This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

LT3SD: Latent Trees for 3D Scene Diffusion

Quan Meng Lei Li Matthias Nießner Angela Dai
Technical University of Munich
quan-meng.github.io/projects/lt3sd

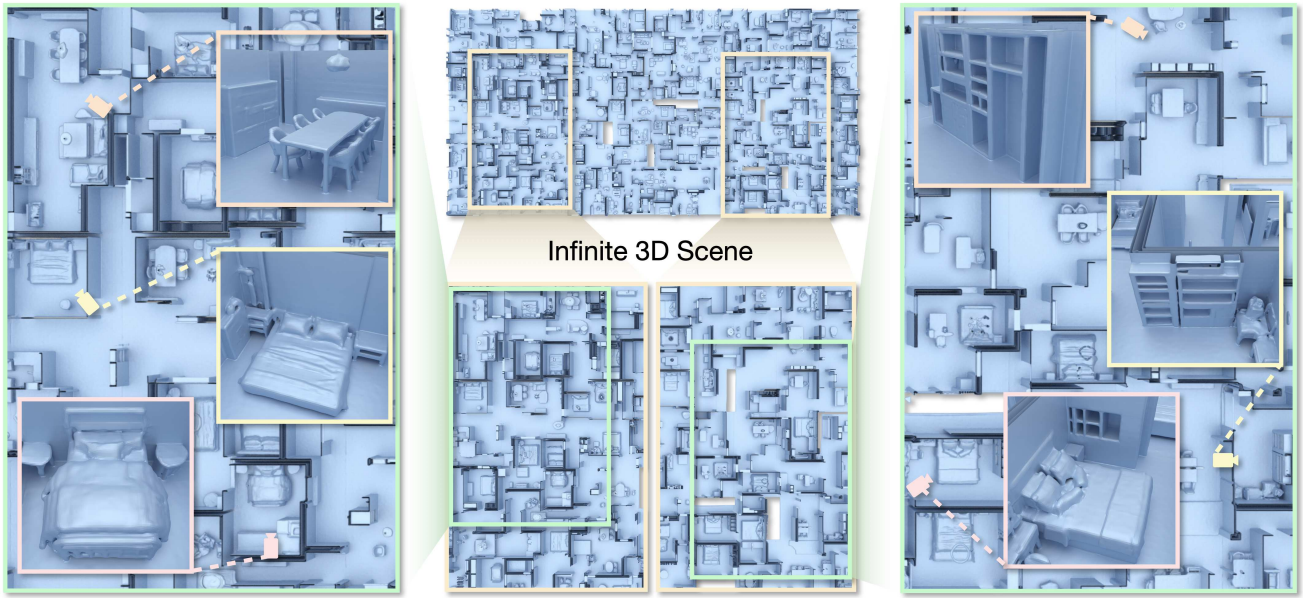


Figure 1. We introduce LT3SD, a novel latent 3D scene diffusion approach enabling high-fidelity generation of infinite 3D environments. We train LT3SD on a latent tree-based 3D scene representation, encoding both lower-frequency geometry and higher-frequency detail, and synthesize infinite scenes in a patch-by-patch and coarse-to-fine fashion.

Abstract

We present LT3SD, a novel latent diffusion model for large-scale 3D scene generation. Recent advances in diffusion models have shown impressive results in 3D object generation, but are limited in spatial extent and quality when extended to 3D scenes. To generate complex and diverse 3D scene structures, we introduce a latent tree representation to effectively encode both lower-frequency geometry and higher-frequency detail in a coarse-to-fine hierarchy. We can then learn a generative diffusion process in this latent 3D scene space, modeling the latent components of a scene at each resolution level. To synthesize large-scale scenes with varying sizes, we train our diffusion model on scene patches and synthesize arbitrary-sized output 3D scenes through shared diffusion generation across multiple scene patches. Through extensive experiments, we demonstrate the efficacy and benefits of LT3SD for large-scale, high-quality unconditional 3D scene generation and for probabilistic completion for partial scene observations.

1. Introduction

In recent years, there has been increasing demand for 3D digital content, both for content creation purposes such as films, video games, and mixed reality, as well as for reconstruction to enable machines to perceive real environments from visual inputs. For instance, video games have driven efforts to produce realistic virtual 3D environments, requiring significant costs for highly-trained 3D artists to create a vast array of 3D scene assets. In particular, AAA video game budgets have risen dramatically in the last five years, with games planned to be released this year and next having development budgets of $200 million or more [54].

Recent advances in diffusion models [23, 36, 43, 45, 47, 51] have achieved remarkable success in generating complex, high resolution, realistic images, and videos. In the 3D domain, however, generative modeling remains a significant challenge due to the data and representation complexity and limited training data availability. In addition, signal in 3D content is highly unevenly distributed, with many re-

gions being predominantly free space, whereas most scene detail is located only around certain surface areas. Existing 3D diffusion models [11, 16, 20, 33, 38, 41, 49, 60] overwhelmingly focus on object-level generation. Following latent diffusion (LDMs) [47] for 2D images, current state-of-the-art methods focus on training diffusion models on learned implicit representations of 3D shapes. However, these methods assume that shapes lie in a canonical and bounded space that is amenable to fixed and compact representation encodings, such as global latent codes [16], hypernetworks [20], and triplanes [22, 49]. While these representations yield impressive results for class-specific distributions, such as chairs or cars, they face challenges when applied to 3D scene environments characterized by highly unstructured geometries, diverse object arrangements, and varying spatial extents. Although early attempts at scene-level synthesis exist, they predominantly focus on single rooms or limited spatial extents [25, 28, 46] with relatively low-resolution [31, 32, 35] 3D scenes.

To this end, we propose LT3SD, a new probabilistic model for generating high-quality, large-scale, arbitrary-sized 3D scenes. Key to our approach is a novel latent tree representation (Fig. 2-right) that progressively decomposes a 3D scene into a tree hierarchy from fine to coarse resolutions. Our latent tree representation efficiently factorizes 3D scenes into geometry (lower-frequency) and latent feature (higher-frequency) volumes, enabling a more compact, effective representation for latent scene diffusion. We then synthesize 3D scenes in a coarse-to-fine, patch-by-patch fashion by reversing the latent tree decomposition.

We introduce a latent tree-based diffusion model to generate the latent feature volume conditioned on the corresponding geometry volume at the same tree level. This conditional learning of scene details helps to reduce the complexity of modeling 3D scene distributions. As 3D scenes can vary significantly in size, we design our diffusion model to learn at the level of scene patches. This strategy helps to shift the learning focus from complex unaligned 3D scenes to local structures with higher shared similarity.

As a result, our approach can generate large-scale, even infinite 3D scenes by coarse-to-fine construction of the latent trees in a patch-wise fashion. Starting from the coarsest level, our method operates patch-wise to generate the overall structure of a 3D scene, which is critical for geometric coherency. Then, in a coarse-to-fine manner, our method conditionally generates the latent feature volume, which represents fine details for each upper level—the final 3D scene results from the inverse process of latent tree decomposition. Experiments show that our method significantly outperforms existing baselines, improving FID scores by 70% on this challenging 3D scene generation task.

