NeuralField-LDM: Scene Generation with Hierarchical Latent Diffusion Models

Seung Wook Kim[1,2,3*] Bradley Brown[1,5*†] Kangxue Yin[1] Karsten Kreis[1] Katja Schwarz[6†]
Daiqing Li[1] Robin Rombach[7†] Antonio Torralba[4] Sanja Fidler[1,2,3]
[1]NVIDIA [2]University of Toronto [3]Vector Institute [4] CSAIL, MIT [5]University of Waterloo
[6]University of Tübingen, Tübingen AI Center [7]LMU Munich

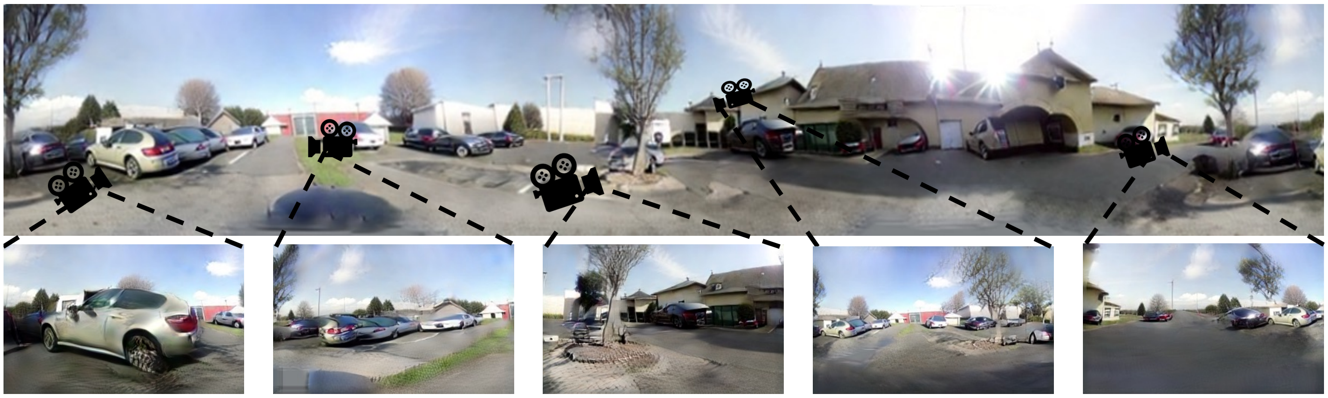

Figure 1. We introduce NeuralField-LDM, a generative model for complex open-world 3D scenes. This figure contains a panorama constructed from NeuralField-LDM's generated scene. We visualize different parts of the scene by placing cameras on them.

Abstract

Automatically generating high-quality real world 3D scenes is of enormous interest for applications such as virtual reality and robotics simulation. Towards this goal, we introduce NeuralField-LDM, a generative model capable of synthesizing complex 3D environments. We leverage Latent Diffusion Models that have been successfully utilized for efficient high-quality 2D content creation. We first train a scene auto-encoder to express a set of image and pose pairs as a neural field, represented as density and feature voxel grids that can be projected to produce novel views of the scene. To further compress this representation, we train a latent-autoencoder that maps the voxel grids to a set of latent representations. A hierarchical diffusion model is then fit to the latents to complete the scene generation pipeline. We achieve a substantial improvement over existing state-of-the-art scene generation models. Additionally, we show how NeuralField-LDM can be used for a variety of 3D content creation applications, including conditional scene generation, scene inpainting and scene style manipulation.

1. Introduction

There has been increasing interest in modelling 3D real-world scenes for use in virtual reality, game design, digital twin creation and more. However, designing 3D worlds by hand is a challenging and time-consuming process, requiring 3D modeling expertise and artistic talent. Recently, we have seen success in automating 3D content creation via 3D generative models that output individual object assets [15, 46, 73]. Although a great step forward, automating the generation of real-world scenes remains an important open problem and would unlock many applications ranging from scalably generating a diverse array of environments for training AI agents (*e.g.* autonomous vehicles) to the design of realistic open-world video games. In this work, we take a step towards this goal with NeuralField-LDM (NF-LDM), a generative model capable of synthesizing complex real-world 3D scenes. NF-LDM is trained on a collection of posed camera images and depth measurements which are easier to obtain than explicit ground-truth 3D data, offering a scalable way to synthesize 3D scenes.

Recent approaches [2, 6, 8] tackle the same problem of generating 3D scenes, albeit on less complex data. In [6, 8], a latent distribution is mapped to a set of scenes using adversarial training, and in GAUDI [2], a denoising diffusion model is fit to a set of scene latents learned using an auto-decoder. These models all have an inherent weakness of attempting to capture the entire scene into a single vector that conditions a neural radiance field. In practice, we find that this limits the ability to fit complex scene distributions.

Recently, diffusion models have emerged as a very powerful class of generative models, capable of generating high-

[*] Equal contribution.
[†] Work done during an internship at NVIDIA.

quality images, point clouds and videos [18, 24, 39, 48, 53, 73, 78]. Yet, due to the nature of our task, where image data must be mapped to a shared 3D scene without an explicit ground truth 3D representation, straightforward approaches fitting a diffusion model directly to data are infeasible.

In NeuralField-LDM, we learn to model scenes using a three-stage pipeline. First, we learn an auto-encoder that encodes scenes into a neural field, represented as density and feature voxel grids. Inspired by the success of latent diffusion models for images [53], we learn to model the distribution of our scene voxels in latent space to focus the generative capacity on core parts of the scene and not the extraneous details captured by our voxel auto-encoders. Specifically, a latent-autoencoder decomposes the scene voxels into a 3D coarse, 2D fine and 1D global latent. Hierarchichal diffusion models are then trained on the tri-latent representation to generate novel 3D scenes. We show how NF-LDM enables applications such as scene editing, birds-eye view conditional generation and style adaptation. Finally, we demonstrate how score distillation [46] can be used to optimize the quality of generated neural fields, allowing us to leverage the representations learned from state-of-the-art image diffusion models that have been exposed to orders of magnitude more data.

Our contributions are: 1) We introduce NF-LDM, a hierarchical diffusion model capable of generating complex open-world 3D scenes and achieving state of the art scene generation results on four challenging datasets. 2) We extend NF-LDM to semantic birds-eye view conditional scene generation, style modification and 3D scene editing.

2. Related Work

2D Generative Models In past years, generative adversarial networks (GANs) [3, 17, 27, 41, 58] and likelihood-based approaches [33, 49, 51, 67] enabled high-resolution photorealistic image synthesis. Due to their quality, GANs are used in a multitude of downstream applications ranging from steerable content creation [30,34,36,37,61,76] to data driven simulation [26,31,32,34]. Recently, autoregressive models and score-based models, e.g. diffusion models, demonstrate better distribution coverage while preserving high sample quality [10, 11, 13, 20, 22, 43, 48, 53, 54, 68]. Since evaluation and optimization of these approaches in pixel space is computationally expensive, [53, 68] apply them to latent space, achieving state-of-the-art image synthesis at megapixel resolution. As our approach operates on 3D scenes, computational efficiency is crucial. Hence, we build upon [53] and train our model in latent space.

