DiffDreamer: Towards Consistent Unsupervised Single-view Scene Extrapolation with Conditional Diffusion Models

Shengqu Cai[1,2*] Eric Ryan Chan[1] Songyou Peng[2,3] Mohamad Shahbazi[2]
Anton Obukhov[2] Luc Van Gool[2,4] Gordon Wetzstein[1]
[1]Stanford University [2]ETH Zürich [3]MPI for Intelligent Systems, Tübingen [4]KU Leuven

Figure 1: **DiffDreamer** (Top) is a novel diffusion-based approach for scene extrapolation. It exhibits high spatio-temporal consistency, a desired property missing in prior art, such as InfNat-0 [39] (Bottom). We check for consistency by extracting keypoints from the sequences with COLMAP, resulting in point clouds of vastly different sizes and sparsity (Right).

Abstract

Scene extrapolation—the idea of generating novel views by flying into a given image—is a promising, yet challenging task. For each predicted frame, a joint inpainting and 3D refinement problem has to be solved, which is ill posed and includes a high level of ambiguity. Moreover, training data for long-range scenes is difficult to obtain and usually lacks sufficient views to infer accurate camera poses. We introduce DiffDreamer*, an unsupervised framework capable of synthesizing novel views depicting a long camera trajectory while training solely on internet-collected images of nature scenes. Utilizing the stochastic nature of the guided denoising steps, we train the diffusion models to refine projected RGBD images but condition the denoising steps on multiple past and future frames for inference. We demonstrate that image-conditioned diffusion models can effectively perform long-range scene extrapolation while preserving both consistency significantly better than prior GAN-based methods. DiffDreamer is a powerful and efficient solution for scene extrapolation, producing impressive results despite limited supervision. Project page:* https://primecai.github.io/diffdreamer.

1. Introduction

3D content creation tools are the foundation of emerging metaverse applications, among many others. Current approaches primarily rely on heavy manual labor, making the process expensive and inefficient. We set out to make 3D content creation automated and accessible. More specifically, an important downstream task we approach is *consistent scene extrapolation*. Given a single image and a long camera trajectory flying into the scene, the goal of *consistent scene extrapolation* is to synthesize a multiview-consistent 3D scene along the camera trajectory. In other words, we want to teach a machine to hallucinate content when flying into the image while maintaining multiview consistency, thereby extrapolating the scene realistically. Successfully addressing this task opens up a wide range of potential applications in virtual reality, 3D content creation, synthetic data creation, and 3D viewing platforms.

Consistent scene extrapolation is extremely challenging as it tries to tackle two difficult tasks simultaneously: consistent single-view novel view synthesis (NVS) and long-range extrapolation. Consistent single-view novel view synthesis has been studied for a long time. Many methods [66, 48] propose utilizing multi-view data to infer the

correspondences between frames, but they generally do not scale down to single- or few-view settings. Recently, there have also been attempts at single-view novel view synthesis. These methods mostly rely on learning a prior [99, 7] or utilizing geometry information [97, 96, 67]. However, they do not generalize to long-range camera movement, as the content of the original image is quickly lost when taking large camera movements. Current methods of long-range extrapolation [41, 39, 64, 35, 69] employ per-frame generation protocols, where the frames are generated in an auto-regressive feed-forward manner. The common downside of these methods is the lack of consistency between subsequent frames due to the per-frame refinement. A few recent methods [19, 4] attempt to generate a whole scene directly using implicit representations. However, this setting is computationally expensive, causing them to fail to achieve photo-realism even on low-resolution synthetic data.

Very recently, efforts have been made to perform scene extrapolation using pre-trained large-scale text-to-image diffusion models [60, 68]. This line of methods relies on prompt engineering and produces results with jittering between frames due to the lack of consistency enforcement. However, image-conditioned diffusion models are naturally suitable for the task of scene extrapolation, as the guided denoising process can be interpreted as a search in the latent space. Compared to feed-forward GAN-based methods [41, 39, 35], this allows the model to preserve high-level semantic meaning and low-frequency features while adding in high-frequency details and in-painting the missing parts. We hence utilize these strengths of diffusion models for scene exploration and further improve its 3D consistency.

In this paper, we propose *DiffDreamer*, a fully unsupervised method capable of consistent scene extrapolation given only a single image as input, and only internet photo collections as training data. Inspired by recent success in diffusion-based image refinement [43, 70], we formulate consistent scene extrapolation as learning a conditional diffusion model from images only. We train the conditional diffusion model to generate the frames in an iterative refinement manner, showing that this allows convergence towards a harmonic set of frames with high fidelity. The consistency achieved by our conditional diffusion model potentially enables one to fuse the outputs as a 3D model, e.g., a NeRF [48] with high consistency score [95]. A key advantage of diffusion models is the flexibility of modifying their sampling behaviors at inference. By stochastic conditioning at inference, we can condition the generation on multiple past and future frames and form a bidirectional pipeline, despite having only single images during training.

Experiments demonstrate that our framework allows one to synthesize a long-range fly-through sequence into an RGB image. We believe our framework not only serves as a starting point for consistent scene extrapolation and diffusion-model-based novel view synthesis but also scene extrapolation on more complex large-scale scenarios, such as autonomous driving scenes.

Our contributions include

- We introduce DiffDreamer, the first single-view scene extrapolation framework based on diffusion models for large-scale scenes.
- We propose an anchored sampling strategy and a lookahead mechanism for long-range scene extrapolation. Combined with diffusion models, we significantly alleviate the well-known domain drifting issue [41, 39] of scene extrapolation.
- We demonstrate a fully automated scene-level novel view synthesis pipeline using conditional diffusion models.

2. Related works

Novel view synthesis from multi-view images Research related to Novel View Synthesis (NVS) has a long history. Traditional multi-view NVS relies on inferring underlying geometry and interpolating the input images [57, 11, 17, 18, 21, 36, 14, 24, 38, 77, 106]. Recent successful attempts utilize deep learning methods to construct scene representations from multi-view data. These scene representations include but are not limited to: depth images [1, 46, 65, 98, 91, 80], multi-plane images [89, 104], voxels [42, 83], and implicit functions [84, 54, 56]. Among these representations, major progress has been made on radiance field approaches. Neural Radiance Fields (NeRFs) [48] have demonstrated encouraging progress for view synthesis by encoding color and transmittance in a multilayer perceptron, hence encoding a scene as an implicit representation. Using volumetric rendering, NeRF can perform photo-realistic novel view synthesis from only multi-view captured images and their poses. The outstanding performance of NeRF attracts tremendous efforts to improve its performance [2, 3, 92, 15], accelerate its training [49, 73, 87, 53], speed up rendering [40, 82], extend or generalize it towards other downstream tasks [44, 47, 8, 9, 76, 50, 25, 105], etc. Unlike these multi-view methods, we assume a single input image at the inference stage, where no geometry or interpolation can be inferred easily. Even with a single input image, our method can produce novel views faithfully.

Novel view synthesis from a single image There has been a vein of research on single-shot NVS. Some of them rely on geometry information or annotations [51, 81, 90]. However, geometry information and annotations are usually expensive to obtain for in-the-wild images. Other methods [16, 99, 7, 91, 84, 32, 88] relax this constraint by learning a prior filling in the missing information. These methods typically either only work well on simple objects (e.g.,

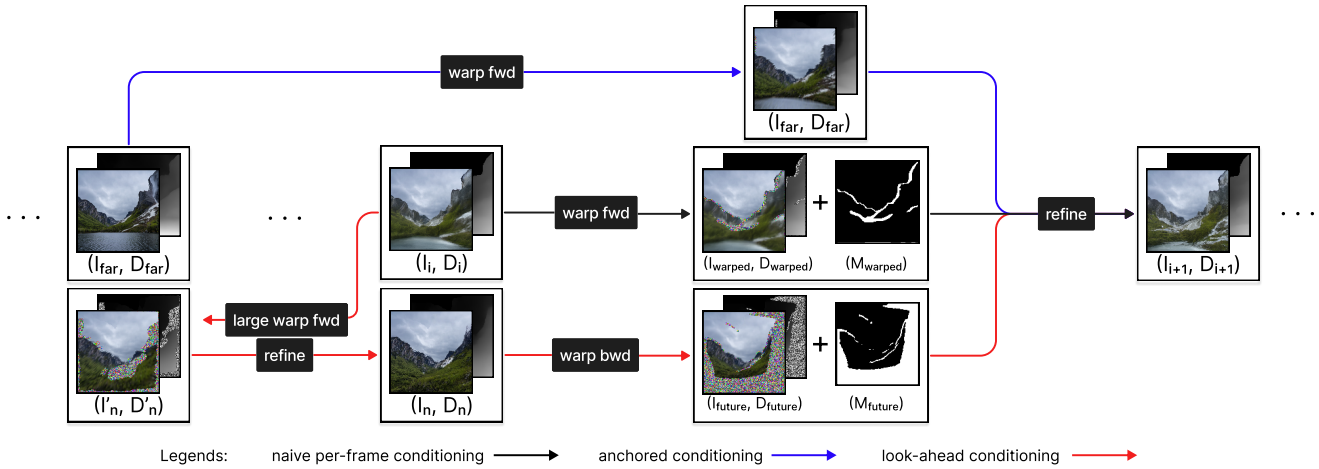

Figure 2: **Overview of our pipeline**. We train an image-conditional diffusion model to perform image-to-image refinement and inpainting given a corrupted image and its missing region mask. At inference, we perform stochastic conditioning on three conditionings: naive forward warping from the previous frame (black arrow), anchored conditioning by warping a further frame (blue arrow), and lookahead conditioning by warping a virtual future frame (red arrow). We repeat this render-refine-repeat pipeline to get sequences extrapolating a given image.

