

Long-Term Photometric Consistent Novel View Synthesis with Diffusion Models

Jason J. Yu^{1,2}, Fereshteh Forghani¹, Konstantinos G. Derpanis^{1,2}, Marcus A. Brubaker^{1,2}

¹York University, ²Vector Institute for AI

{jjyu, forghani, kosta, marcus.brubaker}@yorku.ca

<https://yorkucvil.github.io/Photoconsistent-NVS/>

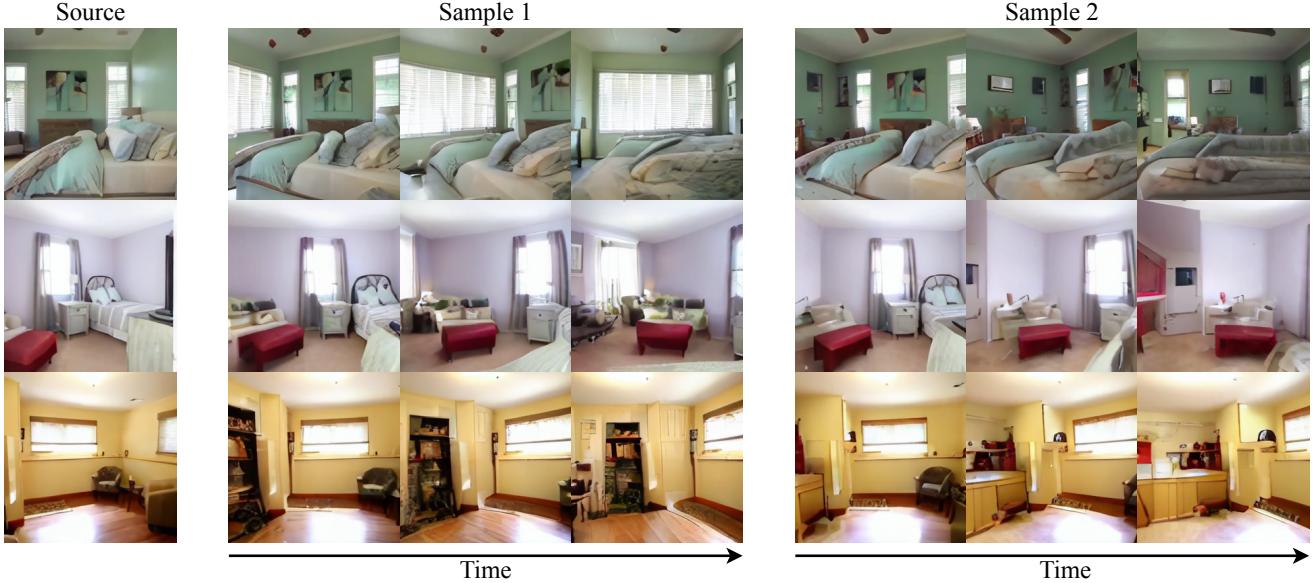


Figure 1: Given a single source view, our model allows us to sample multiple plausible sets of views over a camera trajectory. Here, we show two samples (middle and right) of a sequence using the three source views (left). Our method is able to maintain consistency between observed regions, while plausibly extrapolating unseen regions. Notice that the final frames reveal regions that are largely unseen in the source view, and show different plausible appearances in each sample.

Abstract

Novel view synthesis from a single input image is a challenging task, where the goal is to generate a new view of a scene from a desired camera pose that may be separated by a large motion. The highly uncertain nature of this synthesis task due to unobserved elements within the scene (i.e. occlusion) and outside the field-of-view makes the use of generative models appealing to capture the variety of possible outputs. In this paper, we propose a novel generative model capable of producing a sequence of photorealistic images consistent with a specified camera trajectory, and a single starting image. Our approach is centred on an autoregressive conditional diffusion-based model capable of interpolating visible scene elements, and extrapolating unobserved regions in a view, in a geometrically consistent

manner. Conditioning is limited to an image capturing a single camera view and the (relative) pose of the new camera view. To measure the consistency over a sequence of generated views, we introduce a new metric, the thresholded symmetric epipolar distance (TSED), to measure the number of consistent frame pairs in a sequence. While previous methods have been shown to produce high quality images and consistent semantics across pairs of views, we show empirically with our metric that they are often inconsistent with the desired camera poses. In contrast, we demonstrate that our method produces both photorealistic and view-consistent imagery.

1. Introduction

Novel view synthesis (NVS) methods are generally tasked with generating new scene views, given a set of existing views. NVS has a long history in computer vision

[7, 19, 2] and has recently seen a resurgence of interest with the advent of NeRFs [24, 47, 42]. Most current approaches to NVS (*e.g.* NeRFs) focus on problem settings where generated views remain close to the input and whose content is largely visible from some subset of the given views. This restricted setting makes these methods amenable to direct supervision. In contrast, we consider a more extreme case, where a single view is given as input, and the goal is to generate plausible image sequence continuations from a trajectory of provided camera views. By plausible, we mean that visible portions of the scene should evolve in a 3D consistent fashion, while previously unseen elements (*i.e.* regions occluded or outside of the camera field-of-view) should appear harmonious with the scene. Moreover, regions not visible in the input view are generally highly uncertain; so, there are a variety of plausible continuations that are valid.

To address this challenge, we propose a novel NVS method based on denoising diffusion models [13] to sample multiple, consistent novel views. We condition the diffusion model on both the given view, and a geometrically informed representation of the relative camera settings of both the given and target views. The resulting model is able to produce multiple plausible novel views by simply generating new samples from the model. Further, while the model is trained to generate a single novel view conditioned on an existing view and a target camera pose, we demonstrate that this model can generate a sequence of plausible views, including final views with little or no overlap with the starting view. Fig. 1 shows the outputs of our model for several different starting views, with two samples of plausible sets of views.

Existing NVS techniques have been evaluated primarily in terms of generated image quality (*e.g.* with Fréchet Inception Distance (FID) [12]) but have generally ignored measuring consistency with the camera poses. Based on the epipolar geometry defined by relative camera poses [11], we introduce a new metric which directly evaluates the geometric consistency of generated views independently from the quality of generated imagery. The proposed metric does not require any knowledge of scene geometry, making it widely applicable even on purely generated images. We evaluate the proposed method on both real and synthetic datasets in terms of both generated image quality and geometric consistency. Further, previous work only evaluates performance based on in-distribution camera trajectories. Here, we evaluate the generalization ability of extant models and our own by generating sequences based on novel trajectories (*i.e.* trajectories that differ significantly from those in the training data).

2. Related Work

NVS has been long studied in computer vision (*e.g.* [7, 19, 2]), and a full review is out of scope for this paper.

NVS methods can largely be categorized as those which focus on *view interpolation*, where generated views remain close to the given views, and *view extrapolation*, where the generated field-of-view may contain large amounts of novel content. Many current view interpolation methods are based on NeRFs [24, 47], which leverage neural-network representations of radiance fields fit to the observed images. Others attempt to directly regress novel views [35] from a set-encoded representation of the given views. Alternatively, if depth information is available, images can be back-projected into 3D, and missing regions inpainted [18]. We focus on view extrapolation NVS where significant portions of the generated images are not visible in the inputs.

View extrapolation methods are largely built on probabilistic approaches to capture the high degree of uncertainty. GAUDI [3] learns a latent variable model of entire 3D scenes represented as a neural radiance field and then estimates the latents given observed images. However, the estimated scene representation often has a limited spatial extent, which is in contrast to image-to-image methods [21] which may extend indefinitely. GeoGPT [32] uses an autoregressive likelihood model to sample novel views conditioned on a single source view. In contrast, we use a latent diffusion model [26, 31], and investigate sequential view generation. LookOut [29] extends GeoGPT [32] to generate sequences of views along a trajectory while conditioning on up to two previous views. To enforce consistency, LookOut requires a post-processing step that uses generated outputs as additional conditioning. In contrast, our model is conditioned on a single view, and does not require additional post-processing to achieve consistency. A closely related method [44] also formulates a diffusion model for NVS; however, it was only applied to simple scenes (*i.e.* isolated objects) with constrained camera poses. Here, we consider view extrapolation on real indoor scenes with complex geometry, and without constraints on camera motion.

Conditional generative models are a common approach for view synthesis [32, 29], image editing [23], and video prediction [20]. Recent years has seen significant progress in generative modelling [17, 36, 13, 9, 8] with diffusion models [31, 38, 13] showing promise in many tasks, *e.g.* text-to-image generation [31, 34] and video modeling [15]. In our problem, we utilize latent diffusion models [26, 31], which first compress high dimensional images with an autoencoder and discourage the diffusion model from expending capacity on modeling imperceptible details. The resulting model is more efficient, and uses less computation during training and inference.

Generative methods and 3D capable models are currently a very active research topic and there have been other highly related concurrent works investigating pose-conditional diffusion models. RenderDiffusion [1] uses an explicit 3D tri-plane representation [5] for object-centric NVS, and re-

lies on score-distillation [27] for 3D regularization, rather than relying on multi-view training data. In contrast to our method, RenderDiffusion focuses on object-centric NVS, and utilizes a 3D representation with limited spatial extent, while we focus on extrapolating scenes. The pose-guided diffusion model from Tseng et al. [40] is very similar to our method but uses a cascade diffusion model [14], and only investigates performance on in-distribution trajectories. In contrast, our method uses a latent diffusion model, and investigates generalization to out-of-distribution trajectories.

