Artifacts on long-trajectory NVS. We observe that the results tend to become increasingly saturated, particularly when the target views are far from the input views and share no content overlap, such as in open-ended exploration and navigation. The concurrent work [53] explores the concept of Diffusion Forcing [54] for long video rollouts, achieving high-generation quality. Applying diverse noise to the input views during training can be beneficial, as it enables the refinement of high-level details in all anchor views within the memory bank during sampling, thereby mitigating the accumulation of saturation. We leave this for future work.



Figure 8. **Diverse camera motions and effects.** Single-view conditioning with a text-prompted object-centric scene. The image is generated using SD 3.5 [55] with the text prompt, “A cute firefly dragon in its natural habitat.”

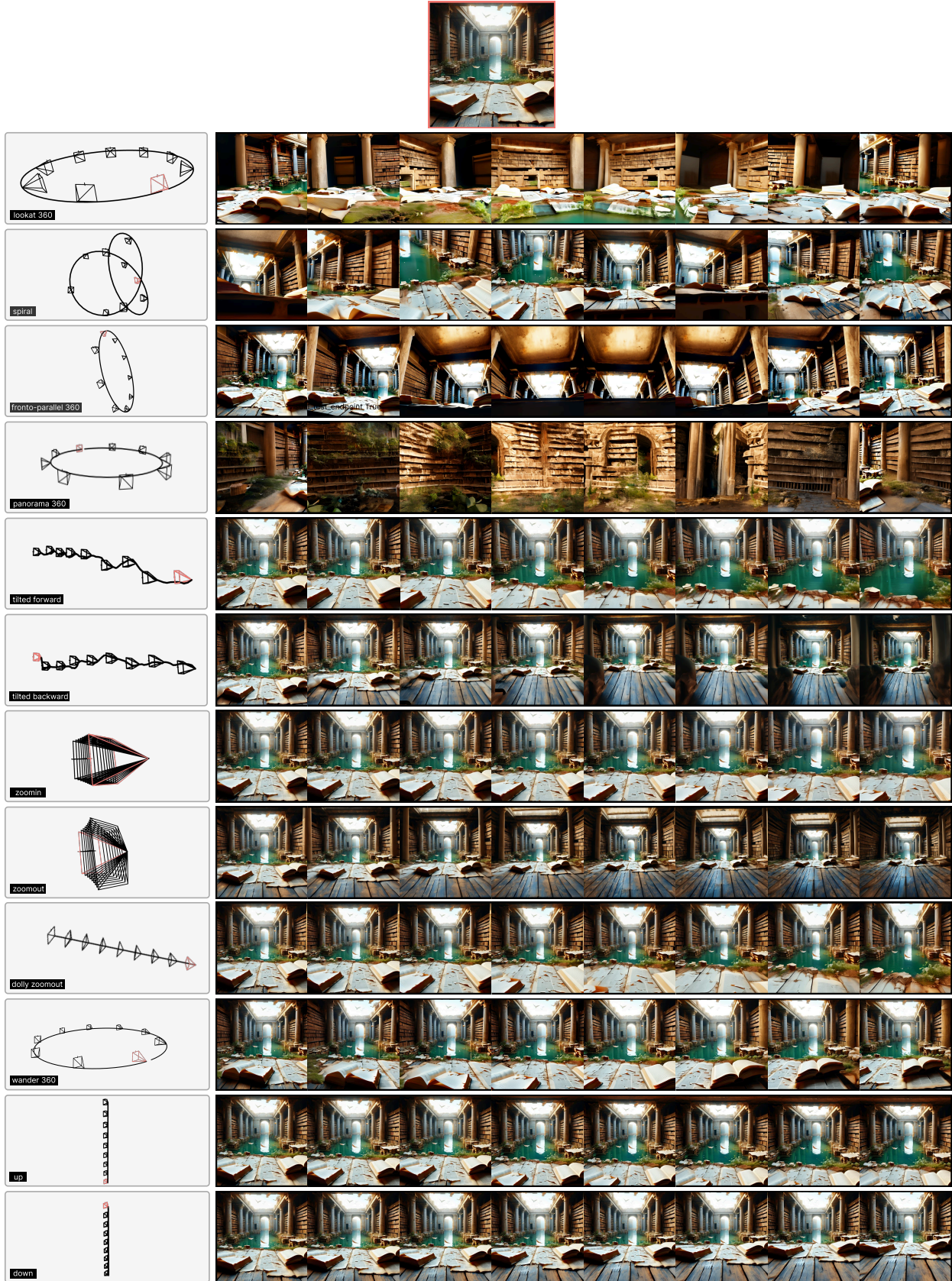


Figure 9. **Diverse camera motions and effects.** Single-view conditioning with a text-prompted scene. The image is generated using SD 3.5 [55] with the text prompt, “Wide view of the interior of the famed Library of Alexandria, elegantly set behind a time-worn wreckage by a lake, hinting at the relentless passage of time. The surroundings are lit by the light of a late afternoon sun, gently cast, immersing the area in a sentimental luminescence.”



Figure 10. Diverse camera motions and effects. Single-view conditioning with a real-life object-centric scene.

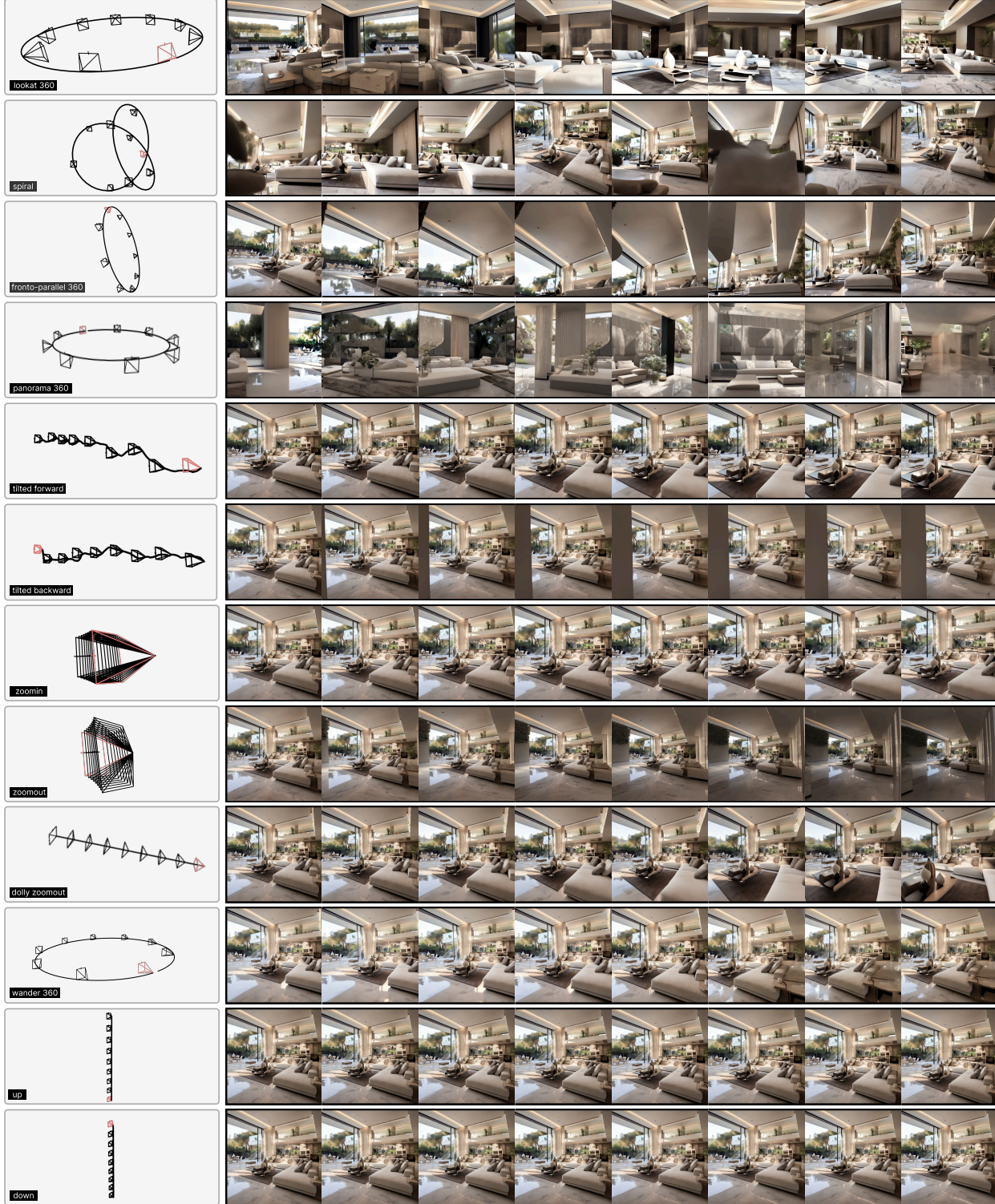
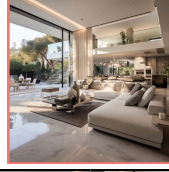


