DiffRoom: Diffusion-based High-Quality 3D Room Reconstruction and Generation with Occupancy Prior

Xiaoliang Ju [1,2] * **Zhaoyang Huang**[1]* **Yijin Li**[3]
Guofeng Zhang[3] **Yu Qiao**[2] **Hongsheng Li**[1]
[1] MMLab, The Chinese University of Hong Kong
[2] Shanghai AI Laboratory [3] Zhejiang University
`{akira,drinkingcoder}@link.cuhk.edu.hk, hsli@ee.cuhk.edu.hk`

(a) The front view. (b) The back view.

Figure 1: Generated scene from noisy occupancy of "Appartment-0" in Replica.

Abstract

We present DiffRoom, a novel framework for tackling the problem of high-quality 3D indoor room reconstruction and generation, both of which are challenging due to the complexity and diversity of the room geometry. Although diffusion-based generative models have previously demonstrated impressive performance in image generation and object-level 3D generation, they have not yet been applied to room-level 3D generation due to their computationally intensive costs. In DiffRoom, we propose a sparse 3D diffusion network that is efficient and possesses strong generative performance for Truncated Signed Distance Field (TSDF), based on a rough occupancy prior. Inspired by KinectFusion's incremental alignment and fusion of local SDFs, we propose a diffusion-based TSDF fusion approach that iteratively diffuses and fuses TSDFs, facilitating the reconstruction and generation of an entire room environment. Additionally, to ease training, we introduce a curriculum diffusion learning paradigm that speeds up the training convergence process and enables high-quality reconstruction. According to the user study, the mesh quality generated by our DiffRoom can even outperform the ground truth mesh provided by ScanNet. Please visit our project page for the latest progress and demonstrations: https://akirahero.github.io/DiffRoom/.

*Both authors contributed equally to this work.

Preprint. Under review.

Figure 2: Our DiffRoom enables 3D high-quality room reconstruction.

1 Introduction

3D scene reconstruction from posed images is a fundamental problem in 3D computer vision with many applications, such as Augmented Reality (AR). The quality of scene reconstruction determines the realistic and immersive effects of AR. While recent works [35, 1] are able to reconstruct coherent scene meshes, represented by Signed Distance Fields (SDF), the quality of the resultant meshes is far from satisfactory. Major mesh details might be lost during iterative fusion. Recently, diffusion models [9, 33] have shown their great ability in generating images and objects of high quality. Here we want to ask a question: *Can we reconstruct high-quality indoor rooms with diffusion models*? In this paper, we propose a novel framework DiffRoom that not only reconstructs but also generates high-quality indoor room geometry with diffusion models (Fig. 2).

Diffusion models are a class of generative models designed for synthesizing data by iterative denoising. The classical Denoising Diffusion Probabilistic Model (DDPM) training paradigm starts the denoising from pure Gaussian noise. Training diffusion models for room-level SDFs are challenging because the size of SDF is quite large. Previous 3D diffusion models only focus on object-level 3D generations. As reported by InstantNGP [20], only 2.57% voxels are valuable in common 3D scenes, we propose a sparse 3D diffusion neural network, SparseDiff, for SDF diffusion. Instead of denoising a dense SDF, SparseDiff only denoises sparse SDFs on occupied voxels with sparse convolutions and attentions, which saves two orders of computational and memory costs.

We propose a curriculum diffusion model learning paradigm for room-level SDF generation. Specifically, we take NeuralRecon [35] to provide a rough SDF and train our diffusion models in two steps. Our SparseDiff accepts a two-channel Sparse SDF including a conditional SDF channel that is fixed during the denoising process and a noise SDF channel that is iteratively denoised. Firstly, we set the SDF predicted by NeuralRecon as the conditional SDF to guide the noise SDF generation. Secondly, we set a schedule that replaces some parts of the conditional SDF with Gaussian noise. Such a conditional noise schedule boosts the convergence of SparseDiff. Moreover, it also enables high-quality scene reconstruction by conditioning the diffusion model on a NeuralRecon-predicted SDF. During inference, the conditioning SDF channel can be used in a plug-and-play manner. We can either adopt NeuralRecon results as the conditional SDFs to reconstruct high-quality room geometries or condition on Gaussian noise to generate room geometries.

Although sparsification of the 3D space significantly reduces the required computational resources, the size of different rooms varies quite significantly, which makes directly training a room-level SDF challenging. There are 3D reconstruction methods [14, 22] incrementally fuse local TSDFs to achieve the whole room reconstruction. Inspired by them, we randomly crop local TSDFs of smaller sizes from the large room-level SDF for training. During inference, we design a stochastic TSDF fusion algorithm that generates the entire room by iterative denoising and fusing local TSDFs.

Our contributions can be summarized as fourfold: 1) We propose a novel framework DiffRoom for room-level SDF diffusion with SparseDiff that saves two orders of resource consumption. 2) We propose a curriculum learning for SparseDiff, which boosts convergence speed and also achieves high-quality reconstruction. 3) We propose a novel algorithm that fuses diffusion-based local TSDFs, which enables large-scale room generation. 4) DiffRoom for the first time achieves high-quality room-level reconstruction and generation.