ShapeNet [10]) or restrict camera motions to small regions around the reference view. In contrast, we aim to relax such constraints and target long-range view extrapolation.

Scene extrapolation Long-range view extrapolation requires going beyond observations. With recent progress in generative modeling, several view extrapolation methods have emerged [6, 13, 37, 31, 34, 55, 78, 93, 102, 69]. Earlier methods such as SynSin [96] perform inpainting after reprojection, which struggles after a very limited range. A follow-up work, PixelSynth [67], works similarly to our DiffDreamer; it performs large-step image outpainting and accumulates a 3D point cloud for intermediate refinement and rendering. However, PixelSynth does not generalize to larger camera movements and requires a refinement module on top of the point cloud to enhance and inpaint, causing severely inconsistent intermediate view synthesis results.

Long-term path synthesis State-of-the-art methods such as InfNat [41], InfNat-0 [39], PathDreamer [35] and LOTR [64] deploy iterative training protocols and achieve perpetual view extrapolation for extremely long camera trajectories. However, these methods work in an auto-regressive per-frame generation framework. As a consequence, severe inconsistency can be observed from their rendered frames. Solving such inconsistency potentially requires generating an entire 3D world model, which is extremely computationally expensive, as shown by previous works [19, 4], whilst feed-forward per-frame generation methods [69, 41, 39] suffer from content drifting and both local and global inconsistency. Therefore, we attempt to benefit from a diffusion-based iterative refinement method to generate consistent content.

Diffusion models The recent development of diffusion models [27, 86] has pushed AI-driven content creation to another level. These methods learn to transform any data distribution into a prior distribution, then sample new data by first sampling a random latent vector from the prior distribution, followed by ancestral sampling with the reverse Markov chain, parameterized by deep neural networks. The powerful diffusion/denoising mechanism enables various traditional image-based tasks, including super-resolution [71, 28], inpainting [43, 70], and editing [45]. Their well-defined steady training protocols enhance the diffusion models' performance for large-scale training [68, 60]. Very recently, success has been made to lift the strength of diffusion models to the 3D domain [58, 95], further demonstrating the potential of 3D-based diffusion models.

We formulate our task similarly to InfNat [41] and InfNat-0 [39], but use a conditional diffusion model instead of GANs and take focus on achieving consistency.

3. DiffDreamer

Given a single input image, the aim of DiffDreamer is to generate a consistent and harmonic 3D camera trajectory that represents flying into the given image. DiffDreamer addresses this task by training a conditional diffusion model to perform image inpainting and refinement concurrently. The overview of our pipeline is illustrated in Fig. 2.

We synthesize frames of a fly-through video with three

steps: *render, refine and repeat*. In detail, given an RGB image I_{i} with its monocular predicted disparity D_{i} located at camera pose c_{i}, we can unproject the colored pixels into 3D space and render the projected view at the next camera pose $c_{\mathrm{i+1}}$ by a 3D renderer [62] π: $(I'_{\mathrm{i+1}}, D'_{\mathrm{i+1}}) = \pi(I_{\mathrm{i}}, D_{\mathrm{i}}, c_{\mathrm{i}}, c_{\mathrm{i+1}})$. With a refinement network F_θ, the warped RGBD image $(I'_{\mathrm{i+1}}, D'_{\mathrm{i+1}})$ can be inpainted and refined to get a fine next frame $(I_{\mathrm{i+1}}, I_{\mathrm{i+1}}) = F_\theta(I'_{\mathrm{i+1}}, D'_{\mathrm{i+1}})$, shown by the black arrow flow in Fig. 2. We then treat $(I_{\mathrm{i+1}}, D_{\mathrm{i+1}})$ as the starting view of the next step and perform the warping and refinement stages repeatedly, yielding a set of frames extrapolating the scene.

3.1. Training

Prior works [41, 39] model the training process exactly as the render-refine-repeat pipeline since the process is fully differentiable. However, this naive approach is not generalizable to diffusion models for two reasons: 1) the training process of diffusion models is split into different noise levels, and 2) sampling from a diffusion model requires up to thousands of denoising steps. This means we need to store thousands of intermediate steps and gradients to perform back-propagation, which is computationally infeasible.

The main function of the "repeat" step during training is to feed the network with its own outputs to ameliorate distribution drifting [41]. Therefore, it becomes critical to replace this step, especially when diffusion models are known to be sensitive to input distribution. Firstly, we create training pairs similar to [39] by projecting a ground truth RGBD image $(I_{\mathrm{gt}}, D_{\mathrm{gt}})$ at initial camera pose c_0 to a pseudo previous camera pose c_{pseudo}: $(I_{\mathrm{pseudo}}, D_{\mathrm{pseudo}}) = \pi(I_{\mathrm{gt}}, D_{\mathrm{gt}}, c_0, c_{\mathrm{pseudo}})$, then project back to c_0 to create a corresponding corrupted RGBD: $(I_{\mathrm{corrupted}}, D_{\mathrm{corrupted}}) = \pi(I_{\mathrm{pseudo}}, D_{\mathrm{pseudo}}, c_{\mathrm{pseudo}}, c_0))$. We thus obtain a pair of ground truth RGBD images $(I_{\mathrm{gt}}, D_{\mathrm{gt}})$ and its corrupted version $(I_{\mathrm{corrupted}}, D_{\mathrm{corrupted}})$, which involves missing parts and warping artifacts simulating a forward motion.

Having these paired data enables training an image-conditioned diffusion model $p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{m})$, where $\boldsymbol{x} = (I_{\mathrm{corrupted}}, D_{\mathrm{corrupted}})$, a corrupted version of ground truth image $\boldsymbol{y} = (I_{\mathrm{gt}}, D_{\mathrm{gt}})$ while $\boldsymbol{m}$ denotes the missing region mask from warping. We train the model with the following objective [27, 70]:

$$L(\theta) = \mathbb{E}_{(\boldsymbol{x},\boldsymbol{m},\boldsymbol{y})}\mathbb{E}_{\boldsymbol{\epsilon}}\mathbb{E}_{\gamma}\left[\|f_\theta(\boldsymbol{x}, \boldsymbol{m}, \widetilde{\boldsymbol{y}}, \gamma) - \boldsymbol{\epsilon}\|_2^2\right], \quad (1)$$

where $\widetilde{\boldsymbol{y}} = \sqrt{\gamma}\,\boldsymbol{y} + \sqrt{1-\gamma}\,\boldsymbol{\epsilon}$, $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$, and γ indicates the noise level. Note that since the diffusion model needs to concurrently learn inpainting and refinement, we additionally condition the neural network on the missing region mask to provide stronger guidance, following prior works [41, 39]. Similar to previous work [39], we assume that the sky region lies infinitely far away and does not change. Therefore, we inject noise only to the ground region to obtain $\widetilde{\boldsymbol{y}}$.

3.2. Inference

Diffusion models trained as described above perform well for a single forward step but do not generalize to long-term due to severe domain drifting after only a few iterations. This causes the extrapolated results to gradually drift away after only a handful of steps (see Sec. 4.2). We propose two strategies at the inference stage to counter this issue and preserve both local and global consistency.

