Supplementary Material:
Diffusion-Based Signed Distance Fields for 3D Shape Generation

Jaehyeok Shim, Changwoo Kang and Kyungdon Joo*
Artificial Intelligence Graduate School, UNIST
{jh.shim,kangchangwoo,kyungdon}@unist.ac.kr
https://kitsunetic.github.io/sdf-diffusion

Overview

This supplementary material contains additional information that could not be included in the main paper due to space limitations. In Sec. 1, we describe more implementation details of our method. Concretely, noise scheduling for continuous time is described in Sec. 1.1. The architecture details and hyperparameters are listed in Sec. 1.2. Lastly, a detailed procedure for metric computation (including data processing) is described in Sec. 1.3.

In Sec. 2, we provide diverse qualitative experiment results. Specifically, in Sec. 2.1, we qualitatively compare reconstructed 3D shapes in the mesh domain (reconstruction from point-cloud-based generative model and reconstruction from our `SDF-Diffusion`). In addition, we utilize another dataset (especially, ShapeNet version 2) and show 3D shape generation results on this dataset, which are available in Sec. 2.2. Various 3D shape generations from our `SDF-Diffusion` in voxel resolution of 32^3, 64^3, and 128^3 are available in Sec. 2.3.

1. Implementation Details

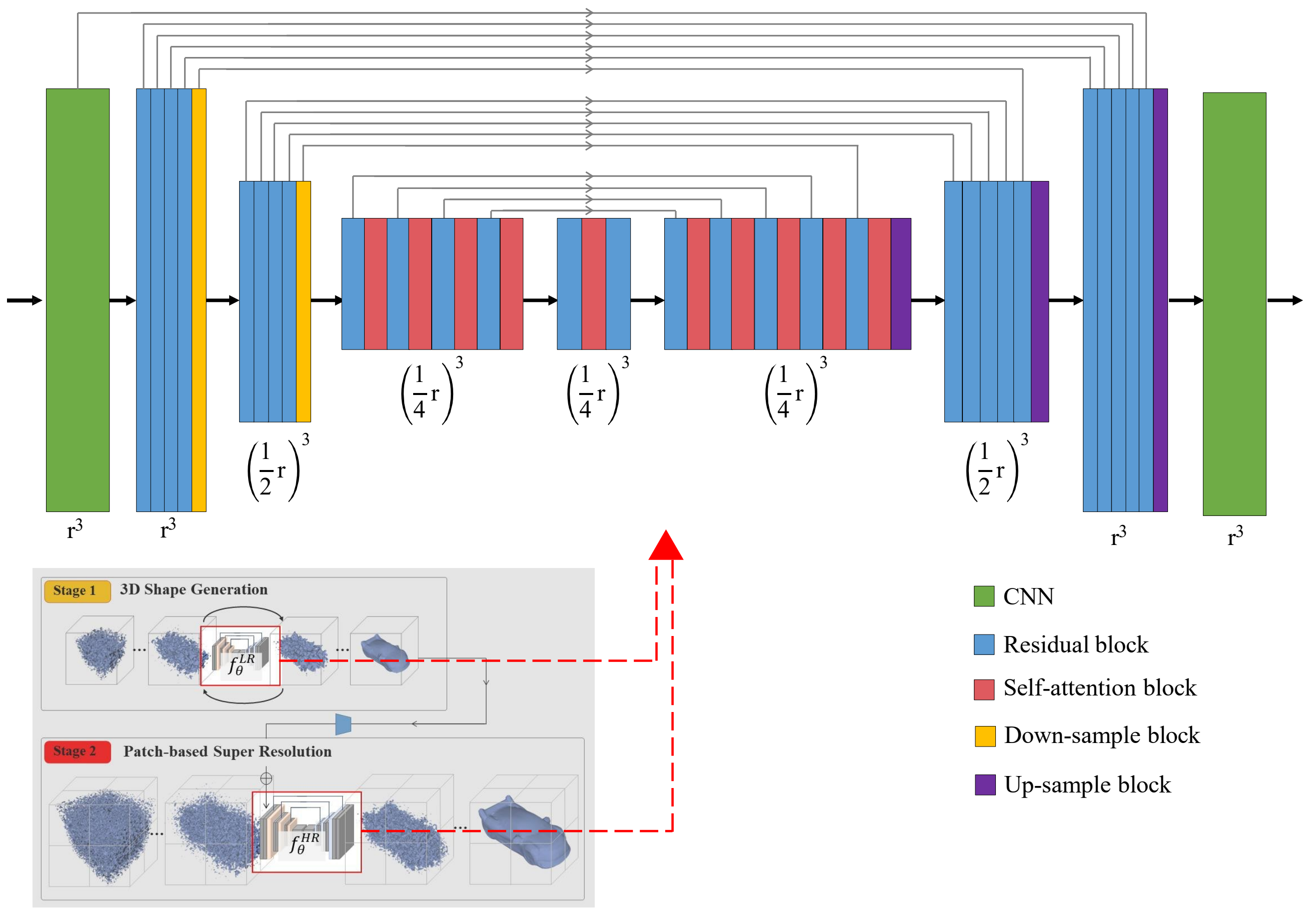
1.1. Noise scheduling for continuous time