2 Related Works

3D Scene Reconstruction. 3D scene reconstruction from multiple images is a classic problem in computer vision that has been extensively studied over the years. Traditionally, methods have relied on a depth map fusion approach [14, 22, 36]. In this process, each keyframe's depth map is first estimated with multi-view depth estimation methods [31, 38, 39, 37]. The estimated depth maps are later filtered and fused into a Truncated Signed Distance Function (TSDF) volume [14], from which the reconstructed mesh can be extracted with the Marching Cubes algorithm [16]. Recently, some methods try to directly regress the TSDF volumetric data end-to-end [35, 1, 21], through which, the network is able to learn the local smoothness and global shape prior of natural 3D surfaces. Nevertheless, the reconstruction results are usually over-smoothed and lack of details. Inspired by the recent success of diffusion models which can synthesize high-resolution images with rich details, in this paper, we propose a novel diffusion-based framework that enables high-quality room-level reconstruction and generation.

3D Scene Synthesis. In recent decades, the field of 3D scene synthesis has experienced extensive investigation, particularly driven by the proliferation of 3D indoor scene datasets [3, 5] and advancements in 3D deep learning [25, 26, 15]. However, current methods mainly focus on synthesizing plausible 3D scene arrangements [4, 27, 6, 7]. They usually learn to synthesize the scene graph as the intermediate scene representation and retrieve objects from available dataset. Needless to say, the capacity of current digital assets limits these methods' generating ability.

Diffusion Models. The diffusion model [32, 9, 33] has emerged as a promising class of generative models for learning data distributions through an iterative denoising process They have shown impressive visual quality in diverse applications of 2D image synthesis, encompassing image inpainting [17], super-resolution [29, 11], editing [19], text-to-image synthesis [23, 28], and video generation [12, 8]. Nevertheless, the application of diffusion models in the 3D domain has received limited attention in comparison to the extensive exploration seen in the 2D domain. In the 3D domain, existing research has focused on the generation of individual objects [18, 40, 24, 13], while less attention has been paid to the synthesis of entire scenes, which possess significantly higher levels of semantic and geometric complexity, as well as the expansive spatial extent present in 3D scene synthesis.

3 Method

Our goal is to generate a 3D Truncated Signed Distance Field (TSDF) $x \in \mathbb{R}^{H \times W \times D}$ with diffusion models. Directly learning to generate room-level TSDF is quite expensive. Given a set of sparse occupied voxels $\mathcal{P} = \{\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_N\}$ containing N coordinates $\mathbf{p}_i \in \mathbb{R}^3$, we choose to generate TSDF on occupied voxels $x(\mathbf{p}_i)$ only and regard the TSDF on empty voxels as 1. We introduce DiffRoom a novel framework for diffusion-based high-quality room reconstruction and generation. DiffRoom mainly consists of three modules: 1) a sparse diffusion network architecture SparseDiff that efficiently denoises 3D TSDFs, 2) a curriculum diffusion learning (CDL) that boosts training convergence speed and achieves high-quality reconstruction and generation, and 3) a stochastic fusion algorithm for diffusion-based local TSDFs, which enables large-scale room reconstruction and generation.

3.1 SparseDiff

We build a novel network architecture SparseDiff for sparse voxel-based TSDF diffusion. As shown in Fig. 3, SparseDiff is a U-Net structure consisting of sparse convolution blocks with attentions. SparseDiff accepts two TSDF volumes, one is the conditional rough TSDF (explained later) and the other one is the noisy TSDF volume to be restored, and predicts the cumulative noise ϵ_θ.

3.2 Curriculum Diffusion Learning

Diffusion models are a class of generative models that produce samples in training distribution from a white Gaussian distribution in an iterative denoising manner. We apply diffusion models for TSDF

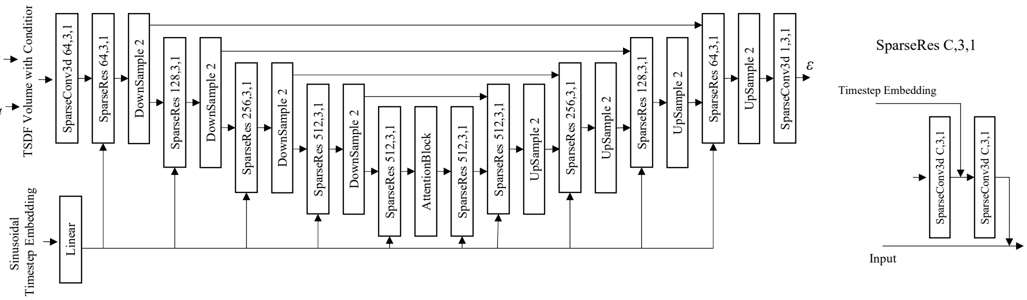


Figure 3: Network Structure of SparseDiff.

