LT3SD: Latent Trees for 3D Scene Diffusion

Supplementary Material

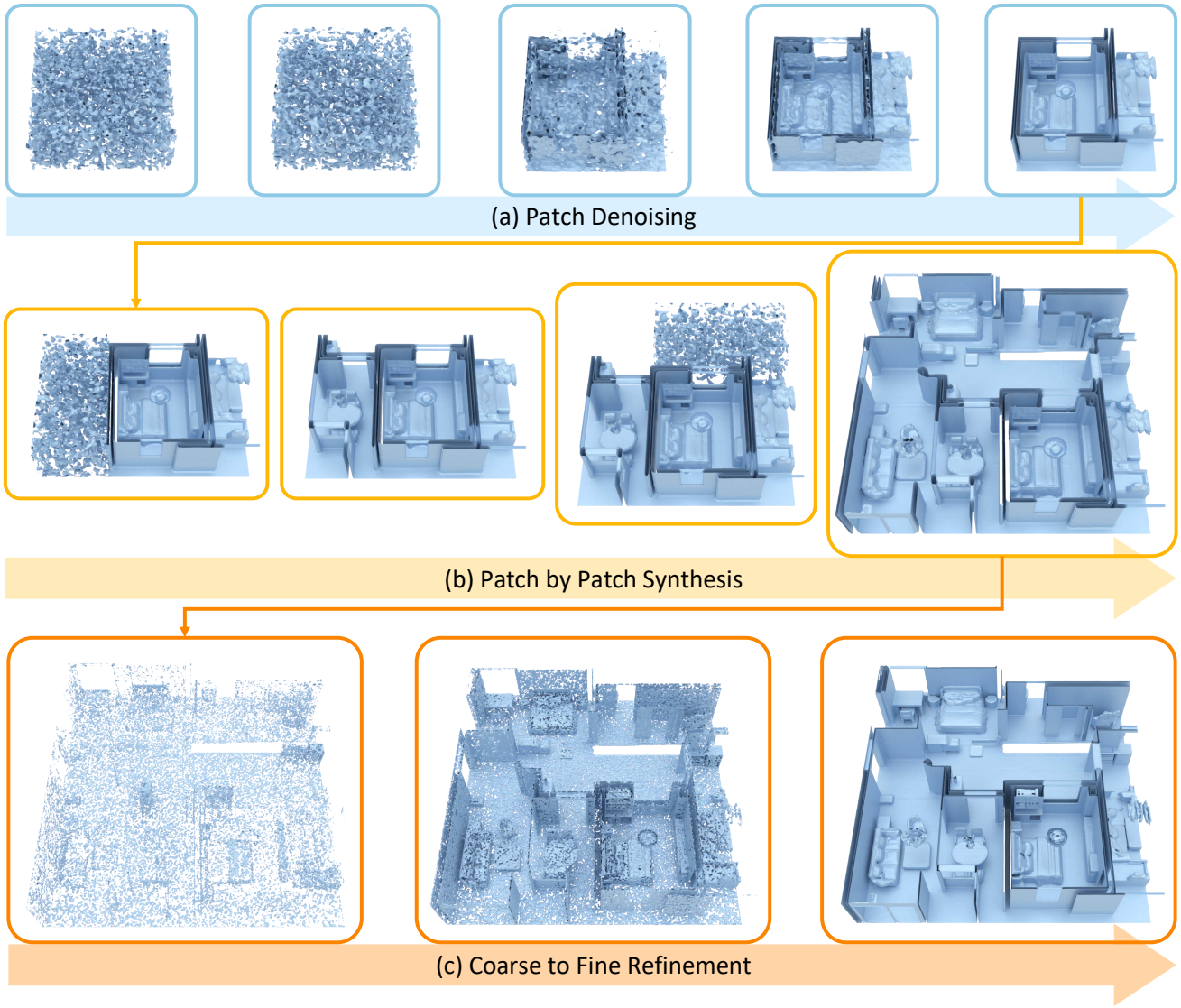

Figure 7. **Intermediate Visualization for 3D Scene Generation.** We visualize the DDIM [52] denoising process for (a) patches, (b) the patch-wise inpainting process on the coarse scene as described in Eq. (5), and (c) the coarse-to-fine refinement process as introduced in Eq. (6).