3. Technical Approach

3.1. Background: Diffusion Models

Here, we provide a brief introduction of diffusion models to ground the following developments but refer interested readers to a recent detailed review [45]. Diffusion models are a class of generative models where sampling is performed by reversing a stochastic diffusion process [13, 38]. The forward process is fixed, typically Gaussian, and discretized into $t \in 1, \dots, T$ timesteps which are defined recursively as

$$q(\mathbf{x}_t | \mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1 - \beta_t} \mathbf{x}_{t-1}, \beta_t \mathbf{I}), \quad (1)$$

where \mathbf{x}_0 is a sample from the data distribution of interest, $q(\mathbf{x}_0)$, \mathbf{I} is an identity matrix, and the values of β_t are dependent on the particular forward process used. Repeatedly applying Eq. 1 adds Gaussian noise with \mathbf{x}_T approximately normally distributed for large values of T . The reverse process is parameterized by θ and takes the form of a Gaussian:

$$p_\theta(\mathbf{x}_{t-1} | \mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \mu_\theta(\mathbf{x}_t, t), \Sigma_\theta(\mathbf{x}_t, t)), \quad (2)$$

where the variance, $\Sigma_\theta(\mathbf{x}_{t-1}, t)$, is generally set as constant. Here, \mathbf{x}_{t-1} is expressed using $\epsilon_\theta(\mathbf{x}_t, t)$ which is implemented as a neural network:

$$\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}_t - \frac{1 - \alpha_t}{\sqrt{1 - \bar{\alpha}_t}} \epsilon_\theta(\mathbf{x}_t, t) \right) + \sigma_t \mathbf{z}_t, \quad (3)$$

where $\alpha_t = 1 - \beta_t$, $\bar{\alpha}_t = \prod_{s=1}^t \alpha_s$, and $\mathbf{z}_t \sim \mathcal{N}(0, I)$. The function $\epsilon_\theta(\mathbf{x}_t, t)$ is referred to as the score function and can be interpreted as a noise estimator which can be used to denoise \mathbf{x}_t to produce \mathbf{x}_{t-1} . Training is performed using denoising score matching [43]:

$$\mathcal{L} = \mathbb{E}_{\mathbf{x}_0, t, \epsilon} [\| \epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_t} \mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t} \epsilon, t) \|^2]. \quad (4)$$

Samples can be drawn from the model by initializing \mathbf{x}_T with Gaussian noise, and iteratively applying the learned reverse process given in Eq. 2. The model is made conditional by providing additional inputs to the score function, $\epsilon_\theta(\mathbf{x}_t, t)$. Due to the redundant and high dimensional nature

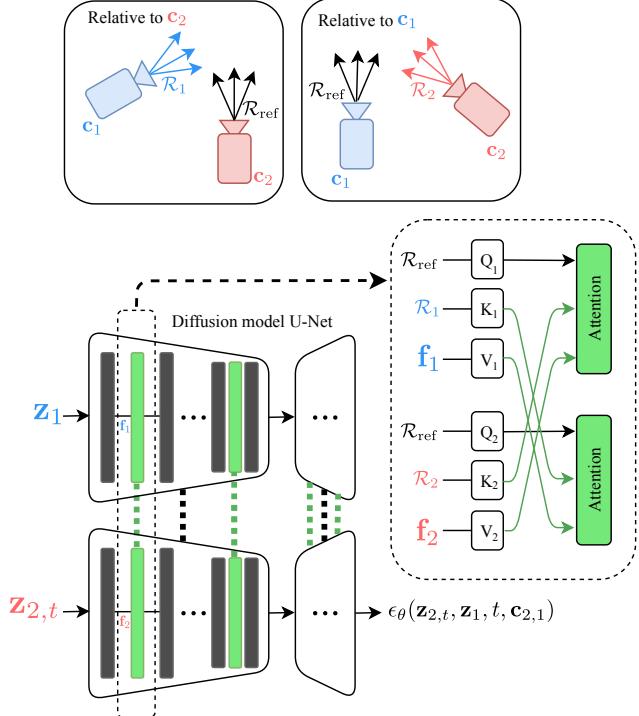


Figure 2: An overview of our model with two streams coupled with cross-attention. Our diffusion model is implemented as a two-stream U-Net [33], where latent representations for the given view, \mathbf{z}_1 (blue) and the generated view at diffusion step t , $\mathbf{z}_{2,t}$ (red), are processed by separate streams consisting of spatial layers with shared parameters (black). The latent of the given view, \mathbf{z}_1 , is used to condition the score of $\mathbf{z}_{2,t}$, and the camera poses are \mathbf{c}_1 and \mathbf{c}_2 . Both streams are conditioned on the noise variance, which is omitted for clarity. The two streams communicate via cross-attention layers (green). The queries are augmented with rays in a canonical reference frame, \mathcal{R}_{ref} . The keys, \mathbf{K}_1 and \mathbf{K}_2 , are augmented with ray information, \mathcal{R}_1 and \mathcal{R}_2 , respectively, which are each localized in the reference frame of the opposite view, \mathbf{c}_2 and \mathbf{c}_1 , illustrated on the top. The inset on the middle-right illustrates the cross-attention layer, where \mathbf{f}_1 and \mathbf{f}_2 are incoming features.

of images, it is beneficial to first reduce their dimensionality. There are several ways to approach the dimensionality reduction task [10, 14, 26, 31]. Here, we use a latent diffusion model [26, 31] that first transforms an image, \mathbf{x} , into a latent representation, \mathbf{z} , with a learned autoencoder, $\mathbf{z} = E(\mathbf{x})$. The diffusion model is then learned in the latent space, \mathbf{z} , and images are recovered by using the corresponding decoder, $\mathbf{x} = D(\mathbf{z})$. Critically for us, the learned latent representation can maintain the spatial structure of the image, e.g. through the use of a convolutional encoder architecture.

3.2. Novel View Synthesis with Diffusion Models

We now describe how we use a diffusion model to sample multiple plausible views in novel view synthesis. A conditioning image, \mathbf{x}_1 , is first mapped into the latent space, $\mathbf{z}_1 = \mathbf{E}(\mathbf{x}_1)$, and then is used to condition the distribution over the latent representation of the desired view:

$$p_\theta(\mathbf{z}_2|\mathbf{z}_1, \mathbf{c}_{2,1}), \quad (5)$$

where $\mathbf{c}_{2,1}$ is the relative camera pose between the source and target views. The distribution is estimated using a diffusion model [38] with score function $\epsilon_\theta(\mathbf{z}_{2,t}, \mathbf{z}_1, t, \mathbf{c}_{2,1})$, where $\mathbf{z}_{2,t}$ is the value of \mathbf{z}_2 at diffusion step t . The novel view, \mathbf{x}_2 , is then decoded from the sampled latent representation: $\mathbf{x}_2 = \mathbf{D}(\mathbf{z}_{2,T})$.

To obtain views along a trajectory with our model we generate them in sequence. Ideally, these would be sampled using the distribution conditioned on all previously generated views:

$$\mathbf{z}_{i+1} \sim p(\mathbf{z}_{i+1}|\mathbf{z}_0, \dots, \mathbf{z}_i, \mathbf{c}_{i+1,0}, \dots, \mathbf{c}_{i+1,i}). \quad (6)$$

We approximate this by assuming a Markov relationship between views in the sequence. That is, given an initial image, \mathbf{x}_0 , samples in the sequence of length L are obtained by encoding the initial image, $\mathbf{z}_0 = \mathbf{E}(\mathbf{x}_0)$, and recursively sampling from:

$$\mathbf{z}_{i+1} \sim p(\mathbf{z}_{i+1}|\mathbf{z}_i, \mathbf{c}_{i+1,i}), \quad (7)$$

with the final image decoded from the sampled latent representation: $\mathbf{x}_{L-1} = \mathbf{D}(\mathbf{z}_{L-1})$. We structure our model specifically for NVS, by equipping it with a specialized representation for relative camera geometry, and a two-stream architecture.

Reasoning about novel views requires knowledge of geometric camera information. To provide this information we augment the input of the score function with a representation of the camera rays for the conditioning and generated views [46, 35]. Our camera model is defined by the intrinsic matrix, \mathbf{K} , and the extrinsics, $\mathbf{c} = [\mathbf{R}|\mathbf{t}]$, where \mathbf{R} and \mathbf{t} are the 3D rotation and translation components, respectively. Given the projection matrix of a camera, $\mathbf{P} = \mathbf{K}[\mathbf{R}|\mathbf{t}]$, the camera center is computed as $\boldsymbol{\tau} = -\mathbf{R}^{-1}\mathbf{t}$. The direction of the camera ray at pixel coordinates (u, v) is given by:

$$\bar{\mathbf{d}}_{u,v} = \mathbf{R}^{-1}\mathbf{K}^{-1} [u \ v \ 1]^\top, \quad (8)$$

which is then normalized to unit length to obtain $\mathbf{d}_{u,v}$. Finally, before being used as conditioning for the diffusion model, the ray direction is concatenated with the camera center, $\mathbf{r}_{u,v} = [\mathbf{d}_{u,v}, \boldsymbol{\tau}]$, and frequency encoded [41]:

$$\mathcal{R} = [\sin(f_1\pi\mathbf{r}), \cos(f_1\pi\mathbf{r}), \dots, \sin(f_K\pi\mathbf{r}), \cos(f_K\pi\mathbf{r})], \quad (9)$$

where K is the number of frequencies, f_k are the frequencies which increase proportionally to 2^k , and the sinusoidal functions are applied element-wise.