generation. During training, we follow DDPM [32, 9] that transforms a sample x_0, a crop of ground truth TSDF, to a white Gaussian noise $x_T \sim \mathcal{N}(0,1)$ in T steps. In each step t, the sample x_t is obtained by adding i.i.d. Gaussian noise with variance β_t and scaling the sample in the previous step x_{t-1} with $\sqrt{1-\beta_t}$:

$$q\left(x_t \mid x_{t-1}\right)=\mathcal{N}\left(x_t ; \sqrt{1-\beta_t} x_{t-1}, \beta_t \mathbf{I}\right), \tag{1}$$

which is also called the forward direction. Diffusion models are trained to reverse the forward process, i.e., predict the parameters $\mu_\theta\left(x_t, t\right)$ and $\Sigma_\theta\left(x_t, t\right)$ of the Gaussian distribution. However, TSDFs of rooms vary quite large, which is challenging for diffusion. We propose a curriculum diffusion learning (CDL) paradigm for room-level TSDF generation. Our CDL adopts NeuralRecon as an off-the-shelf tool to provide rough TSDF initialization and learns to diffuse in two steps. In the first step, we train SparseDiff with condition signals y generated by NeuralRecon. SparseDiff is able to refine TSDFs for achieving high-quality reconstruction. In the second step, we replaces parts of the TSDFs provided by NeuralRecon y with white Gaussian noises ϵ as conditional signals, which encourages SparseDiff to generate high-quality TSDFs from scratch.

Specifically, SparseDiff accepts a two-channel TSDF including a conditional TSDF y provided by NeuralRecon and an TSDF to be diffused x_t, which acts as predicting the parameters $\mu_\theta\left(x_t, y, t\right)$ and $\Sigma_\theta\left(x_t, y, t\right)$ conditioned on a rough TSDF y. The reverse process can be depicted as:

$$p_\theta\left(x_{t-1} \mid x_t, y\right)=\mathcal{N}\left(x_{t-1} ; \mu_\theta\left(x_t, y, t\right), \Sigma_\theta\left(x_t, y, t\right)\right). \tag{2}$$

As reported by Ho et al. [10], directly predicting the cumulative noise ϵ_θ that is added to the TSDF x_t in the current step is better:

$$Loss=E_{t, x_0, \epsilon, y}\left[\left\|\epsilon-\epsilon_\theta\left(x_t, y, t\right)\right\|^2\right]. \tag{3}$$

Then, we can obtain the mean parameters from the predicted noise by

$$\mu_\theta\left(x_t, y, t\right)=\frac{1}{\sqrt{\alpha_t}}\left(x_t-\frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} \epsilon_\theta\left(x_t, y, t\right)\right) \tag{4}$$

with $\alpha_t:=1-\beta_t$ and $\bar{\alpha}_t:=\prod_{s=0}^t \alpha_s$.

The conditioning TSDF y provides a rough guidance for SparseDiff and thus eases the diffusion. After the first step, DiffRoom is able to reconstruct high-quality room TSDF from posed images with the help of an external module NeuralRecon but cannot generate TSDFs from white noises. To achieve so, we start the second step that gradually replaces the conditioning TSDF y with white noises $\epsilon \sim \mathcal{N}(0,1)$. Specifically, we generate random masks $M \in\{0,1\}^{H \times W \times D}$ for the condition TSDF y and replace the TSDF of the masked voxels with white noises ϵ:

$$\hat{y}=M \odot y+(1-M) \odot \epsilon. \tag{5}$$

Also, we randomly generate conditional signals that are pure white noises.

After CDL, the NeuralRecon can be removed. SparseDiff is able to either reconstruct high-quality room geometries with an off-the-shelf NeuralRecon or directly generate high-quality room geometries from white Gaussian noises. CDL also boosts the training convergence speed of SparseDiff.

3.3 Local Fusion for Global Diffusion

Our goal is to generate an entire room. Instead of directly diffusing an entire large room globally, we train SparseDiff to generate TSDF for crops of size $L \times L \times L$ and fuse local crops to obtain a global TSDF. Specifically, given a set of occupied voxels that represent a room of size $H \times W \times D$, we randomly crop spaces with a size of $L \times L \times L$ and retrieve the occupied voxels along with their ground truth TSDFs for training. Cropping a small space not only reduces the necessary computation and memory cost in training but brings a challenge in inference: how to generate a large room with a diffusion-based generator that produces fixed-sizedTSDFs. Previous image generators [17, 28] synthesize large images by generating overlapped local crops in order with a sliding window. However, in this manner, the content generated later conditions on content generated earlier, which does not propagate global information and causes inconsistency. We propose a stochastic fusion for global diffusion, which can simultaneously generate the entire room.

During inference, we split the room space into K overlapped 3D crops $\{\mathcal{P}^0, \mathcal{P}^1, \ldots, \mathcal{P}^{K-1}, \}$ that cover the entire room. We generate the TSDF for the entire room by concurrently diffusing the K crops with stochastic fusion. We denote global TSDF at the timestep t as x_t and the k-th crop at the timestep t as x_t^k and global TSDF at the timestep t as x_t. At the time step t, we need to obtain the global TSDF x_t by fusing local TSDFs x_t^k and then update local TSDFs from the global TSDF: $x_t^k(\mathbf{p}_i) = x_t(\mathbf{p}_i)$. After synchronizing local TSDFs with fusion, each crop step to the next time step individually. Specifically, for a voxel grid $\mathbf{p}$, suppose $\mathcal{G}(\mathbf{p})$ contains the crops that cover $\mathbf{p}$: $\mathcal{G}(\mathbf{p}) = \{k | \mathbf{p} \in \mathcal{P}^k\}$, we need to obtain $x_t(\mathbf{p})$ by fusing the crops $\{x_t^k(\mathbf{p}) | k \in \mathcal{G}(\mathbf{p})\}$ overlapped on $\mathbf{p}$.

