Appendix A: Implementation Details

Textured Shape Generation

This section introduces the implementation details for textured shape generation. For data preparation, we use the same procedure in Section 4.1 to repair all the meshes in ShapeNet to obtain dense on-surface points and off-surface query points and their corresponding SDF values. Then, we reproject these points onto the surface of the original meshes and extract their corresponding RGB values to obtain colored input point clouds. We build octrees with depth 8 (resolution 256^3) with colored point clouds as the input of VAE. The VAE has two separate decoders, and the latent feature dimension is set to 6 with 3 channels for geometry and another 3 channels for color. Then, we train a two-stage OctFusion. Finally, we train an additional octree-based texture diffusion model to generate color latent code based on the geometry latent code to attain a textured 3D mesh followed by the decoder of extended VAE.

Model Architecture

Octree-based VAE

The network architecture of the octree-based latent Variational Autoencoder (VAE), as depicted in Figure 17, is constructed via dual-octree graph convolution network. For two-stage OctFusion model, the VAE has three hierarchical levels, corresponding to octree depths of 8, 7, and 6, with corresponding resolutions of 256^3, 128^3, 64^3. The feature dimensions are set to 24, 32 and 32 respectively.

OctFusion

We present the unified U-Net architecture of OctFusion in Fig. 18. The first stage, denoted as $\mathcal{F}_1$, is designed for generating the splitting signals, using convolutional neural network with residual connection and self-attention block. The U-Net in $\mathcal{F}_1$ is composed of three levels: 16^3, 8^3, 4^3, each associated with model channels of 64, 128, and 256, respectively. For two-stage OctFusion model, the second stage $\mathcal{F}_2$ model is used for predicting clean latent features on each octree leaf node. The U-Net is constructed via dual octree graph convolution and has two levels 64^3, 32^3. The corresponding model channels are 128 and 256. At the bottom of $\mathcal{F}_2$ U-Net, the features are downsampled to 16^3 and fed into $\mathcal{F}_1$.

The deeper OctFusion (such as three-stage) is shown in Fig. 19. The $\mathcal{F}_1$ and $\mathcal{F}_2$ models is used for generating splitting signals and have the same network architecture mentioned above. The $\mathcal{F}_3$ predicts the clean latent code on the octree generated by $\mathcal{F}_1$ and $\mathcal{F}_2$. The U-Net of $\mathcal{F}_3$ has two levels 256^3 and 128^3, with associated model channels of 64 and 128. We also present the network architecture of octree-based texture diffusion model in Fig. 20, which is used for generating color latent codes for existing untextured 3D shape. Our octree-based texture diffusion model has the same architecture as $\mathcal{F}_3$ but is not unified with lower stages.