A. Additional Results

A.1. Intermediate Visualization

We visualize the full inference process of generating a 3D scene unconditionally with LT3SD in Fig. 7. Our method operates patch-wise and coarse-to-fine, as introduced in the main paper. First, starting from the random 3D Gaussian noise on the left of the first row, LT3SD gradually denoises it to the mesh at the end of the row. Then, at the second row, our method autoregressively extrapolates the unknown region in a patch-wise manner with an overlap size of one-half of the patch size, which empirically provides sufficient context and reduces seams, until the 3D scene of the specified spatial extent is complete. Finally, in the third row, conditioned at the coarse 3D scene, LT3SD generates the fine details also in a patch-wise manner with the

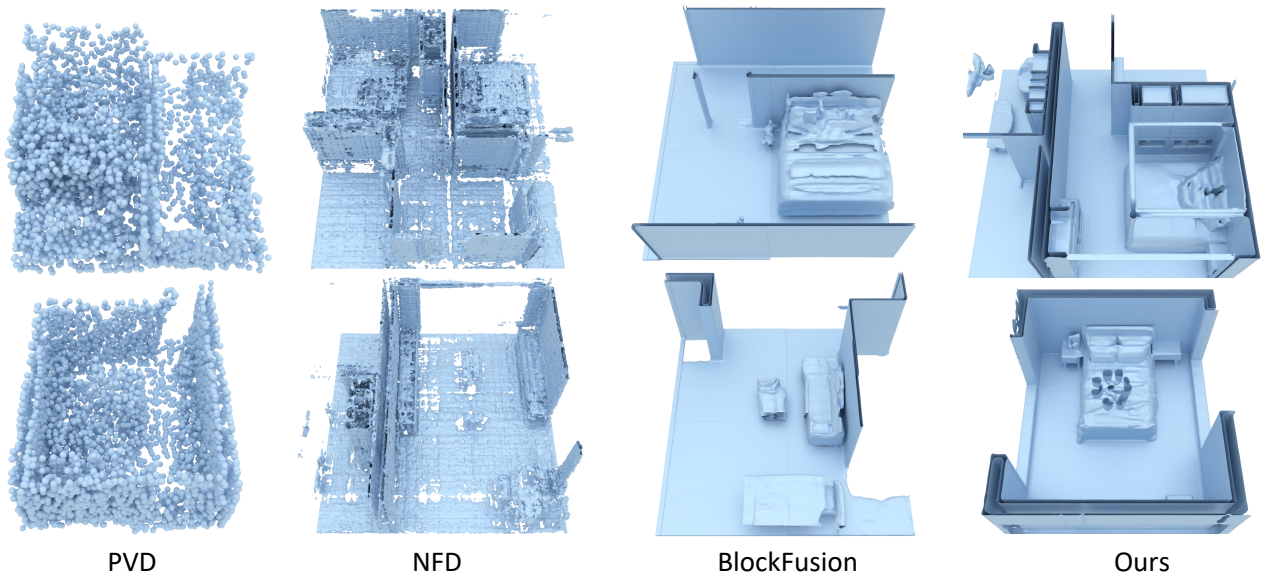

Figure 8. **Additional qualitative comparisons.** We compare unconditional 3D scene generation with state-of-art 3D diffusion methods PVD [62], NFD [49], and BlockFusion [57]. All methods were trained on houses from the 3D-FRONT dataset [21]. Our latent tree-based 3D scene diffusion approach synthesizes cleaner and more detailed surfaces with diverse furniture objects.

same overlap size. This process is implemented in a batch-wise manner as each patch is only dependent on the same patch of the previous timestep, *i.e.*, all patches at the same timestep are independent of each other. Comparing the coarse-level 3D scene and fine-level 3D scene, we can see that the coarse-level scene determines most of the structure, which ensures a plausible scene layout; the fine-level scene adds rich local detail while keeping the scene structure. The patch-wise and coarse-to-fine 3D scene synthesis enables 3D scene generation with both plausible scene layout and high-quality detail.

A.2. Additional Comparisons

We provide additional qualitative comparison results in Fig. 8 using the same model as in the main paper's Fig. 4.