Novel View Synthesis In their seminal work [42], Mildenhall et al. introduce Neural Radiance Fields (NeRF) as a powerful 3D representation. PixelNeRF [72] and IBRNet [70] propose to condition NeRF on aggregated features from multiple views to enable novel view synthesis from a sparse set of views. Another line of works scale NeRF to large-scale indoor and outdoor scenes [40, 50, 74, 75]. Recently, Nerfusion [75] predicts local radiance fields and fuses them into a scene representation using a recurrent neural network. Similarly, we construct a latent scene representation by aggregating features across multiple views. Different from the aforementioned methods, our approach is a generative model capable of synthesizing novel scenes.

3D Diffusion Models A few recent works propose to apply denoising diffusion models (DDM) [20,22,63] on point clouds for 3D shape generation [39,73,78]. While PVD [78] trains on point clouds directly, DPM [39] and LION [73] use a shape latent variable. Similar to LION, we design a hierarchical model by training separate conditional DDMs. However, our approach generates both texture and geometry of a scene without needing 3D ground truth as supervision.

3D-Aware Generative Models 3D-aware generative models synthesize images while providing explicit control over the camera pose and potentially other scene properties, like object shape and appearance. SGAM [62] generates a 3D scene by autoregressively generating sensor data and building a 3D map. Several previous approaches generate NeRFs of single objects with conditional coordinate-based MLPs [7, 44, 59]. GSN [8] conditions a coordinate-based MLP on a "floor plan", i.e. a 2D feature map, to model more complex indoor scenes. EG3D [6] and VoxGRAF [60] use convolutional backbones to generate 3D representations. All of these approaches rely on adversarial training. Instead, we train a DDM on voxels in latent space. The work closest to ours is GAUDI [2], which first trains an auto-decoder and subsequently trains a DDM on the learned latent codes. Instead of using a global latent code, we encode scenes onto voxel grids and train a hierarchical DDM to optimally combine global and local features.

3. NeuralField-LDM

Our objective is to train a generative model to synthesize 3D scenes that can be rendered to any viewpoint. We assume access to a dataset $\{(i, \kappa, \rho)\}_{1..N}$ which consists of N RGB images i and their camera poses κ, along with a depth measurement ρ that can be either sparse (*e.g.* Lidar points) or dense. The generative model must learn to model both the texture and geometry distributions of the dataset in 3D by learning solely from the sensor observations, which is a highly non-trivial problem.

Past work typically tackles this problem with a generative adversarial network (GAN) framework [6, 8, 59, 60]. They produce an intermediate 3D representation and render images for a given viewpoint with volume rendering [25, 42]. Discriminator losses then ensure that the 3D representation produces a valid image from any viewpoint. However, GANs come with notorious training instability

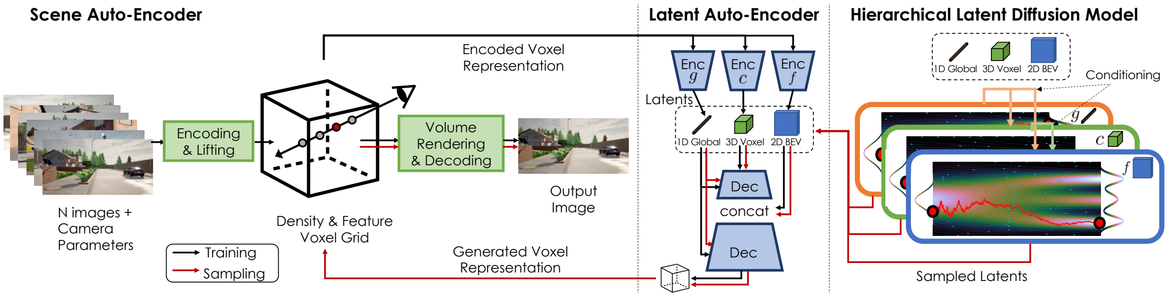

Figure 2. **Overview of NeuralField-LDM**. We first encode RGB images with camera poses into a neural field represented by density and feature voxel grids. We compress the neural field into smaller latent spaces and fit a hierarchical latent diffusion model on the latent space. Sampled latents can then be decoded into a neural field that can be rendered into a given viewpoint.

and mode dropping behaviors [1, 16, 35]. Denoising Diffusion models [20] (DDMs) have recently emerged as an alternative to GANs that avoid the aforementioned disadvantages [53, 56, 57]. However, DDMs model the data likelihood explicitly and are trained to reconstruct the training data. Thus, they have been used in limited scenarios [73, 77] since ground-truth 3D data is not readily available at scale.

To tackle the challenging problem of generating an entire scene with texture and geometry, we take inspiration from latent diffusion models (LDM) [53], which first construct an intermediate latent distribution of the training data then fit a diffusion model on the latent distribution. In Sec. 3.1, we introduce a scene auto-encoder that encodes the set of RGB images into a neural field representation consisting of density and feature voxel grids. To accurately capture a scene, the voxel grids' spatial dimension needs to be much larger than what current state-of-the-art LDMs can model. In Sec. 3.2, we show how we can further compress and decompose the explicit voxel grids into compressed latent representations to facilitate learning the data distribution. Finally, Sec. 3.3 introduces a latent diffusion model that models the latent distributions in a hierarchical manner. Fig. 2 shows an overview of our method, which we name NeuralField-LDM (NF-LDM). We provide training and additional architecture details in the supplementary.

3.1. Scene Auto-Encoder

The goal of the scene auto-encoder is to obtain a 3D representation of the scene from input images by learning to reconstruct them. Fig. 3 depicts the auto-encoding process. The scene encoder is a 2D CNN and processes each RGB image $i_{1..N}$ separately, producing a $\mathbb{R}^{H\times W\times(D+C)}$ dimensional 2D tensor for each image, where H and W are smaller than i's size. We follow a similar procedure to Lift-Splat-Shoot (LSS) [45] to lift each 2D image feature map and combine them in the common voxel-based 3D neural field. We build a discrete frustum of size $H \times W \times D$ with the camera poses κ for each image. This frustum contains image features and density values for each pixel, along a pre-defined discrete set of D depths. Unlike LSS, we take the first D channels of the 2D CNN's output and use them as density values. That is, the $d'th$ channel of the CNN's output at pixel (h, w) becomes the density value of the frustum entry at (h, w, d). Motivated by the volume rendering equation [42], we get the occupancy weight O of each element (h, w, d) in the frustum using the density values $\sigma \geq 0$:

$$O(h,w,d) = \exp(-\sum_{j=0}^{d-1}\sigma_{(h,w,j)}\delta_j)(1-\exp(-\sigma_{(h,w,d)}\delta_d)) \tag{1}$$

where h, w denotes the pixel coordinate of the frustum and δ_j is the distance between each depth in the frustum. Using the occupancy weights, we put the last C channels of the CNN's output into the frustum F:

$$F(h,w,d) = [O(h,w,d)\phi(h,w), \sigma(h,w,d)] \tag{2}$$

where $\phi(h, w)$ denotes the C-channeled feature vector at pixel (h, w) which is scaled by $O(h, w, d)$ for F at depth d.

