DiffRoom: Supplementary Materials

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 Implementation Details

A.1 Dataset and Preprocessing

We use the official train / validation / test split of ScanNet(v2) dataset, including 1201 / 312 / 100 scenes respectively. For there is no TSDF ground truth provided in this dataset, we adopt a TSDF fusion method like [S3] to produce the ground truth as NeuralRecon does. We only use TSDF data without any other data type such as images in the whole training/testing process. To compare the reconstruction results with pretrained NeuralRecon, the grid size of TSDF volume is set to 0.04m, and the truncation distance is set to 0.12m. The default value of the TSDF volume is 1.0.

In the training process, a random volume crop of $96 \times 96 \times 96$ is used as data augmentation, where a random rotation between $[0, 2\pi]$ and a random translation is performed before cropping. To ensure that the sampling crop contains sufficient occupied voxels, the translation is limited in the bounding box of global occupied region, and the entire cropped volume should be within the boundary of this region.

A.2 SparseDiff Model

`TorchSparse` [S6] is used to implement the UNet structure in our SparseDiff model. A group normalization(32 groups) and a SiLU activation are used successively before any layer of sparse convolution. The network parameters is randomly initialized in training process, and we use the Adam optimizer with a fixed learning rate of 1e-4. The model is trained on 8 Nvidia 3090 GPUs with batch size=2 on every GPU.

As for the diffusion framework, the `DDIMScheduler` in the open-source diffusers[S7] is developed as our code-base. Following [S1] and [S4], we adopt the $\alpha-$conditioning to stabilize training, and enable the parameter tuning over the noise schedule and the timesteps during inference stage. More concretely, the cumulative product of α_t namely $\bar{\alpha}_t$ is used as a substitute of the timestep t as time embedding in most existing works. We use a linear noise schedule of $(1e^{-6}, 0.01)$ with 2000 timesteps during training, and the same noise schedule is used with 100 time-step samples during inference within the DDIM framework. The hyper-parameter η for variance level control in the generative process is 0.8, and the clip range for TSDF sample is $[-3.0, 3.0]$.