The standard architecture for a score function is a U-Net architecture [33]. Here, we base our architecture on the Noise Conditional Score Network++ (NCSN++) architecture [38], with a variance exploding forward process. We modify this backbone architecture to incorporate the ray representation and the conditioned view. Inspired by video diffusion models [15], we propose a two-stream architecture using two U-Nets with shared weights to process the novel view, $\mathbf{x}_{2,t}$, and conditioning view, \mathbf{x}_1 . These networks communicate with one another exclusively via cross-attention layers, which are inserted after every spatial attention layer. We also augment the queries and keys of the attention with camera pose information. The output of the novel view stream is used as the output of the score function, $\epsilon_\theta(\mathbf{z}_{2,t}, \mathbf{z}_1, t, \mathbf{c}_{2,1})$. In short, the model contains a stream for each view, and couples them using augmented cross-attention. Our architecture is illustrated in Fig. 2 and more details are given in Appendix A.

3.3. Thresholded Symmetric Epipolar Distance (TSED)

Existing evaluation metrics for NVS primarily focus on the view interpolation case and are based on notions of reconstruction (*e.g.* PSNR and LPIPS) or general image quality (*e.g.* FID); however, reconstruction metrics are inapplicable to view extrapolation, where there is no reasonable expectation of a single ground truth output. General image quality metrics are relevant for view extrapolation but existing measures like FID are insensitive to the accuracy of the geometry. That is, generated images can completely ignore the required camera pose and still achieve excellent FID. To address this issue recent work [44] proposed a metric that is sensitive to accurate camera geometry, but the evaluation involves fitting a NeRF [24] to multiple generated images, and measuring consistency as the FID of unseen interpolated views; however, this evaluation is complex, excessively expensive to compute, and difficult to interpret. Here, we propose the Thresholded Symmetric Epipolar Distance (TSED) as a new lightweight metric for measuring geometric consistency of NVS models.

Our metric is motivated by two consistency criteria. First, the appearance of objects should remain stable between views, and should contain image features that can be identified and matched. Second, these matched features should respect epipolar constraints [11], given by the desired relative camera pose. With the camera poses used to condition the generation of the novel view, we compute the fundamental matrix, \mathbf{F} , which, given a feature point \mathbf{p} in one image, allows us to define the epipolar line $\mathbf{p}'^\top \mathbf{F} \mathbf{p} = 0$ on which its corresponding feature \mathbf{p}' should lie. We de-

fine the symmetric epipolar distance (SED) of corresponding points \mathbf{p} and \mathbf{p}' as:

$$\text{SED}(\mathbf{p}, \mathbf{p}', \mathbf{F}) = \frac{1}{2} [d(\mathbf{p}', \mathbf{F}\mathbf{p}) + d(\mathbf{p}, \mathbf{F}^\top \mathbf{p}')], \quad (10)$$

where $d(\mathbf{p}', \mathbf{F}\mathbf{p})$ is the minimum Euclidean distance between point \mathbf{p}' and the epipolar line induced by $\mathbf{F}\mathbf{p}$. (We note this definition of SED is similar in spirit but slightly different than those found in some standard references.) Given a set of feature correspondences, $M = \{(\mathbf{p}_1, \mathbf{p}'_1), \dots, (\mathbf{p}_n, \mathbf{p}'_n)\}$, between two views (*e.g.* computed with SIFT [22]) we define the pair of images to be consistent if there are a sufficient number of matching features, *i.e.* $n \geq T_{\text{matches}}$, and the median SED over M is less than T_{error} . The median is chosen to mitigate the influence of incorrect correspondences. The threshold T_{matches} makes the metric robust against image pairs with few matches as this likely indicates a low-quality generation, assuming the scenes are not largely textureless and the cameras do not undergo an extreme viewpoint change. The use of epipolar geometry here is key as it does not require knowledge of the scene geometry or scale. It should be noted that using epipolar geometry results in TSED having lower sensitivity to errors when most of the epipolar lines have a similar orientation, because SED for a match is insensitive to errors in 2D correspondence that parallel to the epipolar line. An empirical sensitivity analysis of TSED is provided in Appendix B. Given a NVS model, we evaluate it by generating sequences of images and computing which fraction of neighbouring views are consistent. We use $T_{\text{matches}} = 10$ and explore consistency as a function of different values of T_{error} in our experiments.

4. Experiments

We evaluate and compare to extant methods with a focus on *both* independent image quality and consistency across views. We conduct an ablation study on CLEVR [16], a synthetic dataset, to validate the various components of our model (Sec. 4.2). We further demonstrate the capabilities of our model using RealEstate10K [48], a large dataset of real indoor scenes, Matterport3D [6], a small dataset of building-scale textured meshes, and compare our method with two strong baselines (Sec. 4.3): GeoGPT [32] and LookOut [29].

4.1. Experimental Setup

For our experiments, we implement our model using a latent diffusion model (LDM) [31] with a VQ-GAN [8] as the latent space autoencoder, and a modified architecture as described in the previous section. During inference, we sample with ancestral sampling using a predictor-corrector sampler [38]. Training requires pairs of images along with camera intrinsics, and relative extrinsics. For evaluation

we use the CLEVR [16], RealEstate10K [48], and Matterport3D (MP3D) [6] datasets.

CLEVR [16] is a synthetic dataset consisting of scenes of simple geometric primitives with various materials placed on top of a matte grey surface. We repurpose the Blender based pipeline to uniformly scatter the primitives in the center of the scene in an 8×8 Blender unit area, and render views from a slightly elevated position to prevent the camera from being placed inside an object. The initial camera position is chosen uniformly in the same area that the objects are placed, and oriented towards the center of the scene with a $[-20, 20]$ degree jitter around the yaw axis. For the second view, the camera is randomly translated $[-1, 1]$ units along the ground plane, and jittered $[-20, 20]$ degrees around the yaw axis. Images are rendered at a resolution of 128×128 . The left most panel in Fig. 4 provides an example image.

RealEstate10K [48] consists of publicly available real estate tour videos scraped from YouTube. The videos are partitioned into disjoint sequences, and the camera parameters provided with the dataset were recovered using ORB-SLAM2 [25]. The large amount of real, diverse, and structured environments available in RealEstate10K make it an ideal and commonly used dataset for NVS evaluation, including by the most relevant baselines [32, 29]. Following previous work [29], the videos are obtained at 360p, center cropped, and downsampled to 256×256 . One challenging aspect of using this dataset is the limited diversity in camera motions. Many of the sequences consists of a simple forward motion that travels through and between rooms. This gives us an opportunity to evaluate the generalization of the model to novel camera motions not present in the dataset.

Matterport3D [6] consists of 90 indoor, building-scale environments that have been scanned using RGB-D sensors, and reconstructed as a textured mesh. Following previous work [29], we convert the scenes into videos using an embodied agent in the Habitat [37] simulation platform to navigate between two randomly chosen locations in the scene, and render each frame at a resolution of 256×256 . For each frame of the sequence, the agent chooses one of three actions: move forward, turn left, and turn right. The limited actions that the agent can perform greatly reduces the diversity of camera motions, which is even more limited than those available in RealEstate10K.

For our evaluations, we compare our method with two recent state-of-the-art generative scene extrapolation methods. GeoGPT [32] is an image-to-image NVS method, using a similar probabilistic formulation as our method. Four variants were proposed with options to leverage monocular depth maps provided by MiDaS [28], and perform explicit warping of the source image. For our evaluation, we use their model with implicit geometry and without access to depth maps as this is most similar to our proposed method,

Method	Single view		Last view
	LPIPS ↓	PSNR ↑	FID ↓
Naive concat	0.121	23.24	79.57
Two-Stream SC	-	-	78.11
Two-Stream	0.112	24.20	76.85

Table 1: Reconstruction metrics and FID for single view prediction and sequential prediction on CLEVR. *Two-Stream* is our two stream model, *Two-Stream SC* is our two-stream model sampled with stochastic conditioning, and *Naive concat* is the naive variant where inputs are concatenated along the channel dimension. We evaluate the FID on the last generated image of a trajectory. Stochastic conditioning is only applicable with more than two generated views, no results are provided for this method on single view evaluations.

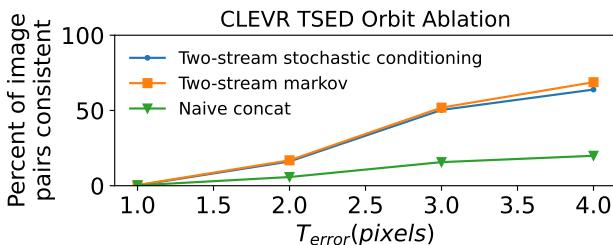


Figure 3: Percent consistent image pairs computed with TSED on different variants of our model, and sampling, on CLEVR.

and can, similarly, be applied autoregressively to generate sequences. LookOut¹ [29] is an extension of GeoGPT with a focus on improving the generation of novel views over a long camera trajectory. The model takes up to two input frames of a sequence, and uses the camera pose information to explicitly bias the attention layers inside the model. The final LookOut model is fine-tuned with a form of simulated error accumulation [21] to make the model robust to errors present in its inputs during autoregressive generation. In our evaluation, we consider two variants of LookOut, one including this post-processing step (LookOut), and one without (LookOut-ne). Note both GeoGPT and our model do not include a post-processing step and could potentially benefit from it. For MP3D, we use the publicly available weights for LookOut.

In addition to our introduced consistency metric (Sec. 3.3), we evaluate the quality of the generated images using standard image-centric metrics, specifically PSNR, LPIPS, and FID. PSNR and LPIPS are standard full reference image reconstruction metrics used to evaluate differences between generated and ground truth views. However, as the camera view changes significantly the space of plausible

¹An official public implementation is available without the pretrained weights on RealEstate10k. After email correspondence with the authors, we were unable to obtain the pretrained model. Reported results are based on a retrained model using the authors' publicly available code.

