Learning Long-range Information with Dual-Scale Transformers for Indoor Scene Completion
Supplementary Material

A. More Real World Scene Completion Results

Fig. 1 and Fig. 2 shows more completion results of the four methods on Matterport3D [1] dataset and ScanNet [2] dataset, respectively. In visual effects, our method completes more parts in different scenes than the other three methods.

B. Experimental Details

Experimental Details of ICL-NUIM. In the evaluation of synthetic dataset ICL-NUIM [4], instead of artificially creating deletions on the mesh, we use the RGB-D frames provided by it to obtain the various TSDF input with 2 cm resolution through voxel fusion. Due to the large overlap between RGB-D frames, we obtain 40 diverse TSDF inputs by randomly sampling different proportions of frames. The depth data of each frame in the synthetic dataset is complete, so we randomly remove about 94% of the pixels for each frame to make TSDF input diverse. Each method continues to use the model trained on Mattetport3D.

Experimental Details of ScanNet. In the evaluation of the real-world dataset ScanNet [2], we use the RGB-D frames provided by it to obtain the TSDF input with 2 cm resolution through voxel fusion. We randomly sample 3% of the RGB-D frames in the Scannet dataset for input, and then we use the model run on the Matterport3D dataset directly for Scannet TSDF input.

C. Evaluation Metrics

We compare all methods with the metrics of Chamfer Distance(d_{CD}) , Recall, and Precision. We ignore the real-world unobserved space for Chamfer Distance evaluation while using the whole predicted scene to calculate Recall and Precision. The S_1 and S_2 denote the predicted and GT point clouds. TP denotes the number of correctly predicted point clouds, FN denotes the number of point clouds that exist in GT but are not predicted, and FP denotes the number of incorrectly predicted point clouds. The matching rule of Recall and Precision is when the distance between two points is less than a predefined threshold τ, and they will be considered the same point. We set τ to be $\sqrt{3}$ times the voxel resolution, which is the length of the diagonal of a voxel.

$$\begin{aligned} d_{\mathrm{CD}}\left(S_1, S_2\right) = & \frac{1}{S_1} \sum_{x \in S_1} \min_{y \in S_2} \|x-y\|_2^2 \\ & + \frac{1}{S_2} \sum_{y \in S_2} \min_{x \in S_1} \|y-x\|_2^2 \end{aligned} \tag{1}$$

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

$$Precision = \frac{TP}{TP + FP} \tag{3}$$

For evaluation of the real-world scans, due to the different sizes of the scans, it is inappropriate to represent different scans with a uniform number of point clouds, so we take 1/4 of the number of vertices on the target mesh as the number of sampling points (N_{sp}), and we uniformly sample N_{sp} points on each of the target mesh and the output mesh, and then calculate metrics. For evaluation on synthetic data, we uniformly sample 30k points in the GT of synthetic data and output meshes for evaluation, as there are sparse vertices in some parts of synthetic data GT (e.g. a wall is described by only two triangles and four vertices).

D. Data Generation Details of Stage 2

The input scene of stage 2 is the predicted scene of stage 1. However, the real-world ground truth (GT) is incomplete, which may not be suitable as a training target in stage 2. To make the network focus on learning inaccurate areas in stage 2, the training target of stage 2 is the fusion of the stage 1 output scene and GT scene. The training blocks are all $128 \times 64 \times 64$ size cut from the scene with the stride of 40 in the x and y directions. The low-resolution supervision information during training is obtained by downsampling the high-resolution supervision information. We use mean-pooling to obtain the supervision of the low-resolution sparse depth TSDF voxel.