After constructing the frustum for each view, we transform the frustums to world coordinates and fuse them into a shared 3D neural field, represented as density and feature voxel grids. Let $V_{\texttt{Density}}$ and $V_{\texttt{Feat}}$ denote the density and feature grid, respectively. This formulation of representing a scene with density and feature grids has been explored before [65] for optimization-based scene reconstruction and we utilize it as an intermediate representation for our scene auto-encoder. $V_{\texttt{Density,Feat}}$ have the same spatial size, and each voxel in V represents a region in the world coordinate system. For each voxel indexed by (x, y, z), we pool all densities and features of the corresponding frustum entries. In this paper, we simply take the mean of the pooled features. More sophisticated pooling functions (*e.g.* attention) can be used, which we leave as future work.

Finally, we perform volume rendering using the camera poses κ to project V onto a 2D feature map. We trilinearly interpolate the values on each voxel to get the feature and density for each sampling point along the camera rays. 2D features are then fed into a CNN decoder that produces the output image $\hat{i}$. We denote rendering of voxels to output images as $\hat{i} = r(V, \kappa)$. From the volume rendering process, we also get the expected depth $\hat{\rho}$ along each ray [50]. The scene

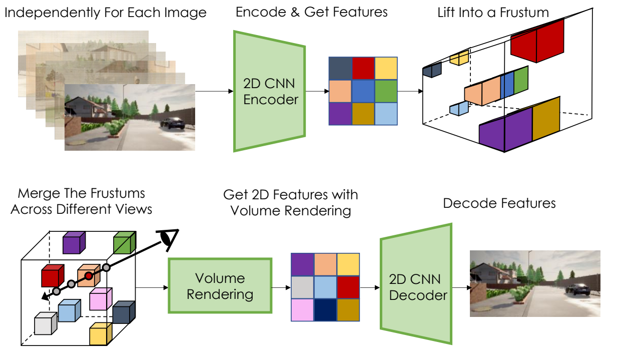
Independently For Each Image
Encode & Get Features
Lift Into a Frustum
2D CNN Encoder
Merge The Frustums Across Different Views
Get 2D Features with Volume Rendering
Decode Features
Volume Rendering
2D CNN Decoder

Figure 3. **Scene Auto-Encoder:** Each input image is processed with a 2D CNN then lifted up to 3D and merged into the shared voxel grids. Density prediction is not shown here for brevity.

auto-encoding pipeline is trained with an image reconstruction loss $||i - \hat{i}||$ and a depth supervision loss $||\rho - \hat{\rho}||$. In the case of sparse depth measurements, we only supervise the pixels with recorded depth. We can further improve the quality of the auto-encoder with adversarial loss as in VQ-GAN [13] or by doing a few optimization steps at inference time, which we discuss in the supplementary.

3.2. Latent Voxel Auto-Encoder

It is possible to fit a generative model on voxel grids obtained from Sec. 3.1. However, to capture real-world scenes, the dimensionality of the representation needs to be much larger than what SOTA diffusion models can be trained on. For example, Imagen [56] trains DDMs on 256×256 RGB images, and we use voxels of size $128 \times 128 \times 32$ with 32 channels. We thus introduce a latent auto-encoder (LAE) that compresses voxels into a 128-dimensional global latent as well as coarse (3D) and fine (2D) quantized latents with channel dimensions of four and spatial dimensions $32 \times 32 \times 16$ and 128×128 respectively.

We concatenate $V_{\texttt{Density}}$ and $V_{\texttt{Feat}}$ along the channel dimension and use separate CNN encoders to encode the voxel grid V into a hierarchy of three latents: 1D global latent g, 3D coarse latent c, and 2D fine latent f, as shown in Fig. 2. The intuition for this design is that g is responsible for representing the global properties of the scene, such as the time of the day, c represents coarse 3D scene structure, and f is a 2D tensor with the same horizontal size $X \times Y$ as V, which gives further details for each location (x, y) in bird's eye view perspective. We empirically found that 2D CNNs perform similarly to 3D CNNs while being more efficient, thus we use 2D CNNs throughout. To use 2D CNNs for the 3D input V, we concatenate V's vertical axis along the channel dimension and feed it to the encoders. We also add latent regularizations to avoid high variance latent spaces [53]. For the 1D vector g, we use a small KL-penalty via the reparameterization trick [33], and for c and f, we impose a vector-quantization [13, 69] layer to regularize them.

The CNN decoder is similarly a 2D CNN, and takes c, concatenated along vertical axis, as the initial input. The decoder uses conditional group normalization layers [71] with g as the conditioning variable. Lastly, we concatenate f to an intermediate tensor in the decoder. The latent decoder outputs $\hat{V}$ which is the reconstructed voxel. LAE is trained with the voxel reconstruction loss $||V - \hat{V}||$ along with the image reconstruction loss $||i - \hat{i}||$ where $\hat{i} = r(\hat{V}, \kappa)$. Note that the image reconstruction loss only helps with the learning of LAE, and the scene auto-encoder is kept fixed.

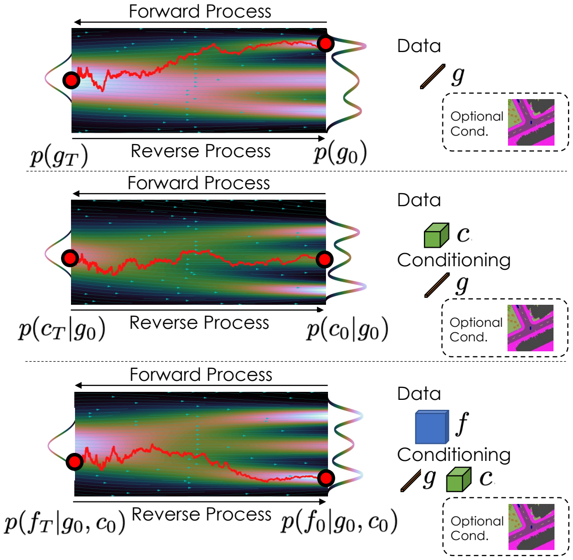

Figure 4. **Hierarchical LDM. Top:** LDM ψ_g for KL-regulairzed global latent g. **Middle:** LDM ψ_c for vector-quantized coarse latent c. **Bottom:** LDM ψ_f for vector-quantized fine latent f. All LDMs optionally take an additional conditioning variable as input, such as a Bird's Eye View segmentation map as depicted here.

3.3. Hierarchical Latent Diffusion Models

Given the latent variables g, c, f that represent a voxel-based scene representation V, we define our generative model as $p(V, g, c, f) = p(V|g, c, f)p(f|g, c)p(c|g)p(g)$ with Denoising Diffusion Models (DDMs) [21]. In general, DDMs with discrete time steps have a fixed Markovian forward process $q(x_t|x_{t-1})$ where $q(x_0)$ denotes the data distribution and $q(x_T)$ is defined to be close to the standard normal distribution, where we use the subscript to denote the time step. DDMs then learn to revert the forward process $p_\theta(x_{t-1}|x_t)$ with learnable parameters θ. It can be shown that learning the reverse process is equivalent to learning to denoise x_t to x_0 for all timesteps t [21, 24] by reducing the following loss:

$$\mathop{\mathbb{E}}_{t,\epsilon,x_0}\left[w(\lambda_t)||x_0 - \hat{x}_\theta(x_t, t)||_2^2\right] \qquad (3)$$

where t is sampled from a uniform distribution for timesteps, ϵ is sampled from the standard normal to noise the data x_0, $w(\lambda_t)$ is a timestep dependent weighting constant, and $\hat{x}_\theta$ denotes the learned denoising model.