Average TSDF Fusion. A straightforward fusion algorithm is taking the average TSDFs of the local crops: $x_t(\mathbf{p}) = \frac{1}{|\mathcal{G}(\mathbf{p})|} \sum_{k \in \mathcal{G}} x_t^k(\mathbf{p})$, which is also adopted by the classical KinectFusion [14]. However, average fusion seriously reduces the variance of the sample distribution. Suppose $x_t^k(\mathbf{p}) \sim \mathcal{N}(\mu_t^k(\mathbf{p}), \Sigma_t^k(\mathbf{p}))$, we have:

$$x_t(\mathbf{p}) \sim \mathcal{N}(\frac{1}{|\mathcal{G}(\mathbf{p})|} \sum_{k \in \mathcal{G}(\mathbf{p})} \mu_t^k(\mathbf{p}), \frac{1}{|\mathcal{G}(\mathbf{p})|^2} \sum_{k \in \mathcal{G}(\mathbf{p})} \Sigma_t^k(\mathbf{p}))). \tag{6}$$

The rapidly decreasing variance impacts generation diversity and quality. We, therefore, propose a stochastic TSDF fusion algorithm.

Stochastic TSDF Fusion. To ensure the generation quality and global consistency, we propose stochastic fusion to keep the distribution and fuse TSDFs in the reverse process. Specifically, we randomly sample an index k from $\mathcal{G}(\mathbf{p})$ in a uniform distribution to update the global TSDF $x_t(\mathbf{p}) = x_t^k(\mathbf{p})$, which remains the distribution:

$$x_t(\mathbf{p}) \sim \mathcal{N}(\mu_t^k(\mathbf{p}), \Sigma_t^k(\mathbf{p})), k = \mathrm{RandomSelect}\left(\mathcal{G}(\mathbf{p})\right). \tag{7}$$

We generate large and high-quality room geometries by generating local crops' TSDF in parallel with the stochastic fusion.

4 Experiments

DiffRoom can reconstruct 3D TSDF from posed images by taking the initial TSDF obtained from NeuralRecon and generating 3D TSDF from white noises, so we provide two versions of our results: Ours Reconstruction (Ours R.) and Ours Generation (Ours G.). We train DiffRoom on the training split of ScanNet dataset [5] and evaluate DiffRoom on the test split. The ScanNet dataset contains 1613 indoor scenes with ground-truth camera poses and surface reconstructions. We follow the data split as NeuralRecon [35]. More implementation details can be found in the first section of the supplementary materials.

Experiment Setup The metrics used in NeuralRecon only evaluate rough geometries while our goal is to reconstruct high-quality meshes. We thus set up three experiments to evaluate the reconstructed mesh quality. We first evaluate the reconstruction accuracy by 3D normal errors, and then quantitatively and qualitatively compare the reconstructed mesh quality with three metrics for triangle meshes.

Table 1: 3D Reconstruction accuracy comparison. We compare the mean and ratio of 3D normal errors with three thresholds: $90°, 45°, 30°$.

	NeuralRecon [35]	NeuralRecon [35] + Lap. Denoising [34]	SimpleRecon [30]	Ours
<90° mean↓	34.65	37.12	35.44	**30.4**
<90° ratio↑	100%	100%	100%	**100%**
<45° mean↓	10.27	12.34	12.51	**8.09**
<45° ratio↑	59.97%	57.97%	60.20%	**65.05%**
<30° mean↓	6.45	8.17	8.31	**5.05**
<30° ratio↑	52.88%	49.89%	51.67%	**59.27%**

Table 2: Mesh quality comparison. We compare the mean and variance of three scores: Aspect Ratio, Circularity, and Shape Regularity.

	Neural-Recon [35]	NeuralRecon [35] + Lap. Denoising [34]	Simple-Recon [30]	Ours G.	Ours R.	GT
Aspe. mean↑	0.459	0.437	0.436	0.469	0.457	**0.477**
Aspe. var↓	0.022	0.023	0.024	0.020	**0.016**	0.022
Circ. mean↑	0.740	0.712	0.708	0.742	**0.763**	0.758
Circ. var↓	0.041	0.052	0.054	0.041	**0.030**	0.034
Shap. mean↑	0.772	0.746	0.739	0.793	**0.797**	0.793
Shap. var↓	0.041	0.052	0.055	0.041	**0.030**	0.031