The conventional denoising diffusion models (DDM) [1] are based on discrete time steps. Instead, following SR3 [4], we extend our model to learn continuous time steps. $\bar{\alpha}_t$ is deformed by uniformly sampled value between $\bar{\alpha}_t$ and $\bar{\alpha}_{t-1}$ so that t is sampled from continuous distribution $\mathcal{U}(0, T)$. Concretely, instead of using $\bar{\alpha}_t$ directly, $\mathcal{U}(\bar{\alpha}_{t-1}, \bar{\alpha}_t)$ is given to the DDM process and network, especially during the training phase. In addition, our DDM implementation follows SR3 with a modification. Following SR3 implementation, its mini-batch uses shared time value, so the noisy data from the forward process of DDM have the same noise strength, it can bother the model to learn about various noise strengths. We modify it so that all data in a mini-batch can have different time values. By doing so, `SDF-Diffusion` can learn continuous time steps and apply flexible inference time steps.

1.2. Architectural Details

Here, we describe the architectural details of `SDF-Diffusion`. In Fig. 1, we visualize the detailed network architecture of `SDF-Diffusion`. Our network architecture is based on a modified version of U-shaped network similar to SR3 [4]. We replace every 2D CNN layer with a 3D CNN so that it can receive 3D voxel-shaped data and decrease the depth of the network from 4 to 3. The network consists of residual blocks and self-attention blocks [5]. The first CNN layer converts the input data channel (1 channel for SDF) into base channels. Then, the down-sample block decreases the voxel resolution in half through a CNN layer with stride. Similarly, the up-sample block doubles the voxel resolution with nearest neighbor interpolation and a CNN layer. The features extracted from the residual blocks and the down-sample blocks on the left side of the network are densely connected to corresponding residual blocks on the right side. In addition, time and category conditions are transferred to every residual block. Table 1 lists the detailed hyperparameters we used. The single-category and multi-category networks have slightly different model sizes due to the presence of a layer that receives the category condition.

*Corresponding author.



r^3
r^3
$\left(\frac{1}{2}r\right)^3$
$\left(\frac{1}{4}r\right)^3$
$\left(\frac{1}{4}r\right)^3$
$\left(\frac{1}{4}r\right)^3$
$\left(\frac{1}{2}r\right)^3$
r^3
r^3
Stage 1
3D Shape Generation
f_θ^{LR}
Stage 2
Patch-based Super Resolution
f_θ^{HR}
CNN
Residual block
Self-attention block
Down-sample block
Up-sample block

Figure 1. **Illustration of our `SDF-Diffusion` network architecture.** Similar network structure is shared in both the first and second stage models. r^3 means the voxel size.

Table 1. **Hyperparameters of `SDF-Diffusion`**. Note that the number of epochs of the single-category super-resolution model for voxel resolution of 128^3 is empty because we have tested only 64^3 resolution on the single-category case.

	Shape generation (32^3)	Super-resolution ($32^3 \rightarrow 64^3$)	Super-resolution ($64^3 \rightarrow 128^3$)
Diffusion steps	$1,000$	$1,000$	$1,000$
Noise schedule	linear	linear	linear
Model size (single-category)	66.95M	66.95M	-
Model size (multi-category)	67.05M	67.05M	40.41M
Base channels	64	64	64
UNet depth	3	3	3
Number of residual blocks	4	4	2
Channels multiple	$1, 2, 4$	$1, 2, 4$	$1, 2, 4$
Input resolution (train / inference)	32 / 32	32 / 64	32 / 128
Attention resolution	8	8	No
Dropout	0.1	0.1	0.1
Epochs (single-category)	$10,000$	$10,000$	-
Epochs (multi-category)	$1,000$	450	150
Learning rate	0.0001	0.0001	0.0001

1.3. Details of Metric Computation Procedure

We use three metrics (*i.e.*, MMD, COV, and 1-NNA) based on both Chamfer Distance(CD) and the Earth Mover's Distance (EMD) as the main evaluation metrics.