We train our hierarchical LDM with the following losses:

$$\mathop{\mathbb{E}}_{t,\epsilon,g_0}\left[w(\lambda_t)||g_0 - \psi_g(g_t,t)||_2^2\right] \quad (4)$$

$$\mathop{\mathbb{E}}_{t,\epsilon,g_0,c_0}\left[w(\lambda_t)||c_0 - \psi_c(c_t,g_0,t)||_2^2\right] \quad (5)$$

$$\mathop{\mathbb{E}}_{t,\epsilon,g_0,c_0,f_0}\left[w(\lambda_t)||f_0 - \psi_f(f_t,g_0,c_0,t)||_2^2\right] \quad (6)$$

where ψ denotes the learnable denoising networks for g, c, f. Fig. 4 visualizes the diffusion models. ψ_g is implemented with linear layers with skip connections and ψ_c and ψ_f adopt the U-net architecture [55]. g is fed into ψ_c and ψ_f with conditional group normalization layers. c is interpolated and concatenated to the input to ψ_f. The camera poses contain the trajectory the camera is travelling, and this information can be useful for modelling a 3D scene as it tells the model where to focus on generating. Therefore, we concatenate the camera trajectory information to g and also learn to sample it. For brevity, we still call the concatenated vector g. For conditional generation, each ψ takes the conditioning variable as input with cross-attention layers [53].

Each ψ can be trained in parallel and, once trained, can be sampled one after another following the hierarchy. In practice, we use the v-prediction parameterization [57] that has been shown to have better convergence and training stability [56, 57]. Once g, c, f are sampled, we can use the latent decoder from Sec. 3.2 to construct the voxel V which represents the neural field for the sampled scene.. Following the volume rendering and decoding step in Sec. 3.1, the sampled scene can be visualized from desired viewpoints.

3.4. Post-Optimizing Generated Neural Fields

Samples generated from our model on real-world data contain reasonable texture and geometry (Fig. 10), but can be further optimized by leveraging recent advances in 2D image diffusion models trained on orders of magnitude more data. Specifically, we iteratively update initially generated voxels, V, by rendering viewpoints from the scene and applying Score Distillation Sampling (SDS) [46] loss on each image independently:

$$\nabla_V L_{SDS} = \mathop{\mathbb{E}}_{\epsilon,t,\kappa}\left[w(\lambda_t)(\epsilon - \hat{\epsilon_\theta}(r(V,\kappa),t))\frac{\partial r(V,\kappa)}{\partial V}\right] \quad (7)$$

where κ is sampled uniformly in a $6m^2$ region around the origin of the scene with random rotation about the vertical axis, $w(\lambda_t)$ is the weighting schedule used to train $\hat{\epsilon_\theta}$ and $t \sim U[0.02T, 0.2T]$ where T is the amount of noise steps used to train $\hat{\epsilon_\theta}$. Note that for latent diffusion models, the noise prediction step is applied after encoding $r(V,\kappa)$ to the LDM's latent space and the partial gradient term is updated appropriately. For $\hat{\epsilon_\theta}$, we use an off-the-shelf latent diffusion model [53], finetuned to condition on CLIP image embeddings [47][1]. We found that CLIP contains a representation of the quality of images that the LDM is able to interpret: denoising an image while conditioning on CLIP image embeddings of our model's samples produced images with similar geometry distortions and texture errors. We leverage this property by optimizing L_{SDS} with negative guidance. Letting y, y' be CLIP embeddings of clean image conditioning (*e.g.* dataset images) and artifact conditioning (*e.g.* samples) respectively, we perform classifier-free guidance [23] with conditioning vector y, but replace the unconditional embedding with y'. As shown in the supplementary, this is equivalent (up to scale) to sampling from $\frac{p(x|y)^\alpha}{p(x|y')}$ at each denoising step where α controls the trade-off between sampling towards dataset images and away from images with artifacts. We stress that this post-optimization is only successful due to the strong scene prior contained in our voxel samples, as shown by our comparison to running optimization on randomly initialized voxels in the supplementary.

[1] https://github.com/justinpinkney/stable-diffusion

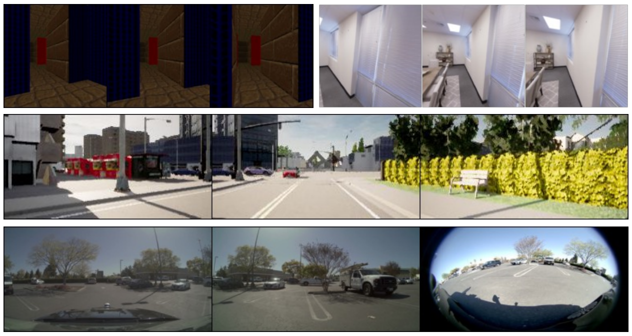

Figure 5. **Datasets:** Top-left: VizDoom [29]. Top-right: Replica [64]. Middle: Carla [12] . Bottom: AVD. For Carla and AVD, we visualize a subset of available cameras.

4. Experiments

We evaluate NeuralField-LDM on the following four datasets (Fig. 5). Each dataset contains RGB images and a depth measurement with their corresponding camera poses.

VizDoom [28] consists of front-view sensor observations obtained by navigating inside a synthetic game environment. We use the dataset provided by [9], which contains 34 trajectories with a length of 600.

Replica [64] is a dataset of high-quality reconstructions of 18 indoor scenes, containing 101 front-view trajectories with a length of 100. We use the dataset provided by [9].

Carla [12] is an open-source simulation platform for self-driving research. We mimic the camera settings for a self-driving car, by placing six cameras *(front-left, front, front-right, back-left, back, back-right)*, covering 360 degrees with some overlaps, and move the car in a randomly sampled direction and distance for 10 timesteps. We sample 43K datapoints, each containing 60 images.