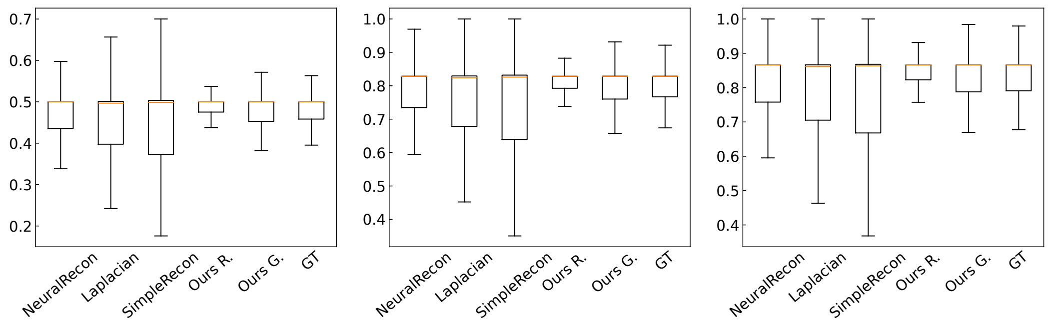
We also conduct a user study to investigate human preferences. Finally, we ablation the proposed modules to show their effectiveness. We compare our methods with three methods, NeuralRecon (our baseline), NeuralRecon [35] with Laplacian denoising [34] (denoted as "NeuralRecon+Lap. Denosing"), and SimpleRecon [30].

4.1 Reconstruction Accuracy

Normals $\mathcal{N}(\mathbf{p}) = \Delta x(\mathbf{p})$ denote the shape variations of surfaces. To evaluate the accuracy of the reconstructed 3D scenes at a detail level, we compute the normal difference between the reconstructed meshes and the ground truth meshes $\mathrm{Error}(\mathbf{p}) = |\mathcal{N}_{\mathrm{pred}}(\mathbf{p}) - \mathcal{N}_{\mathrm{GT}}(\mathbf{p})|_2$. The smaller normal errors, the better the reconstructed mesh variations are aligned to the ground truth mesh. Given a mesh, we compute the normals and compare their normal errors in Table 1. We regard normals whose errors are larger than $90°$ as outliers and filter them out. For the inlier normals, we set a threshold T and compare the percentage of normals whose errors are lower than the threshold ($< T°$ ratio). Besides, we also compare the mean normal error of the normals whose errors are lower than the threshold ($< T°$ mean). Our method significantly outperforms the other three methods on all thresholds. 65.05% of our normals have errors lower than $45°$. Moreover, the average normal error on this threshold is 8.09, a 21.2% error reduction from NeuralRecon, the runner-up. Note that our DiffRoom retains more normals than NeuralRecon (65.05% v.s. 59.97%), but still achieves a large error reduction (8.09 v.s. 10.27), which reveals the superior performance of our DiffRoom in both robustness and accuracy.

4.2 Quantitative Mesh Quality Comparison

Noisy meshes and high-quality meshes present different distributions of triangle shapes. Triangles in noisy meshes and problematic regions have a low aspect ratio, circularity, and regularity as introduced in Brandts et.al. [2]. We take the following three scores to evaluate the mesh quality: *Aspect Ratio.* Low aspect ratio triangles are very long and skinny. Suppose S and l stand for the area and longest side edge of triangles, we compute S/l to evaluate the aspect ratio of triangles. *Circularity.* We denote the largest circle that can be placed inside the triangle (inradius) and the smallest circle that encloses the triangle (circumradius) by R_L and R_S. We compute the ratio R_L/R_S to evaluate the circularity of triangles. High circularity is closer to being equilateral, which is generally considered a desirable mesh property. *Shape Regularity.* Suppose a, b, c respectively denote the three edges of



Aspect Ratio   Circularity   Shape Regularity

Figure 4: Mesh quality comparison in box-and-whisker plots.

Table 3: User Study.

	NeuralRecon [35]	NeuralRecon [35] + Lap. Denoising [34]	Ours	Ground Truth
Details↑	12.26	6.80	17.41	**25.50**
Completeness↑	11.15	8.84	**21.30**	20.76
Tight Plane↑	5.48	12.10	**25.85**	18.17
Sharp Edge↑	8.22	10.20	**22.58**	21.19
Overall (Sum)↑	37.10	37.98	**87.14**	85.62

triangles, we compute $S/(a^2+b^2+c^2)$ to evaluate the shape regularity. *Result Analysis.* We compute the mean and variance of the three scores described above and compare them in Table 2 and Fig. 4. We provide two versions of our DiffRoom: Ours R. and Ours G., which respectively generate rooms from the NeuralRecon initialized TSDFs and white noises. Our reconstruction (denoted as "Ours R.") presents significantly better performance than the other three methods, i.e., higher scores with smaller score variance. Our generation (denoted as "Ours G.") achieves competitive performance with NeuralRecon. Moreover, we also include ground truth mesh for comparison. Interestingly, our DiffRoom also outperforms the ground truth except for the aspect ratio score, which indicates the high quality of our reconstructed meshes.

