# Artist-Created Mesh Generation from Raw Observation

Supplementary Material

## A. Dataset

Our initial dataset for training and evaluation is constructed from ShapeNet for its high data quality within limited categories. We intend to use it as a clean starting point, then progressively fuse meshes from the other repositories. We follow [5] to filter out meshes with more than 800 faces and manually discarded low quality meshes by checking geometric integrity, annotation completeness, and category balance. This produces roughly 5k meshes.

The point clouds generated thus far are high-quality, free from corruption and noise. However, to simulate the challenges posed by real-world lidar data—namely, occlusion and measurement noise—we introduce a synthetic corruption procedure. Specifically, we randomly crop a portion of the point cloud to mimic occlusion and add Gaussian noise to the remaining points to simulate sensor inaccuracies. Our random cropping consists of two strategies: 1) randomly select $20\%$ to crop, and 2) crop $20\%$ of the points around a center region. While this procedure approximates real-world conditions, it may not fully capture the true distribution of lidar artifacts, suggesting a direction for future improvement, such as using a visibility mask to crop out points. After obtaining the point cloud, we extract its corresponding atlas, i.e., position map and normal map, for fine-tuning the Latent Diffusion.

Our dataset contains approximately 5.5k samples, split into $90\%$ for training and $10\%$ for testing, following the convention in [6]. Each sample consists of a clean point cloud, a synthetically corrupted counterpart, and their corresponding atlases. Representative examples are shown in Fig. 3.

## B. Implementation Details

we set the total number of 3D points per atlas to $N = 16384 = 128 \times 128$. This choice of parameters balances computational efficiency for O. T. with the input resolution required for high-quality mesh generation. For point clouds with fewer than $N$ points, we pad with holes (zero positions and normals). For point cloud with more than $N$ points, we apply Farthest Point Sampling (FPS) [15] to downsample.

The inpaint U-Net $\mathcal{U}$ is trained for 100,000 iterations with a learning rate of $1 \times 10^{-5}$, using a batch size of 4 on a single NVIDIA L40S GPU. We use Adam optimizer to train the diffusion model. The total training time is approximately 24 hours. We choose 20 denoising steps to balance the tradeoff between efficiency and generation quality during inference.

## C. Limitations And Future Works

### C.1. Additional Discussions

**Discussion on generated meshes**: The qualitative results in Fig. 2 show that, given noisy point cloud input, our method is able to capture certain topological and geometric structures that MeshAnything V2 fails to recover, demonstrating the effectiveness of our approach. However, our model also exhibits limitations in preserving fine-grained geometry, such as the legs of a table. This is primarily due to noise in the reconstructed point clouds, which adversely affects the performance of the downstream mesh generation model.

**Discussion on generated atlas maps and point clouds**: The comparison between the generated and ground-truth atlas maps shows that our fine-tuned diffusion model can produce visually similar results when conditioned on corrupted inputs, suggesting that it captures key features of the underlying atlas map distribution. Furthermore, the reconstructed point cloud demonstrates the model's ability to recover geometric structure through inpainting, such as filling in missing regions like the hole in a plate.

Despite promising results, our model exhibits notable limitations. A key issue is that the reconstructed point clouds reveal significant noise in the generated atlases compared to the ground truth. This is likely due to our conditioning strategy, where we simply concatenate the input with noise; the denoising process may be insufficient to fully recover clean data, leading to noisy point cloud reconstructions. Such noisy point cloud reconstructions pose challenges for MeshAnything V2 [7] in capturing fine-grained geometric details and introduce ambiguity in surface prediction. As a result, the MeshAnything V2 pipeline tends to prune these uncertain vertices, leading to lower vertex and face counts in the final mesh.

The findings above suggest several directions for improving our methodology. One is to explore more effective conditioning mechanisms that enable the model to better preserve fine-grained structures. Additionally, incorporating a reconstruction loss—alongside the standard diffusion loss, as proposed in [30]—could further encourage the generation of more accurate atlas maps.