3.2.1 Anchored conditioning

We introduce anchored conditioning, which conditions the diffusion model on long-range camera movements in order to enhance consistency over larger distances. As shown in prior work [95], it is feasible to naively approximate true auto-regressive sampling via stochastic conditioning for conditional diffusion models. While moving forward from camera pose c_{i} to $c_{\mathrm{i+1}}$, instead of strictly conditioning on the warped previous image $(I_{\mathrm{warped}}, D_{\mathrm{warped}}) = \pi(I_{\mathrm{i}}, D_{\mathrm{i}}, c_{\mathrm{i}}, c_{\mathrm{i+1}}))$ and mask during the inference denoising stage, we additionally select a frame $(I_{\mathrm{far}}, D_{\mathrm{far}})$ at a previous camera pose c_{far} further away from the current camera position. In practice, we empirically select the current frame $(I_{\mathrm{i}}, D_{\mathrm{i}})$ every 5 steps. We then perform stochastic conditioning on the warped previous frame $(I_{\mathrm{warped}}, D_{\mathrm{warped}})$ and an "anchored" frame by warping $(I_{\mathrm{far}}, D_{\mathrm{far}})$ to the desired camera pose: $(I_{\mathrm{anchored}}, D_{\mathrm{anchored}}) = \pi(I_{\mathrm{far}}, D_{\mathrm{far}}, c_{\mathrm{far}}, c_{\mathrm{i+1}})$. We perform this conditioning without specifying the missing region mask, as anchored conditioning requires longer-range warping, which may introduce more regions as missing, thus undermining the goal of long-term consistency. Conditioning on the warped previous frame naively encourages frame-to-frame consistency, while conditioning on a far-away frame offers long-term consistency. Stochastic conditioning on a largely warped image is also helpful with respect to domain drifting, as it is easier to simulate the same artifacts and blurriness during training by simply warping ground truth images equally further away. Thanks to the diffusion models' steady training protocol, refining and inpainting largely warped images with massive missing areas can be learned jointly during training.

3.2.2 Virtual lookahead conditioning

Prior works [41, 39] deploy per-frame generation; therefore, they suffer from severe inconsistency. A straightforward approach to solve this is to generate a scene representation directly, but this is extremely challenging and expensive to perform. While anchored conditioning solves parts

Figure 3: **Qualitative comparisons of InfNat-0 [39] and our DiffDreamer generation**, for which we ask the models to fly toward a target region and compare the outputs. Note that as InfNat-0 [39] is not 3D consistent and may need more steps even with identical input disparities and camera speed, we manually inserted more refinement steps to our DiffDreamer to ensure it is a fair comparison. Even so, we do not observe significant drifting from our DiffDreamer, while InfNat-0 [39] is incapable of preserving the input domain.

of the global consistency issue, warping an image distorts and stretches the texture, and blurs out fine details.

We notice that compared with flying into an image and refining the artifacts and blurriness, it is significantly easier to zoom out of an image and outpaint the missing regions without suffering from domain drift. This is due to the available regions preserving high-frequency details, which confer a strong signal for filling in missing regions.

Therefore, adding in a "lookahead" mechanism in diffusion models is helpful for both achieving long-term consistency and preventing domain drifting, as we can benefit from conditioning the generation on a future image, whose fine details preserve after warping to the current pose. While flying deep into an image, we observe that the generated content shares little overlap with the input image. Utilizing this fact, we can create a virtual view lying ahead for a sequence of views. With stochastic conditioning, this is as simple as additionally conditioning the generation on $(I_{\text{future}}, D_{\text{future}})$, acquired by warping a shared virtual view $(I_{\text{n}}, D_{\text{n}})$ lying ahead at a shared virtual future camera pose c_{n} warped to each camera pose c_{i}: $(I_{\text{future}}, D_{\text{future}}) = \pi(I_{\text{n}}, D_{\text{n}}, c_{\text{n}}, c_{\text{i}})$. In practice, we empirically generate c_{n} by taking a forward motion 10 times larger than a single step. We update the shared $(I_{\text{n}}, D_{\text{n}})$ every 10 steps and condition the future 10 frames on it. The shared virtual view can be flexibly generated by refining an available view to a future camera pose, a randomly generated view, or even another real image. To preserve consistency, we find it is suitable to warp the current frame $(I_{\text{i}}, D_{\text{i}})$ to a future camera pose c_{n} significantly beyond a single forward motion so that there is enough ambiguity, then refine to get virtual lookahead conditioning $(I_{\text{future}}, D_{\text{future}})$.

With the proposed anchored and virtual lookahead conditioning, we can formulate each denoising step going from camera pose c_{i} to $c_{\text{i}+1}$ as:

$$\boldsymbol{y}_{t-1} \leftarrow \frac{1}{\sqrt{\alpha_t}}\left(\boldsymbol{y}_t - \frac{1-\alpha_t}{\sqrt{1-\gamma_t}} f_\theta(\boldsymbol{x}_{\text{i}}, \boldsymbol{m}_{\text{i}}, \boldsymbol{y}_t, \gamma_t)\right) + \sqrt{1-\alpha_t}\boldsymbol{\epsilon}_t \tag{2}$$

at the inference stage, where $\boldsymbol{x}_{\text{i}}$ is a weighted selection of (0.5, 0.25, 0.25) among $(I_{\text{warped}}, D_{\text{warped}})$, $(I_{\text{anchored}}, D_{\text{anchored}})$ and $(I_{\text{future}}, D_{\text{future}})$, with $\boldsymbol{m}_{\text{i}}$ being the missing region mask. We apply classifier-free guidance [29] during inference as it encourages the denoising process to take more signals from the conditioning.

3.3. Training details

We carefully design the training protocols according to our inference strategies. Firstly, instead of assuming a fixed step size like previous works [41, 39], we randomly choose step size from a range $(-\text{s}, \text{s})$ while training the model to generalize to long-range conditioning, where we empirically choose $\text{s} = 20$. Note that we also train the model to fly out of the image for the purpose of lookahead conditioning. We find injecting random Gaussian noise into the missing regions rather than preserving the stretched details or masking out the missing regions to be very helpful, as it serves as an additional latent space that encourages diversity and effectively reduces the domain drifting between

forward motions and circular motions we used for creating pseudo training pairs. To add support for classifier-free guidance [30] at inference, we zero out all conditioning inputs with 10% probability during training.

4. Experiments

4.1. Evaluation

Datasets We report results on the LHQ [85] dataset, a collection of 90K nature landscape photos. Following prior work [39], we use the full data for training and 100 images provided by [39]'s authors, generated from a pre-trained StyleGAN2 [33], as the test set. Following [41], we also supply quantitative results on the ACID [41] dataset, with evaluation on 50 input images from its test set.

Evaluation metrics Evaluation of scene extrapolation frameworks is non-trivial as there is no single evaluation metric that covers every aspect of the generation quality. We follow the evaluation protocol of prior works [41, 39] on the rendered sequences. For that, we report Inception Score (IS) [72], Frechet Inception Distance (FID) [26] and Kernel Inception Distance (KID) [5] with scaling factor $\times 10$ computed using the torch-fidelity package [52]. We evaluate the models in two settings: a shorter range of 20 refinement steps, a middle range of 50 steps, and a longer range of 100 refinement steps. We additionally report 3D consistency scoring [95], a recent metric for evaluating 3D consistency. We compute this metric by generating a sequence of frames from an input, training a neural field [48] with a fraction of the generated frames, and calculating PSNR (Peak Signal to Noise Ratio), SSIM (Structural Similarity Index Measure) [94] and Perceptual Similarity (LPIPS) [103] against the held-out generated frames. We evaluate 3D consistency scoring over 10 sequences of 30 frames from 10 randomly selected input images. As our prior works [41, 39] also output disparity maps, we use a disparity supervised DVGOv2 [87] as the underlying neural field model. Following [95], we zero out the viewing direction conditioning to avoid overfitting to view-dependencies. To further assess local consistency, we compare the number of points from COLMAP [74, 75] reconstruction on the same 10 rendered video sequences, using the default automatic reconstruction without specifying the camera poses.

Qualitative results Qualitative comparisons are shown in Fig. 3, where we ask the rendering models to fly toward a target. We compare the intermediate frames rendered by the model. Interestingly, we find that even with identical input depth and step size, it takes significantly more refinement steps for InfNat-0 [39] to reach a target due to a tendency to render the far plane to be further than the mesh projection. In contrast, our DiffDreamer maintains high 3D consistency, and because it takes fewer steps to reach a target, it exhibits less drift. However, to ensure a fair comparison with [39] and to evaluate our model's drift under more refinement steps, we manually insert additional intermediate camera poses between each pair of nearby autocruise [41] poses so that the models conduct exactly the same number of refinement steps to reach the final frame.

	first 20 steps			first 50 steps			full 100 steps		
Method	FID↓	KID↓	IS↑	FID↓	KID↓	IS↑	FID↓	KID↓	IS↑
InfNat-0 [39]	39.45	0.12	2.80	**36.53**	**0.11**	2.79	**26.24**	**0.12**	2.72
DiffDreamer	**34.49**	**0.08**	**2.82**	38.86	0.12	**2.90**	51.0	0.28	**2.99**

Table 1: **Quantitative results on LHQ [85]** for 20 steps, 50 steps, and 100 steps generation. To our knowledge, InfNat-0 [39] is the only prior work capable of long-range view synthesis without supervision from sequential data or accurate ground truth depth.

	first 20 steps			first 50 steps			full 100 steps		
Method	FID ↓	KID ↓	IS ↑	FID ↓	KID ↓	IS ↑	FID ↓	KID ↓	IS ↑
SynSin [96]	79.58	0.63	1.90	96.37	0.78	1.71	108.95	1.06	1.75
PixelSynth [67]	89.63	1.10	1.23	97.14	1.32	1.42	107.61	1.20	1.63
3D Photos [80]	99.79	0.80	1.65	123.60	0.79	1.12	111.39	0.87	1.58
InfNat [41]	59.93	0.22	2.36	**57.47**	**0.26**	2.28	**48.27**	**0.27**	2.28
DiffDreamer	**52.81**	**0.12**	**2.69**	61.04	**0.26**	**2.86**	70.11	0.41	**2.82**

Table 2: **Quantitative results on ACID [41]** for 20 steps, 50 steps, and 100 steps generation. Note that all prior works require *posed* multi-view sequences for training, while our DiffDreamer is trained from single image collections.