4.3 Qualitative Mesh Quality Comparison

We qualitatively compare NeuralRecon and our reconstruction results in Fig. 5. We visualize the meshes with their local curvatures. Green and yellow regions indicate those of low and high curvatures. For instance, the curvatures on the floors are expected to be small because they are flat planes. Our reconstruction presents even better quality than the ground truth meshes from this perspective. We also zoom in on the meshes to provide more details. NeuralRecon's reconstructed meshes tend to be noisy and some semantic shapes are corrupted. In contrast, our reconstructions are smooth on flat areas while recovering sharp edges of the scene, e.g., the chairs and desks. This visualization illustrates why our meshes ranks 1st in Table 2 and Fig. 4.

4.4 User Study

We randomly sample 10 scenes of the test split of ScanNet dataset, and visualize the reconstructed meshes generated by the four methods from the same viewpoint. We then request users to rank those methods according to their details, completeness, plane quality, and edge quality. We collect the ranking results from 51 users. The feedback score S_i of the i-th scene is computed as following equation,

$$S_i = \frac{1}{d_i}\sum_{j=1}^{d_i} s(r_{i,j}), \tag{8}$$

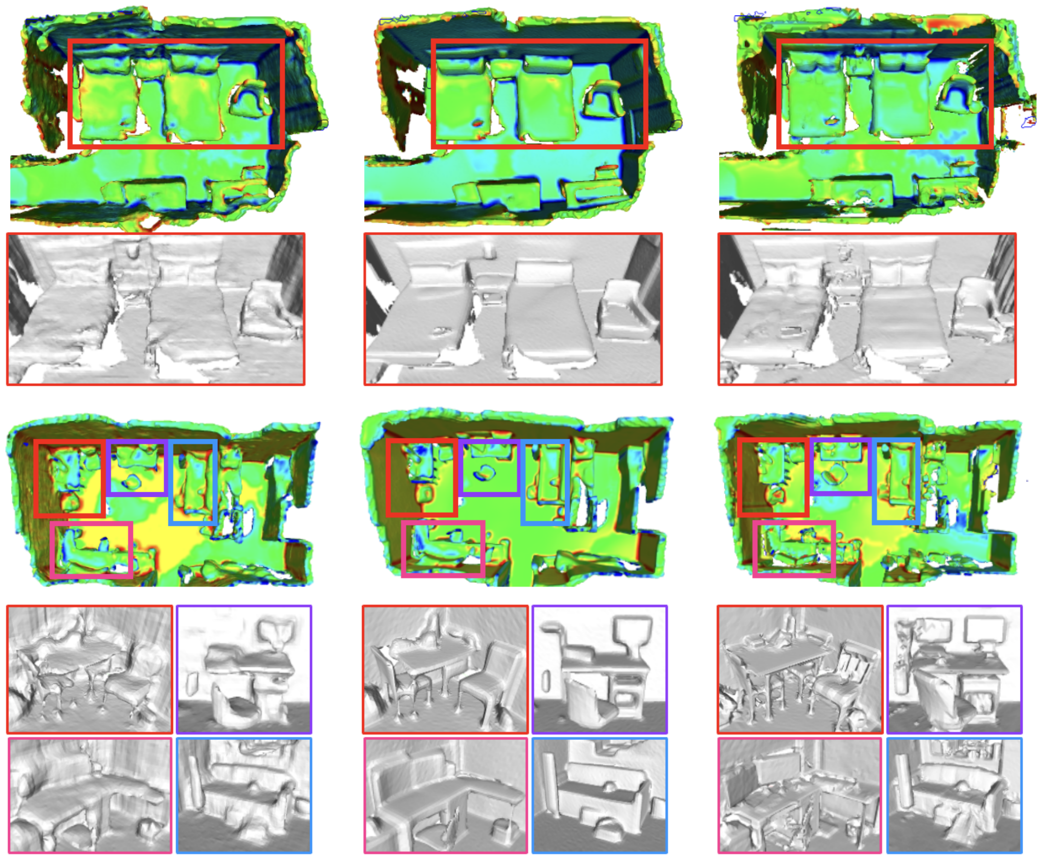


Figure 5: Visualization of mesh quality. The meshes are colored according to curvatures. Green regions denote lower curvatures. We compare result meshes of NeuralRecon, our method, and ground truth.

where $r_{i,j} \in \{1, 2, 3, 4\}$ is the ranking of the j-th user on the i-th scene. $s(r) = 4 - r$ computes the score from the ranking, i.e., the r-th rank worth $4 - r$ score. d_i is the number of total valid feedbacks. We sum up the scores over all scenes to obtain the total score: $S = \sum_{i=1}^{N} S_i$. As shown in Table 3, we compare the 3D reconstructed meshes with such scores S from the user study. Our reconstructed mesh is significantly better than NeuralRecon, "NeuralRecon+Laplacian Denoising" (denoted as "NeuralRecon+Lap.Denoising"), and even better than the ground truth meshes.

4.5 Ablation Study

Sparse Diffusion v.s. Dense Diffusion To compare the dense diffusion and the sparse diffusion, we implement two networks using sparse and dense convolution respectively, with exactly the same structures. Several randomly cropped TSDF volumes (96×96×96) from ScanNet dataset are fed into these two models. The resource consumption of the two strategies are shown in Table 4. This experiment is conducted on the platform equipped with RTX3090 GPU with 24GB memory, with the diffusion model running in training status. The sparse diffusion requires fewer resources and runs faster with fewer parameters, due to the characteristic of the occupancy distribution. In our test data, the largest occupancy rate of the TSDF crops are less than 20%.