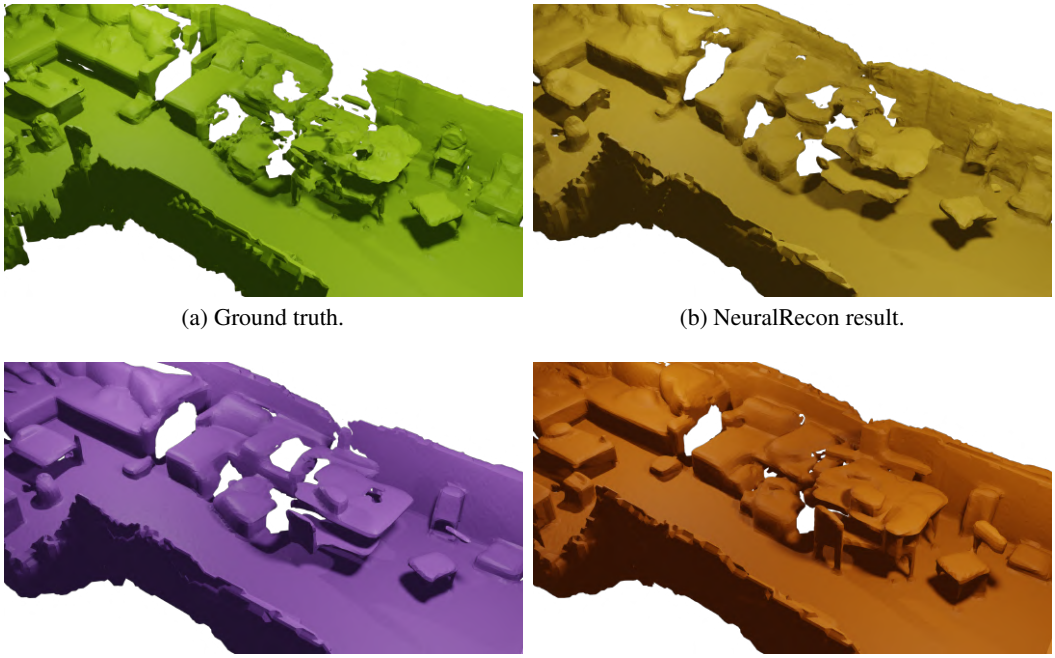
B More Qualitative Results

B.1 Reconstruction & Generation on ScanNet

In this section, all TSDF volumes from different sources are trimmed to have the same occupancy with NeuralRecon, with all other "vacant" grids filled by the default value 1.0. Then the Marching Cube algorithm is used to produce meshes from those TSDF volumes. Therefore, the ground truth may seem to be a little shabby. The results are compared under same camera view in Fig. 1~6.

*Both authors contributed equally to this work.

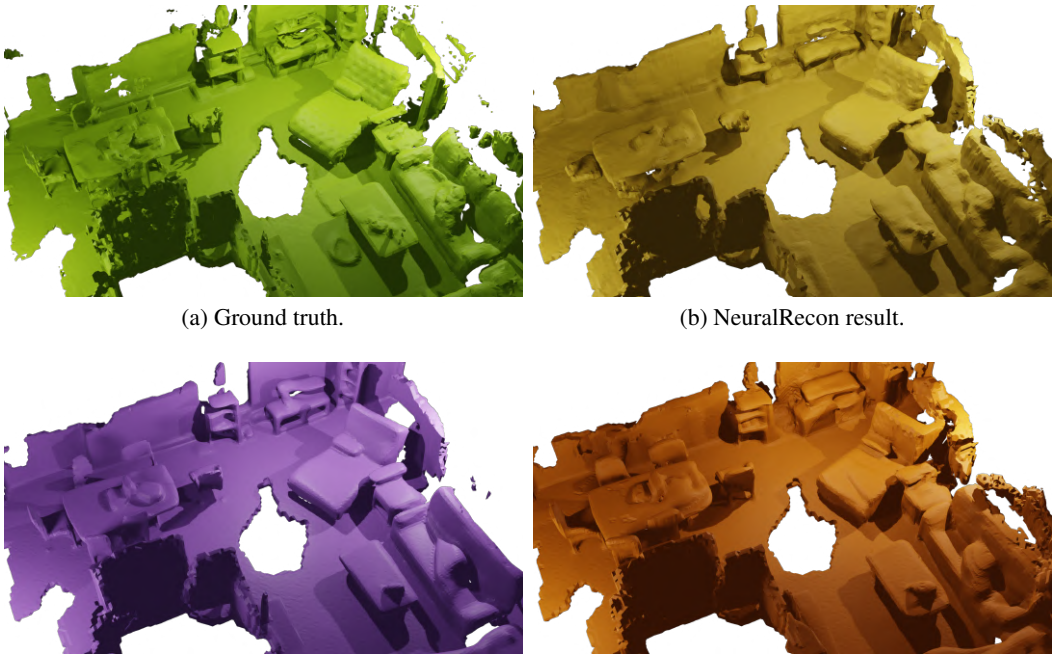
Preprint. Under review.



(a) Ground truth. (b) NeuralRecon result.

(c) Reconstruction. (d) Generation.