A.3. Infinite 3D Scenes

Unlike retrieval-based methods (e.g., DiffuScene [53]), focusing on object layouts for single rooms (without walls/structures), our approach synthesizes large-scale scenes, including objects/walls/floors, while maintaining coherent structures at the room scale. LT3SD extends infinitely through outpainting, making it well-suited for open-world games, large-scale robotic training, and film asset creation–a step toward scalable indoor scene generation.

Here, we present more examples of the infinite 3D scene generation beyond Fig. 1 in the main paper. In Fig. 11, we show two more infinite 3D scenes with the resolution of $[4096, 2048, 128]$ and size of $90.3m \times 45.1m \times 2.8m$. Although trained on the house-level data of 3D-FRONT with only a few connected rooms in each sample, our method generates 3D scenes with diverse room structures, furniture layouts, and varying sizes. We generate infinite 3D scenes in the same manner, *i.e.*, patch-wise and coarse-to-fine. The supplementary video shows more examples of infinite 3D scenes with zoom-ins and intermediate visualization.

A.4. Additional Novelty Analysis Results

In Fig. 9, we provide additional novelty analysis results. The results further show that our method learns to generate novel 3D scenes with different furniture layouts and details. We highlight that our model is only trained on random patches of the 3D scenes with complex furniture layouts but without semantic information.

A.5. Outdoor 3D Scene Generation

In this section, we provide 3D outdoor scene generation visualization in Fig. 10. Unlike methods that assume a specific 3D scene structure, our approach employs a unified voxel-based representation for 3D scenes, allowing seamless extension beyond indoor scenes to diverse environments such as outdoor 3D scenes. As shown in Fig. 10, we train our models on the 3D city asset [1] in a patch-wise manner, as described in the main paper, and generate large 3D scenes with a resolution of $[1536, 1536, 512]$, where 512 is the vertical dimension.