In summary, we present the following contributions:

- We introduce a novel latent 3D scene diffusion approach, leveraging a latent tree representation to compactly encode complex 3D scene geometry. This enables high-fidelity 3D scene generation in a coarse-to-fine fashion along the latent tree resolution levels.
- We characterize 3D scene generation in a patch-wise fashion, enabling efficient training and seamless infinite 3D scene generation through shared diffusion generation across multiple scene patches.

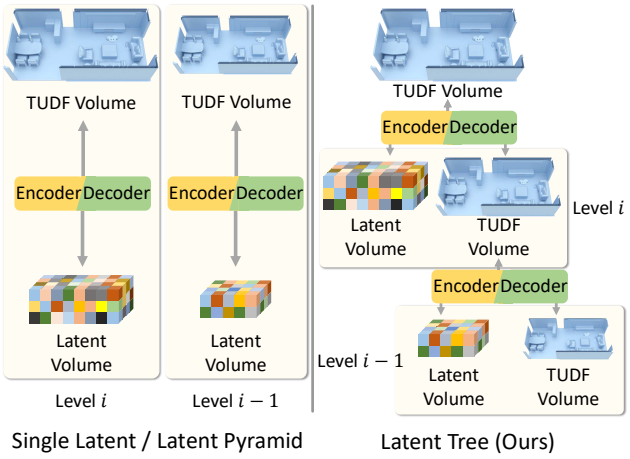


Figure 2. **Latent representations for 3D scenes.** In contrast to encoding a 3D scene into a single latent feature grid or a multi-level latent pyramid on the left, our latent tree representation on the right is a learned hierarchical decomposition with a series of geometry (lower-frequency) and latent feature (higher-frequency) encodings at each resolution level.

2. Related Work

Neural 3D Scene Representations. Various neural representations have been developed to characterize 3D shapes or scenes, with recent works extensively exploring implicit functions [12, 39, 40, 42], feature volumes [26, 34, 44], triplanes [8, 49], and tensor decompositions [10]. These representations are proposed as alternatives to traditional representations, such as voxels [17–19, 50, 56] or point clouds [2, 38, 59], to encode geometry more compactly and while maintaining high resolutions. Earlier approaches model an implicit function as an MLP, though such a global representation can result in difficulty retaining both local and global structures, with limited editability [39, 40, 42]. Tri-planes and tensor decompositions more effectively balance memory cost and performance, making them popular choices for object-level 3D shape generation [22, 49]; however, scene outpainting is not straightforward as the n-planes are highly correlated, requiring intricate synchronization processing for extrapolation [57]. In this work, we use a feature grid representation for capability to handle arbitrary topologies and its effective feature spatial locality to enable patch-based compositions of large-scale scenes.

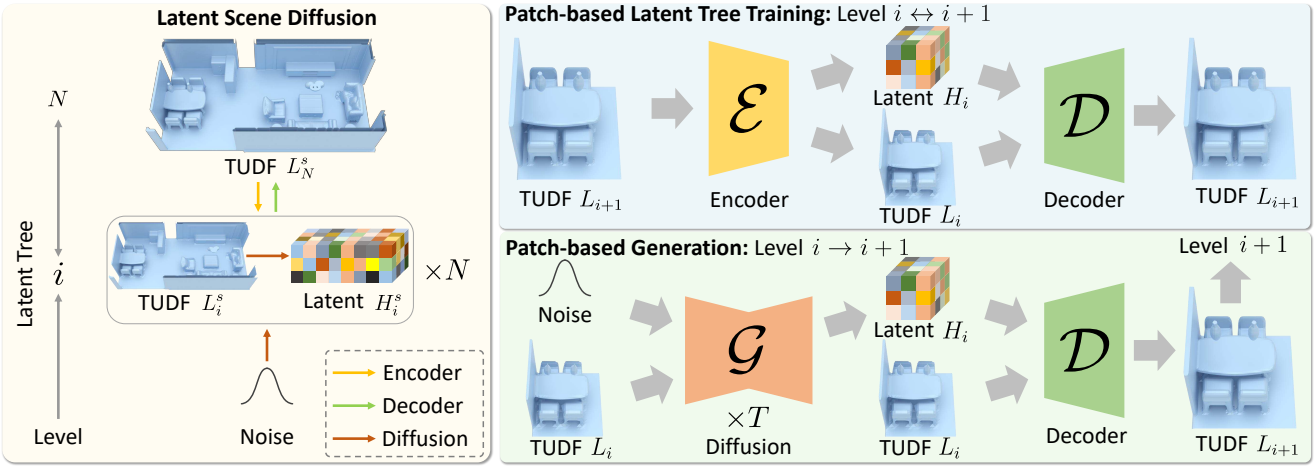


Figure 3. **Overview of LT3SD.** We formulate 3D scene generation as a patch-based latent diffusion process. **Left**: To characterize complex scene geometry, we encode 3D scenes in a novel latent tree representation, where each scene resolution level $i \in [1, N-1]$ is decomposed into a TUDF grid L_i^s and a latent feature grid H_i^s. **Top Right**: During latent tree training, the encoder $\mathcal{E}$ encodes a patch L_{i+1} from the scene grid L_{i+1}^s at resolution level $i+1$ to a coarser TUDF patch L_i and a latent feature patch H_i at level i. The decoder $\mathcal{D}$ then reconstructs the scene patch L_{i+1} based on the factorized grids L_i and H_i. **Bottom Right**: During generation, the diffusion model $\mathcal{G}$ learns to generate a latent feature patch H_i conditioned on a TUDF patch L_i within the same level i. Our method enables arbitrary-sized 3D scene generation at inference time by synthesizing scenes in a coarse-to-fine hierarchy and a patch-by-patch fashion.

3D Diffusion Models for Shape Generation. Diffusion models have shown unprecedented capability in generating 2D images and videos [6, 47]. Existing 3D diffusion work has focused on synthesizing 3D shapes [7, 11, 14–16, 20, 22, 30, 33, 49, 55, 58, 60], showing quite promising results. Following LDMs [47], most 3D diffusion models follow a two-stage approach: first learn a latent space of 3D shapes, and then train diffusion in the latent space. However, extending the same success to complex 3D scenes remains a challenge. Such 3D scene modeling faces higher data dimensionality, but with orders of magnitude less training data. Existing object generation methods assume that shapes lie in a canonical space and do not consider the arbitrary-sized nature of 3D scenes, making extending to 3D scenes difficult. In contrast, our approach enables efficient infinite 3D scene generation via training on latent tree scene patches.