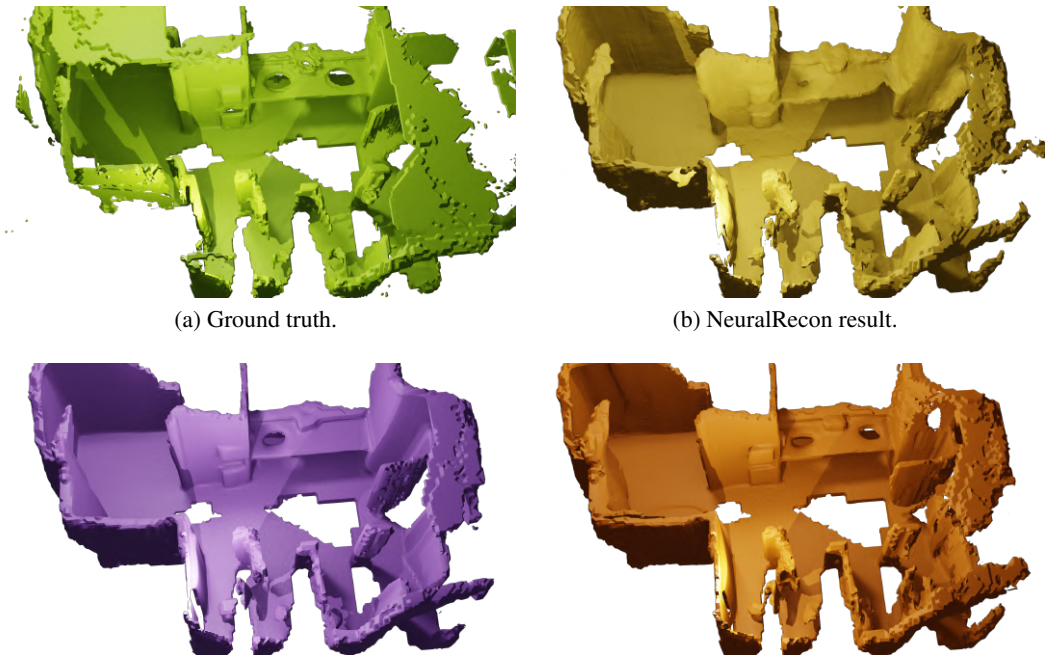
Figure 1: Sample scene0744 in ScanNet. The reconstruction and generation are based on same occupancy with the result from NeuralRecon, and the same applies to the following Fig. 2~6.



(a) Ground truth. (b) NeuralRecon result.

(c) Reconstruction. (d) Generation.

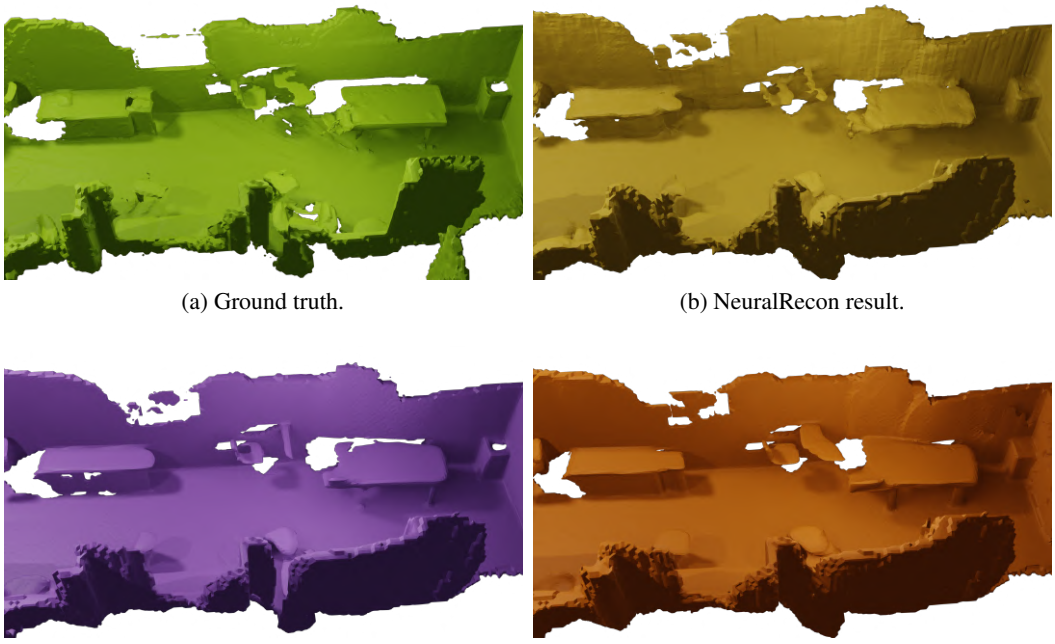
Figure 2: Sample scene0747 in ScanNet.



(a) Ground truth. (b) NeuralRecon result.

(c) Reconstruction. (d) Generation.