AVD is an in-house dataset of human driving recordings in roads and parking lots. It has ten cameras with varying

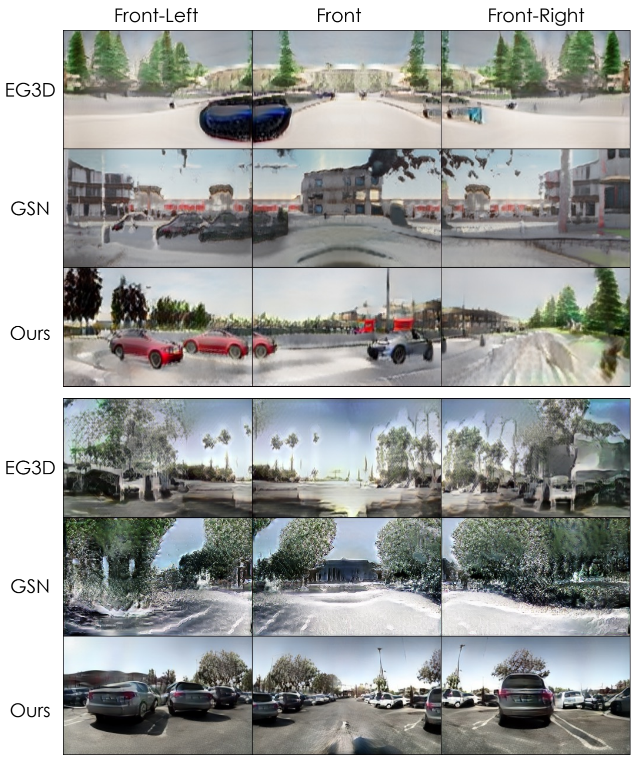

Figure 6. **Generated Scenes:** The top three rows are samples from Carla, and the bottom three rows are samples from AVD.

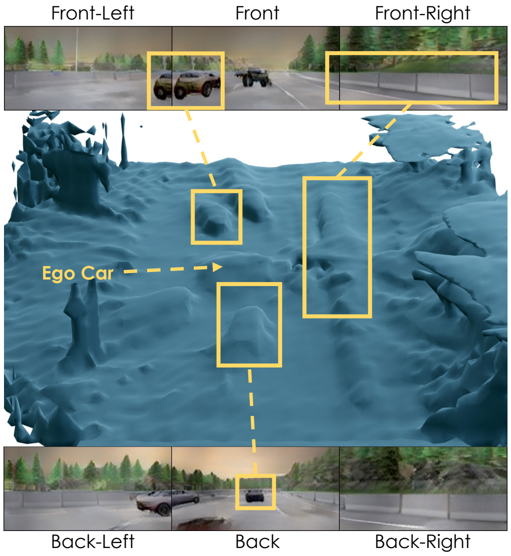

Figure 7. We run marching-cubes [38] on the density voxels to visualize the geometry of the samples generated by NF-LDM. The structure of the scene is reflected well in the mesh.

lens types along with Lidar for depth measurement. It has 73K sequences, each with 8 frames extracted at 10 fps.

Criterion	Method	Depth	VizDoom	Replica
FID (↓)	GRAF [59]	✗	47.50	65.37
	π-GAN [4]	✗	143.55	166.55
	GSN [8]	✓	37.21	41.75
	GAUDI [2]	✓	33.70	18.75
	NF-LDM	✓	**19.54***	**14.59**

Table 1. FID [19] scores on VizDoom and Replica. NF-LDM outperforms all baseline models. Baseline numbers are from [2].

Criterion	Method	Depth	Carla	AVD
FID (↓)	EG3D [5]	✗	76.89	194.34
	GSN [8]	✓	75.45	166.07
	NF-LDM	✓	**35.69**	**54.26**

Table 2. FID [19] scores on Carla and AVD datasets. Baseline models have trouble learning the distribution of complex outdoor datasets, in particular AVD, while NF-LDM models them well.

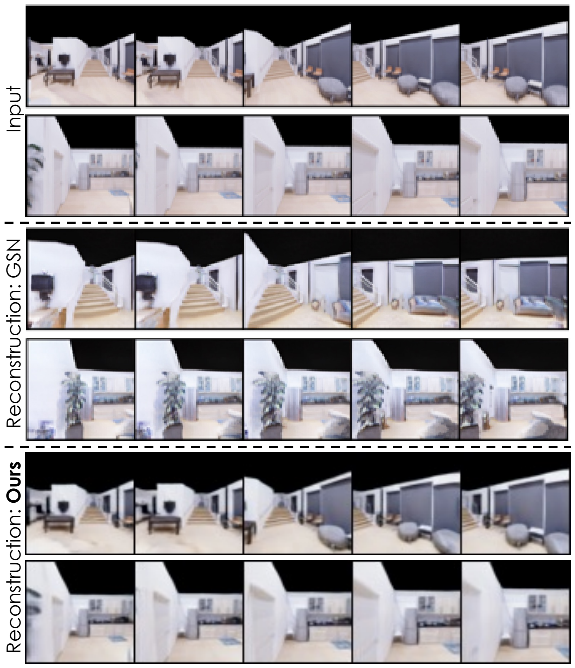

Figure 8. Reconstructing held-out scenes not seen during training.

4.1. Baseline Comparisons

Unconditional Generation We evaluate the unconditional generation performance of NF-LDM by comparing it with baseline models. All results are without the post-optimization step (Sec. 3.4), unless specified. Tab. 1 shows the results on VizDoom and Replica. GRAF [59] and π-GAN [4], which do not utilize ground truth depth in training, have shown successes in modelling single objects, but they exhibit worse performance than others that leverages depth information for modelling scenes. GAUDI [2] is an auto-decoder-based diffusion model. Their auto-decoder optimizes a small per-scene latent to reconstruct its matching scene. GAUDI comes with the advantage that learning the generative model is simple as it only needs to model the small dimensional latent distribution that acts as the key to their corresponding scenes. On the contrary, NF-LDM is trained on the latents that are a decomposition of the explicit 3D neural field. Therefore, GAUDI puts more modelling

Criterion	Method	Depth	Carla	AVD
FVD (↓)	EG3D [5]	✗	134.94	1232.38
	GSN [8]	✓	184.30	1659.81
	NF-LDM	✓	**91.80**	**242.50**

Table 3. FVD [66] scores on Carla and AVD Datasets. As for FID, baseline models have trouble learning to model complex datasets.

capacity into the auto-decoder part, and NF-LDM puts it more into the generative model part. We attribute our improvement over GAUDI to our expressive hierarchical LDM that can model the details of the scenes better. In VizDoom, only one scene exists, and each sequence contains several hundred steps covering a large area in the scene, which our voxels were not large enough to encode. Therefore, we chunked each VizDoom trajectory to be 50 steps long.

Tab. 2 shows results on complex outdoor datasets: Carla and AVD. We compare with EG3D [5] and GSN [8]. Both are GAN-based 3D generative models, but GSN utilizes ground truth depth measurements. Note that we did not include GAUDI [2] as the code was not available. NF-LDM achieves the best performance, and both baseline models have difficulty modelling the real outdoor dataset (AVD). Fig. 6 compares the samples from different models.

Since the datasets are composed of frame sequences, we can treat them as videos and further evaluate with Fréchet Video Distance (FVD) [66] to compare the distributions of the dataset and sampled sequences. This can quantify samples' 3D structure by how natural the rendered sequence from a moving camera is. For EG3D and GSN, we randomly sample a trajectory from the datasets and for NF-LDM, we sample a trajectory from the global latent diffusion model. Tab. 3 shows that NF-LDM achieves the best results. We empirically observed GSN sometimes produced slightly inconsistent rendering, which could attribute to its lower FVD score than EG3D's. We also visualize the geometry of NF-LDM's samples by running marching-cubes [38] on the density voxels. Fig. 7 shows that our samples produce a coarse but realistic geometry.