Figure 11. **Diverse camera motions and effects.** Single-view conditioning with a real-life scene.

References

- [1] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. In *European Conference on Computer Vision*, pages 405–421, 2020. 2
- [2] Gal Metzer, Elad Richardson, Or Patashnik, Raja Giryes, and Daniel Cohen-Or. Latent-NeRF for shape-guided generation of 3D shapes and textures. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12663–12673, 2023. 2
- [3] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 2
- [4] Yang Song and Stefano Ermon. Generative Modeling by Estimating Gradients of the Data Distribution. *arXiv:1907.05600*, 2019. 2
- [5] Mohammed Suhail, Carlos Esteves, Leonid Sigal, and Ameesh Makadia. Light field neural rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8269–8279, 2022. 2
- [6] Haian Jin, Hanwen Jiang, Hao Tan, Kai Zhang, Sai Bi, Tianyuan Zhang, Fujun Luan, Noah Snively, and Zexiang Xu. Lvsm: A large view synthesis model with minimal 3d inductive bias, 2024. 2, 6
- [7] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3D gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4), July 2023. 2, 7, 8
- [8] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelNeRF: Neural radiance fields from one or few images. In *CVPR*, 2021. 2
- [9] Stanislaw Szymanowicz, Christian Rupprecht, and Andrea Vedaldi. Splat image: Ultra-fast single-view 3d reconstruction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10208–10217, 2024. 2
- [10] David Charatan, Sizhe Li, Andrea Tagliasacchi, and Vincent Sitzmann. pixelsplat: 3d gaussian splats from image pairs for scalable generalizable 3d reconstruction. In *arXiv*, 2023. 2, 3, 4, 6
- [11] Yuedong Chen, Haoifei 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, 2025. 2, 6
- [12] 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. 2
- [13] Daniel Watson, William Chan, Ricardo Martin-Brualla, Jonathan Ho, Andrea Tagliasacchi, and Mohammad Norouzi. Novel view synthesis with diffusion models, 2022. 2
- [14] 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. *International Conference on Computer Vision*, 2023.
- [15] Ruoxi Shi, Hansheng Chen, Zhuoyang Zhang, Minghua Liu, Chao Xu, Xinyue Wei, Linghao Chen, Chong Zeng, and Hao Su. Zero123++: a single image to consistent multi-view diffusion base model. *arXiv preprint arXiv:2310.15110*, 2023. 2
- [16] Yichun Shi, Peng Wang, Jianglong Ye, Mai Long, Kejie Li, and Xiao Yang. MVDream: Multi-view diffusion for 3d generation. *arXiv preprint arXiv:2308.16512*, 2023. 2
- [17] Yuan Liu, Cheng Lin, Zijiao Zeng, Xiaoxiao Long, Lingjie Liu, Taku Komura, and Wenping Wang. SyncDreamer: Generating multiview-consistent images from a single-view image. *arXiv preprint arXiv:2309.03453*, 2023. 2
- [18] Antoine Mercier, Ramin Nakhli, Mahesh Reddy, Rajeev Yasarla, Hong Cai, Fatih Porikli, and Guillaume Berger. HexaGen3D: Stablediffusion is just one step away from fast and diverse Text-to-3D generation. *arXiv preprint arXiv:2401.07727*, 2024. 2
- [19] Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. Dreamfusion: Text-to-3D using 2D diffusion. *arXiv*, 2022. 2