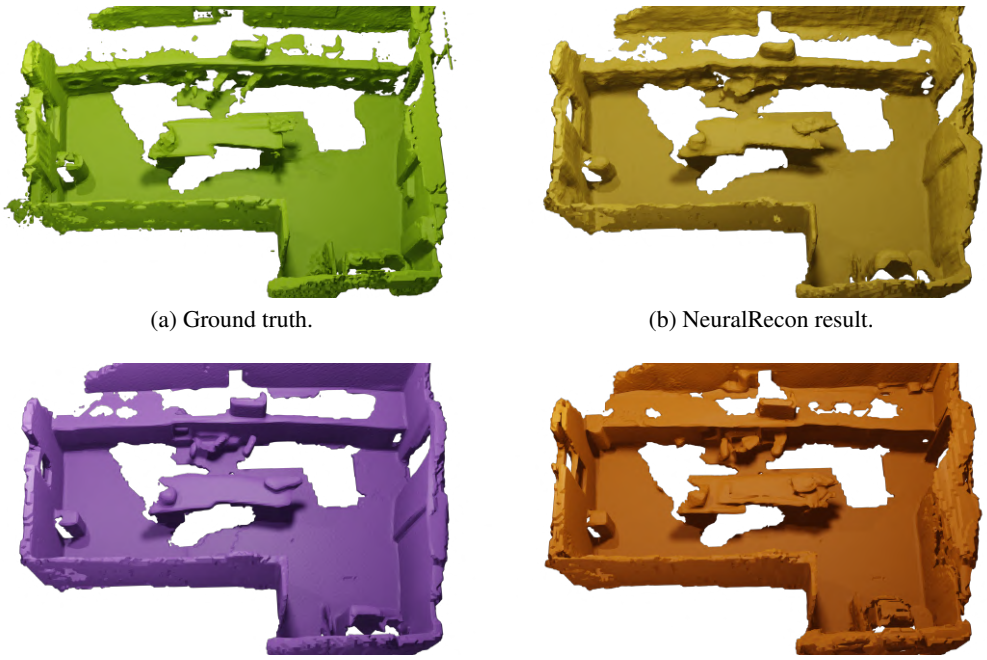
Figure 3: Sample scene0775 in ScanNet.



(a) Ground truth. (b) NeuralRecon result.

(c) Reconstruction. (d) Generation.

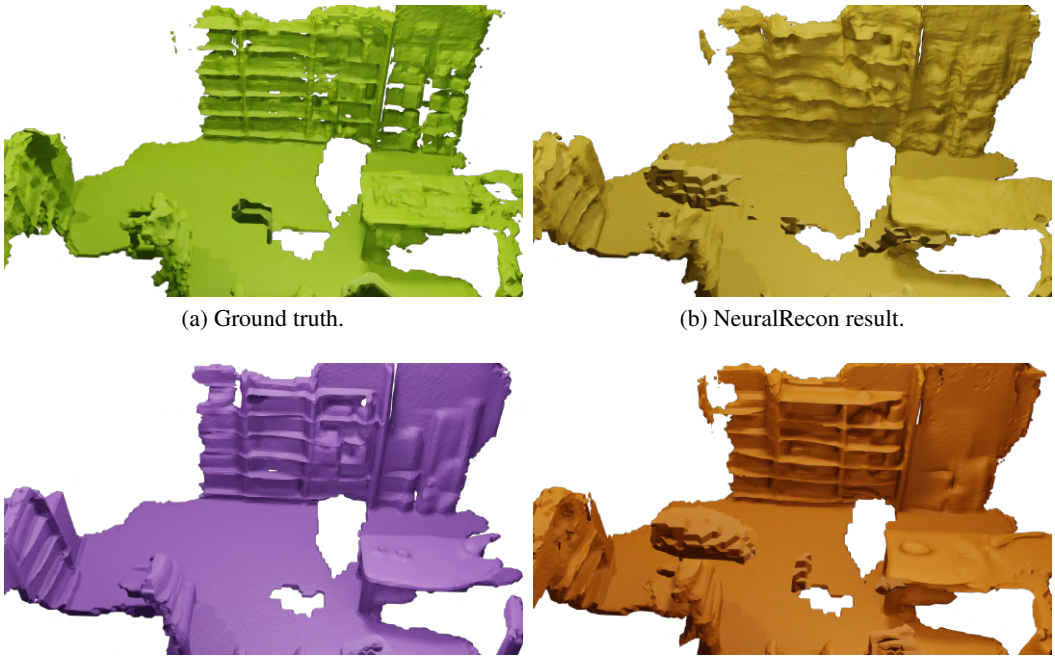
Figure 4: Sample scene0777 in ScanNet.