The two distance metrics are defined as:

$$\mathrm{CD}(X,Y) = \sum_{x\in X} \min_{y\in Y} \|x-y\|_2^2 + \sum_{y\in Y} \min_{x\in X} \|x-y\|_2^2, \tag{1}$$

$$\mathrm{EMD}(X,Y) = \min_{\phi:X\to Y} \sum_{x\in X} \|x-\phi(x)\|_2, \tag{2}$$

where both X and Y are point clouds, and ϕ is a bijection between two point clouds. The main metrics compare the distributional similarity between sets of point clouds. Thus, we need to sample 3D point clouds from the generated meshes by `SDF-Diffusion`.

Specifically, given the trained `SDF-Diffusion`, we first sample 3D shapes with the same amount for each category in the validation set. We then randomly sample $2,048$ points from each generated mesh. In addition, the point cloud of each object is shifted so that all objects are uniformly zero-centered before computing metrics similar to [6]. By removing the arbitrary offset of the generated samples, we can conduct a fair comparison with only the generated shape.

2. Additional Experiments

2.1. Qualitative Comparison with Reconstructed Meshes

In the manuscript, we compared the qualitative results with other approaches in each domain (point clouds for the other methods and mesh for ours), as shown in Table 1 and Table 2 of the main paper. Here, we additionally compare the qualitative results on the mesh domain, which is a continuous surface and suitable for real-world applications. To this end, we compare with reconstructed meshes from the output of point voxel diffusion (PVD) [7] with shape as points (SAP) [3]. SAP consists of two modules: a neural network that estimates point normals from point clouds, and a differentiable Poisson solver that reconstructs meshes from points and their normals through the marching cube algorithm [2]. The multi-category version of PVD* is used, and SAP is the pre-trained version that the authors have offered.

Figure 2 shows the reconstructed meshes for both PVD* and ours. The surfaces of reconstructed meshes of PVD* are often uneven and rough because of sparse point clouds and the limitation of point normal estimation quality. Concretely, thin and sharp object (chair legs in row 4 column 3) is distorted and has uneven thickness. In addition, concave object (cabinet in row 3 column 3) sometimes collapses because it fails to determine the surface boundary between two close planes. On the other hand, meshes generated from `SDF-Diffusion` show smooth and sharp surface details. Result meshes of ours are reconstructed from the voxel resolution of 64^3 for a fair comparison.

2.2. Generation Results on ShapeNet Dataset Version 2

Figure 3 shows generated results of our `SDF-Diffusion` trained on the ShapeNet dataset version 2. We pre-process the ShapeNet dataset version 2 with the same method depicted in Sec. 4 of the main paper. We train our first-stage model for several unseen categories in ShapeNet version 1 (bathtub, bookshelf, bottle, faucet, jar, knife, laptop, motorcycle, mug, pistol, and printer), where we use a single-category scheme for training. Then, the second stage model is trained on every 55-category of the ShapeNet dataset version 2 without category conditions (we do not test training multi-category such as Sec. 4.2 in the main paper because the dataset has strong category imbalance). As you can see, `SDF-Diffusion` generates diverse and high-quality 3D shapes. The visualization shows generated 3D shapes in voxel resolution of 128^3.

2.3. More Generation Results

This section shows additional 3D shapes generated by `SDF-Diffusion` trained on the 13-category ShapeNet dataset. We show meshes reconstructed from the voxel resolutions of 32^3, 64^3, and 128^3, where 32^3 is a low-resolution shape generated from our first-stage model, and both 64^3 and 128^3 are super-resolution versions of our second-stage models. The visualization is available in Fig. 4, Fig. 5, and Fig. 6. `SDF-Diffusion` generates coarse but diverse 3D shapes in 32^3 resolution and makes them sharper and more detailed by increasing the resolution to 64^3 and 128^3.