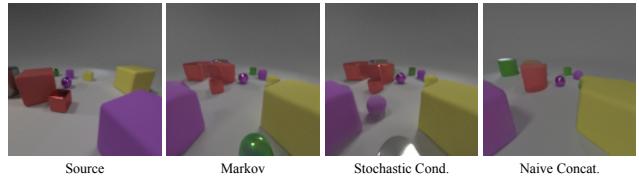


Figure 4: Samples of the sixth generated frame from the initial image on the left. Note the small red cube visible in the initial image disappears for the naive model.

views increases dramatically and reconstruction metrics like PSNR and LPIPS become less relevant due to a lack of single ground truth reference. While these metrics are not suitable for evaluating view extrapolation tasks [39, 30], they can still provide some sense of consistency for short-term generation, where uncertainty in the novel views is low. FID [12] is a standard reference-free metric for generative methods which measures sample quality of a set of i.i.d. samples, compared to a set of real samples. While FID does not provide a measure of consistency between images, it gives a sense of the overall realism of the generated images.

4.2. Ablations

Here, we explore variations on model architecture and sampling, and compare performance. First, we compare our two-stream architecture with a naive conditional diffusion model architecture, where both source and target views are concatenated to create a six channel image, and the model estimates the score for the target view. The results are shown in Tab. 1, which shows clearly that our proposed architecture is effective.

We also explore an alternative strategy for sampling trajectories of novel views. Previous work [44] proposed a heuristic for extending a single source view novel view diffusion model to use an arbitrary number of source views called *stochastic conditioning*. Given m possible source views, each iteration of the diffusion sampling process is modified to be randomly conditioned on one of the m views. We consider this heuristic for generating sets of views, conditioning on up to two of the previous frames. For these ablations, we sample ten images from a trajectory orbiting the center of the scene, using 100 different starting images.

We evaluate consistency using TSED; quantitative results are provided in Fig. 3, and qualitative results are shown in Fig. 4. We find that the naive model can generate images where clearly visible objects may disappear, leading to less consistency qualitatively and quantitatively. Sampling with stochastic conditioning is qualitatively similar to Markov sampling. Quantitatively, stochastic conditioning is less consistent when T_{error} is high, which is the result of fewer matches being made. In general, recovering correspondences on CLEVR is challenging due to few distinct features. Despite the challenges presented by this dataset,

	Method	Short-term		Long-term	
		LPIPS ↓	PSNR ↑	LPIPS ↓	PSNR ↑
RealEstate10K	GeoGPT [32]	0.444	13.35	0.674	9.54
	LookOut-ne [29]	0.390	14.19	0.688	9.65
	LookOut [29]	0.378	14.43	0.658	10.51
	Ours	0.333	15.51	0.588	11.54
MP3D	LookOut [29]	0.604	12.76	0.739	10.60
	Ours	0.504	14.83	0.674	13.00

Table 2: RealEstate10K and MP3D reconstruction metrics with in-distribution trajectories. LookOut-ne refers to the LookOut method without the final error accumulation training step.

	Method	Short-term FID ↓	Long-term FID ↓
RealEstate10K	GeoGPT [32]	26.72	41.87
	LookOut-ne [29]	30.38	72.01
	LookOut [29]	28.86	58.12
	Ours	26.76	41.95
MP3D	LookOut [29]	80.97	132.36
	Ours	73.16	100.99

Table 3: RealEstate10K and MP3D FIDs with in-distribution trajectories. FID scores between generated images at short-term and long-term generations, and a fixed set of randomly selected images from the test set.

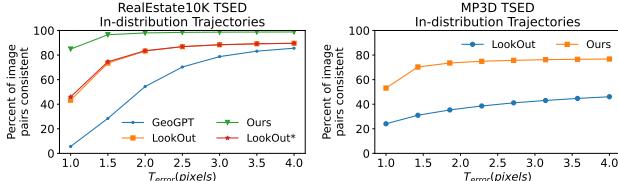


Figure 5: RealEstate10K TSED on in-distribution trajectories. Consistency is measured as the average percent of consistent image pairs in the generated sequences. We set $T_{\text{matches}} = 10$.

our metric is still able to provide a measure of consistency. Overall, these results show that in contrast to previous work [44], stochastic conditioning has no benefit to our approach and may actually hurt performance. We also attempt to perform stochastic conditioning on RealEstate10K, but the images are qualitatively poor; results are available in the Appendix C.

4.3. Generation with In-Distribution Trajectories

For our initial set of experiments on RealEstate10K and MP3D, we consider the generation of novel views along in-distribution trajectories. To generate representative, in-distribution trajectories, given a start image, we randomly sample camera trajectories from the test set, as done in previous work [29].

Image quality. We evaluate the reconstruction performance of novel views using PSNR and LPIPS, across short-term and long-term generations. Following previous work [29], we only consider test sequences where at least 200 frames are available, for RealEstate10K. This choice ensures that

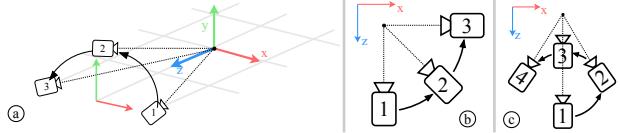


Figure 6: A visualization of our custom trajectories: Hop (a), Orbit (b), and Spin (c). All cameras point towards a pivot in the scene, and the dotted lines represent the optical axes of the cameras. We use a coordinate space where x is right, y is up, and z is backward.

there are ground truth images to evaluate against. Starting with the first frame from the ground truth test sequences as our initial images, we generate 20 images of a sequence using 20 camera poses of the ground truth trajectory. The camera poses are spaced ten frames apart with respect to the sequence’s native frame rate, yielding a final camera pose that is 200 frames from the initial view. Short-term evaluations are performed over the fifth generated image, and long-term evaluations are performed on the final generated image. Quantitative results for RealEstate10K are provided in Tab. 2. Compared to the baselines, our method has the lowest reconstruction error in all cases. We also evaluate LookOut [29] without their additional post-processing step (LookOut-ne), and find that it yields slightly worse reconstruction results.

Similar to our full reference metric evaluation, we evaluate short-term and long-term quality with the no-reference metric FID. To measure the generation image quality over time, we evaluate the FID between generated views at a specific time, and a fixed set of randomly selected views from the test set. Tab. 3 presents quantitative results for RealEstate10K. As seen from the table, all methods suffer from some level of error accumulation, and yield worse performance as the sequence length increases. We find that LookOut produces images with significantly higher FID without the final error accumulation step. For in-distribution trajectories, our method generates images with comparable quality as GeoGPT, and outperforms LookOut in terms of FID. Notably, GeoGPT has the tendency to generate viewpoint-inconsistent images, where the semantics remain the same but the content changes. This point is examined later using our viewpoint consistency metric.

In addition to RealEstate10K, we evaluate on MP3D with a similar setup, except the images in the sequence are neighboring frames since the rendered images from MP3D differ by larger camera motions. We also provide reconstruction-based results for MP3D in Tab. 2, and FID-based results in Tab. 3, with LookOut as the baseline. Quantitatively, we find the results with MP3D are similar to RealEstate10K, where our method outperforms LookOut on all standard metrics for in-distribution trajectories.

Consistency over long-term generations. We evaluate the

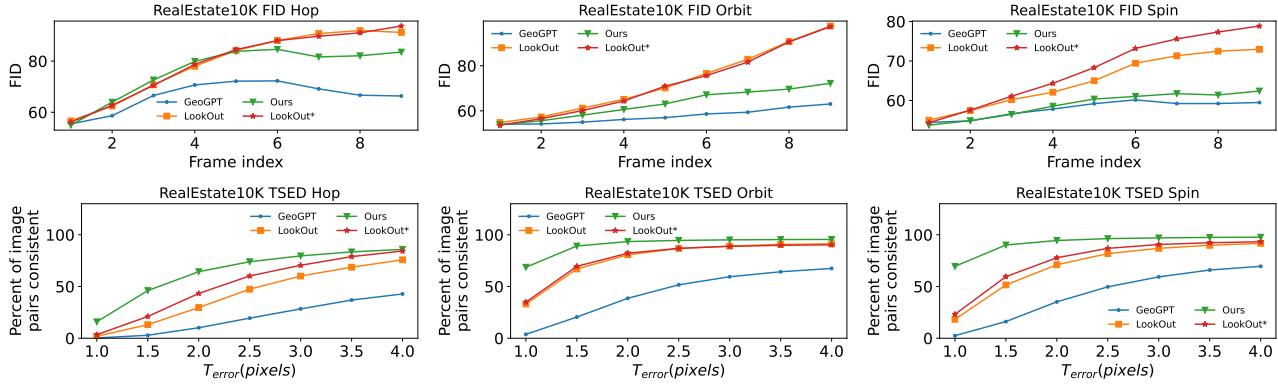


Figure 7: RealEstate10K FID (top), and TSED (bottom), on custom trajectories. Sequences are sampled using three novel trajectories designed to differ from the dominant modes in the dataset: Hop, Orbit, and Spin. *LookOut** is a version of LookOut without error accumulation post-processing. For TSED, we set $T_{\text{matches}} = 10$ while sweeping over a range of T_{error} values.