Diffusion with Fusion. We fuse TSDFs in overlapped crops during the diffusion iteration. To investigate the effectiveness of our stochastic fusion, we compare stochastic fusion, average fusion, and individual diffusion without fusion in Fig. 6. Individual diffusion presents significant inconsistency between adjacent crops. Average fusion generates meshes of lower quality because the sample

Table 4: Resource Consumption Comparison.

	TFLOPs	Parameters(M)	GPU Memory(GB)		
Batch Size	1	1	1	2	4
Sparse	0.008	161.5	11.8	15.3	22.8
Dense	3.290	161.5	22.8	-	-

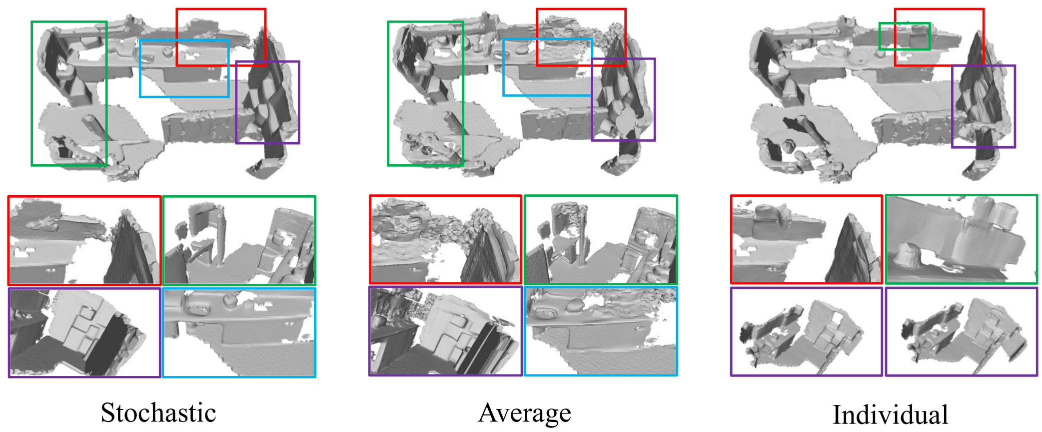


Figure 6: Comparison of diffusion with different fusion strategies.

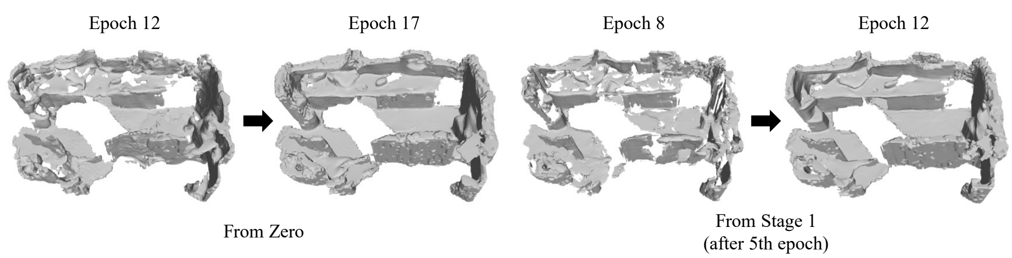


Figure 7: Curricular learning speed up the training process.

distribution during diffusion is disturbed. Our stochastic diffusion remains global consistency and generates high-quality meshes.

Curriculum Diffusion Learning (CDL). We conduct a two-step CDL that first learns to generate finer meshes from a rough mesh and then gradually replaces parts of the rough meshes with white noises. To investigate the effectiveness of CDL, we compare the results of SparseDiff models trained with two policies. One is trained from scratch and the other one is trained in two steps as CDL. Both trained with 12 epochs, the SparseDiff trained with CDL converges faster and presents better details as shown in Fig. 7.

5 Conclusions

We have presented DiffRoom a novel framework for diffusion-based high-quality room reconstruction and generation. DiffRoom mainly consists of three modules: 1) a sparse diffusion network architecture SparseDiff that efficiently denoises 3D SDFs, 2) a curriculum diffusion learning (CDL) that boosts training convergence speed and achieves high-quality reconstruction and generation, and 3) a stochastic fusion algorithm for diffusion-based local SDFs, which enables large-scale room generation. **Limitations**. The proposed method relies on a pre-defined occupancy field, which provide sparse occupied voxels. In the future, we will explore to generate occupancy field so that DiffRoom can generate high-quality large room geometries from scratch.

References

[1] Aljaz Bozic, Pablo Palafox, Justus Thies, Angela Dai, and Matthias Nießner. Transformerfusion: Monocular rgb scene reconstruction using transformers. *Advances in Neural Information Processing Systems*, 34:1403–1414, 2021.

[2] Jan Brandts, Sergey Korotov, and Michal Křížek. On the equivalence of regularity criteria for triangular and tetrahedral finite element partitions. *Computers & Mathematics with Applications*, 55(10):2227–2233, 2008. Advanced Numerical Algorithms for Large-Scale Computations.