3D Scene Generation. Inspired by the success of 2D image generation, various recent methods have proposed reconstructing 3D scenes using 2D generative models [9, 13]. These approaches leverage 2D image generative models to learn to generate consistent novel views; however, designing the camera trajectory and generating a complete 3D scene is challenging due to occlusions. Scene graphs-based 3D generation methods [53, 61] offer an abstract representation for 3D scene generation. Several recent works have introduced diffusion models for 3D scenes [3, 27, 28, 31, 32, 35, 46, 57, 58]. DiffRoom [27], NeuralField-LDM [28], CC3D [3], XCube [46] and TRELLIS [58] focus on bounded generation of 3D scenes, while SSG [31], [35], and SemCity [32] focus on outdoor 3D semantic scene generation, they offer limited geometry detail due to low-resolution data or explicit representation. Text-to-3D-based methods [25] typically require hours of optimization. Very recently, BlockFusion [57] proposed to generate larger-scale 3D scenes through latent triplane extrapolation, conditioned on 2D layout inputs. However, 3D scene generation is still in its infancy, and it remains a challenge to efficiently generate unbounded 3D scenes with high-quality, coherent structures, high-frequency details, and plausible layouts. In this work, we propose an efficient unconditional 3D scene generation pipeline that enables seamless infinite 3D scene generation at high quality.

3. Method

3.1. Overview

Our goal is to develop a probabilistic model for generating 3D scenes encompassing complex geometric structures, diverse object arrangements, and varying spatial extents. We propose LT3SD, a novel coarse-to-fine, patch-by-patch 3D scene generation approach by leveraging diffusion models in the space of a learned latent tree representation for 3D scenes. Our method has two main stages: latent tree encoding and patch-based latent diffusion, as illustrated in Fig. 3.

In the first stage (Sec. 3.2), we progressively decompose a 3D scene into a hierarchical latent tree from fine to coarse levels. Each tree level is a factorization of the scene into a geometry volume and a latent feature volume at the corresponding resolution level (Fig. 3-left). The geometry volume, represented as a truncated unsigned distance field (TUDF), captures lower-frequency scene information,

while the latent feature volume compactly encodes higher-frequency details. As shown in Fig. 3-top right, we learn this factorization with a 3D CNN-based encoder $\mathcal{E}$. The inverse process is built upon a decoder $\mathcal{D}$ that combines the two decomposed volumes to accurately recover the geometry volume at a higher resolution level.

In the second stage (Sec. 3.3), we train a 3D UNet-based diffusion network on the latent tree encodings. At each level, we randomly crop fixed-size patches from the scene geometry volumes and predict their corresponding latent feature patches (Fig. 3-bottom right). At inference time, we can effectively synthesize large-scale 3D scenes by progressively generating latent trees in a patch-by-patch fashion from coarse-to-fine levels.

3.2. Latent Tree Representation for 3D Scenes

To enable high-quality 3D scene generation, we propose to transform a 3D scene into a latent tree representation. Each level of the hierarchy comprises factorized geometry and latent feature encodings of the scene at the corresponding resolution. We represent the geometry encoding as truncated unsigned distance field (TUDF) grids, characterizing lower-frequency scene information, while higher-frequency scene details are encoded into a latent feature grid.

Our latent tree representation, as depicted in Fig. 2-right, offers the following advantages: (1) It allows for generating complex scene geometry in a coarse-to-fine fashion, in contrast to a single latent (Fig. 2-left), which does not well-represent shared local structures; (2) Compared to a multi-level latent pyramid (Fig. 2-left), our latent tree offers better encoding of 3D scenes with higher reconstruction quality.

Construction. To construct the latent tree representations, we first compute UDFs for 3D scenes. UDFs can easily encode arbitrary 3D scene geometry with various topologies. To focus on surface geometry regions and reduce data distribution complexity, we clip distances exceeding a specified threshold τ to obtain TUDF voxel grids.

Concretely, we decompose the TUDF grid of a 3D scene into a latent tree with N levels, where each scene resolution level $i \in [1, N-1]$ consists of a TUDF grid L_i^s and a latent feature grid H_i^s. We denote the root of the tree (*i.e.*, the highest resolution) as L_N^s. As illustrated in Fig. 3-top right, we learn this decomposition process with fixed-size patches L_{i+1} randomly cropped from the scene grid L_{i+1}^s for computational efficiency. All scene levels can be trained in parallel through patch-based training.

As shown in Fig. 3-top right, at resolution level $i+1$, we use the encoder $\mathcal{E}_{i+1}$ to factorize the TUDF patch $L_{i+1} \in \mathbb{R}^{1\times D_{i+1}\times H_{i+1}\times W_{i+1}}$ into a lower-resolution TUDF patch L_i and a latent feature patch H_i at the next coarser level i:

$$\mathcal{E}_{i+1}(L_{i+1}) \Rightarrow [L_i, H_i], \tag{1}$$

where $L_i \in \mathbb{R}^{1\times D_i\times H_i\times W_i}$ represents the coarser patch geometry, and $H_i \in \mathbb{R}^{C\times D_i\times H_i\times W_i}$ with a latent feature dimension C encodes higher-frequency details. In practice, we compute L_i by downsampling L_{i+1} with average pooling and predict H_i with a 3D CNN.

We then train a decoder network $\mathcal{D}_{i+1}$ to reconstruct the TUDF patch L_{i+1} by combining the factorized grids L_i and H_i as follows:

$$\mathcal{D}_{i+1}([L_i, H_i]) \Rightarrow L_{i+1}. \tag{2}$$

The decoder $\mathcal{D}_{i+1}$ is also implemented as a 3D CNN.

Learning. To train the encoders and decoders, our loss measures the ℓ_2 error between the reconstructed TUDFs and the ground truth TUDFs:

$$\mathcal{L}_{\text{latent}} = \Big(L_{i+1} - \mathcal{D}_{i+1}\big(\mathcal{E}_{i+1}\left(L_{i+1}\right)\big)\Big)^2. \tag{3}$$

Given the trained encoders, we can progressively decompose a high-resolution, arbitrary-sized 3D scene into a compact latent tree $\{L_1^s, H_1^s, \cdots, L_{N-1}^s, H_{N-1}^s\}$.

3.3. Patch-Based Latent Scene Diffusion

We learn a denoising diffusion probabilistic model $\mathcal{G}_i$ [23] for each resolution level i of the latent tree to generate high-fidelity 3D scenes. We leverage the explicit factorization in our latent tree (Sec. 3.2) and train $\mathcal{G}_i$ to generate the latent feature grid H_i^s, conditioned on geometry grid L_i^s. Since 3D scenes typically consist of local structures with shared similarities, our latent diffusion models are trained on scene patches randomly cropped from scene grids.