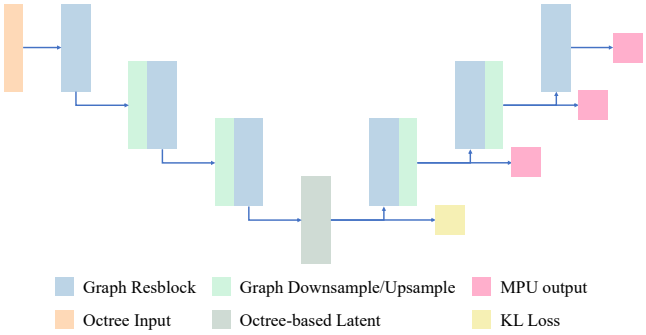

Figure 17: *The network architecture of Octree-based VAE.*

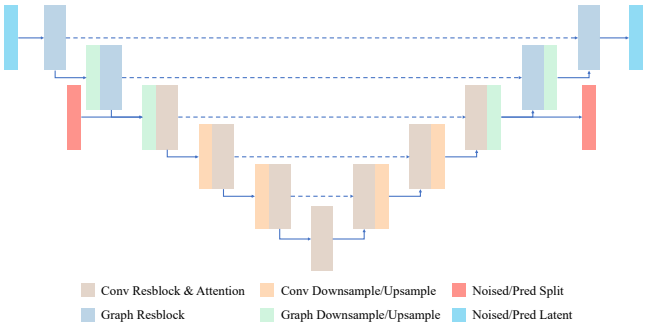

Figure 18: *The network architecture of OctFusion U-Net.*

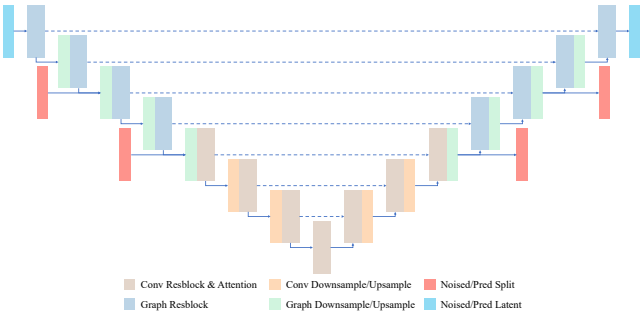

Figure 19: *The network architecture of deeper OctFusion U-Net.*

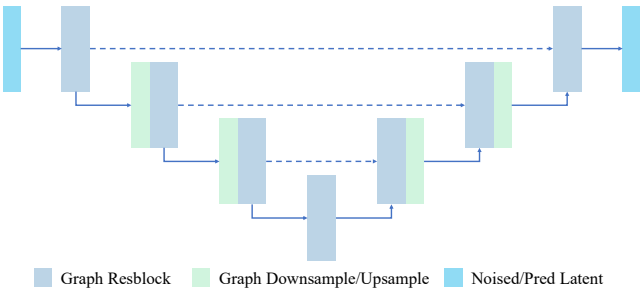

Figure 20: *The network architecture of OctFusion U-Net for color generation.*

Appendix B: Metric Definition

Distance

We begin by sampling points from the surfaces of both the generated mesh and the reference mesh in dataset, resulting in the point clouds denoted as S_g and S_r, respectively. Distance between two point clouds can be evaluated by Chamfer Distance(CD) and Earth

Mover's Distance (EMD). Chamfer Distance is a symmetric measure that calculates the average distance from points in S_g to the nearest points in S_r, and vice versa. The formula for Chamfer Distance is given by:

$$\mathrm{CD}(S_g, S_r) = \sum_{x \in S_g} \min_{y \in S_r} ||x-y||_2^2 + \sum_{y \in S_r} \min_{x \in S_g} ||x-y||_2^2. \quad (6)$$

where $d(x,y)$ is the Euclidean distance between X and Y. Earth Mover's Distance can be treated of as the minimum transportation from one point cloud into another. The formula of Earth Mover's Distance (EMD) is defined as:

$$\mathrm{EMD}(S_g, S_r) = \min_{\phi: S_g \to S_r} \sum_{X \in S_g} \|X - \phi(X)\|_2 \quad (7)$$

where ϕ is a bijection.

Coverage (COV)

Coverage is calculated as the fraction of point clouds in the reference set that are matched to at least one point cloud in the generated set. For each point cloud in the generated set, its near neighbor in the reference set is marked as a match:

$$COV(S_g, S_r) = \frac{|\{\mathrm{argmin}_{Y \in S_r} D(X,Y) | X \in S_g\}|}{|S_r|} \quad (8)$$

where $D(\cdot,\cdot)$ can be either CD or EMD. A high coverage score indicated that most of reference set is roughly represented within generated set.

Minimum Matching Distance(MMD)

Minimum Matching Distance is proposed to complement coverage as a metric that measures quality. For each point cloud in the reference set, the distance to its nearest neighbor in the generated set is computed and averaged:

$$\mathrm{MMD}(S_g, S_r) = \frac{1}{|S_g|} \sum_{Y \in S_r} \min_{X \in S_g} D(X,Y) \quad (9)$$

where $D(\cdot,\cdot)$ can be either CD or EMD. Since MMD relies directly on the distances of the matching, it correlates well with how faithful (with respect to the reference set) the elements of generated set are.

1-NNA

The 1-Nearest Neighbor Assignment (1-NNA) metric evaluates the classification accuracy when employing the nearest neighbor criterion under distance measure D to indicate whether a point cloud is synthetic or not. Ideally, if the generated point cloud closely mirrors the distribution of the reference set, the classification accuracy should be around 50%. The formula for 1-NNA is defined as follows.

$$\text{1-NNA}(S_g, S_r) = \frac{\sum_{X \in S_g} \mathbb{1}[N_x \in S_g] + \sum_{Y \in S_r} \mathbb{1}[N_y \in S_r]}{|S_g| + |S_r|} \quad (10)$$

where N_X is the closest point cloud to X under distance metric $D(\cdot,\cdot)$, and $\mathbb{1}[\cdot]$ is the indicator function.