- [20] Kyle Sargent, Zizhang Li, Tanmay Shah, Charles Herrmann, Hong-Xing Yu, Yunzhi Zhang, Eric Ryan Chan, Dmitry Lagun, Li Fei-Fei, Deqing Sun, et al. Zeronvs: Zero-shot 360-degree view synthesis from a single real image. *arXiv preprint arXiv:2310.17994*, 2023. 2, 7
- [21] Rundi Wu, Ben Mildenhall, Philipp Henzler, Keunhong Park, Ruiqi Gao, Daniel Watson, Pratul P Srinivasan, Dor Verbin, Jonathan T Barron, Ben Poole, et al. Reconfusion: 3d reconstruction with diffusion priors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21551–21561, 2024. 2, 3, 4, 6, 7
- [22] Ruiqi Gao*, Aleksander Holynski*, Philipp Henzler, Arthur Brussee, Ricardo Martin-Brualla, Pratul P. Srinivasan, Jonathan T. Barron, and Ben Poole*. Cat3d: Create anything in 3d with multi-view diffusion models. *arXiv*, 2024. 2, 3, 7
- [23] Vikram Voleti, Chun-Han Yao, Mark Boss, Adam Letts, David Pankratz, Dmitrii Tochilkin, Christian Laforte, Robin Rombach, and Varun Jampani. SV3D: Novel multi-view synthesis and 3D generation from a single image using latent video diffusion. In *European Conference on Computer Vision*, 2024. 2, 3, 4, 7
- [24] Sherwin Bahmani, Ivan Skorokhodov, Victor Rong, Gordon Wetzstein, Leonidas Guibas, Peter Wonka, Sergey Tulyakov, Jeong Joon Park, Andrea Tagliasacchi, and David B Lindell. 4d-fy: Text-to-4d generation using hybrid score distillation sampling. *arXiv preprint arXiv:2311.17984*, 2023. 2
- [25] Julius Plucker. Xvii. on a new geometry of space. *Philosophical Transactions of the Royal Society of London*, (155):725–791, 1865. 3
- [26] Chuanxia Zheng and Andrea Vedaldi. Free3D: Consistent novel view synthesis without 3D representation. *arXiv preprint arXiv:2312.04551*, 2023. 3
- [27] 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. 3, 8
- [28] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 3
- [29] Tong Wu, Jiarui Zhang, Xiao Fu, Yuxin Wang, Jiawei Ren, Liang Pan, Wayne Wu, Lei Yang, Jiaqi Wang, Chen Qian, et al. Omniobject3D: Large-vocabulary 3D object dataset for realistic perception, reconstruction and generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 803–814, 2023. 3, 4
- [30] Laura Downs, Anthony Francis, Nate Koenig, Brandon Kinman, Ryan Hickman, Krista Reymann, Thomas B McHugh, and Vincent Vanhoucke. Google Scanned Objects: A high-quality dataset of 3D scanned household items. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 2553–2560. IEEE, 2022. 3, 4
- [31] 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. 3, 4
- [32] Daniel Watson, Saurabh Saxena, Lala Li, Andrea Tagliasacchi, and David J Fleet. Controlling space and time with diffusion models. *arXiv preprint arXiv:2407.07860*, 2024. 3, 4, 5, 6
- [33] Wangbo Yu, Jinbo Xing, Li Yuan, Wenbo Hu, Xiaoyu Li, Zhipeng Huang, Xiangjun Gao, Tien-Tsin Wong, Ying Shan, and Yonghong Tian. ViewCrafter: Taming video diffusion models for high-fidelity novel view synthesis. *arXiv preprint arXiv:2409.02048*, 2024. 3, 4, 6, 7
- [34] Ben Mildenhall, Pratul P Srinivasan, Rodrigo Ortiz-Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. *ACM Transactions on Graphics (ToG)*, 38(4):1–14, 2019. 3, 4
- [35] Rasmus Jensen, Anders Dahl, George Vogiatzis, Engin Tola, and Henrik Aanaes. Large scale multi-view stereopsis evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 406–413, 2014. 3, 4