We denote the diffusion model as $\mathcal{G}_i(z_t, t, c)$, denoising a noisy latent patch z_t at each time step t based on condition c. At each training step, a patch of Gaussian noise $\epsilon \sim \mathcal{N}(0,1)$ of the same size and a time step $t \in [1, T]$ are randomly sampled and applied to a latent patch z to obtain a noisy latent patch z_t. At level $i > 1$, we train the diffusion model $\mathcal{G}_i$ such that given a coarse geometry patch L_i as condition, it predicts its corresponding latent feature patch H_i, *i.e.*, $z = H_i$ and $c = L_i$. At level $i = 1$, the coarsest level of the latent tree, the diffusion model generates both grids L_i and H_i unconditionally, *i.e.*, $z = [L_i, H_i]$ and $c = \varnothing$. The diffusion model $\mathcal{G}_i$ learns to denoise z_t using the following training loss:

$$\mathcal{L}_{\text{diff}} = \mathbb{E}_{z,c,\epsilon,t}\big[\,\|\epsilon - \mathcal{G}_i\left(z_t, t, c\right)\|_2^2\,\big]. \tag{4}$$

Our latent diffusion models $\{\mathcal{G}_i\}$ build upon 3D UNet backbones. The patch-wise training strategy also serves as data augmentation, helping our model avoid overfitting and learn shared local structures across scenes. Our conditional diffusion modeling can capture both lower-frequency structures as well as higher-frequency detail, due to our decomposition of 3D scenes into complementary geometry and latent feature grids at each resolution level. Our latent diffusion model can then synthesize arbitrary-sized 3D scene outputs by generating scenes in a patch-by-patch fashion.

3.4. Large-Scale Scene Generation

At inference time, the generation of large-scale 3D scenes is enabled by our hierarchical latent tree representation (Sec. 3.2) and the latent diffusion models trained on scene patches (Sec. 3.3). The generation process reverses the latent tree decomposition and constructs the hierarchy from coarse to fine levels. At each resolution level, the scene is synthesized in a patch-wise manner, allowing for arbitrary-sized 3D outputs.

Patch-by-Patch Scene Generation. To construct a latent tree from scratch, we first use the learned diffusion model $\mathcal{G}_1$ to create a coarse scene structure at the lowest resolution level $i = 1$. That is, starting from random Gaussian noise, we apply the diffusion model $\mathcal{G}_1$ to unconditionally generate both the geometry L_1^s and latent feature H_1^s grids of the scene. We find that using a simple inpainting-based patch-generation algorithm [28, 47], without requiring additional networks, works well to produce 3D scenes without seams.

Patches of L_1^s and H_1^s are synthesized autoregressively on the ground (*i.e.*, xy) plane. We adopt a breadth-first patch ordering scheme. From each known patch z_0, we first generate its adjacent patches in the x direction and then the y direction; this is then repeated in a "dilated" scene generation process. To ensure a smooth transition between patches, we require that each patch to be generated by the diffusion model $\mathcal{G}_1$ has partial overlap with existing patches [37]. We use the same partial overlap size for patches at all resolution levels. Specifically, we follow LDMs [47] and adopt the autoregressive Stable Inpainting scheme:

$$z_{t-1} = m \odot z_{t-1}^{\text{known}} + (1 - m) \odot z_{t-1}^{\text{unknown}}, \tag{5}$$

where z_{t-1}^{known} is sampled from the existing patch z_0, while z_{t-1}^{unknown} is sampled from the previous denoising iteration z_t. These two are then combined using the inpainting mask m to form the new sample z_{t-1}, keeping the known region unchanged.

Coarse-to-Fine Refinement. We then refine the generated coarse scene at higher resolution levels. At each level $i > 1$, the scene grid L_i^s is first reconstructed from the geometry grid L_{i-1}^s and the latent feature grid H_{i-1}^s at the lower-resolution level $i - 1$ using the trained decoder $\mathcal{D}_i$, as defined in Eq. (2). Next, the diffusion model $\mathcal{G}_i$ generates the latent feature grid H_i^s conditioned on the geometry grid L_i^s at patch-level with overlap.

However, generating patches sequentially like Stable Inpainting [47] at higher-resolution levels or infinite scenes with significantly more patches can be extremely time-consuming [57]. To speed up inference, we adapt the denoising fusion scheme from MultiDiffusion [4], which takes each denoising step on all patches simultaneously. First, we divide the generated coarse 3D scene into n patches with the same overlap size as previous levels. Then, the model $\mathcal{G}_i$ performs a denoising step in parallel across all patches from the noisy scene grid z_t^s with geometry patches from L_i^s as the condition. Finally, an aggregation step $\mathcal{A}$ blends the patches by averaging the predictions from overlapping regions, ensuring smooth transitions across patches and producing seamless 3D scenes. Concretely, each denoising step t at scene resolution level i is formulated as:

$$z_{t-1}^j = \mathcal{G}_i\Big(F_j(z_t^s), t, F_j(L_i^s)\Big), \tag{6}$$

$$z_{t-1}^s = \mathcal{A}\left(\left\{z_{t-1}^j\right\}_{j=1}^n\right), \tag{7}$$

where $\forall j \in [1, n]$, F_j is an operation that crops the j-th patches from the scene grids z_t^s and L_i^s, respectively. The latent feature volume H_i^s of the scene is obtained after completing all denoising time steps, *i.e.*, $H_i^s = z_0^s$.

The decoders $\mathcal{D}_i$ and diffusion models $\mathcal{G}_i$ are then applied alternately, as illustrated in Fig. 3-bottom right, until the scene geometry L_N^s at the highest level N is synthesized as the final output.

4. Experiments

Experimental Setup. We train LT3SD on house-level scene data from the 3D-FRONT dataset [21], a large-scale indoor scene dataset with diverse structures and layouts. Each house contains various room types (*e.g.*, bedrooms, living rooms, kitchens) with different layouts. After filtering out rooms with incorrect furniture scales or no furniture, we retained 6,479 houses. We used an $80\%/5\%/15\%$ train/validation/test split for both stages of LT3SD and the baselines. Due to open surfaces in scene meshes, consistent signed distance computation is challenging, so we compute UDFs with a voxel size of $2.2cm$ and a truncation value of $\tau = 10cm$. Instead of preprocessing scenes into fixed patches, we continuously sample chunks from 3D scenes during training to reduce overfitting, applying random flips and rotations for data augmentation.

Implementation Details. We implemented our method in Pytorch and ran all experiments on NVIDIA RTX A6000. To first construct latent trees for 3D scenes, we train the encoders and decoders with Adam optimizer [29], using a batch size of 4 and a learning rate of $1e^{-4}$, which takes from 5 hours to one day for different resolution levels until convergence. We use $N = 3$ levels in our latent tree. For all levels, the latent feature volumes have 4 channels. We use the Adam optimizer to train the diffusion models, with a batch size of 8 and a learning rate of $1e^{-4}$, which takes approximately 6 days until convergence on 2 GPUs. Note that during the second stage, we infer the latent of each TUDF patch on the fly without storing them on the disk in practice.