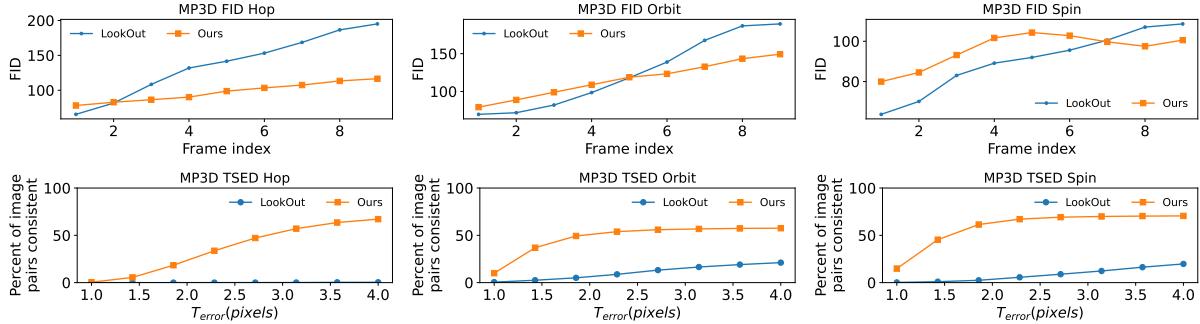


Figure 8: MP3D FID (top), and TSED (bottom), on custom trajectories. Evaluation on MP3D is performed using the same custom trajectories and TSED parameters as RealEstate10K.

percent of consistent pairs of neighboring views out of 20 total pairs using our proposed metric, TSED (Sec. 3.3). Quantitative results for RealEstate10K are shown in Fig. 5, where we evaluate the consistency over a range of values for T_{error} , with $T_{\text{matches}} = 10$. The average number of matches per pair on RealEstate10K is 33, 87, and 94, for GeoGPT, LookOut, and our method, respectively. The lower consistency of GeoGPT is partly due to fewer matches per image pair. Samples from LookOut have a comparable number of matches to our method, suggesting that the inconsistency is due to larger violations of the epipolar constraints. Compared to the baselines, our method can generate better views with consistent appearances, and motion that respects epipolar constraints. LookOut performs similarly on our consistency metric with and without error accumulation training. We also compare LookOut and our method using TSED on MP3D, shown in Fig. 5, and find that our method is more consistent on this dataset as well.

4.4. Generation with Novel Trajectories

Previous work limited evaluation to the ground truth trajectories in the RealEstate10K and MP3D datasets. Consequently, given the biased nature of the trajectories, this may lead to overfitting. Here, we explore the generalization ca-

pability of both our method and the baselines by evaluating on out-of-distribution trajectories.

As mentioned in Section 4.1, the camera motions available in RealEstate10K, and MP3D are limited. We sample novel views over three manually defined trajectories distinct from those found in the training data: (i) a 90-degree orbit around the azimuth (Orbit), (ii) a vertical orbit along a semi-circular path (Hop), and (iii) a translation along a circular path parallel to the ground plane (Spin). These trajectories are illustrated in Fig. 6.

As ground truth images are not available, we evaluate performance using the reference-free metrics, FID and TSED. Quantitative results for RealEstate10K are summarized in Fig. 7. In terms of FID, our model’s generation quality degrades faster than GeoGPT but slower than LookOut. Qualitative results such as those shown in Fig. 9 suggest that when the baseline methods fail, they favour generating good-quality images, even though they may not be photometrically consistent with the other views. The consistency of our generated sequences, evaluated using TSED, is higher than the baselines on all trajectories. Between the three custom trajectories, *Hop* is the most novel as it contains a vertical motion that is rare in RealEstate10K, while *Spin* is the closest to the training trajectories, which contain

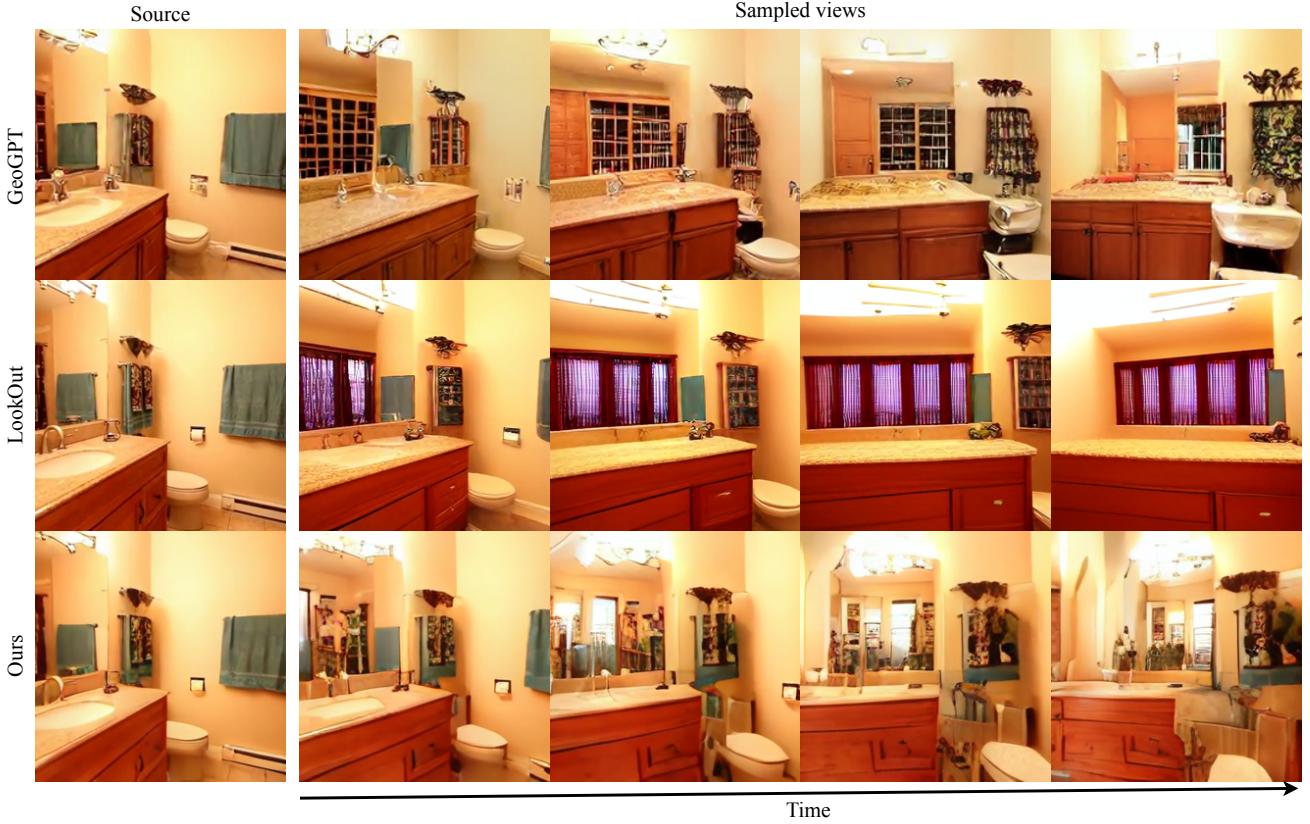


Figure 9: Samples from the *Orbit* trajectory. Each row presents a generated image sequence from our method and the baselines, GeoGPT [32] and LookOut [29]. The columns, are sampled views along the trajectory with the left most image being given. Notice both baselines give the impression of an orbiting camera motion, but parts of the visible scene in both views change between frames, *i.e.* the cabinet under the sink. Images generated from LookOut tend to lose details in subsequent frames. Our method tends to maintain photometric consistency across the sequence.

many forward and backward motions. Interestingly, LookOut without error accumulation performs better in the *Hop* trajectory on TSED. This suggests that the error accumulation post-processing may trade off generalization for higher image quality. Overall, our method provides the best trade-off of photometric quality and consistency.

Fig. 8 shows quantitative results on MP3D comparing LookOut, and our method. LookOut is significantly less consistent than our method in terms of TSED, especially on *Hop*. Qualitative inspection reveals that LookOut generalizes poorly to our custom trajectories, and often does not generate images that respect the requested camera motion.

5. Conclusion and Discussion

We addressed the most challenging setting for NVS, *i.e.* generative view extrapolation from a single image. Our method exploits recent advancements in diffusion-based generative models to sample multiple consistent novel views. Empirically, we presented a finer-grained evaluation of the task compared to previous studies. In particular,

reported results of previous work focus on generated image quality of each image but ignore geometric consistency. Here, we introduced a new metric based on epipolar geometry, which directly evaluates geometric consistency of generated views independent of image quality. Based on both new and standard metrics, we showed that our method generates images that are more consistent than current methods, while maintaining high image quality. Further, on camera trajectories that are atypical of the training data, we showed that our method generates images that are more consistent than the baselines.

Acknowledgements. This work was funded in part by the Canada First Research Excellence Fund (CFREF) for the Vision: Science to Applications (VISTA) program, the NSERC Discovery Grant program, the NSERC Canada Graduate Scholarships – Doctoral program, and the Vector Institute for AI.