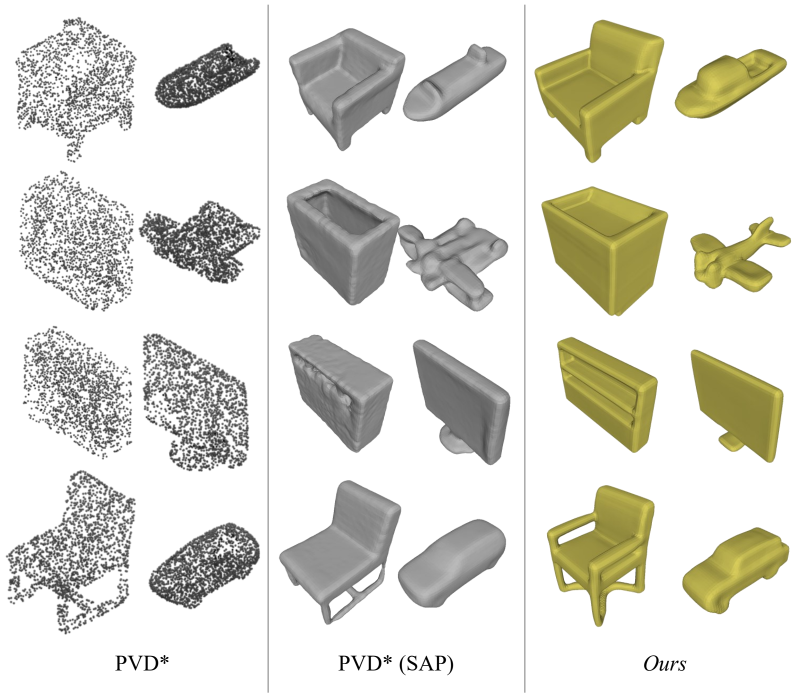


Figure 2. **Qualitative comparison of reconstructed meshes between PVD* and ours.**

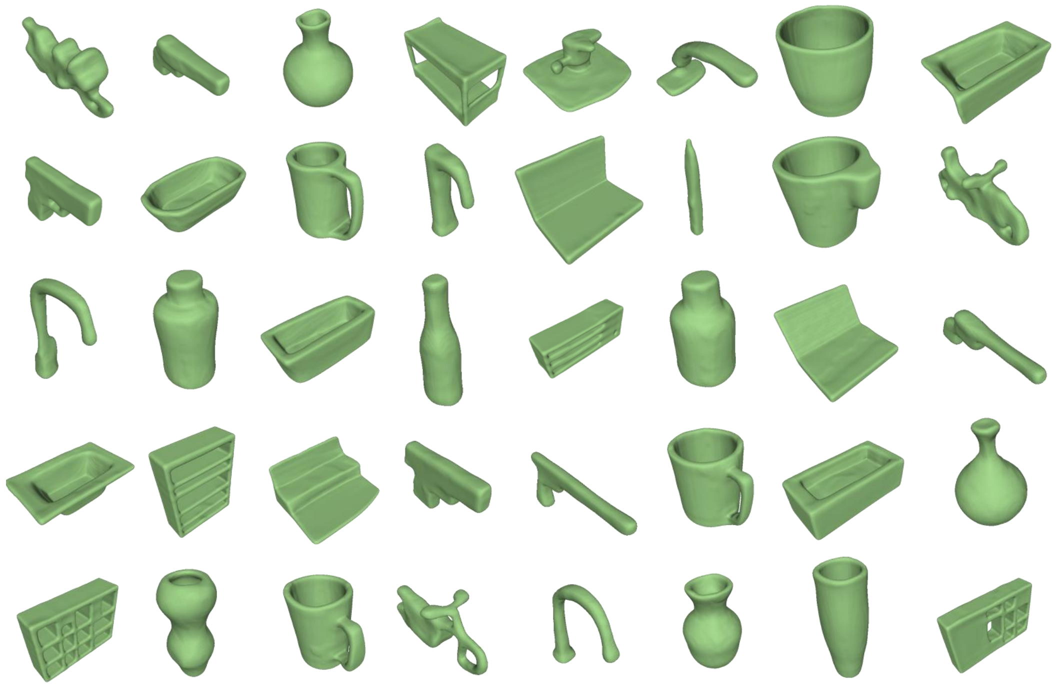


Figure 3. **Generated shapes of `SDF-Diffusion` trained on ShapeNet dataset version 2.**

