Supplementary material:

PU-Net: Point Cloud Upsampling Network

Submission ID: 1248

A. Overview

In this supplementary material, we first provide more details about our collected dataset in Section B. Then, we show the details of our network architecture as well as the baseline networks employed in the experiments in Section C. Finally, we provide visualization results on ModelNet40 and ShapeNet tested by our network in Section D.

B. Details of our Collected Dataset

We collect 60 different 3D models to form our training and testing datasets. The specific name of each model is shown in Table 1. We also present the shapes of some training and testing 3D models in our dataset in Fig. 1 and Fig. 2, respectively. As we can see, our collected datasets have a large variation in geometry shapes, containing 3D models with smooth surface regions (first row) and 3D models with sharp corners and edges (second row). There is also a large variation between training and testing 3D models, indicating a good generalization ability of our proposed method.

Table 1. The complete name list of the 3D models in our training and testing datasets.

	Model Names
Training	Armadillo, Boy1, Boy2, Bumpy_torus, Bunny, Cad, Cylinder, Child1, Child2, Chinese_lion, Cone, Cup, Dino, Egea, Ellipsoid, Eros, Fish, Focal_octa, Gargoyle, Girl1, Girl2, Hand, Joint, Julius, Nicolo, Octa_flower, Pierrot, Pulley, Pyramid1, Pyramid2, Retinal, Rolling_stage, Screwdriver, Sharp_sphere, Special_cube, Star1, Turbine, Twirl, Vaselion
Testing	Camel, Casting, Chair, Cover_rear, Cow, Duck, Eight, Elephant, Elk, Fandisk, Genus, Horse, Icosahedron, Kitten, Moai, Octahedron, Pig, Quadric, Sculpt, Star2

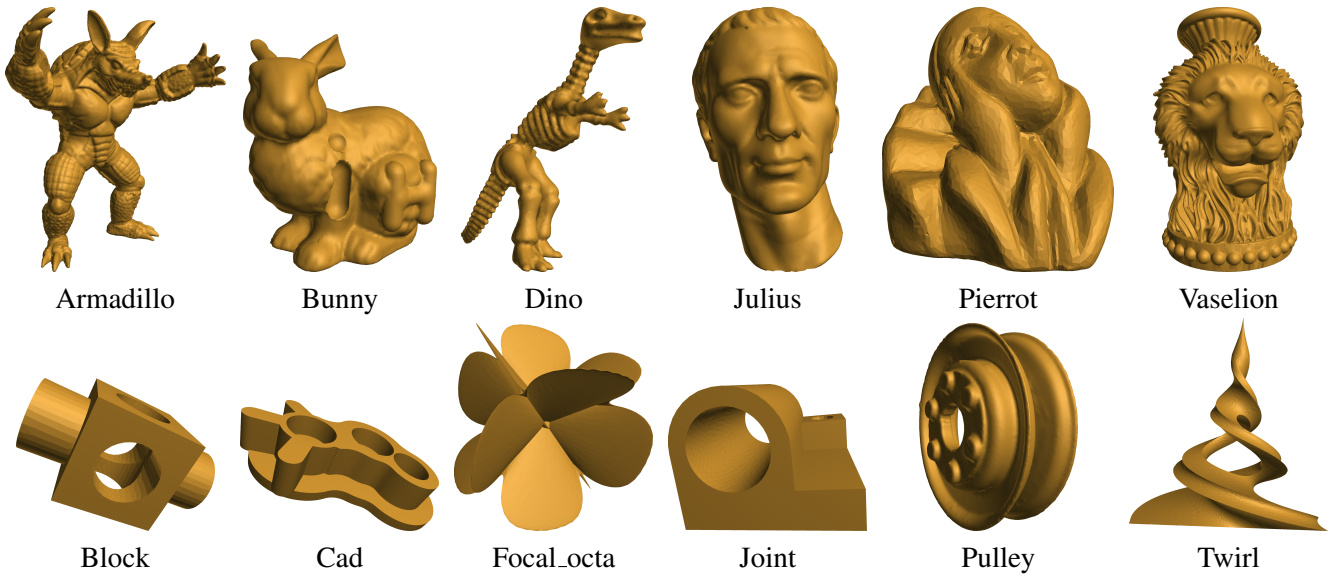

Figure 1. Examples 3D models in our training dataset. The first row shows 3D models with smooth surface regions, while the second row shows 3D models with sharp corners and edges.