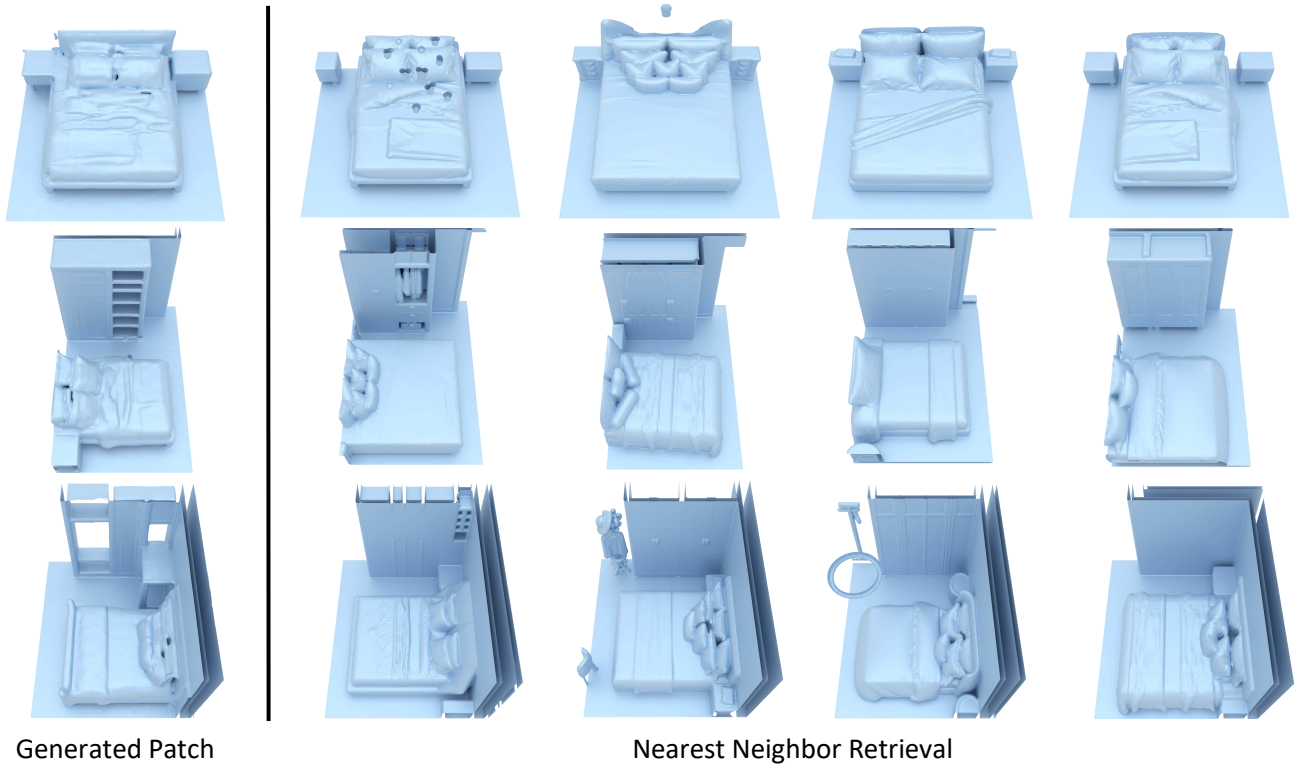

Figure 9. **Generation novelty analysis.** Our generated scene patches (left) are compared with the nearest neighbor's training patches retrieved by Chamfer distance.

Figure 10. **Outdoor 3D Scene Generation.**

B. Evaluation Metrics

For FID, we set the patch size to be $5.6m \times 5.6m \times 2.8m$ to consider both geometry quality and scene layout. For point clouds-based metrics, we split each generated and ground-truth patch into four smaller ones with the size of $2.8m \times 2.8m \times 2.8m$ to make sure complex structures are represented with a limited number of sampled points.

The MMD, COV, and 1-NNA metrics are formally defined as:

$$\text{MMD}(S_g, S_r) = \frac{1}{|S_r|} \sum_{Y \in S_r} \min_{X \in S_g} D(X, Y),$$

$$\text{COV}(S_g, S_r) = \frac{\left|\{\arg\min_{Y \in S_r} D(X, Y) \mid X \in S_g\}\right|}{|S_r|},$$

$$\text{1-NNA}(S_g, S_r) = \frac{\sum_{X \in S_g} \mathbb{1}[N_X \in S_g] + \sum_{Y \in S_r} \mathbb{1}[N_Y \in S_r]}{|S_g| + |S_r|},$$

where $D(\cdot, \cdot)$ represents the CD or EMD distance, S_g and S_r are the sets of generated point clouds and reference point clouds, and X and Y are samples from the generated set and reference set. $\mathbb{1}[.]$ is the indicator function. In the 1-NNA metric, N_X is a point cloud that is closest to X in both

Table 3. **Ablation over coarse-to-fine refinement**. Our coarse-to-fine scheme significantly increases the efficiency of large-scale 3D scene generation.

	Only Sequential Generation	Full Model (Ours)
Time	5 Hours	2 Hours

generated and reference dataset, *i.e.*,

$$N_X = \underset{K \in S_r \cup S_g}{\arg\min}\ D(X, K). \tag{8}$$

C. Ablation Study

C.1. Hierarchical Latent Representation

Here, we provide detailed statistics for Tab. 2 in the main paper: We compare the performance of our latent tree representation against the latent cascaded representation for encoding TUDF voxel grids, using a downsampling/upsampling factor of 4. The latent tree representation requires 20 hours of training and consumes 33KB per room sample for storage, while the latent cascaded representation takes 24 hours and 41KB per sample. After training, we evaluate reconstruction performance on the test set by calculating the l_2 error relative to the ground-truth TUDF voxel grids.

C.2. Coarse-to-Fine Refinement

For large-scale 3D generation, generating patches autoregressively can be highly time-consuming. In the 3D scene depicted in Fig. 4, our full model initially generates the scene patch-by-patch, as outlined in Eq. (5), focusing on capturing the coarse structure. We then apply a coarse-to-fine refinement process, adding high-frequency details using the parallel algorithm, as described in Eq. (6). Fig. 4 shows that our highest resolution synthesis notably improves details in object structures such as chairs and pillows, compared to the coarser resolution shown on the left. In the ablation study (Tab. 3), we replaced the parallel algorithm with the autoregressive one Eq. (5) to generate the finer levels. The results show that this approach uses $2.5\times$ inference time, compared to our full model. This demonstrates that the batch-wise coarse-to-fine approach significantly reduces overall inference time for large-scale 3D scene generation, in contrast to the purely patch-by-patch method. Furthermore, our approach can be accelerated using multi-GPU setups, a capability that previous works [35, 57], which employ naive autoregressive patch-wise outpainting, do not support.