[3] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgb-d data in indoor environments. *arXiv preprint arXiv:1709.06158*, 2017.

[4] Angel Chang, Manolis Savva, and Christopher D Manning. Learning spatial knowledge for text to 3d scene generation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 2028–2038, 2014.

[5] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5828–5839, 2017.

[6] Matthew Fisher, Manolis Savva, Yangyan Li, Pat Hanrahan, and Matthias Nießner. Activity-centric scene synthesis for functional 3d scene modeling. *ACM Transactions on Graphics (TOG)*, 34(6):1–13, 2015.

[7] Qiang Fu, Xiaowu Chen, Xiaotian Wang, Sijia Wen, Bin Zhou, and Hongbo Fu. Adaptive synthesis of indoor scenes via activity-associated object relation graphs. *ACM Transactions on Graphics (TOG)*, 36(6):1–13, 2017.

[8] Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P Kingma, Ben Poole, Mohammad Norouzi, David J Fleet, et al. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022.

[9] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

[10] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

[11] 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.*, 23(47):1–33, 2022.

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

[13] Ka-Hei Hui, Ruihui Li, Jingyu Hu, and Chi-Wing Fu. Neural wavelet-domain diffusion for 3d shape generation. In *SIGGRAPH Asia 2022 Conference Papers*, pages 1–9, 2022.

[14] Shahram Izadi, David Kim, Otmar Hilliges, David Molyneaux, Richard Newcombe, Pushmeet Kohli, Jamie Shotton, Steve Hodges, Dustin Freeman, Andrew Davison, et al. Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pages 559–568, 2011.

[15] Zhijian Liu, Haotian Tang, Yujun Lin, and Song Han. Point-voxel cnn for efficient 3d deep learning. *Advances in Neural Information Processing Systems*, 32, 2019.

[16] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. *ACM siggraph computer graphics*, 21(4):163–169, 1987.

[17] 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*, pages 11461–11471, 2022.

[18] 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*, pages 2837–2845, 2021.

[19] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. Sdedit: Guided image synthesis and editing with stochastic differential equations. In *International Conference on Learning Representations*, 2021.

[20] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics (ToG)*, 41(4):1–15, 2022.

[21] Zak Murez, Tarrence Van As, James Bartolozzi, Ayan Sinha, Vijay Badrinarayanan, and Andrew Rabinovich. Atlas: End-to-end 3d scene reconstruction from posed images. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VII 16*, pages 414–431. Springer, 2020.

[22] Richard A Newcombe, Dieter Fox, and Steven M Seitz. Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 343–352, 2015.

[23] Alex 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. *arXiv preprint arXiv:2112.10741*, 2021.

[24] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022.

[25] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017.

[26] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017.

[27] Siyuan Qi, Yixin Zhu, Siyuan Huang, Chenfanfu Jiang, and Song-Chun Zhu. Human-centric indoor scene synthesis using stochastic grammar. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5899–5908, 2018.

[28] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022.

[29] Chitwan Saharia, Jonathan Ho, William Chan, Tim Salimans, David J Fleet, and Mohammad Norouzi. Image super-resolution via iterative refinement. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[30] Mohamed Sayed, John Gibson, Jamie Watson, Victor Prisacariu, Michael Firman, and Clément Godard. Simplerecon: 3d reconstruction without 3d convolutions. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXIII*, pages 1–19. Springer, 2022.

[31] Johannes L Schönberger, Enliang Zheng, Jan-Michael Frahm, and Marc Pollefeys. Pixelwise view selection for unstructured multi-view stereo. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III 14*, pages 501–518. Springer, 2016.

[32] 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.

[33] Yang Song and Stefano Ermon. Improved techniques for training score-based generative models. *Advances in neural information processing systems*, 33:12438–12448, 2020.

[34] Olga Sorkine, Daniel Cohen-Or, Yaron Lipman, Marc Alexa, Christian Rössl, and H-P Seidel. Laplacian surface editing. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, pages 175–184, 2004.

[35] Jiaming Sun, Yiming Xie, Linghao Chen, Xiaowei Zhou, and Hujun Bao. Neuralrecon: Real-time coherent 3d reconstruction from monocular video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15598–15607, 2021.

[36] Thomas Whelan, Stefan Leutenegger, Renato Salas-Moreno, Ben Glocker, and Andrew Davison. Elasticfusion: Dense slam without a pose graph. Robotics: Science and Systems, 2015.

[37] Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. Mvsnet: Depth inference for unstructured multi-view stereo. In *Proceedings of the European conference on computer vision (ECCV)*, pages 767–783, 2018.

[38] Guofeng Zhang, Jiaya Jia, Tien-Tsin Wong, and Hujun Bao. Consistent depth maps recovery from a video sequence. *IEEE Transactions on pattern analysis and machine intelligence*, 31(6):974–988, 2009.

[39] Enliang Zheng, Enrique Dunn, Vladimir Jojic, and Jan-Michael Frahm. Patchmatch based joint view selection and depthmap estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1510–1517, 2014.

[40] 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*, pages 5826–5835, 2021.