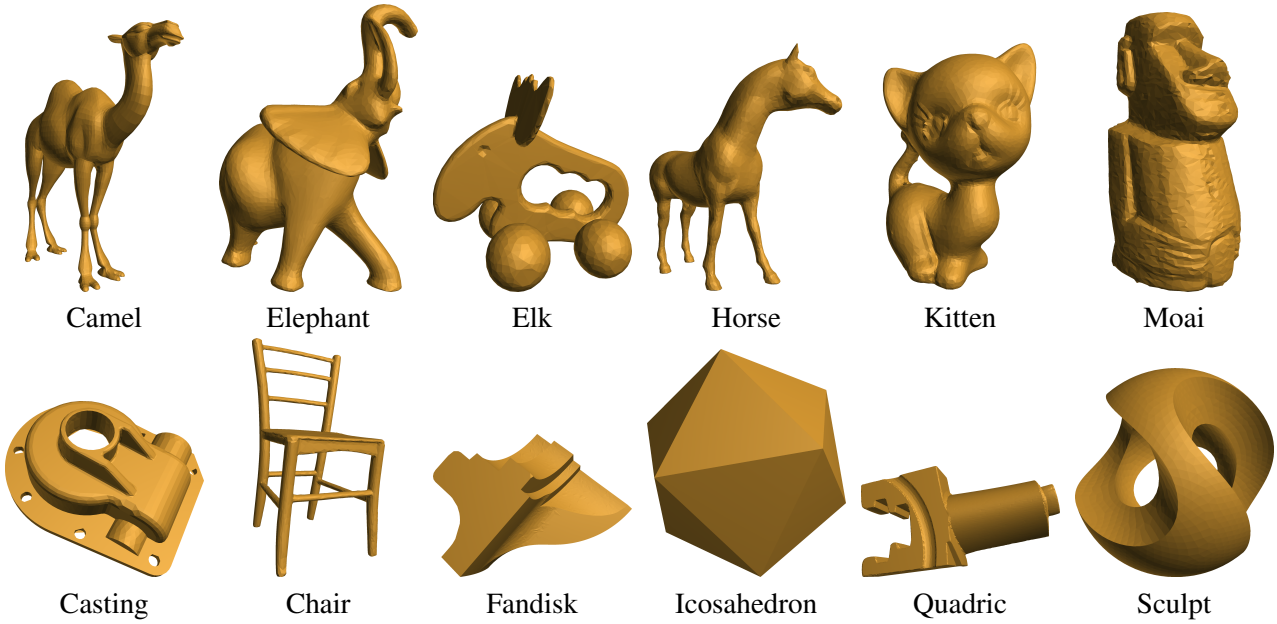

Figure 2. Examples 3D models in our testing dataset. The first row shows 3D models with smooth surface regions, while the second row shows 3D models with sharp corners and edges.

C. Details of Network Architectures

The details of our network architecture are listed as follows.

- In the hierarchical feature learning component, we use four levels to extract local features. Following the notations in PointNet++, we use (K, r, [$l_1, ..., l_d$]) to represent a level with K local regions of ball radius r, and [$l_1, ..., l_d$] the d fully connected layers with width l_i ($i = 1, ..., d$). Therefore, the parameters we use are $(N, 0.05, [32, 32, 64])$, $(N/2, 0.1, [64, 64, 128])$, $(N/4, 0.2, [128, 128, 256])$ and $(N/8, 0.3, [256, 256, 512])$.
- In the multi-level feature aggregation component, we use interpolation to restore the feature of each level and use a convolution to reduce the restored feature to 64 dimensions. Therefore, $\tilde{C}$ = 259 in our experiments.
- In the feature expansion component, the output feature channel numbers $\tilde{C}_1$ and $\tilde{C}_2$ are 256 and 128, respectively.
- In the coordinate reconstruction component, we use two fully connected layers with 64 and 3 output channels, respectively.

The details of the baseline architectures are illustrated in Fig. 3, Fig. 4 and Fig. 5.

All the convolution layers and fully connected layers in the above networks are followed by the ReLU operator, except for the last coordinate regression layer.

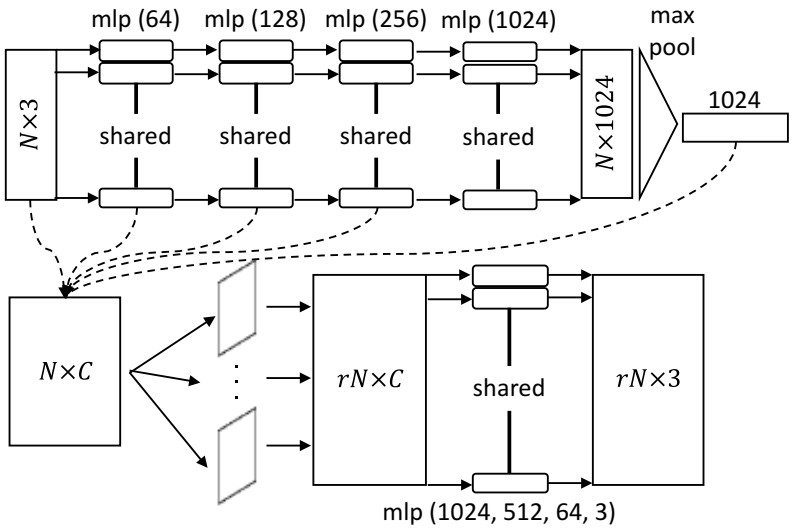

Figure 3. The network architecture of PointNet for point cloud upsampling.