(a) Ground truth. (b) NeuralRecon result.

(c) Reconstruction. (d) Generation.

Figure 5: Sample scene0780 in ScanNet.



(a) Ground truth. (b) NeuralRecon result.

(c) Reconstruction. (d) Generation.

Figure 6: Sample scene0799 in ScanNet.

B.2 Generation on Replica

Replica [S5] is a dataset of 18 synthetic 3D indoor scene reconstructions, including dense meshes of high quality. First, we process those meshes by cutting off the ceiling layers, so that the inner scene can be exposed to camera view. Then those meshes are optimized to be watertight using the method proposed by [S2]. Finally, the SDF volumes can be generated from those scenes. To make the ground truth for our model, we produce TSDF from SDF with a truncation value of 0.12m, and the occupancy provided for the generation task is the grids with a absolute TSDF value less than 1.0. To make the generation more challenging, a Gaussian noise is added to the grids of SDF volumes with absolute value in $[0.12, 0.20]$ before the calculating the occupancy.

In this section, we use a model pretrained on ScanNet and finetuned on 17 scenes of Replica dataset. We test the model on the largest scene "Appartment0", and the results shown as Fig. 7~9.

Supplementary Reference

[S1] Nanxin Chen, Yu Zhang, Heiga Zen, Ron J Weiss, Mohammad Norouzi, and William Chan. Wavegrad: Estimating gradients for waveform generation. *arXiv preprint arXiv:2009.00713*, 2020.

[S2] Jingwei Huang, Hao Su, and Leonidas Guibas. Robust watertight manifold surface generation method for shapenet models. *arXiv preprint arXiv:1802.01698*, 2018.

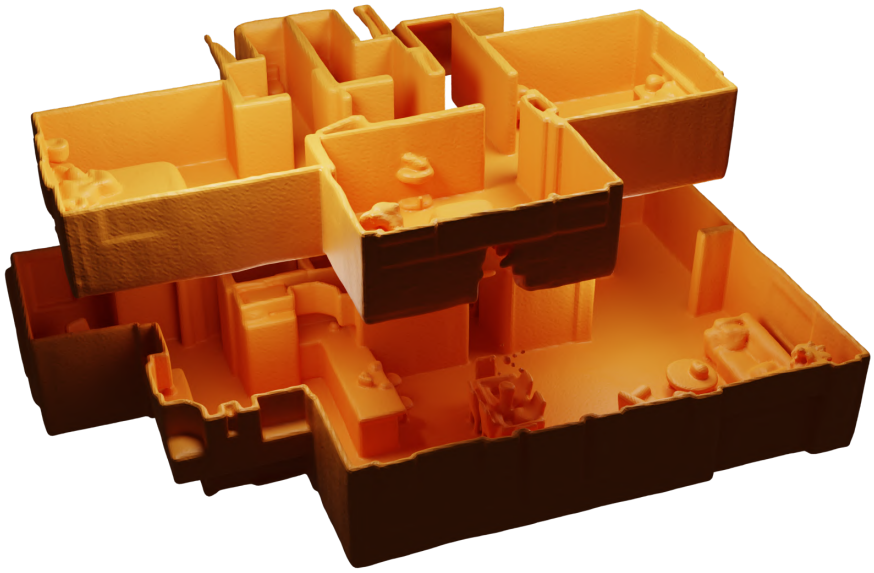
[S3] Shahram Izadi, David Kim, Otmar Hilliges, David Molyneaux, Richard Newcombe, Pushmeet Kohli, Jamie Shotton, Steve Hodges, Dustin Freeman, Andrew Davison, et al. Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pages 559–568, 2011.