Ablations We evaluate the hierarchical structure of NF-LDM. Tab. 4 shows an ablation study on Carla. The model with the full hierarchy achieves the best performance. The global latent makes it easier for the other LDMs to sample as conditioning on the global properties of the scene (*e.g.* time of day) narrows down the distribution they need to model. The 2D fine latent helps retain the residual information missing in the 3D coarse latent, thus improving the latent auto-encoder and, consequently, the LDMs.

Scene Reconstruction Unlike previous approaches, NF-LDM has an explicit scene auto-encoder that can be used for scene reconstruction. GAUDI [2] is auto-decoder based, so it is not trivial to infer a latent for a new scene. GSN [8] can invert a new scene using a GAN inversion method [52, 79], but as Fig. 8 shows, it fails to get the details of the scene correct. Our scene auto-encoder generalizes well and is scalable as the number of scenes grow.

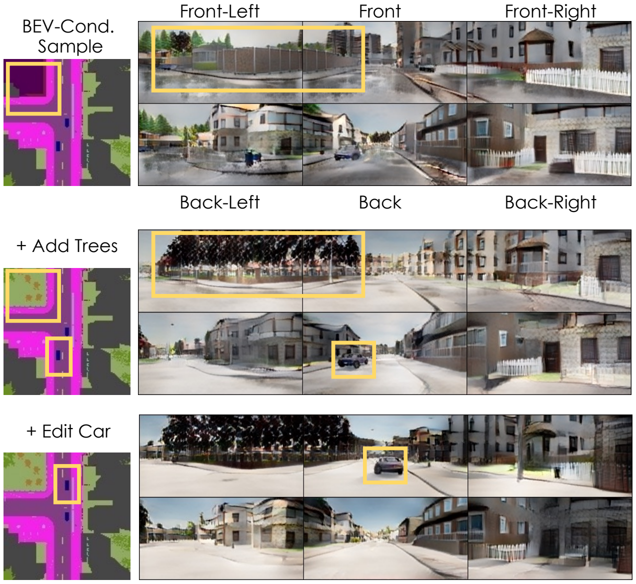

Figure 9. **BEV-Conditioned Synthesis**: NF-LDM allows controllable generation by editing the BEV segmentation map. From the initial sample, we add trees (green) and then edit the location of the car (blue). Note the ego car is at the center and thus not rendered.

	Coarse lat. c	+ Fine lat. f	+ Global lat. g
FID (↓)	46.43	43.52	**35.69**

Table 4. FID [19] on ablating the choice of hierarchy on the Carla dataset. The first column is for training both LAE and LDM only with the coarse latent. The last column is our full model.

4.2. Applications and Limitations

Conditional Synthesis NF-LDM can utilize additional conditioning signals for controllable generation. In this paper, we consider Bird's Eye View (BEV) segmentation maps, but our model can be extended to other conditioning variables. We use cross attention layers [53], which have been shown to be effective for conditional synthesis. Fig. 9 shows that NF-LDM follows the given BEV map faithfully and how the map can be edited for controllable synthesis.

Scene Editing Image diffusion models can be used for image inpainting without explicit training on the task [53, 63]. We leverage this property to edit scenes in 3D by resampling a region in the 3D coarse latent c. Specifically, at each denoising step, we noise the region to be kept and concatenate with the region being sampled, and pass it through the diffusion model. We use reconstruction guidance [24] to better harmonize the sampled and kept regions. After we get a new c, the fine latent is also re-sampled conditioned on c. Fig. 11 shows results on scene editing with NF-LDM.

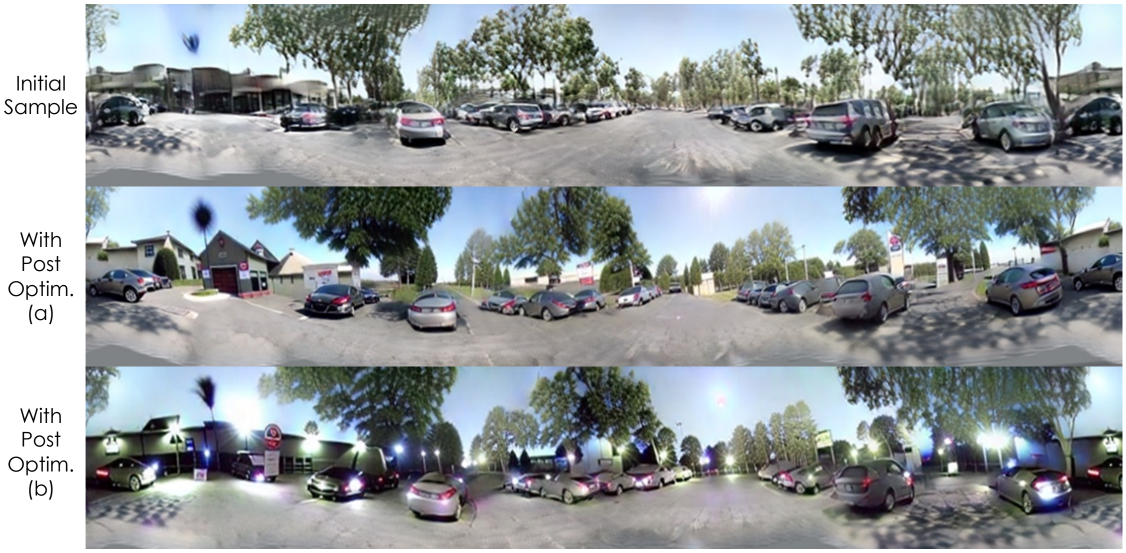

Figure 10. **Panoramas from NF-LDM's samples:** From the initial sample at the top, we apply post-optimization with Score Distillation Sampling [46] (Sec. 3.4). (a) demonstrates improved sample quality. (b) showcases style modification by conditioning on evening scenes.

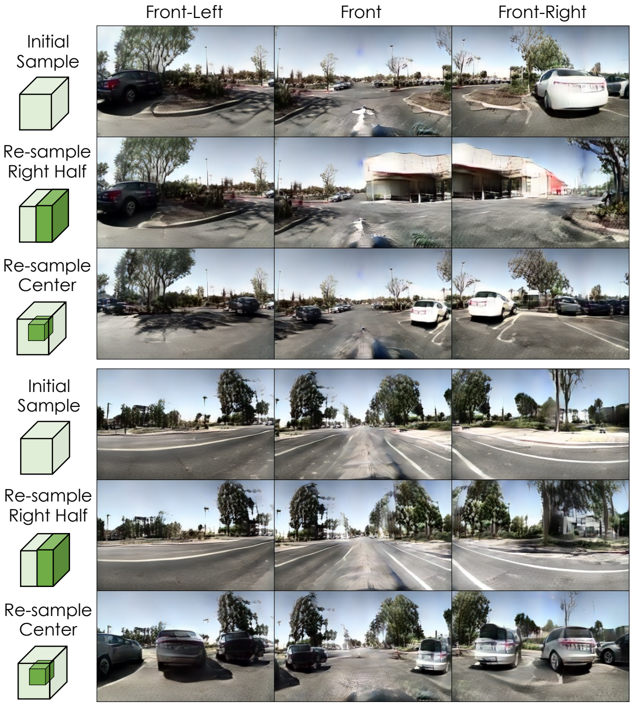

Figure 11. **Scene Editing**: We use the 3D coarse latent c for scene editing. From the initial sample indicated by light green, we re-sample a part of the latent, indicated by dark green.