We compare DiffDreamer's consistency against InfNat-0 in Fig. 5, along with a reference mesh projection created by warping the initial image into the final view according to the estimated disparity map. With identical disparity map and step size provided by InfNat-0 [39]'s authors, our model shows significantly better alignment with the projected mesh. While InfNat-0 [39] renders decent intermediate frames, it demonstrates only limited consistency with the mesh. Although it accurately models the expected movement of the foreground, the hill in the mid-distance remains static, whereas we expect it to fill the frame as the camera flies forward. This artifact may be due to a bias that encourages maintaining useful distant content while training scene extrapolation models. By enforcing the lookahead mechanism, which enforces future frames to be consistent with the mesh projection, our DiffDreamer does not suffer from this issue. We show an example of the detail-preserving ability of DiffDreamer in Fig 6.

We additionally visualize examples of scene extrapolation of 50 steps in Fig. 4. Despite only seeing single images during training, the learned generative prior enables DiffDreamer to perform long-range extrapolation.

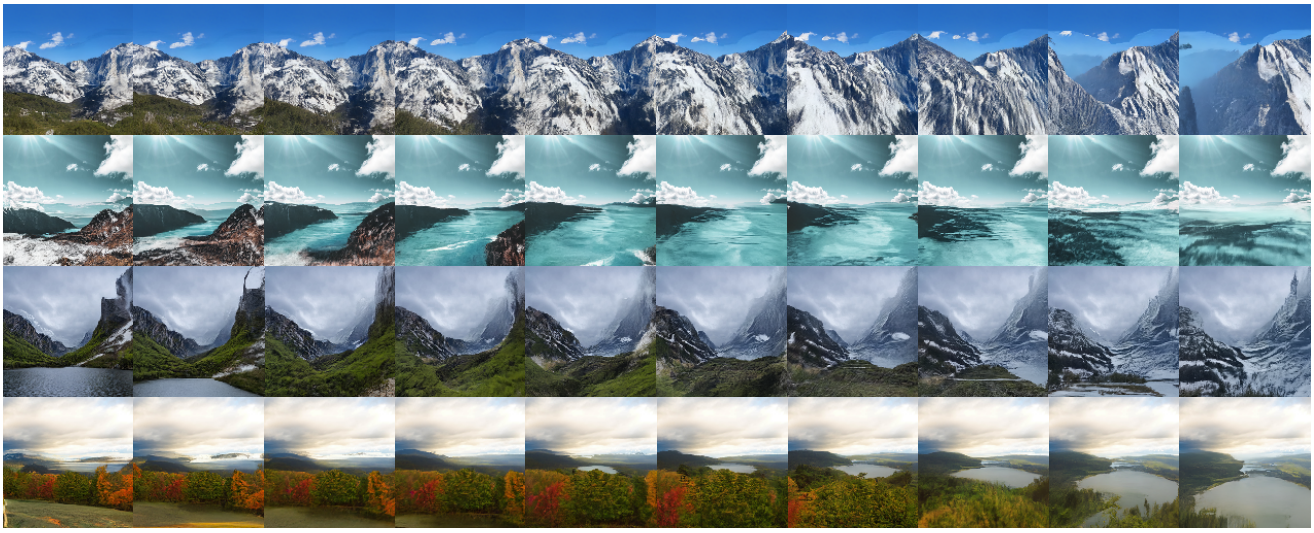

Figure 4: **Long-range view extrapolation** of over 50 steps forward.

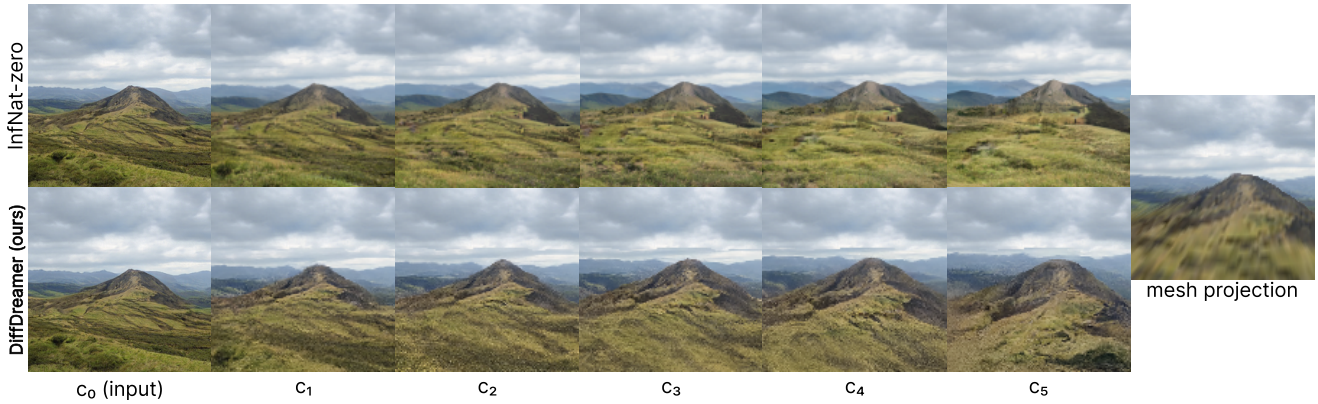

Figure 5: **Comparison of 3D consistency** achieved by our DiffDreamer and InfNat-0 [39], where we ask the camera to fly towards the top of the hill and show the intermediate renderings at camera positions c_0 to c_5.

Method	PSNR↑	SSIM↑	LPIPS↓
InfNat [41]	19.94±1.63	0.55±0.07	0.18±0.04
InfNat-0 [39]	18.92±1.42	0.41±0.08	0.20±0.02
DiffDreamer	**23.56**±3.30	**0.68**±0.04	**0.12**±0.02

Table 3: **3D consistency scoring [95]**, where we train disparity supervised DVGOv2 [87] using 10 sequences generated by the models and report the mean±std novel view synthesis metrics.

Method	Avg. points reconstructed
InfNat [41]	1476±477
InfNat-0 [39]	612±104
DiffDreamer	**3124**±622

Table 4: **Number of reconstructed points via COLMAP [74, 75]**, where we run COLMAP on 10 generated sequences, count the number of reconstructed points and report mean±std.

Quantitative results We show the quantitative evaluation on LHQ [85] in Tab. 1 and ACID [41] in Tab. 2. We observe that our 20-step generation outperforms prior works on all metrics by a relatively large margin. Our 50-step generation also has a significant advantage over prior works except for InfNat [41] and InfNat-0 [39], which are on par with our model. Our DiffDreamer's 100-step generation is not as good as [41] and [39] on FID [26] and KID [5] while achieving higher IS [72]. However, we achieve significantly better 3D consistency metrics, as shown in Tab. 3 and Tab. 4. Our model performs scene extrapolation based on the presented content from the input image very well, but we do not enforce it to generate diverse content. Therefore, the model may output blander frames after it goes significantly beyond the input. We also notice that our better 3D consistency makes autocruise [41] fail and hit the ground/hills

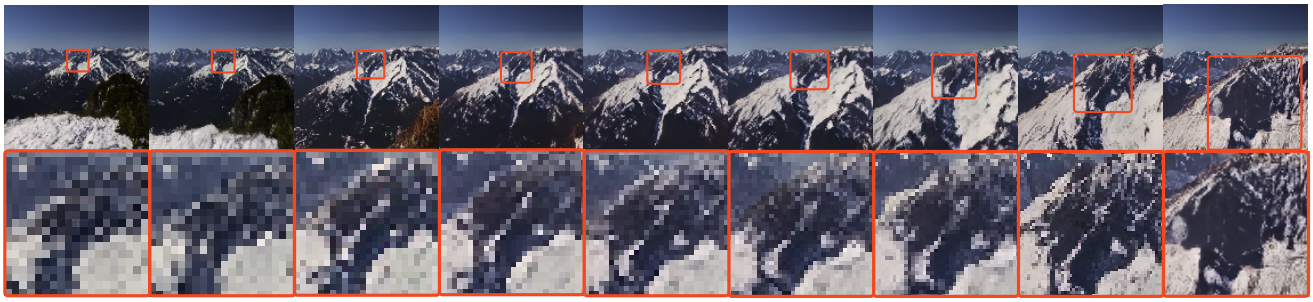

Figure 6: **Detail preservation** of over 30 steps. DiffDreamer can preserve details upon long-range extrapolation.