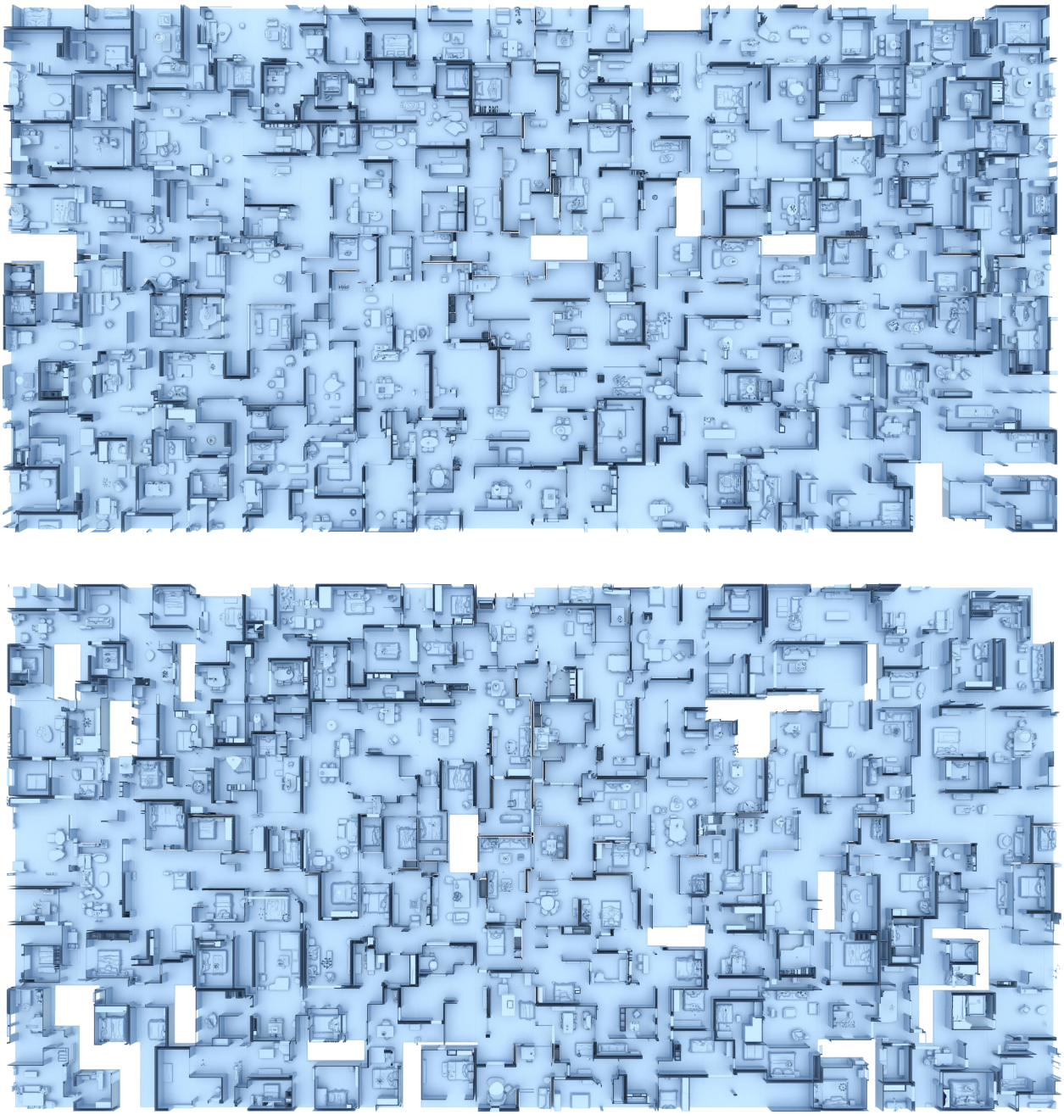

Figure 11. **More Infinite 3D Scene Generation Results.**