Post-Optimization Fig. 10 shows how post-optimization (Sec. 3.4) can improve the quality of NF-LDM's initial sample while retaining the 3D structure. In addition to improving quality, we can also modify scene properties, such as time of day and weather, by conditioning the LDM on images with the desired properties. SDS-based style modification is effective for changes where a set of clean image data is available with the desired property and is reasonably close to our dataset's domain (*e.g*. street images for AVD). In the supplementary, we also provide results experimenting with directional CLIP loss [14] to quickly finetune our scene decoder for a given text prompt.

Limitations NF-LDM's hierarchical structure and three stage pipeline allows us to achieve high-quality generations and reconstructions, but it comes with a degradation in training time and sampling speed. In this work, the neural field representation is based on dense voxel grids, and it becomes expensive to volume render and learn the diffusion models as they get larger. Therefore, exploring alternative sparse representations is a promising future direction. Lastly, our method requires multi-view images which limits data availability and therefore risks universal problems in generative modelling of overfitting. For example, we found that output samples in AVD had limited diversity because the dataset itself was recorded in a limited number of scenes.

5. Conclusion

We introduced NeuralField-LDM (NF-LDM), a generative model for complex 3D environments. NF-LDM first constructs an expressive latent distribution by encoding input images into a 3D neural field representation which is further compressed into more abstract latent spaces. Then, our proposed hierarchical LDM is fit onto the latent spaces, achieving state-of-the-art performance on 3D scene generation. NF-LDM enables a diverse set of applications, including controllable scene generation and scene editing. Future directions include exploring more efficient sparse voxel representations, training on larger-scale real-world data and learning to continously expand generated scenes.

References

[1] Martin Arjovsky and Léon Bottou. Towards principled methods for training generative adversarial networks. In *ICLR*, 2017. 3

[2] Miguel Ángel Bautista, Pengsheng Guo, Samira Abnar, Walter Talbott, Alexander Toshev, Zhuoyuan Chen, Laurent Dinh, Shuangfei Zhai, Hanlin Goh, Daniel Ulbricht, Afshin Dehghan, and Josh M. Susskind. GAUDI: A neural architect for immersive 3d scene generation. 2022. 1, 2, 6, 7

[3] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *ICLR*, 2019. 2

[4] Eric Chan, Marco Monteiro, Petr Kellnhofer, Jiajun Wu, and Gordon Wetzstein. pi-gan: Periodic implicit generative adversarial networks for 3d-aware image synthesis. *arXiv*, 2012.00926, 2020. 6

[5] 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 *arXiv*, 2021. 6, 7

[6] 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. 1, 2

[7] Eric R. 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

[8] Terrance DeVries, Miguel Ángel Bautista, Nitish Srivastava, Graham W. Taylor, and Joshua M. Susskind. Unconstrained scene generation with locally conditioned radiance fields. In *ICCV*, 2021. 1, 2, 6, 7

[9] Terrance DeVries, Miguel Angel Bautista, Nitish Srivastava, Graham W. Taylor, and Joshua M. Susskind. Unconstrained scene generation with locally conditioned radiance fields. *arXiv*, 2021. 5

[10] Prafulla Dhariwal and Alexander Quinn Nichol. Diffusion models beat gans on image synthesis. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, editors, *NeurIPS*, 2021. 2

[11] Tim Dockhorn, Arash Vahdat, and Karsten Kreis. Score-based generative modeling with critically-damped langevin diffusion. In *ICLR*, 2022. 2

[12] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator. 2017. 5

[13] Patrick Esser, Robin Rombach, and Björn Ommer. Taming transformers for high-resolution image synthesis. In *CVPR*, 2021. 2, 4

[14] Rinon Gal, Or Patashnik, Haggai Maron, Gal Chechik, and Daniel Cohen-Or. Stylegan-nada: Clip-guided domain adaptation of image generators, 2021. 8

[15] Jun Gao, Tianchang Shen, Zian Wang, Wenzheng Chen, Kangxue Yin, Daiqing Li, Or Litany, Zan Gojcic, and Sanja Fidler. Get3d: A generative model of high quality 3d textured shapes learned from images. In *Advances In Neural Information Processing Systems*, 2022. 1

[16] Ian Goodfellow. Nips 2016 tutorial: Generative adversarial networks. *arXiv preprint arXiv:1701.00160*, 2016. 3

[17] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. Generative adversarial nets. In *NeurIPS*, 2014. 2

[18] William Harvey, Saeid Naderiparizi, Vaden Masrani, Christian Weilbach, and Frank Wood. Flexible diffusion modeling of long videos, 2022. 2

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

[20] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *NeurIPS*, 2020. 2, 3

[21] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 6840–6851. Curran Associates, Inc., 2020. 4

[22] Jonathan Ho, Chitwan Saharia, William Chan, David J. Fleet, Mohammad Norouzi, and Tim Salimans. Cascaded diffusion models for high fidelity image generation. *J. Mach. Learn. Res.*, 2022. 2

[23] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. 5

[24] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *arXiv:2204.03458*, 2022. 2, 4, 7

[25] James T Kajiya and Brian P Von Herzen. Ray tracing volume densities. *ACM SIGGRAPH computer graphics*, 18(3):165–174, 1984. 2

[26] Amlan Kar, Aayush Prakash, Ming-Yu Liu, Eric Cameracci, Justin Yuan, Matt Rusiniak, David Acuna, Antonio Torralba, and Sanja Fidler. Meta-sim: Learning to generate synthetic datasets. In *ICCV*, 2019. 2

[27] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of StyleGAN. In *CVPR*, 2020. 2

[28] Michał Kempka, Marek Wydmuch, Grzegorz Runc, Jakub Toczek, and Wojciech Jaśkowski. Vizdoom: A doom-based ai research platform for visual reinforcement learning. In *2016 IEEE conference on computational intelligence and games (CIG)*, pages 1–8. IEEE, 2016. 5

[29] Margret Keuper, Siyu Tang, Bjoern Andres, Thomas Brox, and Bernt Schiele. Motion segmentation & multiple object tracking by correlation co-clustering. *IEEE TPAMI*, 2018. 5

[30] Seung Wook Kim, Karsten Kreis, Daiqing Li, Antonio Torralba, and Sanja Fidler. Polymorphic-gan: Generating aligned samples across multiple domains with learned morph maps. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2022. 2

[31] Seung Wook Kim, Jonah Philion, Antonio Torralba, and Sanja Fidler. Drivegan: Towards a controllable high-quality neural simulation. In *CVPR*, 2021. 2

[32] Seung Wook Kim, Yuhao Zhou, Jonah Philion, Antonio Torralba, and Sanja Fidler. Learning to simulate dynamic environments with gamegan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1231–1240, 2020. 2