- [36] Jeremy Reizenstein, Roman Shapovalov, Philipp Henzler, Luca Sbordone, Patrick Labatut, and David Novotny. Common objects in 3d: Large-scale learning and evaluation of real-life 3d category reconstruction. In *International Conference on Computer Vision*, 2021. 3, 4
- [37] Hongchi Xia, Yang Fu, Sifei Liu, and Xiaolong Wang. Rgb-d objects in the wild: Scaling real-world 3d object learning from rgb-d videos, 2024. 3, 4, 5
- [38] Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. Mip-NeRF: A multiscale representation for anti-aliasing neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5855–5864, 2021. 3, 4
- [39] Lu Ling, Yichen Sheng, Zhi Tu, Wentian Zhao, Cheng Xin, Kun Wan, Lantao Yu, Qianyu Guo, Zixun Yu, Yawen Lu, et al. D13dv-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. 3, 4, 5
- [40] Chen Ziwen, Hao Tan, Kai Zhang, Sai Bi, Fujun Luan, Yicong Hong, Li Fuxin, and Zexiang Xu. Long-lrm: Long-sequence large reconstruction model for wide-coverage gaussian splats. *arXiv preprint arXiv:2410.12781*, 2024. 3, 4, 6
- [41] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics (ToG)*, 36(4):1–13, 2017. 3, 4, 5
- [42] Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, Yaohui Wang, Xinyuan Chen, Limin Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. VBench: Comprehensive benchmark suite for video generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. 5, 6
- [43] Jason J Yu, Fereshteh Forghani, Konstantinos G Derpanis, and Marcus A Brubaker. Long-term photometric consistent novel view synthesis with diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7094–7104, 2023. 5
- [44] Huiyu Zhou, Yuan Yuan, and Chunmei Shi. Object tracking using sift features and mean shift. *Computer vision and image understanding*, 113(3):345–352, 2009. 5
- [45] Haofei Xu, Songyou Peng, Fangjinhua Wang, Hermann Blum, Daniel Barath, Andreas Geiger, and Marc Pollefeys. Depthsplat: Connecting gaussian splatting and depth. *arXiv preprint arXiv:2410.13862*, 2024. 6, 7
- [46] Tsai-Shien Chen, Chieh Hubert Lin, Hung-Yu Tseng, Tsung-Yi Lin, and Ming-Hsuan Yang. Motion-conditioned diffusion model for controllable video synthesis. *arXiv preprint arXiv:2304.14404*, 2023. 6, 7
- [47] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Zip-nerf: Anti-aliased grid-based neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19697–19705, 2023. 7
- [48] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 6
- [49] 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. 6
- [50] 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. 6

- [51] Jonathan Ho and Tim Salimans. Classifier-Free Diffusion Guidance. *arXiv:2207.12598*, 2022. 7
- [52] Petar Veličković, Christos Perivolaropoulos, Federico Barbero, and Razvan Pascanu. softmax is not enough (for sharp out-of-distribution). *arXiv preprint arXiv:2410.01104*, 2024. 9
- [53] Kiwhan Song, Boyuan Chen, Max Simchowitz, Yilun Du, Russ Tedrake, and Vincent Sitzmann. History-guided video diffusion, 2025. 9
- [54] Boyuan Chen, Diego Marti Monso, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion forcing: Next-token prediction meets full-sequence diffusion, 2024. 9
- [55] Stability AI. Stable diffusion 3.5. <https://stability.ai/news/introducing-stable-diffusion-3-5>, 2024. 10, 11