Baselines. We compare our method against 3D object diffusion models PVD [62] and NFD [49], as well as the recent

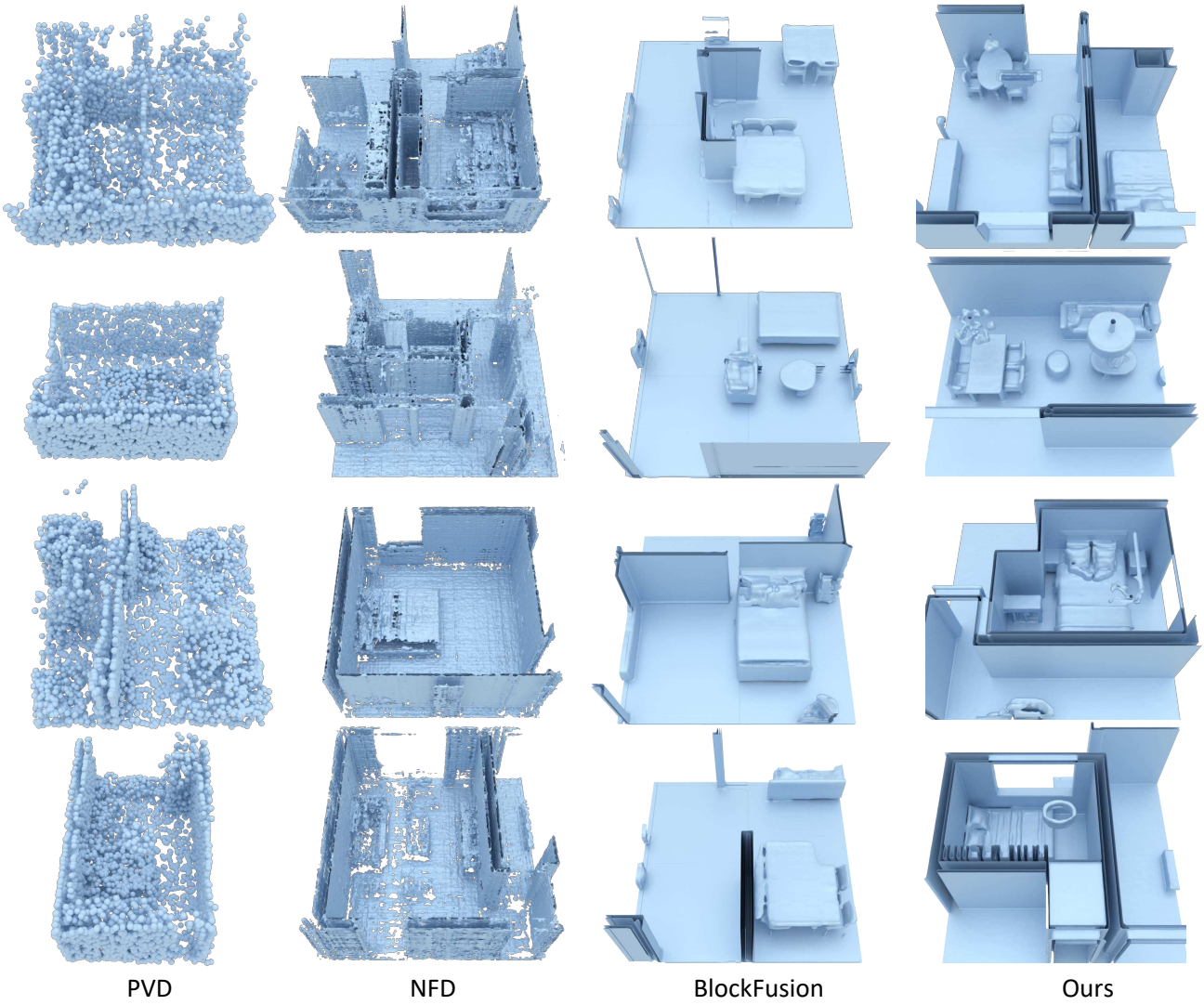


Figure 4. **Qualitative comparison.** We compare unconditional 3D scene generation with diverse 3D diffusion methods PVD [62], NFD [49], and BlockFusion [57]. All methods were trained on houses from the 3D-FRONT dataset [21]. Our latent tree-based 3D scene diffusion approach synthesizes cleaner surfaces with more geometric details and captures diverse furniture objects.

3D scene diffusion approach BlockFusion [57]. PVD is a point-voxel-based diffusion model, while NFD and BlockFusion are triplane-based diffusion models. All baselines were trained on 3D-FRONT. We uniformly sampled 8,192 points on the 3D scene surface for PVD. We used the same TUDF voxel grids for NFD as in our approach. The trained model of BlockFusion is provided by its authors, and we followed the authors' instructions for comparisons.

Evaluation Metrics. We follow NFD [48] to evaluate generation ability and use an adapted Fréchet Inception Distance (FID) metric based on rendered images of generated meshes to measure perceptual distribution similarity. Following prior work [20, 46, 62], we also assess coverage and diversity using the Coverage (COV), Minimum Matching Distance (MMD), and 1-Nearest Neighbor Accuracy (1-NNA) metrics. COV measures the fraction of test set shapes matched to at least one in the generated set, indicating diversity representation. MMD reflects the average distance from each test shape to its nearest neighbor in the generated set, measuring quality and fidelity. The 1-NNA metric evaluates distribution similarity, with a value closer to 50% indicating statistical indistinguishability between the generated and test distributions.

For FID, we render images of generated and ground-truth 3D meshes by placing cameras at the scene center and rotating them horizontally to capture 20 images per scene. To compute COV, MMD, and 1-NNA, we measure point cloud similarity using Chamfer Distance (CD) and Earth Mover's Distance (EMD). Points are uniformly sampled from the generated meshes of NFD, BlockFusion, our method, and GT, while the same number of points are randomly sampled from PVD's generated point clouds.

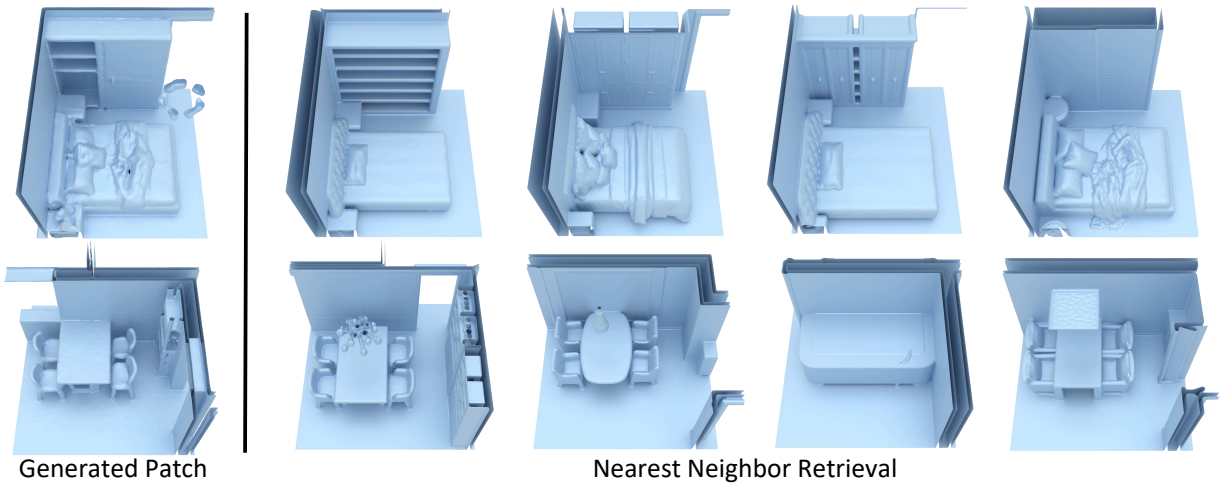


Figure 5. **Generation novelty analysis.** Our generated scene patches (left), compared with their nearest-neighbor retrieved training patches by Chamfer distance. Our approach can synthesize novel patches with different geometric structures than their training set neighbors.