shading-image-based FID

shading-image-based FID is a more robust measure for evaluating both the quality and diversity of generated shapes. To compute the FID metric, each generated shape is rendered from 20 uniformly distributed viewpoints around the shape. These rendered shading images are then used to calculate the FID scores on the rendered image set of the original training dataset. The formula for FID is defined as follows.

$$\mathrm{FID} = \frac{1}{20} \sum_{i=1}^{20} \|\mu_g^i - \mu_r^i\|^2 + \mathrm{Tr}(\Sigma_g^i + \Sigma_r^i - 2(\Sigma_g^i \Sigma_r^i)^{1/2}) \quad (11)$$

where μ^i and Σ^i denote the mean and covariance of the i-th view's shading images.

Appendix C: More Results on ShapeNet

We present more unconditional generation results on ShapeNet category `chair`, `table`, `airplane`, `car`, and `rifle` in the following pages. These results demonstrate the quality and diversity of our proposed OctFusion.

Figure 21: *More generative results on airplane*

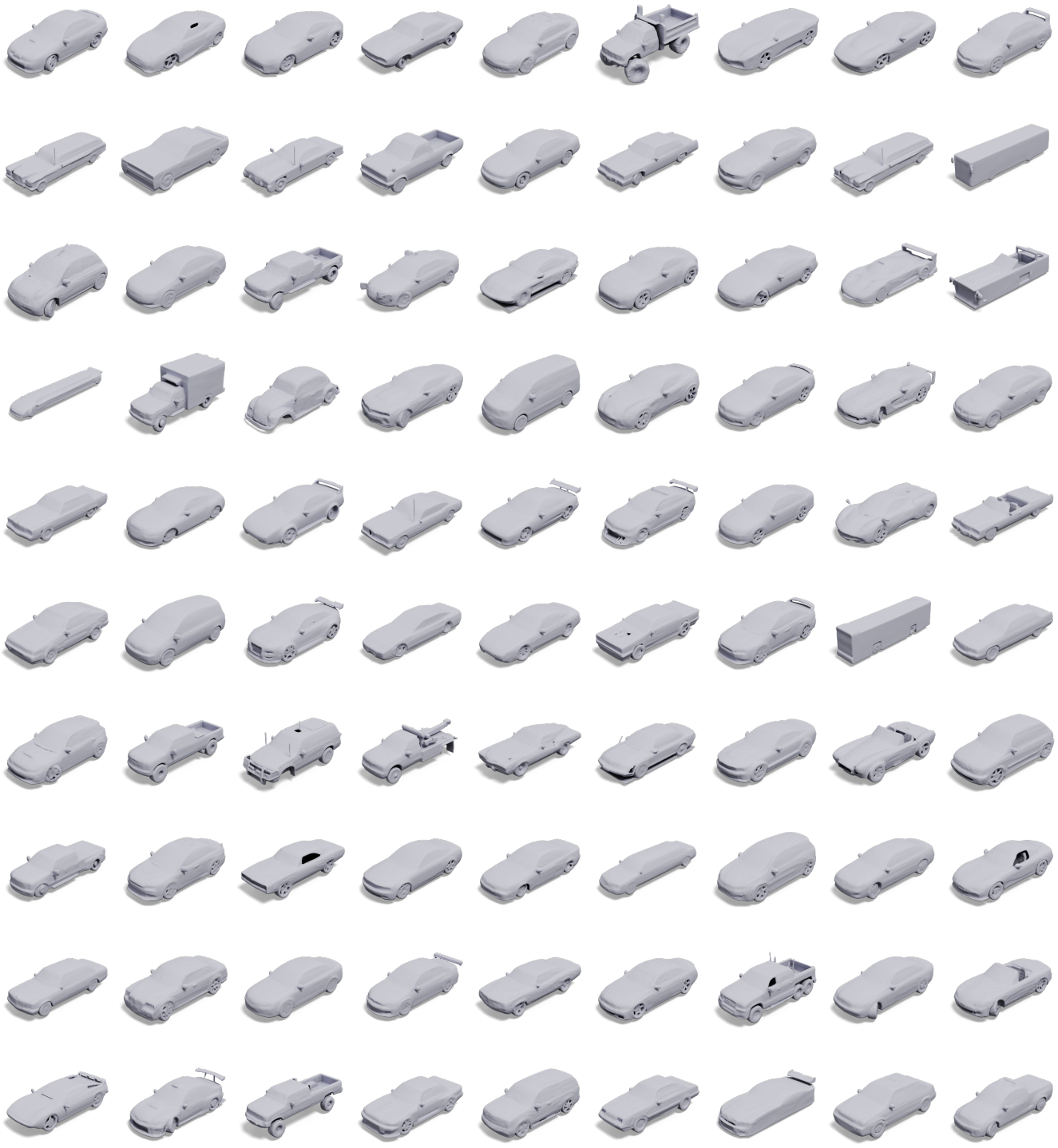

Figure 22: *More generative results on car*

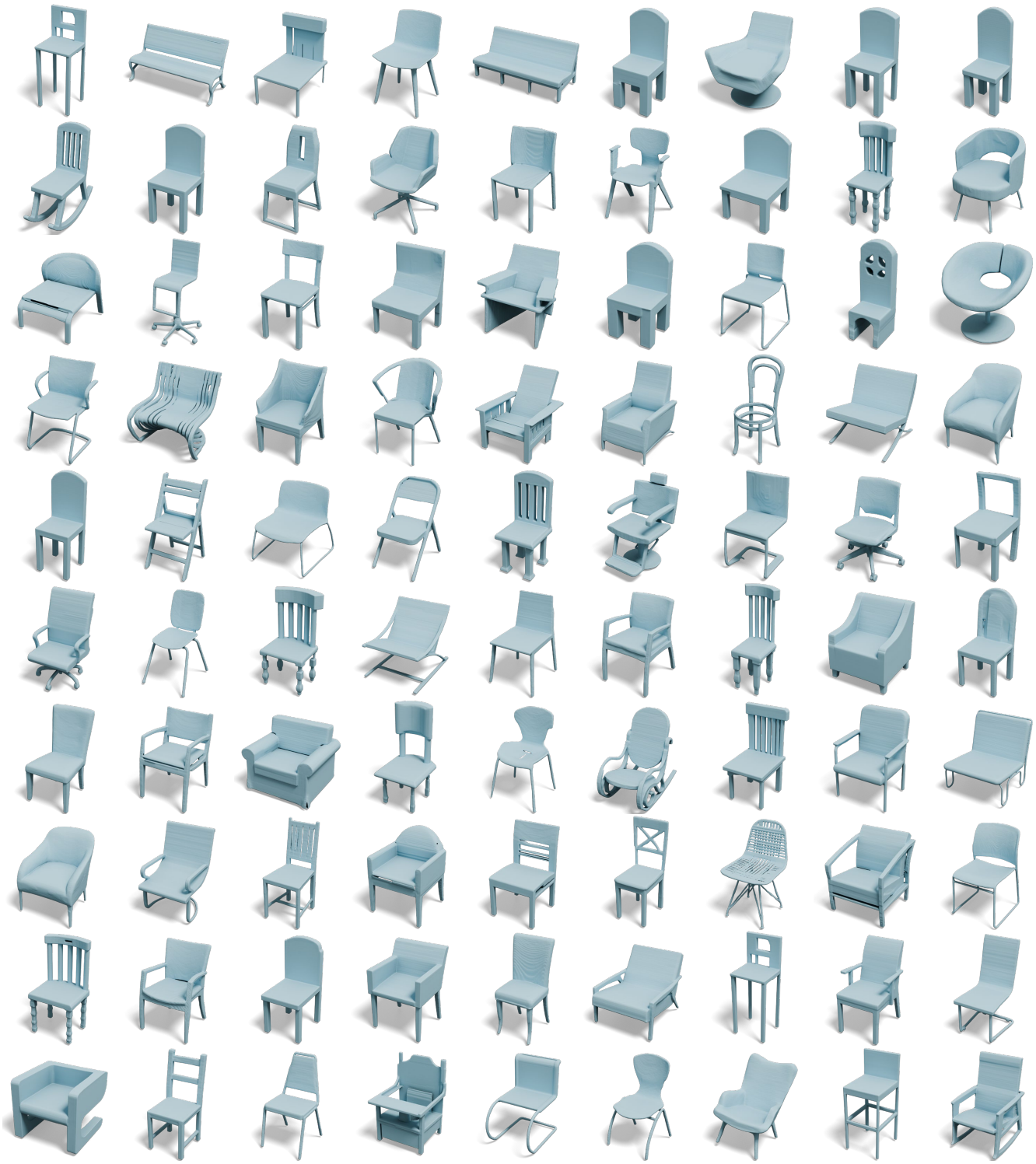

Figure 23: *More generative results on chair*

Figure 24: *More generative results on rifle*

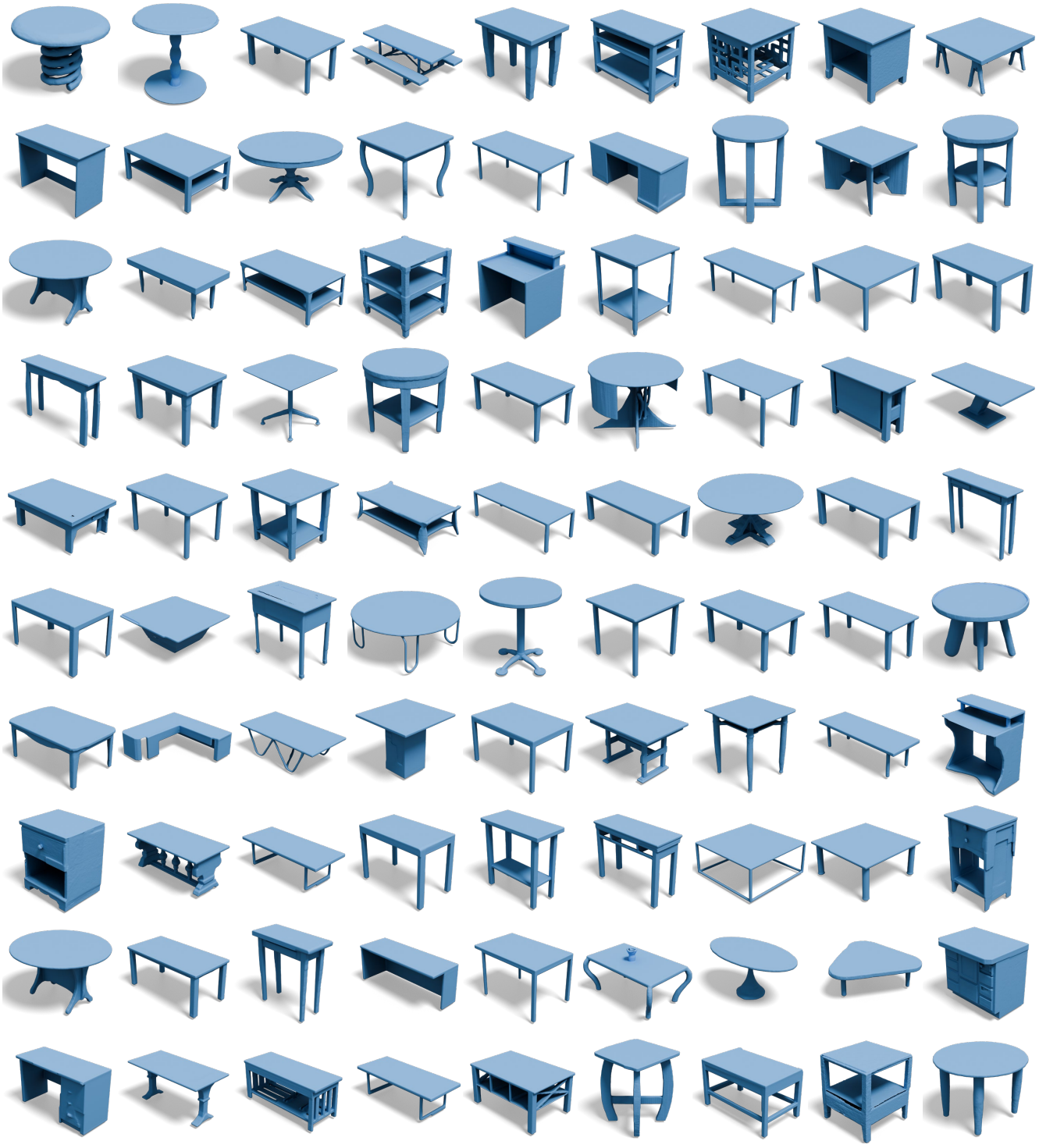

Figure 25: *More generative results on table*