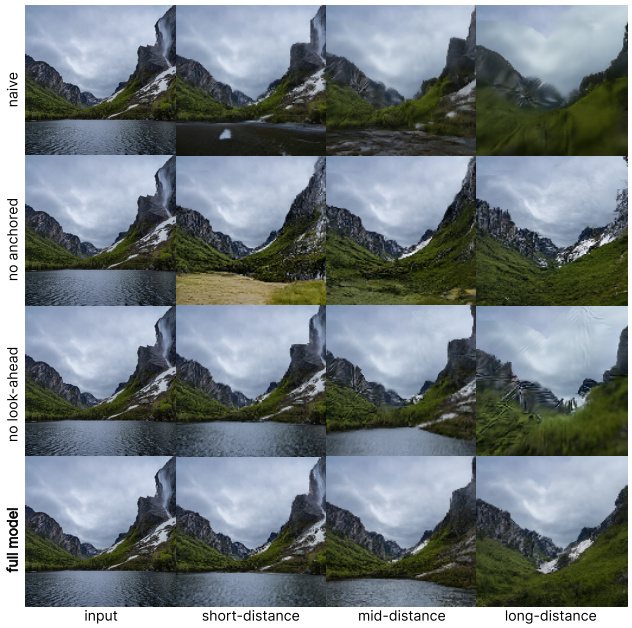

Figure 7: **Ablation study**, where we disabled our key building blocks. We observe clear artifacts (naive, no lookahead) or inconsistency (no anchored) in all ablations.

more often, constituting a large portion of the failed scenes. This is due to our model preserving geometry cues—it does not refine regions to be further away from their actual positions. Note that though InfNat [41] and InfNat-0 [39] can synthesize sequences with hundreds of frames, they resemble complete trade-offs with consistency. We argue that without accurate geometry preservation, scene extrapolation models [41, 39] will converge towards random latent space walk using pretrained GANs.

4.2. Ablation studies

We perform a thorough ablation study to verify our design choices, shown in Fig. 7. The setups are: 1) Naive auto-regressive: we first set up a baseline by performing the simplest per-step generation. The naive method fails after only a few refinement steps. This is due to the input domain drifting as observed in prior works [41, 39]. 2) Without anchored conditioning: next, we disable anchored conditioning: we observe more severe 3D inconsistency while moving forward, as the conditioning signal is purely from the past frame. 3) Without lookahead conditioning: we proceed with removing the lookahead mechanism. We observe significant domain drifting and artifacts as the model no longer takes advantage of the easier flying-out task. We supply the corresponding quantitative ablations in Appendix Sec. B.

5. Discussion

In this paper, we introduced DiffDreamer, a novel unsupervised pipeline based on conditional diffusion models for scene extrapolation. Diffdreamer can conduct scene extrapolation capable of "flying" into the image while training only from internet-collected single images. The key idea of DiffDreamer is to utilize a conditional diffusion model to simultaneously inpaint and refine a corrupted image obtained by warping a previous image. This is accomplished by training a conditional diffusion model that is capable of performing image-to-image translation under various corruption augmentations and utilizing stochastic conditioning to refine a corrupted image given multiple conditioning. Our model demonstrated comparable generation quality with GAN-based methods while maintaining significantly better consistency than prior works.

Limitation and future work DiffDreamer cannot synthesize novel views in real time due to the heavy inference of diffusion models. However, speeding up diffusion models' inference has been a very active area, and we expect advances in this area to speed up our approach directly. In addition, we do not enforce the diversity of content while going significantly beyond the input image, causing degradation for a true perpetual generation. We believe adding CLIP [59] conditioning to be a very exciting extension.

Conclusion Diffusion models are emerging as state-of-the-art generative 2D methods. DiffDreamer is the first approach to apply them to 3D scene extrapolation, demonstrating a high amount of view consistency that is crucial for many downstream tasks.

Acknowledgement We thank Zhengqi Li for providing all the technical details and data samples related to InfNat-0 [39]. Gordon Wetzstein was supported by Samsung, Stanford HAI, and a PECASE from the ARO.

References

[1] Kara-Ali Aliev, Artem Sevastopolsky, Maria Kolos, Dmitry Ulyanov, and Victor Lempitsky. Neural point-based graphics. In *ECCV*, 2020. 2

[2] 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 *ICCV*, 2021. 2

[3] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *CVPR*, 2022. 2

[4] Miguel Angel Bautista, Pengsheng Guo, Samira Abnar, Walter Talbott, Alexander Toshev, Zhuoyuan Chen, Laurent Dinh, Shuangfei Zhai, Hanlin Goh, Daniel Ulbricht, Afshin Dehghan, and Josh Susskind. Gaudi: A neural architect for immersive 3d scene generation. *arXiv*, 2022. 2, 3

[5] Mikołaj Bińkowski, Danica J Sutherland, Michael Arbel, and Arthur Gretton. Demystifying mmd gans. *arXiv*, 2018. 6, 7

[6] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. In *arXiv*, 2018. 3

[7] Shengqu Cai, Anton Obukhov, Dengxin Dai, and Luc Van Gool. Pix2nerf: Unsupervised conditional p-gan for single image to neural radiance fields translation. In *CVPR*, 2022. 2

[8] Eric Chan, Marco Monteiro, Petr Kellnhofer, Jiajun Wu, and Gordon Wetzstein. pi-gan: Periodic implicit generative adversarial networks for 3d-aware image synthesis. In *CVPR*, 2021. 2

[9] Eric R. Chan, Connor Z. Lin, Matthew A. Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas Guibas, Jonathan Tremblay, Sameh Khamis, Tero Karras, and Gordon Wetzstein. Efficient geometry-aware 3D generative adversarial networks. In *CVPR*, 2022. 2

[10] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. In *arXiv*, 2015. 3

[11] Gaurav Chaurasia, Sylvain Duchêne, Olga Sorkine-Hornung, and George Drettakis. Depth synthesis and local warps for plausible image-based navigation. *TOG*, 2013. 2

[12] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *CVPR*, 2017. 1

[13] Qifeng Chen and Vladlen Koltun. Photographic image synthesis with cascaded refinement networks. In *ICCV*, 2017. 3

[14] Shenchang Eric Chen and Lance Williams. View interpolation for image synthesis. In *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, 1993. 2

[15] Tianlong Chen, Peihao Wang, Zhiwen Fan, and Zhangyang Wang. Aug-nerf: Training stronger neural radiance fields with triple-level physically-grounded augmentations. In *CVPR*, 2022. 2

[16] Xu Chen, Jie Song, and Otmar Hilliges. Monocular neural image based rendering with continuous view control. In *ICCV*, 2019. 2

[17] Paul Debevec, Yizhou Yu, and George Borshukov. Efficient view-dependent image-based rendering with projective texture-mapping. In *Eurographics Workshop on Rendering Techniques*, 1998. 2

[18] Paul E Debevec, Camillo J Taylor, and Jitendra Malik. Modeling and rendering architecture from photographs: A hybrid geometry-and image-based approach. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, 1996. 2

[19] Terrance DeVries, Miguel Angel Bautista, Nitish Srivastava, Graham W. Taylor, and Joshua M. Susskind. Unconstrained scene generation with locally conditioned radiance fields. In *ICCV*, 2021. 2, 3

[20] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. In *NeurIPS*, 2021. 1

[21] Andrew Fitzgibbon, Yonatan Wexler, and Andrew Zisserman. Image-based rendering using image-based priors. *IJCV*, 2005. 2

[22] Rafail Fridman, Amit Abecasis, Yoni Kasten, and Tali Dekel. Scenescape: Text-driven consistent scene generation. *arXiv*, 2023. 1

[23] Kyle Genova, Forrester Cole, Aaron Maschinot, Aaron Sarna, Daniel Vlasic, and William T. Freeman. Unsupervised training for 3d morphable model regression. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 2

[24] Steven J Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F Cohen. The lumigraph. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, 1996. 2

[25] Zekun Hao, Arun Mallya, Serge Belongie, and Ming-Yu Liu. GANcraft: Unsupervised 3D Neural Rendering of Minecraft Worlds. In *ICCV*, 2021. 2

[26] 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. *NeurIPS*, 2017. 6, 7

[27] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *NeurIPS*, 2020. 3, 4

[28] Jonathan Ho, Chitwan Saharia, William Chan, David J Fleet, Mohammad Norouzi, and Tim Salimans. Cascaded diffusion models for high fidelity image generation. *JMLR*, 2022. 3

[29] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021. 5

[30] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. 6

[31] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, 2017. 3

[32] Wonbong Jang and Lourdes Agapito. Codenerf: Disentangled neural radiance fields for object categories. In *ICCV*, 2021. 2