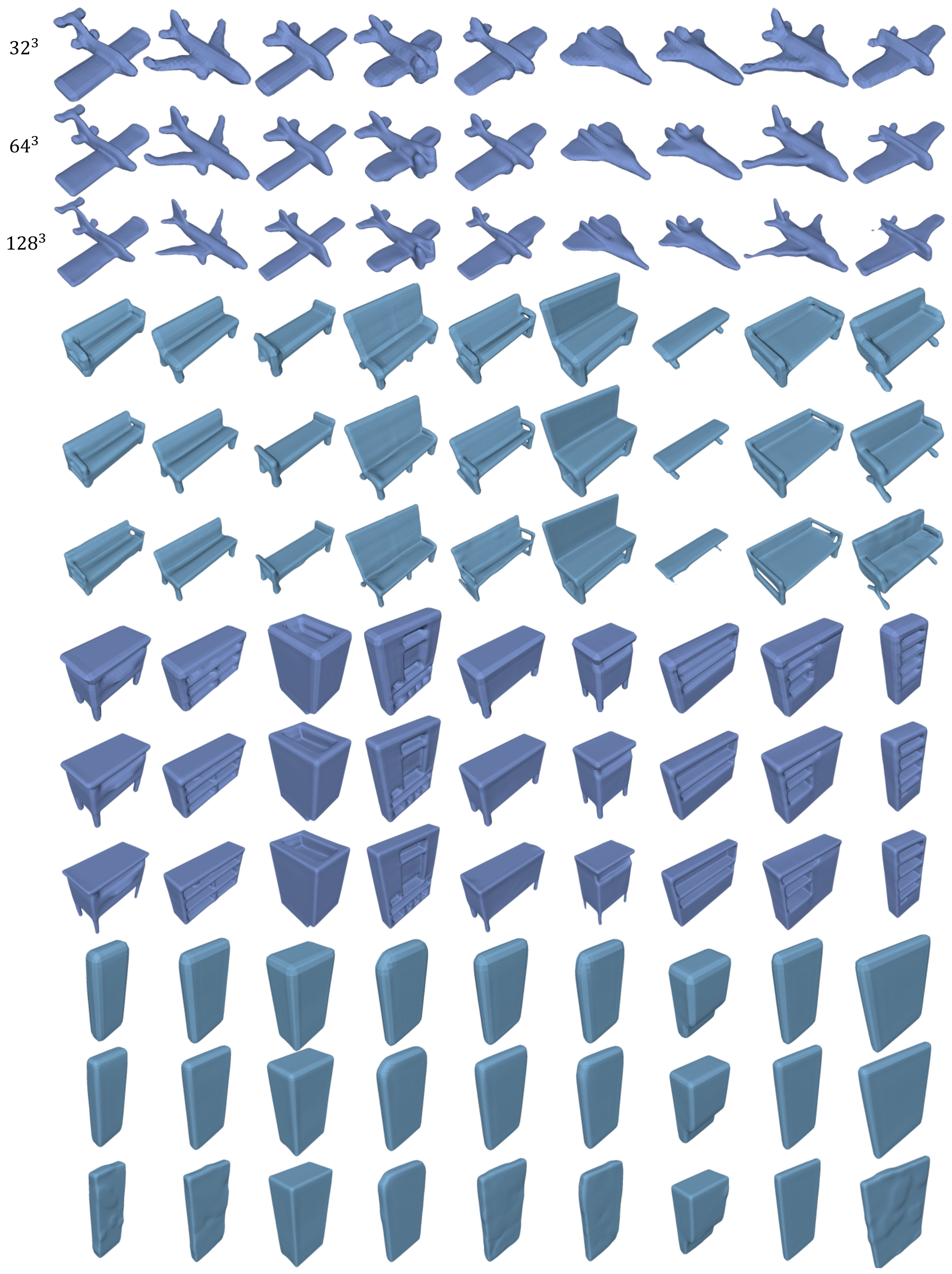


Figure 4. **Visualization of generated 3D shapes in four categories (airplane, bench, cabinet, and telephone).**