[S4] Chitwan Saharia, William Chan, Huiwen Chang, Chris Lee, Jonathan Ho, Tim Salimans, David Fleet, and Mohammad Norouzi. Palette: Image-to-image diffusion models. In *ACM SIGGRAPH 2022 Conference Proceedings*, pages 1–10, 2022.

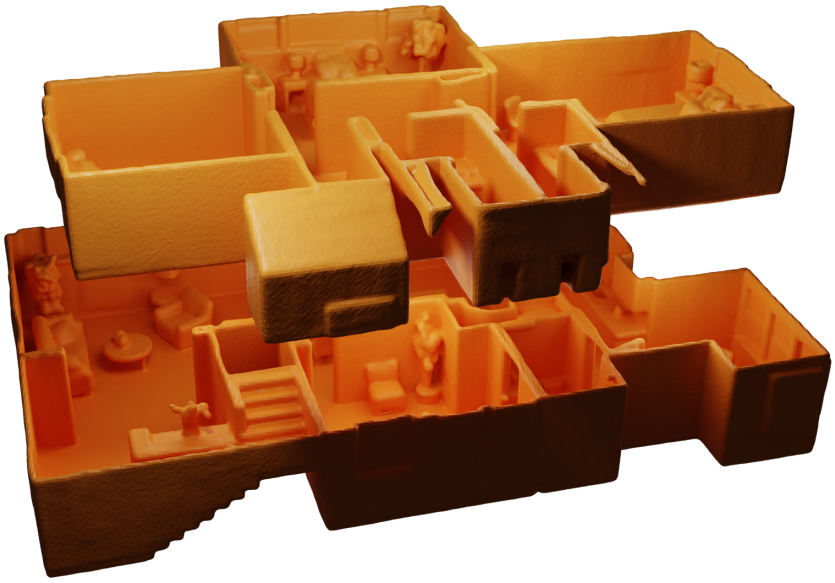
[S5] Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J. Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, Anton Clarkson, Mingfei Yan, Brian Budge, Yajie Yan, Xiaqing Pan, June Yon, Yuyang Zou, Kimberly Leon, Nigel Carter, Jesus Briales, Tyler Gillingham, Elias Mueggler, 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 preprint arXiv:1906.05797*, 2019.

[S6] Haotian Tang, Zhijian Liu, Xiuyu Li, Yujun Lin, and Song Han. TorchSparse: Efficient Point Cloud Inference Engine. In *Conference on Machine Learning and Systems (MLSys)*, 2022.

[S7] Patrick von Platen, Suraj Patil, Anton Lozhkov, Pedro Cuenca, Nathan Lambert, Kashif Rasul, Mishig Davaadorj, and Thomas Wolf. Diffusers: State-of-the-art diffusion models. `https://github.com/huggingface/diffusers`, 2022.

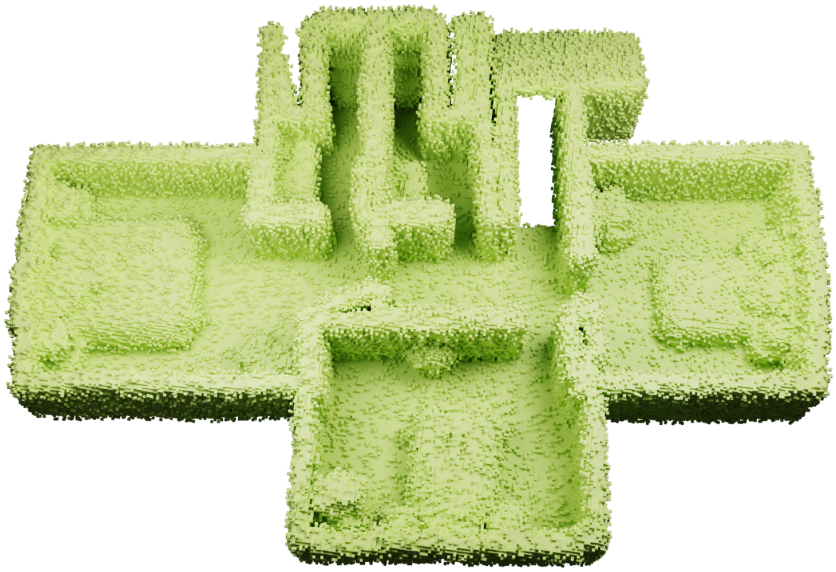


(a) The front view.