[33] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. In *NeurIPS*, 2020. 6

[34] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *CVPR*, 2020. 3

[35] Jing Yu Koh, Honglak Lee, Yinfei Yang, Jason Baldridge, and Peter Anderson. Pathdreamer: A world model for indoor navigation. In *ICCV*, 2021. 2, 3

[36] Johannes Kopf, Fabian Langguth, Daniel Scharstein, Richard Szeliski, and Michael Goesele. Image-based rendering in the gradient domain. *TOG*, 2013. 2

[37] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *CVPR*, 2017. 3

[38] Marc Levoy and Pat Hanrahan. Light field rendering. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, 1996. 2

[39] Zhengqi Li, Qianqian Wang, Noah Snavely, and Angjoo Kanazawa. Infinitenature-zero: Learning perpetual view generation of natural scenes from single images. In *ECCV*, 2022. 1, 2, 3, 4, 5, 6, 7, 8, 9

[40] D. B.* Lindell, J. N. P.* Martel, and G. Wetzstein. Autoint: Automatic integration for fast neural volume rendering. In *CVPR*, 2021. 2

[41] 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 *ICCV*, October 2021. 2, 3, 4, 5, 6, 7, 8, 1

[42] Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural sparse voxel fields. *NeurIPS*, 2020. 2

[43] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *CVPR*, 2022. 2, 3

[44] Ricardo Martin-Brualla, Noha Radwan, Mehdi S. M. Sajjadi, Jonathan T. Barron, Alexey Dosovitskiy, and Daniel Duckworth. NeRF in the Wild: Neural Radiance Fields for Unconstrained Photo Collections. In *CVPR*, 2021. 2

[45] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. Sdedit: Guided image synthesis and editing with stochastic differential equations. In *ICLR*, 2021. 3

[46] Moustafa Meshry, Dan B. Goldman, Sameh Khamis, Hugues Hoppe, Rohit Pandey, Noah Snavely, and Ricardo Martin-Brualla. Neural rerendering in the wild. *CoRR*, 2019. 2

[47] Ben Mildenhall, Peter Hedman, Ricardo Martin-Brualla, Pratul P. Srinivasan, and Jonathan T. Barron. NeRF in the dark: High dynamic range view synthesis from noisy raw images. *CVPR*, 2022. 2

[48] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020. 1, 2, 6

[49] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *SIGGRAPH*, 2022. 2

[50] Michael Niemeyer and Andreas Geiger. Giraffe: Representing scenes as compositional generative neural feature fields. In *CVPR*, 2021. 2

[51] Simon Niklaus, Long Mai, Jimei Yang, and Feng Liu. 3d ken burns effect from a single image. *TOG*, 2019. 2

[52] Anton Obukhov, Maximilian Seitzer, Po-Wei Wu, Semen Zhydenko, Jonathan Kyl, and Elvis Yu-Jing Lin. High-fidelity performance metrics for generative models in pytorch, 2020. 6

[53] Anton Obukhov, Mikhail Usvyatsov, Christos Sakaridis, Konrad Schindler, and Luc Van Gool. Tt-nf: Tensor train neural fields. *arXiv*, 2022. 2

[54] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *CVPR*, 2019. 2

[55] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *CVPR*, 2019. 3

[56] Songyou Peng, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, and Andreas Geiger. Convolutional occupancy networks. In *ECCV*, 2020. 2

[57] Eric Penner and Li Zhang. Soft 3d reconstruction for view synthesis. *TOG*, 2017. 2

[58] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv*, 2022. 3

[59] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. *CoRR*, 2021. 8

[60] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv*, 2022. 2, 3

[61] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *TPAMI*, 2022. 1, 2

[62] Nikhila Ravi, Jeremy Reizenstein, David Novotny, Taylor Gordon, Wan-Yen Lo, Justin Johnson, and Georgia Gkioxari. Accelerating 3d deep learning with pytorch3d. *arXiv*, 2020. 4

[63] Fitsum Reda, Janne Kontkanen, Eric Tabellion, Deqing Sun, Caroline Pantofaru, and Brian Curless. Film: Frame interpolation for large motion. In *European Conference on Computer Vision (ECCV)*, 2022. 1

[64] Xuanchi Ren and Xiaolong Wang. Look outside the room: Synthesizing a consistent long-term 3d scene video from a single image. In *CVPR*, 2022. 2, 3

[65] Gernot Riegler and Vladlen Koltun. Free view synthesis. In *ECCV*, 2020. 2

[66] Gernot Riegler and Vladlen Koltun. Stable view synthesis. In *CVPR*, 2021. 1

[67] Chris Rockwell, David F. Fouhey, and Justin Johnson. Pixelsynth: Generating a 3d-consistent experience from a single image. In *ICCV*, 2021. 2, 3, 6

[68] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. 2, 3

[69] Robin Rombach, Patrick Esser, and Björn Ommer. Geometry-free view synthesis: Transformers and no 3d priors, 2021. 2, 3

[70] Chitwan Saharia, William Chan, Huiwen Chang, Chris Lee, Jonathan Ho, Tim Salimans, David Fleet, and Mohammad Norouzi. Palette: Image-to-image diffusion models. In *SIGGRAPH*, 2022. 2, 3, 4, 1

[71] Chitwan Saharia, Jonathan Ho, William Chan, Tim Salimans, David J Fleet, and Mohammad Norouzi. Image super-resolution via iterative refinement. *TPAMI*, 2022. 3

[72] Tim Salimans, Ian J. Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *CoRR*, 2016. 6, 7

[73] Sara Fridovich-Keil and Alex Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. In *CVPR*, 2022. 2

[74] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, 2016. 6, 7

[75] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *ECCV*, 2016. 6, 7

[76] Katja Schwarz, Yiyi Liao, Michael Niemeyer, and Andreas Geiger. Graf: Generative radiance fields for 3d-aware image synthesis. In *NeurIPS*, 2020. 2

[77] Steven M Seitz, Brian Curless, James Diebel, Daniel Scharstein, and Richard Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *CVPR*, 2006. 2

[78] Tamar Rott Shaham, Tali Dekel, and Tomer Michaeli. Singan: Learning a generative model from a single natural image. *ICCV*, 2019. 3

[79] Tianyang Shi, Zhengxia Zou, Xinhui Song, Zheng Song, Changjian Gu, Changjie Fan, and Yi Yuan. Neutral face game character auto-creation via pokerface-gan. In *arXiV*, 2020. 2

[80] Meng-Li Shih, Shih-Yang Su, Johannes Kopf, and Jia-Bin Huang. 3d photography using context-aware layered depth inpainting. In *CVPR*, 2020. 2, 6

[81] Daeyun Shin, Zhile Ren, Erik B Sudderth, and Charless C Fowlkes. 3d scene reconstruction with multi-layer depth and epipolar transformers. In *CVPR*, 2019. 2

[82] Vincent Sitzmann, Semon Rezchikov, William T. Freeman, Joshua B. Tenenbaum, and Fredo Durand. Light field networks: Neural scene representations with single-evaluation rendering. In *NeurIPS*, 2021. 2

[83] Vincent Sitzmann, Justus Thies, Felix Heide, Matthias Nießner, Gordon Wetzstein, and Michael Zollhofer. Deepvoxels: Learning persistent 3d feature embeddings. In *CVPR*, pages 2437–2446, 2019. 2

[84] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene representation networks: Continuous 3d-structure-aware neural scene representations. In *NeurIPS*, 2019. 2

[85] Ivan Skorokhodov, Grigorii Sotnikov, and Mohamed Elhoseiny. Aligning latent and image spaces to connect the unconnectable. *arXiv*, 2021. 6, 7, 1, 2

[86] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *ICML*, 2015. 3

[87] Cheng Sun, Min Sun, and Hwann-Tzong Chen. Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In *CVPR*, 2022. 2, 6, 7

[88] Alex Trevithick and Bo Yang. Grf: Learning a general radiance field for 3d representation and rendering. In *ICCV*, 2021. 2

[89] Richard Tucker and Noah Snavely. Single-view view synthesis with multiplane images. In *CVPR*, 2020. 2

[90] Shubham Tulsiani, Saurabh Gupta, David Fouhey, Alexei A. Efros, and Jitendra Malik. Factoring shape, pose, and layout from the 2d image of a 3d scene. In *CVPR*, 2018. 2

[91] Shubham Tulsiani, Richard Tucker, and Noah Snavely. Layer-structured 3d scene inference via view synthesis. In *ECCV*, 2018. 2

[92] Dor Verbin, Peter Hedman, Ben Mildenhall, Todd Zickler, Jonathan T Barron, and Pratul P Srinivasan. Ref-nerf: Structured view-dependent appearance for neural radiance fields. *arXiv*, 2021. 2