### C.2. Future Works

Our initial results validate the soundness of the proposed approach and point to several directions for future work. First, we plan to expand the dataset to better reflect diverse real-world distributions. We also aim to explore alternative corruption methods, such as estimating point cloud visibility

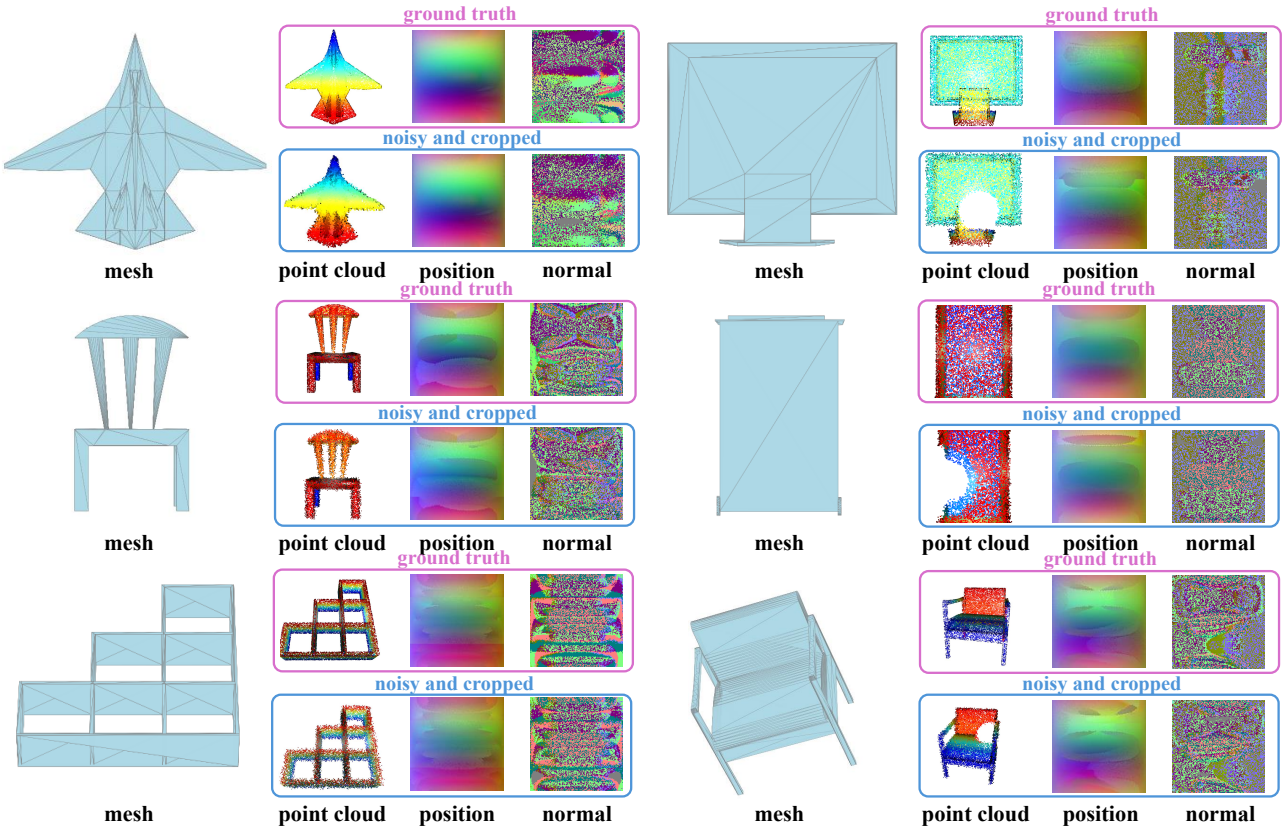


Figure 3. **Overview of Dataset**. Each data point from our processed ShapeNet includes a ground-truth mesh, ground-truth point cloud, ground-truth atlas, and their noisy and cropped counterparts. The dataset comprises approximately 5.5k samples.

maps to more accurately simulate lidar scan artifacts. To improve atlas quality, we will investigate more effective conditioning strategies and incorporate additional loss terms. Finally, we propose bypassing the point cloud reconstruction step by directly encoding the atlas and fine-tuning the mesh generation pipeline. This would mitigate error propagation introduced by the optimal transport step during point cloud reconstruction.