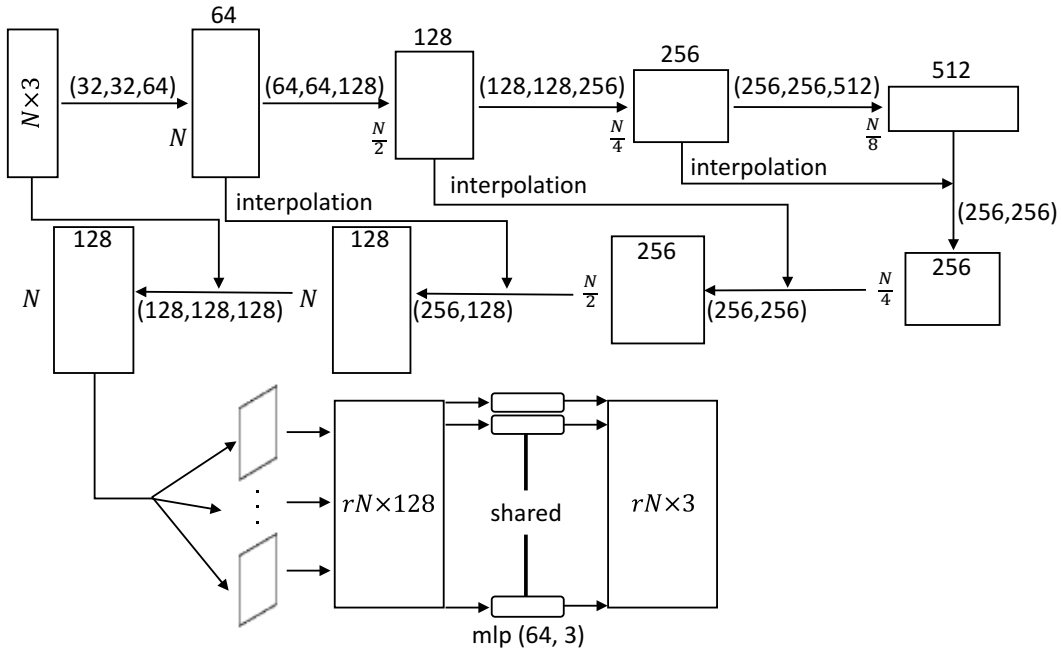

Figure 4. The network architecture of PointNet++ for point cloud upsampling.

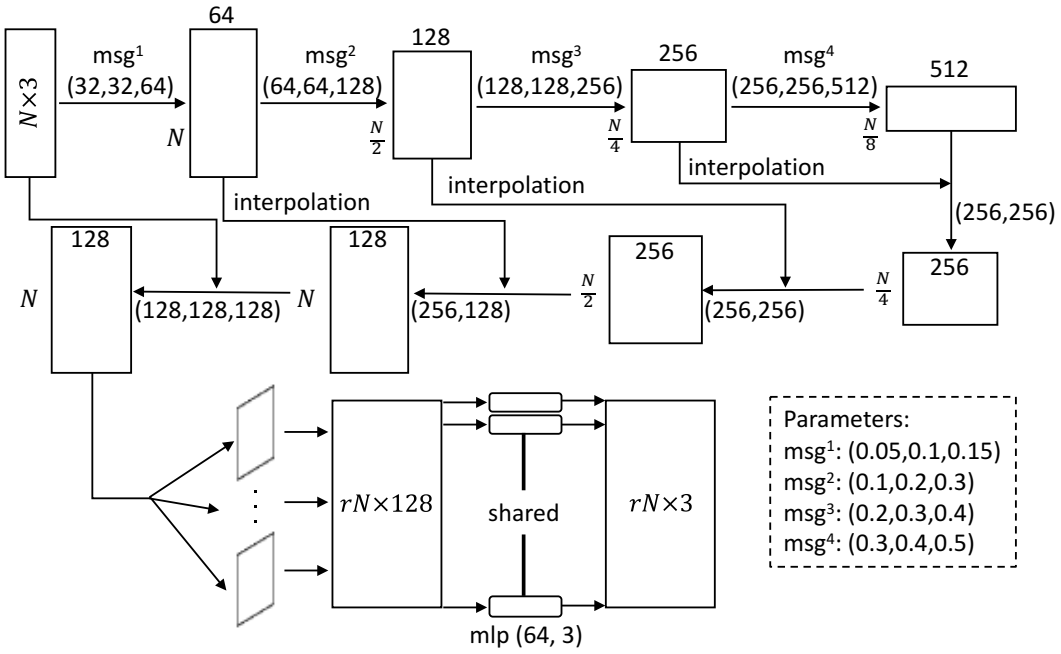

Figure 5. The network architecture of PointNet++(MSG) for point cloud upsampling.