[93] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *CVPR*, 2018. 3

[94] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 2004. 6

[95] Daniel Watson, William Chan, Ricardo Martin-Brualla, Jonathan Ho, Andrea Tagliasacchi, and Mohammad Norouzi. Novel view synthesis with diffusion models. *arXiv*, 2022. 2, 3, 4, 6, 7

[96] Olivia Wiles, Georgia Gkioxari, Richard Szeliski, and Justin Johnson. SynSin: End-to-end view synthesis from a single image. In *CVPR*, 2020. 2, 3, 6

[97] Dejia Xu, Yifan Jiang, Peihao Wang, Zhiwen Fan, Humphrey Shi, and Zhangyang Wang. Sinnerf: Training neural radiance fields on complex scenes from a single image. *arXiv*, 2022. 2

[98] Jae Shin Yoon, Kihwan Kim, Orazio Gallo, Hyun Soo Park, and Jan Kautz. Novel view synthesis of dynamic scenes with globally coherent depths from a monocular camera. *CoRR*, 2020. 2

[99] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelNeRF: Neural radiance fields from one or few images. In *CVPR*, 2021. 2

[100] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Free-form image inpainting with gated convolution. *arXiv*, 2018. 1

[101] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Generative image inpainting with contextual attention. *arXiv*, 2018. 1

[102] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. In *ICML*, 2019. 3

[103] 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. 6

[104] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. In *SIGGRAPH*, 2018. 2

[105] Zihan Zhu, Songyou Peng, Viktor Larsson, Weiwei Xu, Hujun Bao, Zhaopeng Cui, Martin R. Oswald, and Marc Pollefeys. Nice-slam: Neural implicit scalable encoding for slam. In *CVPR*, 2022. 2

[106] C. Lawrence Zitnick, Sing Bing Kang, Matthew Uyttendaele, Simon Winder, and Richard Szeliski. High-quality video view interpolation using a layered representation. *TOG*, 2004. 2

DiffDreamer: Towards Consistent Unsupervised Single-shot Scene Extrapoaltion with Conditional Diffusion Models

Supplementary Material

A. Additional qualitative results

Figures 11, 12, 13, and 14 show additional scene extrapolation results from our model with 50 steps of forward motion. The task of scene extrapolation has a multi-modal nature: given a single input image, there could be infinite ways of generation. Therefore, we show multiple rendering trajectories of over 50 steps for each input image and supporting videos with framerates upsampled using [63] (note that the videos are rendered at 128×128 and may appear blurry under higher resolution). To encourage diversity and prevent hitting mountains/the ground while generating longer sequences, we can additionally condition the diffusion model on randomly selected patterns from the input image while generating the pseudo future frames with a weight of 0.2. We select these patterns by simply performing free-form brush stroke masking, using the algorithm provided in [100, 101] and refer to this diversity-focused version as "DiffDreamer-diverse", encouraging diversity over long-range at a price of trading-off consistency. We supply results for this diversity-focused setting and show frames from a generated 500-step sequence in Fig. 8.

B. Additional quantitative results

We supply quantitative ablation comparisons including DiffDreamer-diverse on LHQ [85] in Tab. 5, and additional quantitative results of DiffDreamer-diverse on ACID [41] in Tab. 6.

	20 steps			50 steps			100 steps			COLMAP
Method	FID ↓	KID ↓	IS ↑	FID ↓	KID ↓	IS ↑	FID ↓	KID ↓	IS ↑	
InfNat-0	39.45	0.12	2.80	36.53	0.11	2.79	26.24	**0.12**	2.72	612
Auto-regressive	70.53	0.53	1.99	77.81	0.63	1.91	90.69	0.81	2.14	2030
No anchored	38.41	0.17	2.70	46.40	0.24	2.63	58.67	0.40	2.79	1543
No lookahead	68.30	0.46	1.76	75.18	0.74	1.85	92.85	0.81	2.06	2457
DiffDreamer	**34.49**	**0.08**	2.82	38.86	0.12	2.90	51.0	0.28	2.99	**3124**
DiffDreamer-diverse	34.92	0.09	**3.19**	**30.78**	**0.10**	**3.27**	**24.04**	**0.12**	**3.26**	1403

Table 5: Quantitative ablation studies.

	20 steps			50 steps			100 steps			COLMAP
Method	FID ↓	KID ↓	IS ↑	FID ↓	KID ↓	IS ↑	FID ↓	KID ↓	IS ↑	
InfNat	59.93	0.22	2.36	57.47	0.26	2.28	48.27	0.27	2.28	1476
DiffDreamer	52.81	**0.12**	**2.69**	61.04	0.26	**2.86**	70.11	0.41	**2.82**	**3423**
DiffDreamer-diverse	**51.28**	0.15	2.37	**44.44**	**0.19**	2.40	**42.97**	**0.21**	2.43	1883

Table 6: Quantitative comparison of DiffDreamer-diverse's performance on ACID [41].

C. Flying-out

Even though we do not design our model specifically for flying-out setting, DiffDreamer has a significant advantage over naïve autoregression. Since we are working on outdoor scenes, dramatic depth discontinuities will appear [22]. This is especially obvious when the flying-out motion is not just a straight translation. Our bi-directional method is a good counter to this issue since future frame guidance and simultaneous refinement can alleviate the artifacts. We show example flying-out sequences in Fig. 10 and include accompanying videos with 100 steps.

D. Technical details

We use the U-Net backbone from [20] and train all models for 1M iterations with a mini-batch size of 128. We trained our model for roughly a week and 3 days respectively for LHQ [85] and ACID [41], on 2 NVIDIA RTX 8000 GPUs. We compare against the released pretrained InfNat and InfNat-zero models, which were trained for 8 days on 10 GPUs, and 6 days on 8 GPUs respectively. We build our model on top of Palette [70] and use the Adam optimizer with a learning rate of 1e-4 and a 10k linear learning rate warm-up schedule. We also employ 0.9999 EMA for our model. During both training and inference, we use a linear noise schedule of (1e-6, 0.01) with 2000 time steps. Following prior works [41, 39], we extract monocular-predicted disparity maps with MiDaS [61], and sky region masks using DeepLab [12]. We adopt the autocruise algorithm from [41] to sample the camera path for both training and inference. The autocruise algorithm uses the disparity map to estimate the skyline and horizon, then generate a camera trajectory that avoids hitting the ground or hills. We follow [39] during inference and use a camera speed of 0.1875. We train and evaluate our model on image resolution of 128×128 to be consistent with prior work [39].

E. Autocruise specifics

We use the autocruise algorithm from [41] to generate camera trajectories for both training and evaluation. As we only have raw images as training data, whose intrinsics are unknown and cannot be easily inferred, we follow [39] and randomly sample the field of view (FoV) between 45° and 70°, and fix to 55° during testing. Autocruise algorithm deploys a mechanism to predict the next camera pose by encouraging the next view to have a τ_{sky} fraction of sky regions (determined by thresholding disparity less than 0.08) and a fraction of τ_{near} fraction of nearby regions (determined by thresholding disparity larger than 0.4). We follow

Figure 8: **Perpetual view synthesis** of a sequence over 500 steps.

Figure 9: **Failure cases** when the model's output is not diverse enough to support future frames (left) or the autocruise algorithm gets too close to the mountains/ground (right).

[39] to uniformly sample τ_{near} from [0.2, 0.4] and τ_{sky} from [0.25, 0.45] during training, and fix them to be 0.25 and 0.1 respectively during inference. In contrast to [41, 39], which only moves a small fraction $\tau_{\text{lerp}} = 0.05$ of the way to the target directions at each frame to ensure smooth camera pose changing, we only use $\tau_{\text{lerp}} = 0.05$ during inference of our next frame and increase $\tau_{\text{lerp}} = 0.3$ for generating the pseudo future frame. We uniformly sample τ_{lerp} from [0.0, 0.3] during training. We direct readers to [41, 39] for further specifics of the autocruise algorithm.

F. Mesh renderer specifics

We use a PyTorch implementation [79] of a 3D mesh renderer [23]. Following [41], each pixel is projected into the 3D space using its disparity and is then treated as a vertex connected with its neighbors to form a triangle mesh. To obtain the missing region masks, we follow [41] and threshold the gradient of the input disparity by 0.3 to make a mask, which refers to the regions with sharp disparity change. We project the mask to target the camera pose to get the final missing region mask.

G. Dataset pre-processing

Both of the LHQ [85] dataset and the ACID [41] dataset contains many samples unsuitable for training scene extrapolation models. This includes images focusing on the foreground and images of the ground, with camera poses pointed downward. Following [39], we filter out images whose minimum MiDaS [61] predicted disparity value is larger than 200.