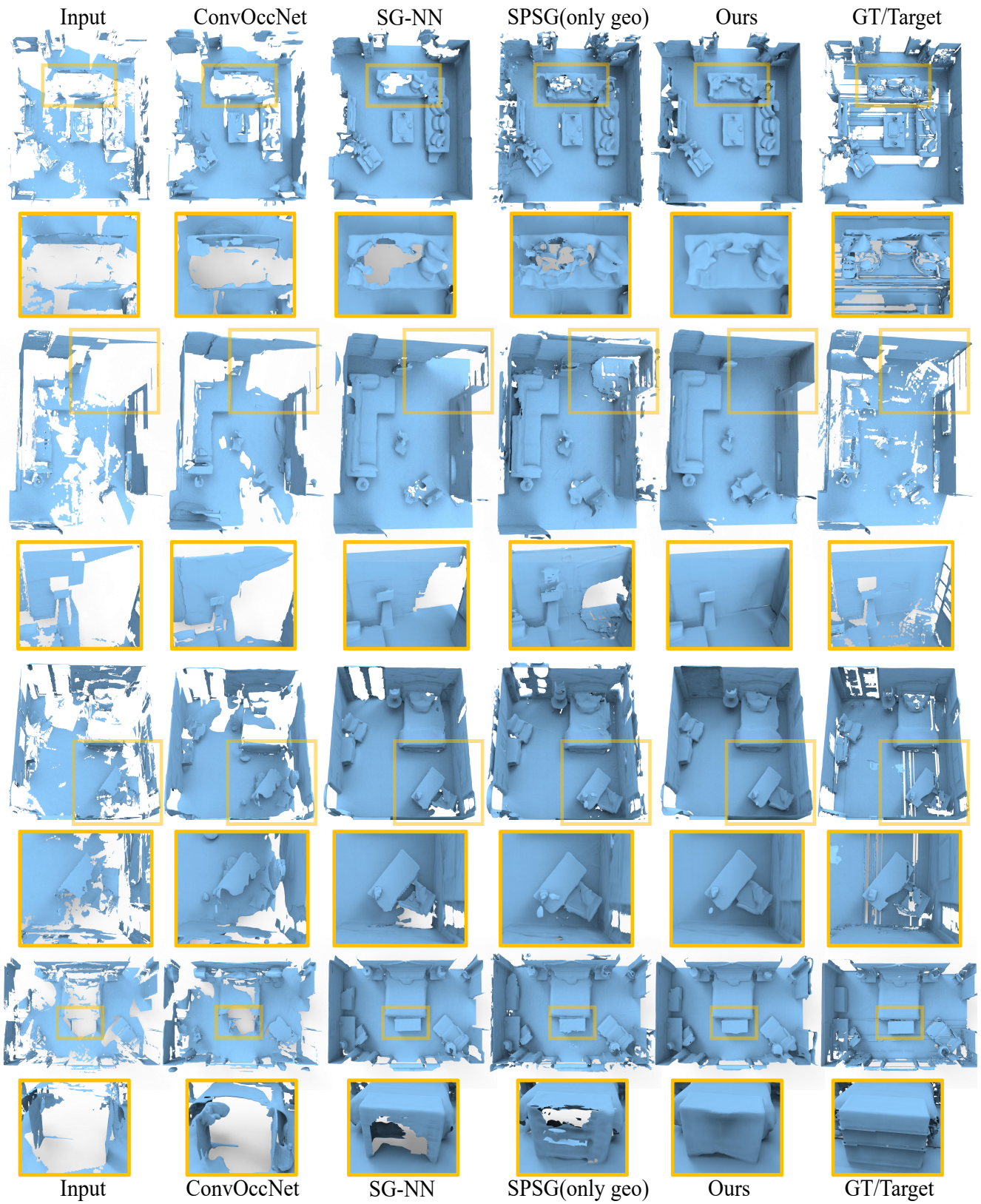


Figure 1: More qualitative comparison with state-of-the-art methods on Matterport3D.