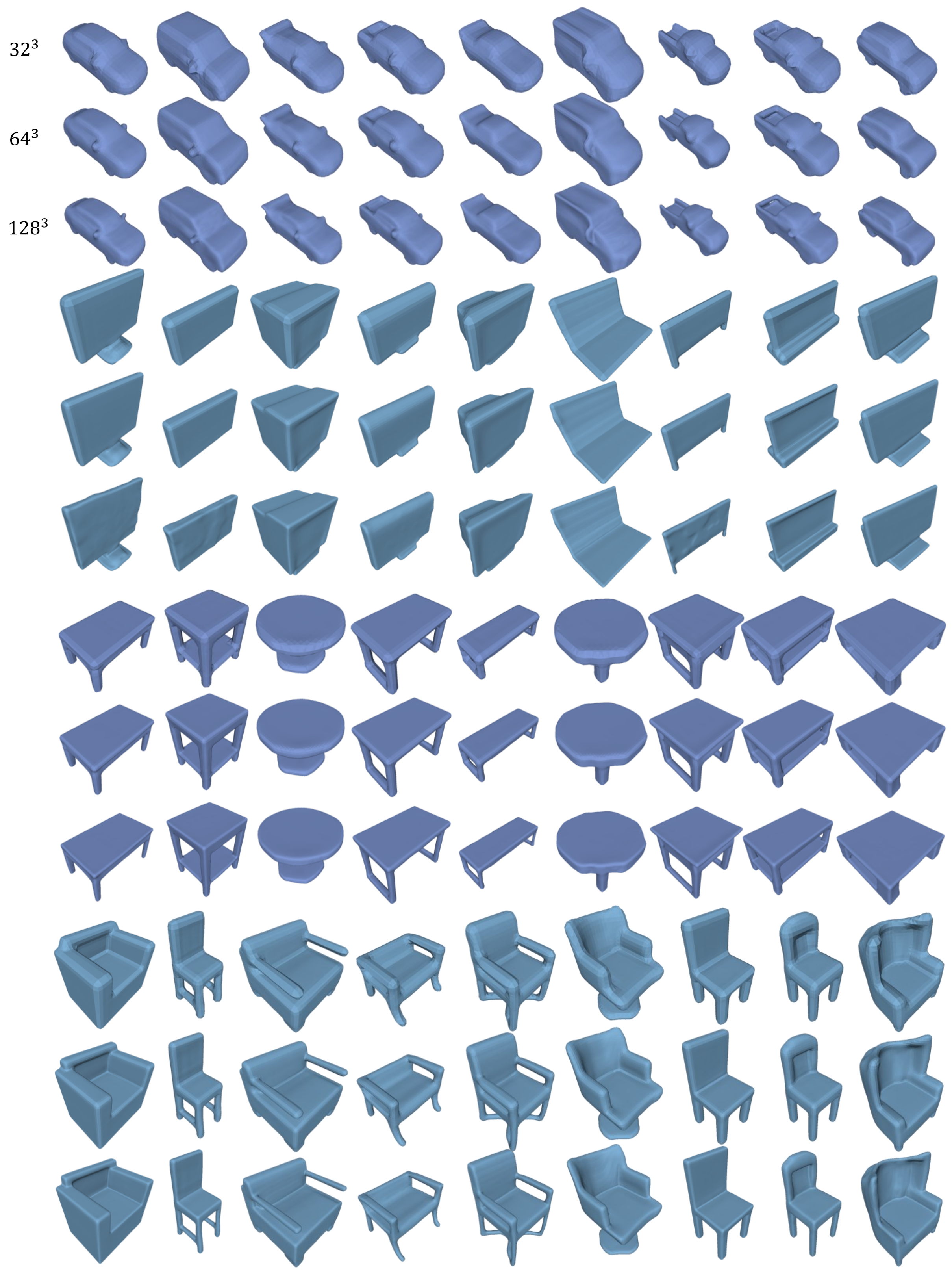


Figure 5. **Visualization of generated 3D shapes in four categories (car, display, table, chair).**