[33] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. *ICLR*, 2014. 2, 4

[34] Daiqing Li, Huan Ling, Seung Wook Kim, Karsten Kreis, Adela Barriuso, Sanja Fidler, and Antonio Torralba. Bigdatasetgan: Synthesizing imagenet with pixel-wise annotations. *arXiv*, 2201.04684, 2022. 2

[35] Ke Li and Jitendra Malik. On the implicit assumptions of gans. *arXiv preprint arXiv:1811.12402*, 2018. 3

[36] Yiyi Liao, Katja Schwarz, Lars Mescheder, and Andreas Geiger. Towards unsupervised learning of generative models for 3d controllable image synthesis. In *CVPR*, 2020. 2

[37] Huan Ling, Karsten Kreis, Daiqing Li, Seung Wook Kim, Antonio Torralba, and Sanja Fidler. Editgan: High-precision semantic image editing. In *NeurIPS*, 2021. 2

[38] William E. Lorensen and Harvey E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *ACM Trans. on Graphics*, 1987. 6, 7

[39] Shitong Luo and Wei Hu. Diffusion probabilistic models for 3d point cloud generation. In *CVPR*, 2021. 2

[40] 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

[41] Lars Mescheder, Andreas Geiger, and Sebastian Nowozin. Which training methods for gans do actually converge? In *Proc. of the International Conf. on Machine learning (ICML)*, 2018. 2

[42] 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. 2, 3

[43] Alexander Quinn Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. GLIDE: towards photorealistic image generation and editing with text-guided diffusion models. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato, editors, *Proc. of the International Conf. on Machine learning (ICML)*, 2022. 2

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

[45] Jonah Philion and Sanja Fidler. Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In *ECCV*, 2020. 3

[46] Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv*, 2022. 1, 2, 5, 8

[47] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. *arXiv*, 2103.00020, 2021. 5

[48] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with CLIP latents. *arXiv*, abs/2204.06125, 2022. 2

[49] Ali Razavi, Aäron van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with VQ-VAE-2. In *NeurIPS*, 2019. 2

[50] Konstantinos Rematas, Andrew Liu, Pratul P. Srinivasan, Jonathan T. Barron, Andrea Tagliasacchi, Thomas A. Funkhouser, and Vittorio Ferrari. Urban radiance fields. In *CVPR*, 2022. 2, 3

[51] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *Proc. of the International Conf. on Machine learning (ICML)*, 2014. 2

[52] Elad Richardson, Yuval Alaluf, Or Patashnik, Yotam Nitzan, Yaniv Azar, Stav Shapiro, and Daniel Cohen-Or. Encoding in style: a stylegan encoder for image-to-image translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2287–2296, 2021. 7

[53] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. 2, 3, 4, 5, 7

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

[55] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 5

[56] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S Sara Mahdavi, Rapha Gontijo Lopes, et al. Photorealistic text-to-image diffusion models with deep language understanding. *arXiv preprint arXiv:2205.11487*, 2022. 3, 4, 5

[57] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. *arXiv preprint arXiv:2202.00512*, 2022. 3, 5

[58] Axel Sauer, Katja Schwarz, and Andreas Geiger. Stylegan-xl: Scaling stylegan to large diverse datasets. *ACM Trans. on Graphics*, 2022. 2

[59] Katja Schwarz, Yiyi Liao, Michael Niemeyer, and Andreas Geiger. Graf: Generative radiance fields for 3d-aware image synthesis. 2020. 2, 6

[60] Katja Schwarz, Axel Sauer, Michael Niemeyer, Yiyi Liao, and Andreas Geiger. Voxgraf: Fast 3d-aware image synthesis with sparse voxel grids. In *NeurIPS*, 2022. 2

[61] Yujun Shen, Jinjin Gu, Xiaoou Tang, and Bolei Zhou. Interpreting the latent space of gans for semantic face editing. In *CVPR*, 2020. 2

[62] Yuan Shen, Wei-Chiu Ma, and Shenlong Wang. SGAM: Building a virtual 3d world through simultaneous generation and mapping. In *Thirty-Sixth Conference on Neural Information Processing Systems*, 2022. 2

[63] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020. 2, 7

[64] Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J. Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, Anton Clarkson, Mingfei Yan, Brian Budge, Yajie Yan, Xiaqing Pan, June Yon, Yuyang Zou, Kimberly Leon, Nigel Carter, Jesus Briales, Tyler Gillingham, Elias Mueggler, Luis Pesqueira, Manolis Savva, Dhruv Batra, Hauke M. Strasdat, Renzo De Nardi, Michael Goesele, Steven Lovegrove, and Richard Newcombe. The Replica dataset: A digital replica of indoor spaces. *arXiv*, 1906.05797, 2019. 5

[65] Cheng Sun, Min Sun, and Hwann-Tzong Chen. Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. *CVPR*, 2022. 3

[66] Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphael Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric & challenges. *arXiv preprint arXiv:1812.01717*, 2018. 7

[67] Arash Vahdat and Jan Kautz. NVAE: A deep hierarchical variational autoencoder. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *NeurIPS*, 2020. 2

[68] Arash Vahdat, Karsten Kreis, and Jan Kautz. Score-based generative modeling in latent space. In *NeurIPS*, 2021. 2

[69] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017. 4

[70] Qianqian Wang, Zhicheng Wang, Kyle Genova, Pratul P. Srinivasan, Howard Zhou, Jonathan T. Barron, Ricardo Martin-Brualla, Noah Snavely, and Thomas A. Funkhouser. Ibrnet: Learning multi-view image-based rendering. In *CVPR*, 2021. 2

[71] Yuxin Wu and Kaiming He. Group normalization. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018. 4

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

[73] Xiaohui Zeng, Arash Vahdat, Francis Williams, Zan Gojcic, Or Litany, Sanja Fidler, and Karsten Kreis. LION: latent point diffusion models for 3d shape generation. In *NeurIPS*, 2022. 1, 2, 3

[74] Kai Zhang, Gernot Riegler, Noah Snavely, and Vladlen Koltun. Nerf++: Analyzing and improving neural radiance fields. *arXiv*, 2010.07492, 2020. 2

[75] Xiaoshuai Zhang, Sai Bi, Kalyan Sunkavalli, Hao Su, and Zexiang Xu. Nerfusion: Fusing radiance fields for large-scale scene reconstruction. In *CVPR*, 2022. 2

[76] Yuxuan Zhang, Wenzheng Chen, Huan Ling, Jun Gao, Yinan Zhang, Antonio Torralba, and Sanja Fidler. Image gans meet differentiable rendering for inverse graphics and interpretable 3d neural rendering. In *ICLR*, 2021. 2

[77] Yan Zheng, Lemeng Wu, Xingchao Liu, Zhen Chen, Qiang Liu, and Qixing Huang. Neural volumetric mesh generator. *arXiv preprint arXiv:2210.03158*, 2022. 3

[78] Linqi Zhou, Yilun Du, and Jiajun Wu. 3d shape generation and completion through point-voxel diffusion. In *ICCV*, 2021. 2

[79] Jiapeng Zhu, Yujun Shen, Deli Zhao, and Bolei Zhou. In-domain gan inversion for real image editing. In *European conference on computer vision*, pages 592–608. Springer, 2020. 7