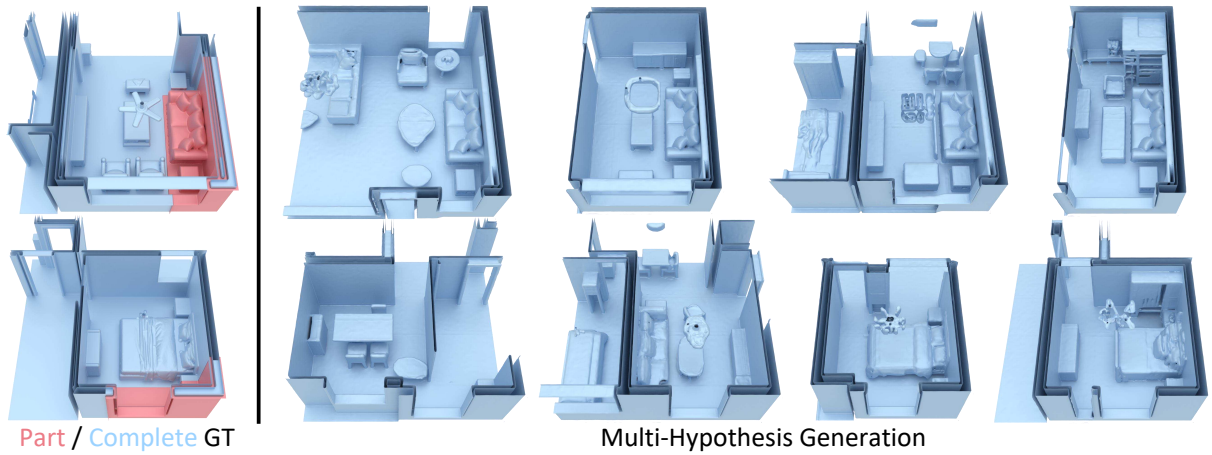


Figure 6. **Probabilistic Scene Completion.** Given an initial part of a complete ground-truth 3D scene from the test split, LT3SD can synthesize various plausible hypotheses of a complete scene by sampling various diffusion outputs.

4.1. 3D Scene Generation

We evaluate our method on 3D scene generation, both quantitatively and qualitatively in comparison with state-of-the-art 3D diffusion alternatives. We generate 3D scenes of a fixed size for each method for evaluation.

We show qualitative comparisons in Fig. 4. Our results demonstrate superior surface quality and object detail: PVD generates plausible global structures but produces incomplete geometric information in point clouds; NFD [48] generates plausible scene structures and scales well for complex 3D scenes, but struggles to produce complex arrangements of furniture and smooth surfaces; BlockFusion [57] achieves smooth local surfaces and structural diversity, but struggles in generating meaningful higher-level structures (e.g., Fig. 4-1st row, where walls are misaligned despite improved surface quality). In contrast, our approach generates scenes with both plausible room structures and layouts while maintaining the highest quality in finer-scale details.

Table 1. **Quantitative comparison.** We compare unconditional 3D scene generation with state-of-the-art 3D diffusion methods PVD [62], NFD [49], and BlockFusion [57] trained on 3D-FRONT data [21]. Our method significantly outperforms baselines in all metrics.

Method	COV (%) ↑		MMD (×1e2)↓		1-NNA (%) ↓		FID ↓
	CD	EMD	CD	EMD	CD	EMD	
PVD [62]	43.82	32.52	3.69	29.36	70.83	82.24	237.85
NFD [49]	44.65	45.48	3.65	27.98	62.86	63.62	266.27
BlockFusion [57]	24.32	22.85	5.10	31.82	89.01	90.85	45.55
Ours (17.6 - 2.2) cm	28.61	29.98	4.93	30.17	74.51	69.39	59.23
Ours (8.8 - 2.2) cm	40.87	44.21	4.90	31.50	84.34	81.57	50.07
Ours (17.6 - 8.8 - 2.2) cm	**53.10**	**54.66**	**3.51**	**27.30**	**53.22**	**52.56**	**13.39**

Tab. 1 shows a quantitative comparison of our LT3SD against baselines. Our approach significantly improves both point-based and perceptual-based metrics, demonstrating its effectiveness in 3D scene generation.

Since the point-cloud-based PVD [62] generates only sparse point clouds, we convert PVD points to triangle meshes with ball-pivoting [5] for FID computation. The

Table 2. **Ablation over latent representation**. Our latent tree-based representation achieves higher reconstruction accuracy while using less training time and latent storage by factorizing 3D scenes into complementary components.

Representation	# Params	Training	Storage	Recon. Error (ℓ_2)
Latent CM	13.3 M	$\times$ 1.00	$\times$ 1.00	4.91e-4
Latent Tree (Ours)	13.3 M	$\times$ 0.87	$\times$ 0.80	3.20e-4

triplane-based NFD [48] produces good scene structures, though lacks perceptual quality in local surface smoothness. As shown in Fig. 4, BlockFusion [57] produces recognizable furniture and smooth surfaces, but with quite fragmented room structures (*e.g.*, in Fig. 4, the walls in the 1st and 3rd rows are fragmented and unlikely in structure), resulting in a good FID score but a relatively lower point cloud metrics that rely on more strongly reflecting global structures in the ground-truth distribution.

In contrast, our method follows a hierarchical approach on our latent tree encoding, starting with a good coarse scene structure (reflected in point-cloud metrics) and progressively adding high-fidelity local detail, as indicated by the FID score. Additionally, our method is also much more efficient while achieving superior quality: the large-scale scene generation in Fig. 1 of size $45.1m \times 90.3m \times 2.8m$ containing $\approx$ 170 rooms takes 2 hours on a single GPU, while BlockFusion [57] takes 3 hours for the generation of 7 rooms.

4.2. Ablation Study

Latent Tree Structure. We validate the latent tree structure of our method in Tab. 1 for 3D scene generation. In this ablation, we compare our full model with latent trees containing only one level. As shown in Tab. 1-bottom, the multi-level latent tree-based diffusion model outperforms the single-level ones. Rather than using a single level of the tree and generating a 3D scene in a single step, our LT3SD adopts a coarse-to-fine strategy for encoding; the diffusion model only generates the higher-frequency component of 3D scenes in each step and, finally, both lower-frequency and higher-frequency components are gracefully combined together and decoded to a high-quality 3D scene. This ablation demonstrates that the hierarchical tree structure is essential for effective generative modeling with diffusion for complex 3D scenes.