References

- [1] Titas Auciukevičius, Zexiang Xu, Matthew Fisher, Paul Henderson, Hakan Bilen, Niloy J Mitra, and Paul Guerrero. Renderdiffusion: Image diffusion for 3D reconstruction, inpainting and generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 2
- [2] Shai Avidan and Amnon Shashua. Novel view synthesis in tensor space. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1997. 2
- [3] Miguel Ángel Bautista, Pengsheng Guo, Samira Abnar, Walter Talbott, Alexander T Toshev, Zhiyuan Chen, Laurent Dinh, Shuangfei Zhai, Hanlin Goh, Daniel Ulbricht, Afshin Dehghan, and Joshua M. Susskind. GAUDI: A neural architect for immersive 3D scene generation. In *Neural Information Processing Systems (NeurIPS)*, 2022. 2
- [4] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019. 12
- [5] Eric R Chan, Connor Z Lin, Matthew A Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas J Guibas, Jonathan Tremblay, Sameh Khamis, Tero Karras, and Gordon Wetzstein. Efficient geometry-aware 3D generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2
- [6] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3D: Learning from RGB-D data in indoor environments. *Proceedings of the International Conference on 3D Vision (3DV)*, 2017. 5, 13
- [7] Shenchang Eric Chen and Lance Williams. View interpolation for image synthesis. In *Proceedings of SIGGRAPH*, pages 279–288, 1993. 2
- [8] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2, 5
- [9] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Neural Information Processing Systems (NeurIPS)*, 2014. 2
- [10] Florentin Guth, Simon Coste, Valentin De Bortoli, and Stephane Mallat. Wavelet score-based generative modeling. *Neural Information Processing Systems (NeurIPS)*, 2022. 3
- [11] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003. 2, 4, 15, 16
- [12] 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. In *Neural Information Processing Systems (NeurIPS)*, 2017. 2, 6
- [13] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Neural Information Processing Systems (NeurIPS)*, 2020. 2, 3
- [14] Jonathan Ho, Chitwan Saharia, William Chan, David J Fleet, Mohammad Norouzi, and Tim Salimans. Cascaded diffusion models for high fidelity image generation. *Journal of Machine Learning Research (JMLR)*, 2022. 3
- [15] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2022. 2, 4
- [16] Justin Johnson, Bharath Hariharan, Laurens Van Der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 5, 13
- [17] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2014. 2
- [18] Jing Yu Koh, Harsh Agrawal, Dhruv Batra, Richard Tucker, Austin Waters, Honglak Lee, Yinfei Yang, Jason Baldridge, and Peter Anderson. Simple and effective synthesis of indoor 3D scenes. *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, 2023. 2
- [19] Stephane Laveau and Olivier D Faugeras. 3-D scene representation as a collection of images. In *Proceedings of the International Conference on Pattern Recognition (ICPR)*, volume 1, pages 689–691, 1994. 2
- [20] Alex X. Lee, Richard Zhang, Frederik Ebert, Pieter Abbeel, Chelsea Finn, and Sergey Levine. Stochastic adversarial video prediction. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019. 2
- [21] Andrew Liu, Richard Tucker, Varun Jampani, Ameesh Makadia, Noah Snavely, and Angjoo Kanazawa. Infinite nature: Perpetual view generation of natural scenes from a single image. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021. 2, 6
- [22] David G Lowe. Object recognition from local scale-invariant features. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 1999. 5, 15
- [23] 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 *Proceedings of the International Conference on Learning Representations (ICLR)*, 2022. 2
- [24] 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 *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 2, 4
- [25] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. ORB-SLAM: A versatile and accurate monocular SLAM system. *IEEE Transactions on Robotics (T-RO)*, 31(5):1147–1163, 2015. 5
- [26] Weili Nie, Arash Vahdat, and Anima Anandkumar. Controllable and compositional generation with latent-space energy-based models. In *Neural Information Processing Systems (NeurIPS)*, 2021. 2, 3

- [27] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. DreamFusion: Text-to-3D using 2D diffusion. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2022. 3
- [28] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 44(3), 2022. 5
- [29] Xuanchi Ren and Xiaolong Wang. Look outside the room: Synthesizing a consistent long-term 3D scene video from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2, 5, 6, 7, 9, 15
- [30] Chris Rockwell, David F Fouhey, and Justin Johnson. Pixel-synth: Generating a 3D-consistent experience from a single image. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021. 6
- [31] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2, 3, 5
- [32] Robin Rombach, Patrick Esser, and Björn Ommer. Geometry-free view synthesis: Transformers and no 3D priors. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021. 2, 5, 7, 9, 15
- [33] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. In *Proceedings of the International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI)*, pages 234–241, 2015. 3, 4
- [34] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Raphael Gontijo-Lopes, Burcu Karagol Ayan, Tim Salimans, Jonathan Ho, David J. Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding. In *Neural Information Processing Systems (NeurIPS)*, 2022. 2
- [35] Mehdi S. M. Sajjadi, Henning Meyer, Etienne Pot, Urs Bergmann, Klaus Greff, Noha Radwan, Suhani Vora, Mario Lucic, Daniel Duckworth, Alexey Dosovitskiy, Jakob Uszkoreit, Thomas Funkhouser, and Andrea Tagliasacchi. Scene representation transformer: Geometry-free novel view synthesis through set-latent scene representations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2, 4
- [36] Tim Salimans, Andrej Karpathy, Xi Chen, and Diederik P. Kingma. PixelCNN++: Improving the PixelCNN with discretized logistic mixture likelihood and other modifications. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017. 2
- [37] Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, et al. Habitat: A platform for embodied AI research. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2019. 5
- [38] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021. 2, 3, 4, 5, 12
- [39] Piotr Teterwak, Aaron Sarna, Dilip Krishnan, Aaron Maschinot, David Belanger, Ce Liu, and William T Freeman. Boundless: Generative adversarial networks for image extension. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2019. 6
- [40] Hung-Yu Tseng, Qinbo Li, Changil Kim, Suhib Alsisan, Jia-Bin Huang, and Johannes Kopf. Consistent view synthesis with pose-guided diffusion models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 3
- [41] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Neural Information Processing Systems (NeurIPS)*, 2017. 4
- [42] 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. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2
- [43] Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural computation*, (7):1661–1674, 2011. 3
- [44] Daniel Watson, William Chan, Ricardo Martin-Brualla, Jonathan Ho, Andrea Tagliasacchi, and Mohammad Norouzi. Novel view synthesis with diffusion models. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2023. 2, 4, 6, 7, 13
- [45] Ling Yang, Zhilong Zhang, Yang Song, Shenda Hong, Runsheng Xu, Yue Zhao, Yingxia Shao, Wentao Zhang, Ming-Hsuan Yang, and Bin Cui. Diffusion models: A comprehensive survey of methods and applications. *CoRR*, abs/2209.00796, 2022. 3
- [46] Wang Yifan, Carl Doersch, Relja Arandjelović, João Carreira, and Andrew Zisserman. Input-level inductive biases for 3D reconstruction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 4
- [47] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelNeRF: Neural radiance fields from one or few images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2
- [48] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. In *ACM Transactions on Graphics (TOG)*, 2018. 5, 12, 13

Appendix

A. Architecture Details

In this section, we provide additional details of our model described in Section 3.2 of the main paper. Our model is based on *Noise Conditional Score Network++* (NCSN++) [38]. An overview of the main backbone is provided in Tables A.4 and A.5. Two streams of the backbone are used to process the conditioning and generated image. We modify the original architecture by adding cross-attention layers throughout the backbone, which attend to features in the opposite stream. The residual blocks are based on the residual blocks used in BigGAN [4]. Upsampling and downsampling is also performed in the network using BigGAN residual blocks [4]. Inputs to the backbone encoder are provided at various layers using a multi-scale pyramid. Outputs of the network are accumulated from multiple layers of the decoder using a multi-scale residual pyramid. Specific implementation details can be found in the code release: <https://yorkucvlib.github.io/Photoconsistent-NVS/>.

Layer	Output size	Additional inputs	Additional outputs
Input image	$4 \times 32 \times 32$		Skip 0, In Pyramid
ResBlock	$256 \times 32 \times 32$	Time emb.	
Spatial Attn.	$256 \times 32 \times 32$		
Cross Attn.	$256 \times 32 \times 32$	Cross, Rays	Skip 1, Cross
ResBlock	$256 \times 32 \times 32$	Time emb.	
Spatial Attn.	$256 \times 32 \times 32$		
Cross Attn.	$256 \times 32 \times 32$	Cross, Rays	Skip 2, Cross
ResBlockDown Combiner	$256 \times 16 \times 16$	Time emb.	
	$256 \times 16 \times 16$	In Pyramid 1	Skip 3
ResBlock	$256 \times 16 \times 16$	Time emb.	
Spatial Attn.	$256 \times 16 \times 16$		
Cross Attn.	$256 \times 16 \times 16$	Cross, Rays	Skip 4, Cross
ResBlock	$256 \times 16 \times 16$	Time emb.	
Spatial Attn.	$256 \times 16 \times 16$		
Cross Attn.	$256 \times 16 \times 16$	Cross, Rays	Skip 5, Cross
ResBlockDown Combiner	$256 \times 8 \times 8$	Time emb.	
	$256 \times 8 \times 8$	In Pyramid 2	Skip 6
ResBlock	$256 \times 8 \times 8$	Time emb.	
Spatial Attn.	$256 \times 8 \times 8$		
Cross Attn.	$256 \times 8 \times 8$	Cross, Rays	Skip 7, Cross
ResBlock	$256 \times 8 \times 8$	Time emb.	
Spatial Attn.	$256 \times 8 \times 8$		
Cross Attn.	$256 \times 8 \times 8$	Cross, Rays	Skip 8, Cross
ResBlockDown Combiner	$256 \times 4 \times 4$	Time emb.	
	$256 \times 4 \times 4$	In Pyramid 3	Skip 9
ResBlock	$256 \times 4 \times 4$	Time emb.	
Spatial Attn.	$256 \times 4 \times 4$		
Cross Attn.	$256 \times 4 \times 4$	Cross, Rays	Skip 10, Cross
ResBlock	$256 \times 4 \times 4$	Time emb.	
Spatial Attn.	$256 \times 4 \times 4$		
Cross Attn.	$256 \times 4 \times 4$	Cross, Rays	Skip 11, Cross
ResBlock	$256 \times 4 \times 4$	Time emb.	
Spatial Attn.	$256 \times 4 \times 4$		
ResBlock	$256 \times 4 \times 4$	Time emb.	