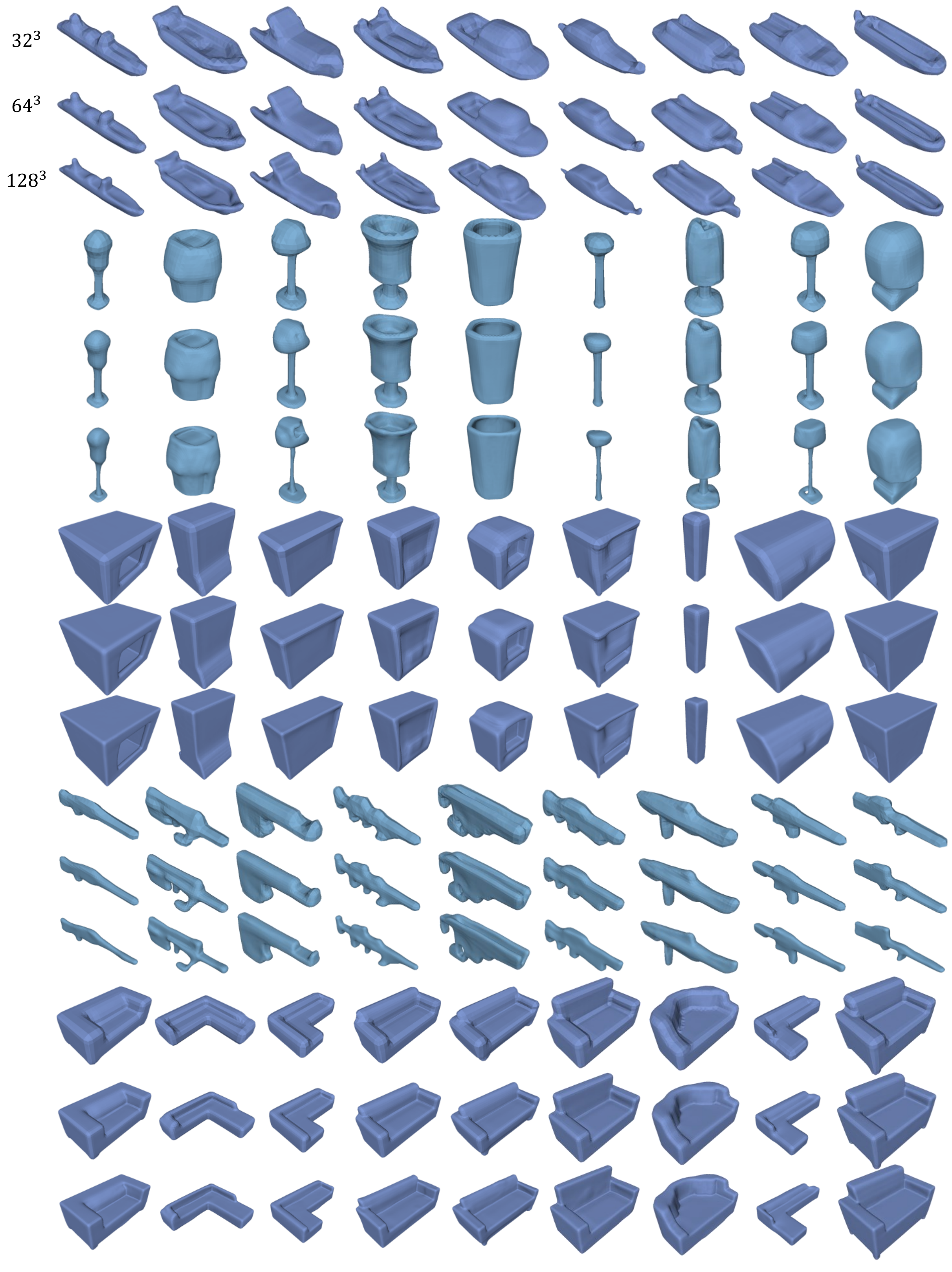


Figure 6. **Visualization of generated 3D shapes in five categories (vessel, lamp, loudspeaker, rifle, sofa).**

References

[1] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *NeurIPS*, 2020. 1

[2] William E. Lorensen and Harvey E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. *ACM SIGGRAPH*, 1987. 3

[3] Songyou Peng, Chiyu Jiang, Yiyi Liao, Michael Niemeyer, Marc Pollefeys, and Andreas Geiger. Shape as points: A differentiable poisson solver. In *NeurIPS*, 2021. 3

[4] Chitwan Saharia, Jonathan Ho, William Chan, Tim Salimans, David J Fleet, and Mohammad Norouzi. Image super-resolution via iterative refinement. *IEEE TPAMI*, 2022. 1

[5] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *NeurIPS*, 2017. 1

[6] Dongsu Zhang, Changwoon Choi, Jeonghwan Kim, and Young Min Kim. Learning to generate 3d shapes with generative cellular automata. In *ICLR*, 2021. 3

[7] Linqi Zhou, Yilun Du, and Jiajun Wu. 3D shape generation and completion through point-voxel diffusion. In *CVPR*, 2021. 3