Hierarchical Latent Representation. We compare our latent tree with latent Cascaded Models (CMs) (Fig. 2-left), commonly used in 2D image synthesis [24] and 3D generation [35, 46]. Unlike CMs, which independently model redundant information at each level, our latent tree decomposes 3D scenes into coarse geometry and fine latent components, capturing residual high-frequency details. As shown in Tab. 2, with the same autoencoder and parameter count, our model trains more efficiently (requiring less time per level) and achieves higher reconstruction accuracy (lower l_2 error) on test data. Additionally, its more compact latent representation (fewer latent channels) reduces storage requirements when converting TUDF grids to latent grids. Training time and storage statistics are reported as ratios w.r.t. latent CM; please see the supplemental for more details.

4.3. Novelty of Generated Scene Patches

An important consideration in unconditional generation is the ability of trained models to produce novel outputs, including new shapes and scene layouts that differ from the training data. We retrieve the nearest training patch to each generated scene patch based on Chamfer Distance to evaluate this. We perform retrieval from the entire training set, applying all permutations of the training augmentations (flip and rotation). In Fig. 5, we show scene patches generated by our LT3SD, along with the top-4 nearest neighbor training patches. We observe that our generated patches produce new structures different from the training patches. Our method generates strong differences in local object characteristics (top row) as well as differences in layout arrangements (bottom row). We attribute this partly to the random, continuous patch-wise training strategy employed on the fly during the two training stages and the augmentations.

4.4. Probabilistic Scene Completion

We further demonstrate the generative capability of LT3SD for scene completion from partial observations. Fig. 6 shows that from small portions of a given scene, our diffusion-based approach enables sampling a diverse set of possible completed scenes that explain the partial geometry input. Our approach can preserve the initial scene geometry while generating completed scenes of various sizes. Although LT3SD is trained without semantic annotations, our completion results respect the input object semantic structures (*e.g.*, given a sofa, generating various living rooms).

5. Conclusion

We have presented LT3SD, a novel approach for high-quality 3D scene generation through latent scene diffusion. We propose decomposing high-resolution 3D scene voxel grids to compact latent trees for efficient diffusion training in this latent space. This strategy enables seamless infinite 3D scene generation in a patch-wise manner. Experimental validation demonstrates the capacity of LT3SD for large-scale 3D scene generation, significantly outperforming state-of-the-art alternatives for unconditional 3D scene generation. We hope our method provides an avenue for future exploration towards the challenging task of 3D scene generation and automated 3D content creation.

Acknowledgements. This work was supported by the ERC Starting Grant SpatialSem (101076253). Matthias Nießner was supported by the ERC Starting Grant Scan2CAD (804724).

References

[1] Full gameready city buildings. https://skfb.ly/p8Qzx, 2024. Accessed: 19-Feb-2025. 13

[2] Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas Guibas. Learning representations and generative models for 3d point clouds. In *International conference on machine learning*, pages 40–49. PMLR, 2018. 2

[3] Sherwin Bahmani, Jeong Joon Park, Despoina Paschalidou, Xingguang Yan, Gordon Wetzstein, Leonidas Guibas, and Andrea Tagliasacchi. Cc3d: Layout-conditioned generation of compositional 3d scenes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7171–7181, 2023. 3

[4] Omer Bar-Tal, Lior Yariv, Yaron Lipman, and Tali Dekel. Multidiffusion: Fusing diffusion paths for controlled image generation. *arXiv preprint arXiv:2302.08113*, 2023. 5

[5] F. Bernardini, J. Mittleman, H. Rushmeier, C. Silva, and G. Taubin. The ball-pivoting algorithm for surface reconstruction. *IEEE Transactions on Visualization and Computer Graphics*, 5(4):349–359, 1999. 7

[6] Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. Align your latents: High-resolution video synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22563–22575, 2023. 3

[7] Ruojin Cai, Guandao Yang, Hadar Averbuch-Elor, Zekun Hao, Serge Belongie, Noah Snavely, and Bharath Hariharan. Learning gradient fields for shape generation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 3

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

[9] Eric R. Chan, Koki Nagano, Matthew A. Chan, Alexander W. Bergman, Jeong Joon Park, Axel Levy, Miika Aittala, Shalini De Mello, Tero Karras, and Gordon Wetzstein. GeNVS: Generative novel view synthesis with 3D-aware diffusion models. In *arXiv*, 2023. 3

[10] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. In *European Conference on Computer Vision (ECCV)*, 2022. 2

[11] Hansheng Chen, Jiatao Gu, Anpei Chen, Wei Tian, Zhuowen Tu, Lingjie Liu, and Hao Su. Single-stage diffusion nerf: A unified approach to 3d generation and reconstruction. In *ICCV*, 2023. 2, 3

[12] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[13] Z. Chen, G. Wang, and Z. Liu. Scenedreamer: Unbounded 3d scene generation from 2d image collections. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45 (12):15562–15576, 2023. 3

[14] Zhaoxi Chen, Jiaxiang Tang, Yuhao Dong, Ziang Cao, Fangzhou Hong, Yushi Lan, Tengfei Wang, Haozhe Xie, Tong Wu, Shunsuke Saito, Liang Pan, Dahua Lin, and Ziwei Liu. 3dtopia-xl: High-quality 3d pbr asset generation via primitive diffusion. *arXiv preprint arXiv:2409.12957*, 2024. 3

[15] Yen-Chi Cheng, Hsin-Ying Lee, Sergey Tulyakov, Alexander G Schwing, and Liang-Yan Gui. Sdfusion: Multimodal 3d shape completion, reconstruction, and generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4456–4465, 2023.

[16] Gene Chou, Yuval Bahat, and Felix Heide. Diffusion-sdf: Conditional generative modeling of signed distance functions. In *The IEEE International Conference on Computer Vision (ICCV)*, 2023. 2, 3

[17] Angela Dai, Charles Ruizhongtai Qi, and Matthias Nießner. Shape completion using 3d-encoder-predictor cnns and shape synthesis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5868–5877, 2017. 2

[18] Angela Dai, Christian Diller, and Matthias Nießner. Sg-nn: Sparse generative neural networks for self-supervised scene completion of rgb-d scans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 849–858, 2020.

[19] Angela Dai, Yawar Siddiqui, Justus Thies, Julien Valentin, and Matthias Nießner. Spsg: Self-supervised photometric scene generation from rgb-d scans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1747–1756, 2021. 2

[20] Ziya Erkoç, Fangchang Ma, Qi Shan, Matthias Nießner, and Angela Dai. Hyperdiffusion: Generating implicit neural fields with weight-space diffusion, 2023. 2, 3, 6

[21] Huan Fu, Bowen Cai, Lin Gao, Ling-Xiao Zhang, Jiaming Wang, Cao Li, Qixun Zeng, Chengyue Sun, Rongfei Jia, Binqiang Zhao, et al. 3d-front: 3d furnished rooms with layouts and semantics. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10933–10942, 2021. 5, 6, 7, 13

[22] Anchit Gupta, Wenhan Xiong, Yixin Nie, Ian Jones, and Barlas Oğuz. 3dgen: Triplane latent diffusion for textured mesh generation. *arXiv preprint arXiv:2303.05371*, 2023. 2, 3