H. Failure cases

There are two main causes for failures. First, we do not enforce diversity of outputs. During training, the model always sees real images. This means during our pseudo pairs generation, the corrupted version of the ground truth image will still be diverse, even if it is under a lower frequency due to warping artifacts. However, while we are going significantly beyond the input image's content, any future frame will solely rely on the model's outputs, which may not exhibit enough diverse content for moving forward. We show an example of this case in Fig. 9. We believe it is exciting to extend DiffDreamer to support vector conditioning, e.g., CLIP embedding conditioning, to enforce output diversity.

Second, as our model has significantly better geometry alignment than [41], the autocruise algorithm fails more often, causing the camera trajectory to hit mountains or the ground, despite our best efforts in tuning its parameters. We show an example of this failure case in Fig. 9.

Figure 10: **Flying-out** 100 steps of the input images.

Figure 11: **Additional qualitative results:** Six distinct realizations, synthesized over 50 steps of forward motion.

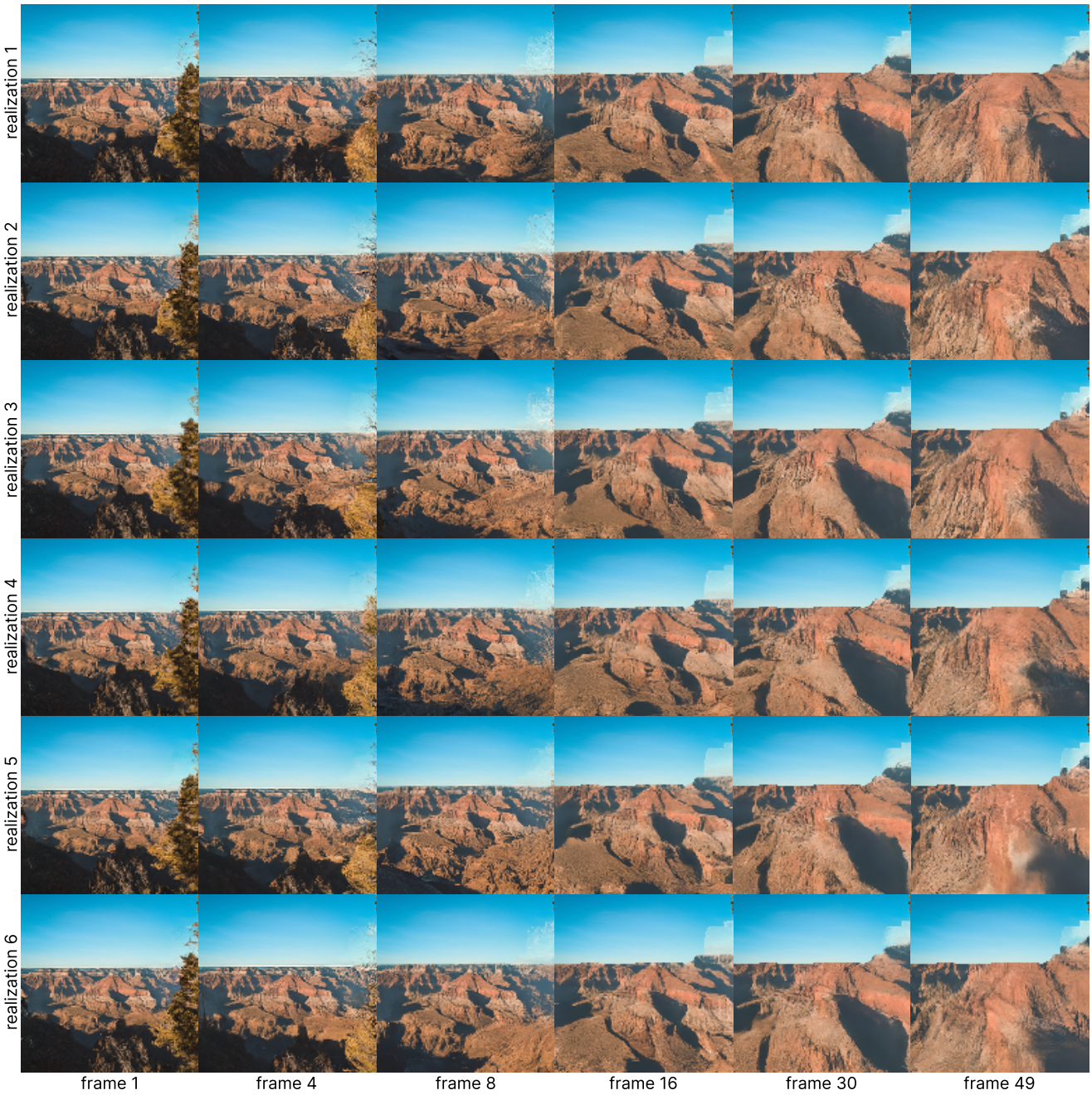

Figure 12: Additional qualitative results: Six distinct realizations, synthesized over 50 steps of forward motion.

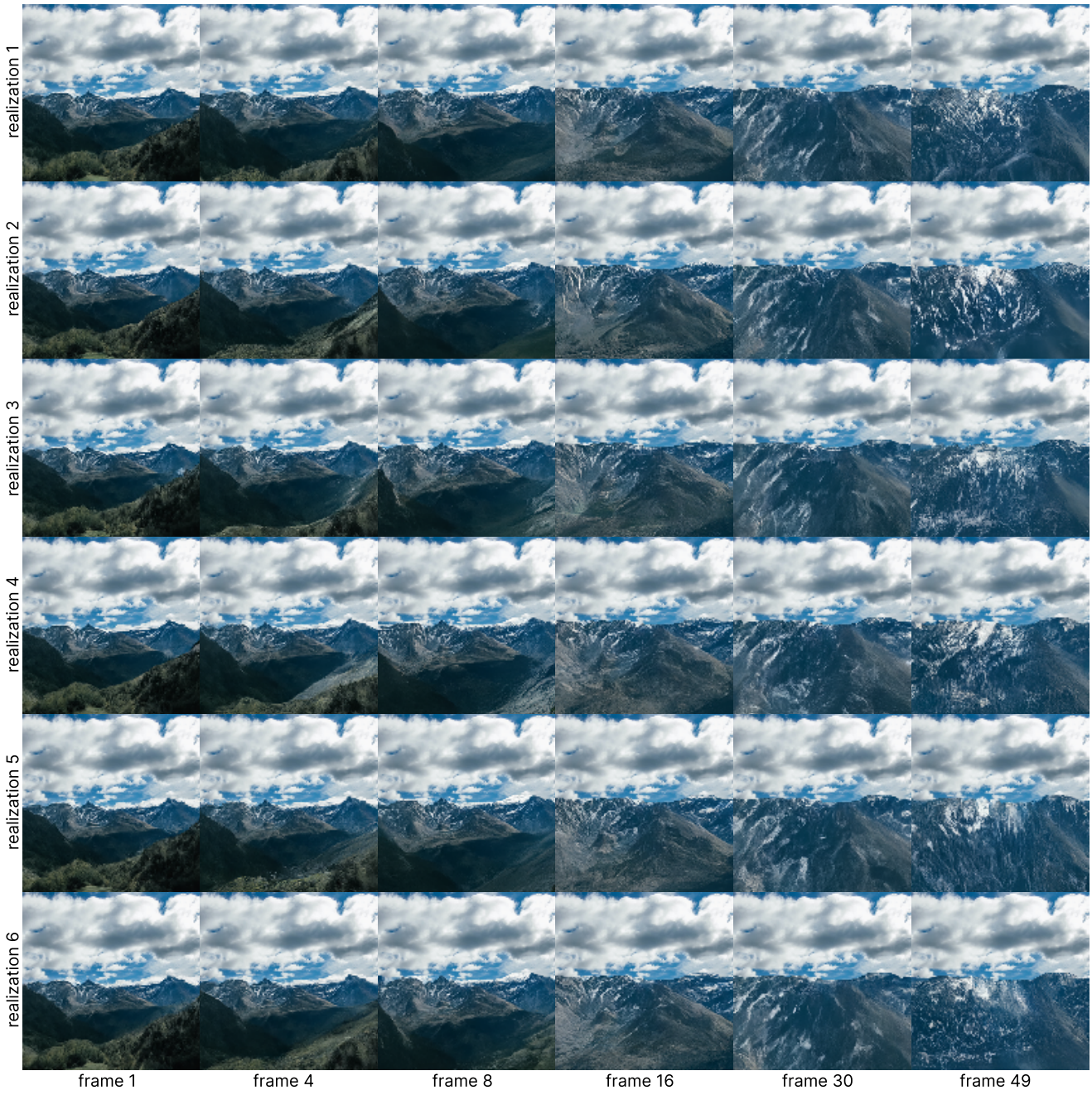

Figure 13: **Additional qualitative results:** Six distinct realizations, synthesized over 50 steps of forward motion. DiffDreamer is able to preserve consistency when there is no significant refinement needed.

Figure 14: **Additional qualitative results:** Six distinct realizations, synthesized over 50 steps of forward motion, where we encourage output diversity by additionally conditioning on input patterns.