Table A.4: NCSN++ U-Net backbone encoder. ResBlocks are BigGAN [4] style residual blocks, ResBlocksDown layers are the same, but configured with a downsampling option. Time emb. is the time information provided for the diffusion model. Skip inputs are skip connections that go to the decoder. Rays are the camera ray conditioning, and Cross is a cross-attention connection to the other stream.

B. TSED Sensitivity Analysis.

A drawback to using epipolar geometry to measure consistency between correspondences and the camera poses is the potential for TSED to be insensitive to positional errors in the correspondences along epipolar lines. We empirically analyse the sensitivity of TSED on ground truth image pairs from RealEstate10K [48] under three classes of camera motion: dominant forward-backward motions, dominant left-right motions, and motion that contains more than ten degrees of azimuth rotation. Using $T_{\text{error}} = 2$, we compute TSED over 100 random image pairs in each class while adding perturbations to the 2D positions

Layer	Output size	Additional inputs	Additional outputs
Encoder input	$256 \times 4 \times 4$		
ResBlock	$256 \times 4 \times 4$	Time emb., Skip 11	
ResBlock	$256 \times 4 \times 4$	Time emb., Skip 10	
ResBlock	$256 \times 4 \times 4$	Time emb., Skip 9	
Spatial Attn.	$256 \times 4 \times 4$		
Cross Attn.	$256 \times 4 \times 4$	Cross, Rays	Cross
Conv3 × 3	$256 \times 4 \times 4$	Out Pyramid 1	
ResBlockUp	$256 \times 8 \times 8$	Time emb.	
ResBlock	$256 \times 8 \times 8$	Time emb., Skip 8	
ResBlock	$256 \times 8 \times 8$	Time emb., Skip 7	
ResBlock	$256 \times 8 \times 8$	Time emb., Skip 6	
Spatial Attn.	$256 \times 8 \times 8$		
Cross Attn.	$256 \times 8 \times 8$	Cross, Rays	Cross
Conv3 × 3	$256 \times 8 \times 8$	Out Pyramid 2	
ResBlockUp	$256 \times 16 \times 16$	Time emb.	
ResBlock	$256 \times 16 \times 16$	Time emb., Skip 5	
ResBlock	$256 \times 16 \times 16$	Time emb., Skip 4	
ResBlock	$256 \times 16 \times 16$	Time emb., Skip 3	
Spatial Attn.	$256 \times 16 \times 16$		
Cross Attn.	$256 \times 16 \times 16$	Cross, Rays	Cross
Conv3 × 3	$256 \times 16 \times 16$	Out Pyramid 3	
ResBlockUp	$256 \times 32 \times 32$	Time emb.	
ResBlock	$256 \times 32 \times 32$	Time emb., Skip 2	
ResBlock	$256 \times 32 \times 32$	Time emb., Skip 1	
ResBlock	$256 \times 32 \times 32$	Time emb., Skip 0	
Spatial Attn.	$256 \times 32 \times 32$		
Cross Attn.	$256 \times 32 \times 32$	Cross, Rays	Cross
Conv3 × 3	$256 \times 32 \times 32$	Out Pyramid 4	

Table A.5: NCSN++ U-Net backbone decoder. ResBlocks are BigGAN [4] style residual blocks, ResBlocksUp layers are the same, but configured with an upsampling option. Time emb. is the time information provided for the diffusion model. Skip inputs are skip connections coming from the encoder. Rays are the camera ray conditioning, and Cross is the cross-attention connection to the other stream.

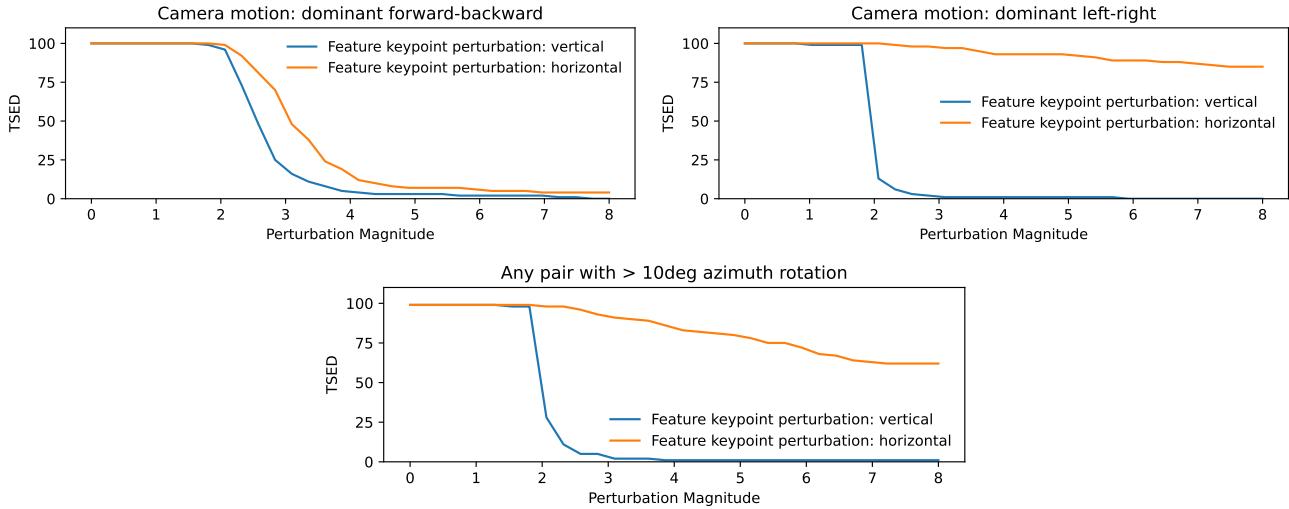


Figure B.10: TSED sensitivity analysis for image pairs with different dominant camera motions using $T_{\text{error}} = 2$. TSED scores are plotted for perturbations to the 2D correspondence locations with constant magnitude along horizontal and vertical directions. Camera motion determines the orientations of the epipolar lines, which can make the metric insensitive in some cases when many epipolar lines share the same orientation.

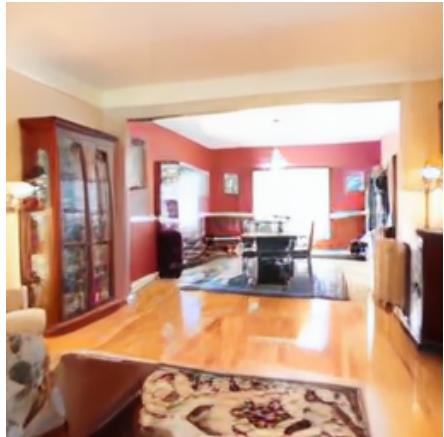
of the correspondences in each view by a constant magnitude along horizontal and vertical directions. In the ideal case when TSED is maximally sensitive, it should show a sharp reduction when the perturbations have a magnitude of T_{error} or greater. Results from our sensitivity analysis are shown in Fig. B.10. As expected, TSED is least sensitive to horizontal perturbations for when there are left-right camera motions since most of the epipolar lines are horizontal. For image-pairs with greater than 10 degrees of azimuth rotation, there are fewer horizontal epipolar lines, and TSED is more sensitive to horizontal perturbations than with dominant left-right motion. The results also show that TSED is most sensitive for forward-backward motions since the epipolar lines have a variety of orientations.

C. Stochastic Conditioning on RealEstate10K

Previous work [44] proposed a heuristic for extending a novel view diffusion model to use an arbitrary number of source views, called *stochastic conditioning*. Given m possible source views, each iteration of the diffusion sampling process is modified to be randomly conditioned on one of the m views. Results using stochastic conditioning on CLEVR [16] are provided in the main paper in Section 4.2. Previous work [44] used stochastic conditioning to condition on all previous frames. We also apply this heuristic for generating sets of views on RealEstate10K [48], but we conditioned on up to two of the previous frames. Qualitative results shown in Figure C.11 exhibit a significant reduction in quality, and contain noticeable artifacts. As a consequence, we did not include results based on stochastic conditioning with our method.

D. Additional Qualitative Results

Additional qualitative results are provided with an interactive viewer on our project page, <https://yorkucvil.github.io/Photoconsistent-NVS/>, under the **RealEstate10K Qualitative Results - Out-of-Distribution Trajectories** and **RealEstate10K Qualitative Results - In-Distribution Trajectories** sections. The viewer allows the images along a trajectory to be explored for multiple scenes, and sampling instances. Due to the stochastic nature of our model and the baselines, different plausible extrapolations of the scene are shown in the different instances of sampling. Additional qualitative results for Matterport3D [6] are also available on our project page, <https://yorkucvil.github.io/Photoconsistent-NVS/>, under the **Matterport3D Qualitative Results - Out-of-Distribution Trajectories** and **Matterport3D Qualitative Results - In-Distribution Trajectories** sections.



(a) Source image.



(b) Frame 5 of Markov sampling.



(c) Frame 7 of Markov sampling.



(d) Frame 5 with stochastic conditioning.



(e) Frame 7 with stochastic conditioning.