[23] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 1, 4

[24] Jonathan Ho, Chitwan Saharia, William Chan, David J Fleet, Mohammad Norouzi, and Tim Salimans. Cascaded diffusion models for high fidelity image generation. *arXiv preprint arXiv:2106.15282*, 2021. 8

[25] Lukas Höllein, Ang Cao, Andrew Owens, Justin Johnson, and Matthias Nießner. Text2room: Extracting textured 3d meshes from 2d text-to-image models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7909–7920, 2023. 2, 3

[26] Chiyu Jiang, Avneesh Sud, Ameesh Makadia, Jingwei Huang, Matthias Nießner, Thomas Funkhouser, et al. Local implicit grid representations for 3d scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6001–6010, 2020. 2

[27] Xiaoliang Ju, Zhaoyang Huang, Yijin Li, Guofeng Zhang, Yu Qiao, and Hongsheng Li. Diffroom: Diffusion-based high-quality 3d room reconstruction and generation. *arXiv preprint arXiv:2306.00519*, 2023. 3

[28] Seung Wook Kim, Bradley Brown, Kangxue Yin, Karsten Kreis, Katja Schwarz, Daiqing Li, Robin Rombach, Antonio Torralba, and Sanja Fidler. Neuralfield-ldm: Scene generation with hierarchical latent diffusion models. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 2, 3, 5

[29] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[30] Yushi Lan, Fangzhou Hong, Shuai Yang, Shangchen Zhou, Xuyi Meng, Bo Dai, Xingang Pan, and Chen Change Loy. Ln3diff: Scalable latent neural fields diffusion for speedy 3d generation. In *ECCV*, 2024. 3

[31] Jumin Lee, Woobin Im, Sebin Lee, and Sung-Eui Yoon. Diffusion probabilistic models for scene-scale 3d categorical data. *arXiv preprint arXiv:2301.00527*, 2023. 2, 3

[32] Jumin Lee, Sebin Lee, Changho Jo, Woobin Im, Juhyeong Seon, and Sung-Eui Yoon. Semcity: Semantic scene generation with triplane diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 28337–28347, 2024. 2, 3

[33] Muheng Li, Yueqi Duan, Jie Zhou, and Jiwen Lu. Diffusion-sdf: Text-to-shape via voxelized diffusion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 2, 3

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

[35] Yuheng Liu, Xinke Li, Xueting Li, Lu Qi, Chongshou Li, and Ming-Hsuan Yang. Pyramid diffusion for fine 3d large scene generation. *arXiv preprint arXiv:2311.12085*, 2023. 2, 3, 8, 15

[36] Yixin Liu, Kai Zhang, Yuan Li, Zhiling Yan, Chujie Gao, Ruoxi Chen, Zhengqing Yuan, Yue Huang, Hanchi Sun, Jianfeng Gao, Lifang He, and Lichao Sun. Sora: A review on background, technology, limitations, and opportunities of large vision models, 2024. 1

[37] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11461–11471, 2022. 5

[38] Shitong Luo and Wei Hu. Diffusion probabilistic models for 3d point cloud generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2

[39] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[40] 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

[41] Gimin Nam, Mariem Khlifi, Andrew Rodriguez, Alberto Tono, Linqi Zhou, and Paul Guerrero. 3d-ldm: Neural implicit 3d shape generation with latent diffusion models. *arXiv preprint arXiv:2212.00842*, 2022. 2

[42] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 165–174, 2019. 2

[43] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4195–4205, 2023. 1

[44] Songyou Peng, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, and Andreas Geiger. Convolutional occupancy networks. In *European Conference on Computer Vision (ECCV)*, 2020. 2

[45] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, pages 8821–8831. PMLR, 2021. 1

[46] Xuanchi Ren, Jiahui Huang, Xiaohui Zeng, Ken Museth, Sanja Fidler, and Francis Williams. Xcube: Large-scale 3d generative modeling using sparse voxel hierarchies. *arXiv preprint*, 2023. 2, 3, 6, 8

[47] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2021. 1, 2, 3, 5

[48] Maximilian Seitzer. pytorch-fid: FID Score for PyTorch. https://github.com/mseitzer/pytorch-fid, 2020. Version 0.3.0. 6, 7, 8

[49] J Ryan Shue, Eric Ryan Chan, Ryan Po, Zachary Ankner, Jiajun Wu, and Gordon Wetzstein. 3d neural field generation using triplane diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20875–20886, 2023. 2, 3, 5, 6, 7, 13

[50] Yawar Siddiqui, Justus Thies, Fangchang Ma, Qi Shan, Matthias Nießner, and Angela Dai. Retrievalfuse: Neural 3d scene reconstruction with a database. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12568–12577, 2021. 2

[51] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. PMLR, 2015. 1

[52] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv:2010.02502*, 2020. 12

[53] Jiapeng Tang, Yinyu Nie, Lev Markhasin, Angela Dai, Justus Thies, and Matthias Nießner. Diffuscene: Denoising diffusion models for gerative indoor scene synthesis. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, 2024. 3, 13

[54] UKCMA. Anticipated acquisition by microsoft of activision blizzard, inc., 2022. https://www.gov.uk/cma-cases/microsoft-slash-activision-blizzard-merger-inquiry. 1

[55] Tengfei Wang, Bo Zhang, Ting Zhang, Shuyang Gu, Jianmin Bao, Tadas Baltrusaitis, Jingjing Shen, Dong Chen, Fang Wen, Qifeng Chen, et al. Rodin: A generative model for sculpting 3d digital avatars using diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4563–4573, 2023. 3

[56] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1912–1920, 2015. 2

[57] Zhennan Wu, Yang Li, Han Yan, Taizhang Shang, Weixuan Sun, Senbo Wang, Ruikai Cui, Weizhe Liu, Hiroyuki Sato, Hongdong Li, et al. Blockfusion: Expandable 3d scene generation using latent tri-plane extrapolation. *arXiv preprint arXiv:2401.17053*, 2024. 2, 3, 5, 6, 7, 8, 13, 15

[58] Jianfeng Xiang, Zelong Lv, Sicheng Xu, Yu Deng, Ruicheng Wang, Bowen Zhang, Dong Chen, Xin Tong, and Jiaolong Yang. Structured 3d latents for scalable and versatile 3d generation. *arXiv preprint arXiv:2412.01506*, 2024. 3

[59] Guandao Yang, Xun Huang, Zekun Hao, Ming-Yu Liu, Serge Belongie, and Bharath Hariharan. Pointflow: 3d point cloud generation with continuous normalizing flows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4541–4550, 2019. 2

[60] 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 *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 2, 3

[61] Guangyao Zhai, Evin Pinar Örnek, Shun-Cheng Wu, Yan Di, Federico Tombari, Nassir Navab, and Benjamin Busam. Commonscenes: Generating commonsense 3d indoor scenes with scene graphs. *arXiv preprint arXiv:2305.16283*, 2023. 3

[62] Linqi Zhou, Yilun Du, and Jiajun Wu. 3d shape generation and completion through point-voxel diffusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5826–5835, 2021. 5, 6, 7, 13