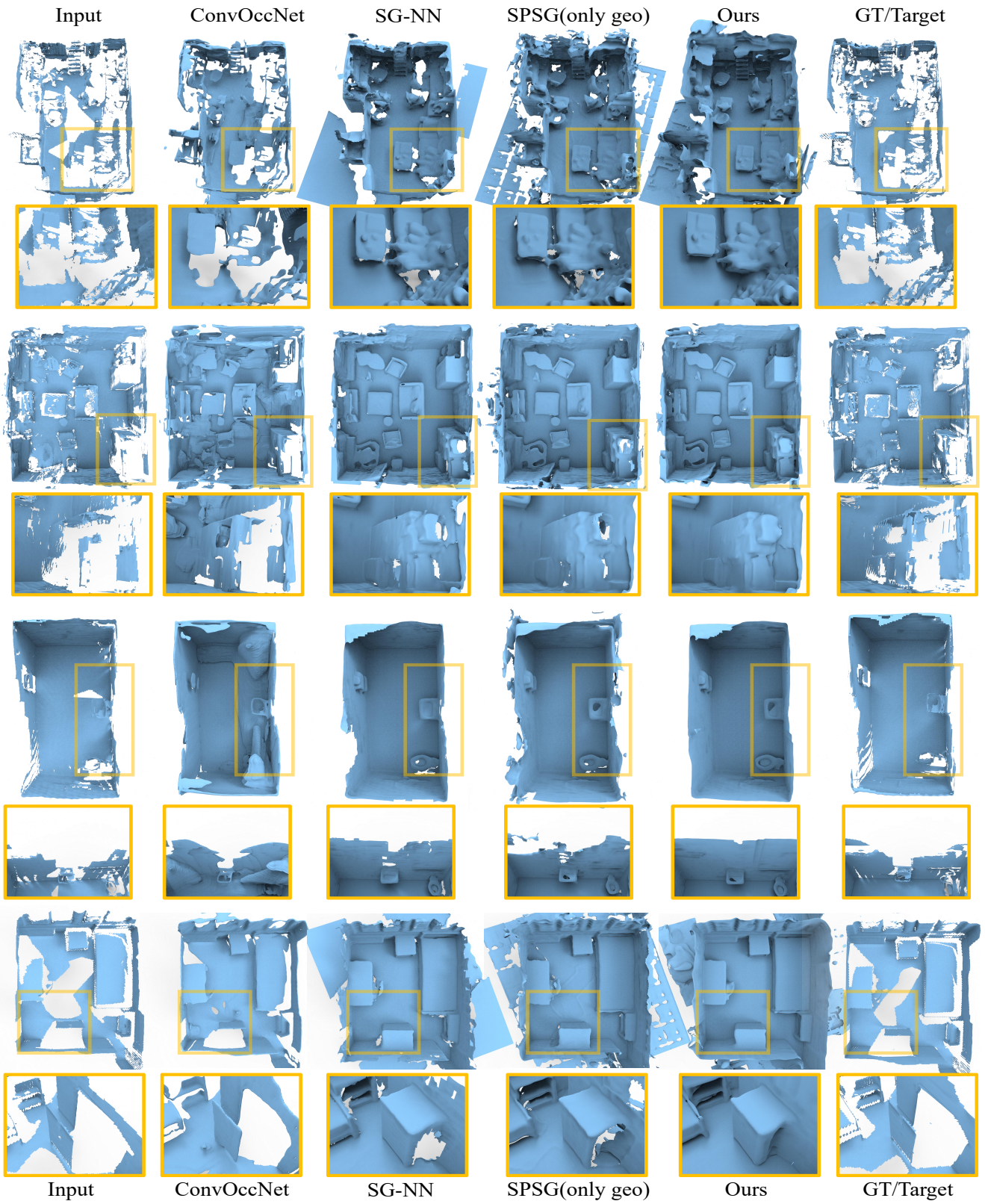
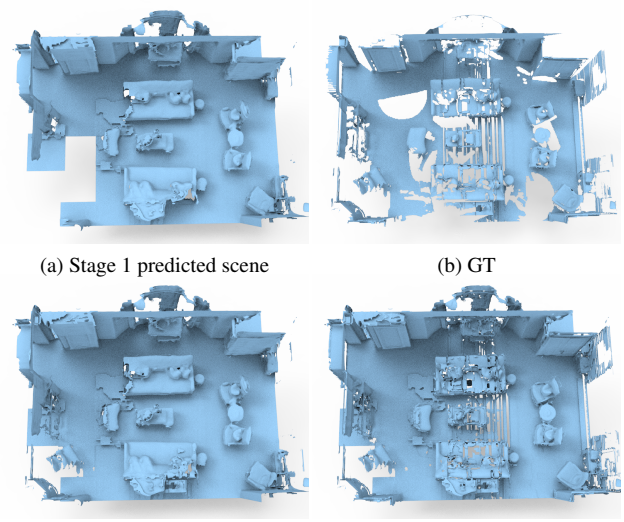


Figure 2: More qualitative comparison with state-of-the-art methods on ScanNet.



(a) Stage 1 predicted scene (b) GT

(c) Replace the predicted scene with GT (d) Replace GT with the predicted scene

Figure 3: The visual comparison of the first-stage output and second-stage target obtained in three different ways.