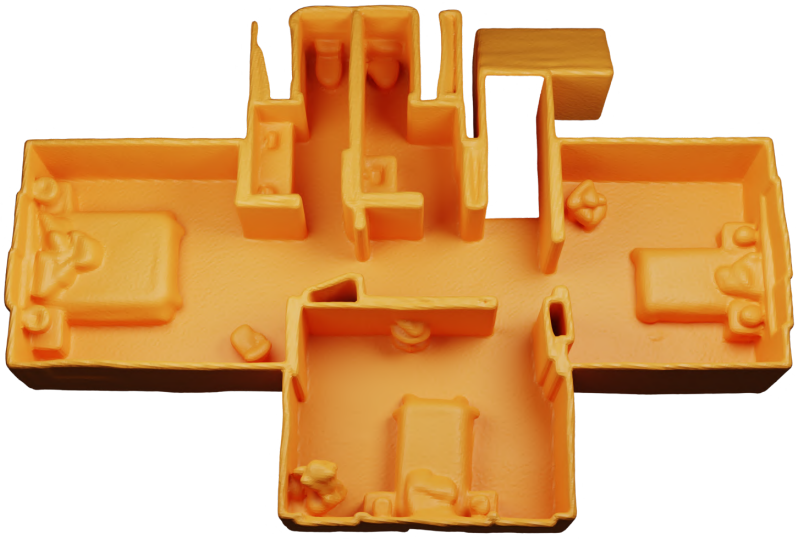


(b) The back view.

Figure 7: Generated scene from noisy occupancy of "Appartment0" in Replica, which is the largest scene in the dataset.



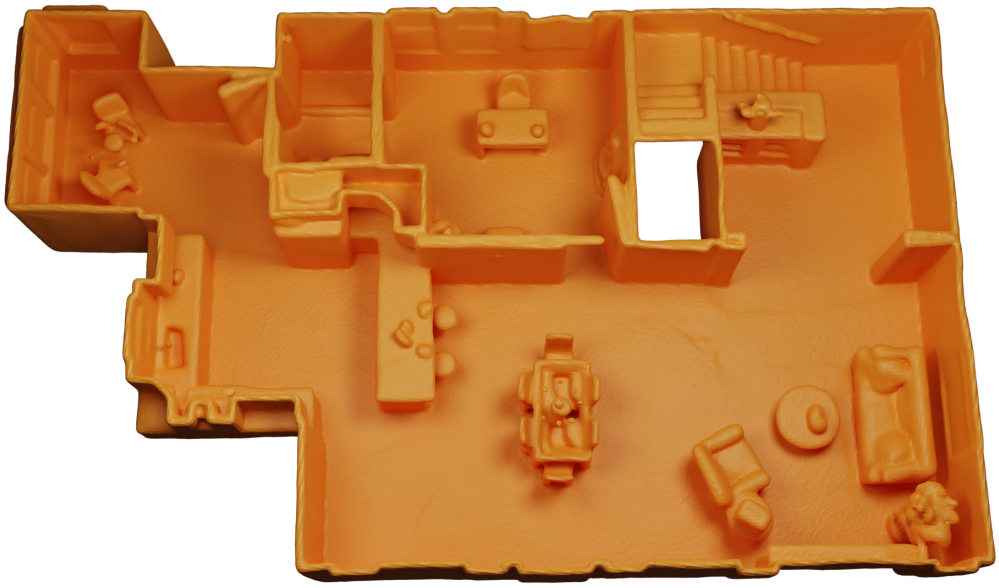
(a) The input to our model: noisy occupancy grids.



(b) The generated result.

Figure 8: Generated scene from noisy occupancy of "Appartment0 1/F" in Replica. The size of this scene is $9.0m \times 12.3m \times 2.1m$.

(a) The input to our model: noisy occupancy grids.



(b) The generated result.

Figure 9: Generated scene from noisy occupancy of "Appartment0 G/F" in Replica. The size of this floor is $9.4m \times 14.9m \times 2.4m$.