Figure C.11: Comparison of generation using a Markov dependency vs stochastic conditioning with the previous two frames as input. Both methods were generated using the same trajectory and source image. Notice the reduction of quality when stochastic conditioning is applied.

E. Additional Results with TSED

We provide additional quantitative results using TSED in Figures E.12, E.13, E.14, and E.15 for images generated using in-distribution trajectories, and the orbit, spin, hop out-of-distribution trajectories, respectively. We sweep across a range of values for both T_{error} and T_{matches} . Pairs of images with less than T_{matches} SIFT [22] matches, or a median SED [11] lower than T_{error} , are considered not consistent. In all trajectory types, GeoGPT [32] is the most affected by T_{matches} due to a lack of photometric consistency, which leads to a low number of SIFT correspondences. The TSED for both variants of Lookout [29] do not vary as severely as GeoGPT with respect to T_{matches} . Image pairs generated with our method tend to yield more SIFT matches, and are mainly affected by T_{error} . The quantitative TSED results in the main paper were evaluated at $T_{\text{matches}} = 10$, but these extended results show that our method yields higher TSED scores, remains consistent over a range of T_{matches} values, in all cases.

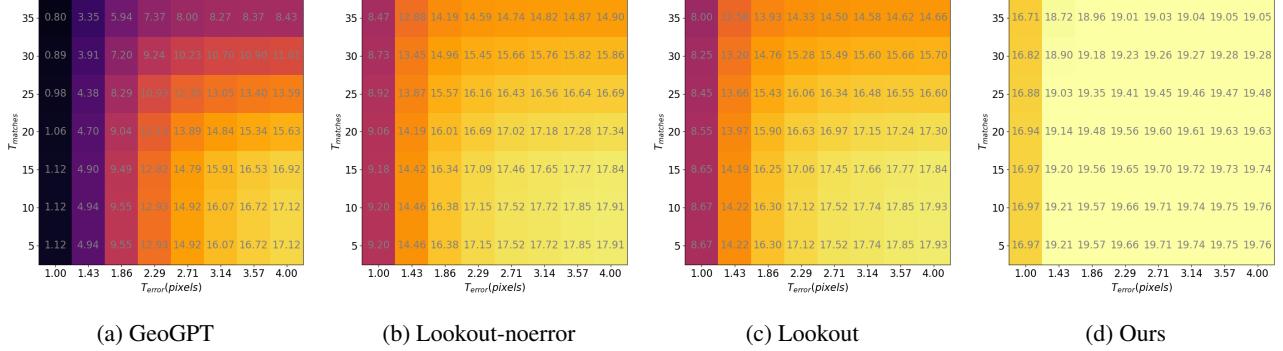


Figure E.12: TSED computed using images generated over in-distribution trajectories. We sweep over a range of values for T_{matches} and T_{error} . The values are provided as the average number of consistent pairs per sequence out of 20.

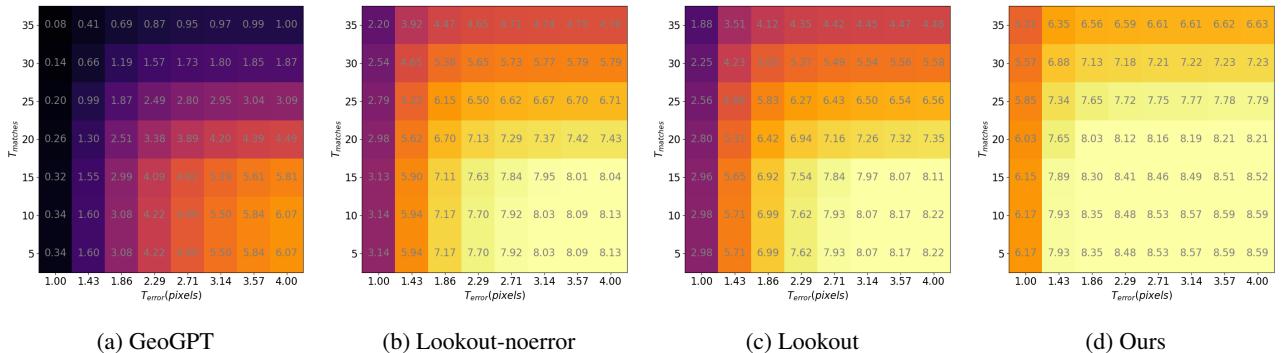


Figure E.13: TSED computed using images generated over *orbit* trajectory. We sweep over a range of values for T_{matches} and T_{error} . The values are provided as the average number of consistent pairs per sequence out of 9.

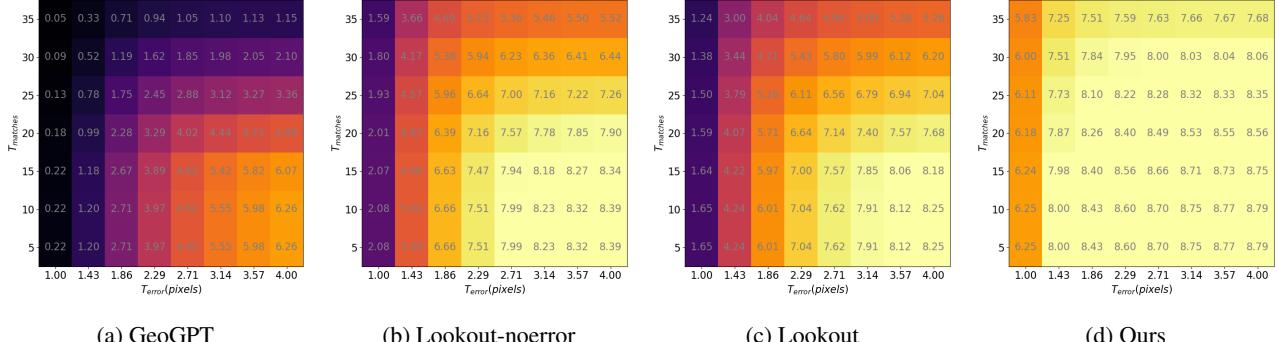


Figure E.14: TSED computed using images generated over *spin* trajectory. We sweep over a range of values for T_{matches} and T_{error} . The values are provided as the average number of consistent pairs per sequence out of 9.

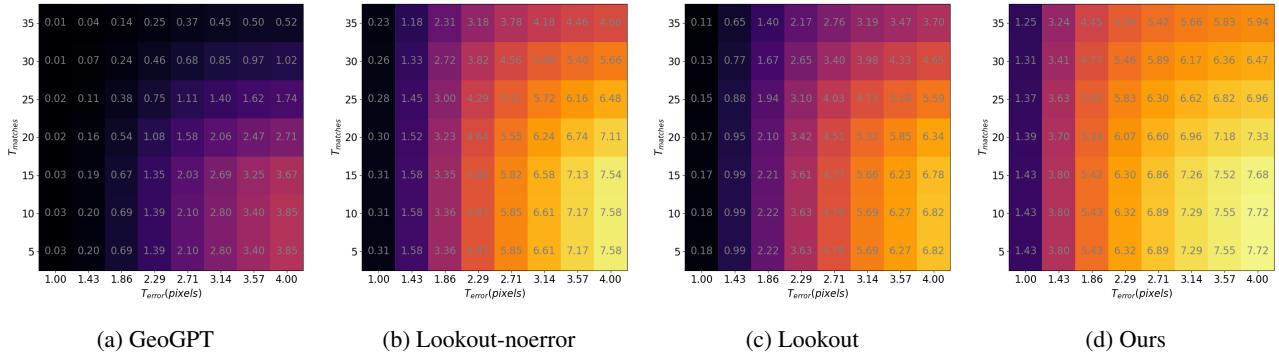


Figure E.15: TSED computed using images generated over our *hop* trajectory. We sweep over a range of values for T_{matches} and T_{error} . The values are provided as the average number of consistent pairs per sequence out of 9.

To provide a better intuition on how symmetric epipolar distance (SED) [11] provides a measure of consistency, we provide an interactive demo on our project page, <https://yorkucvlib.github.io/Photoconsistent-NVS/>, under the **Visualization of SED** section. The demo visualizes how SED varies in response to the positions of two correspondences in a pair of views with known relative camera geometry. Each point creates an epipolar line on the opposite image, and the minimal distance line between a point and a line on the same image is shown.

F. Limitations of Autoregressive Sampling

Our method and the baselines are limited by the use of sequential generation with a fixed budget for conditioning images. Regions that become occluded and subsequently disoccluded in a sequence are very likely to change appearance. For example, conditioning on one image prevents information about previously disoccluded regions from informing the generation of those same regions beyond one frame. Qualitative examples of this phenomenon can be seen on our project page, <https://yorkucvlib.github.io/Photoconsistent-NVS/>, under the **RealEstate10K Qualitative Results - Out-of-Distribution Trajectories** section, with the **Spin** motion. The described phenomenon can be observed at the edges of the images with **Spin** motion, where those regions of the scene often move beyond the image boundaries before returning in the future. A qualitative example of this is shown in Figure F.16.

Conditioning on an arbitrary number of frames could theoretically solve this problem. However, the practicality of this solution is limited by the ability to design models that can process an arbitrary number of inputs, and the model’s ability to generalize to out-of-distribution camera poses (*e.g.*, far away cameras in large scenes). Leveraging many images for generation is a potentially significant direction for future work.



(a) Initial image



(b) Image after returning close to the initial camera position.

Figure F.16: The initial frame and the final frame from a generated sequence with the *spin* motion. Notice the final frame has returned to a location similar to the initial frame, but the bottom left region on the floor has changed appearance.