Three different ways to get training target in stage 2. Considering the sparse predicted and GT voxels of stage 1 may coincide in the same coordinates in fusion, we evaluate three different ways of fusion to obtain the target values in stage 2: 1) using the GT TSDF values directly, 2) replacing the predicted TSDF values with GT values, and 3) replacing the GT TSDF values with the predicted values. The visual comparisons are shown in Fig. 3.

The impact of training targets obtained in three ways on the results. In our evaluation, we find that it is difficult to achieve good completion results by directly using TSDF values of GT because real-world GT is incomplete. In the case of replacing the predicted TSDF values with GT values in fusion, there is no information to correct the output TSDF of stage 1, so it will greatly accumulate errors, and it is difficult to have further completion or correction. In the case of replacing the predicted TSDF values with GT values in fusion, we can get a good complement effect. The network can learn the precise geometry information of the target in the changed area, and maintain better consistency in the area that has not changed.

The obtained training target further influences the training process. Furthermore, we notice that it is difficult to obtain relatively good results by using only depth l_1 loss in the third case in stage 2 because the learning for the problem areas is more difficult. It is more about learning a modification and refinement process in stage 2 rather than just a process of completion in stage 1. But with all the loss items, we can make it learn enough information to handle the problem area in the output of stage 1 while minimizing the accumulation of errors.

E. Network Structure Details

Our DST-Net basically consists of four modules: geometry encoder, global structure extractor, region geometry generator, and local detail generator. The region geometry generator has been described in detail in our paper. Tab. 1 shows the details of the geometry encoder, global structure extractor (GSE), and local detail generator. The geometry encoder and local detail generator are similar to the work [3], and the GSE (global) is inspired by the PVT [5], which is used for 2D dense prediction. Linear parameters are given as (nf_in, nf_out), and convolution parameters are given as (nf_in, nf_out, kernel_size, stride, padding). Each convolutional layer is followed by batch normalization and a ReLU, except for the convolutional layer in front of the FullyConvolutionalNet and in front of the Attention.

Module	Layer and Parameter
geometry encoder(n_{in},n_{out})	SubmanifoldConvolution (n_{in},n_{out},3,2) SubmanifoldConvolution (n_{out},n_{out},3,2) SubmanifoldConvolution (n_{out},n_{out},3,2) Convolution(n_{out},n_{out},2,2)
GSE(global)	Conv3d(16,24,4,2,1) Attention(Linear(24,24), Linear(24,24), Linear(24,24)) Linear(24,24), GELU Linear(24,24) Conv3d(24,32,4,2,1) Attention(Linear(32,32), Linear(32,32), Linear(32,32)) Linear(32,32), GELU Linear(32,32) Conv3d(32,32,1,1) Cat ConvTranspose3d(64,32,4,2,1) Cat ConvTranspose3d(56,28,4,2,1) Conv3d(28,16,1,1) Conv3d(16,1,1,1)
GSE(local)	Conv3d(16,24,4,2,1) Conv3d(24,32,4,2,1) Conv3d(32,32,1,1) Cat ConvTranspose3d(64,32,4,2,1) Cat ConvTranspose3d(56,28,4,2,1) Conv3d(28,16,1,1) Conv3d(16,1,1,1)
local detail generator(n_{in},n_{out})	SubmanifoldConvolution(n_{in},n_{out},3,2) FullyConvolutionalNet (n_{out}, $3n_{out}$) Upsample(2) SubmanifoldConvolution($3n_{out}$, n_{out})

Table 1: Detailed network architectures of geometry encoder, global structure extractor (GSE), and local detail generator

References

[1] 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. 1

[2] 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. 1

[3] Angela Dai, Christian Diller, and Matthias Nießner. SG-NN: sparse generative neural networks for self-supervised scene completion of RGB-D scans. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 846–855. Computer Vision Foundation / IEEE, 2020. 4

[4] A. Handa, T. Whelan, J.B. McDonald, and A.J. Davison. A benchmark for RGB-D visual odometry, 3D reconstruction and SLAM. In *IEEE Intl. Conf. on Robotics and Automation, ICRA*, Hong Kong, China, May 2014. 1

[5] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 548–558. IEEE, 2021. 4