D. Results on ModelNet40 and ShapeNet

In this section, we show the visual results tested on ModelNet40 and ShapeNet with our network.

Fig. 6 and Fig. 7 demonstrate the results from ModelNet40, while Fig. 8 and Fig. 9 demonstrate the results from ShapeNet. In each row, the left three images show the input point cloud in three different views, while the right three images show the output point cloud in corresponding views.

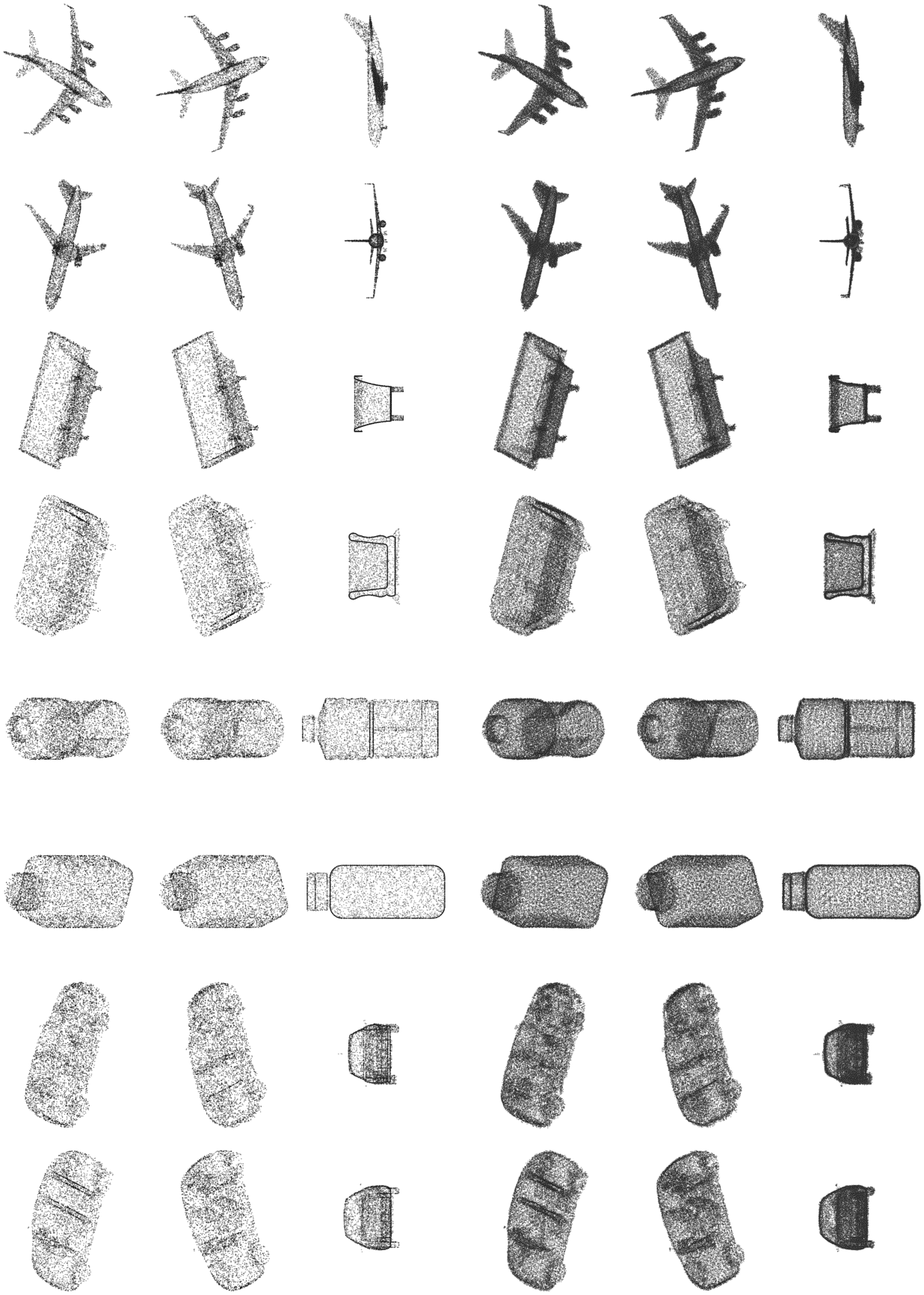

Figure 6. Visual comparison of input and output point clouds on objects from ModelNet40.

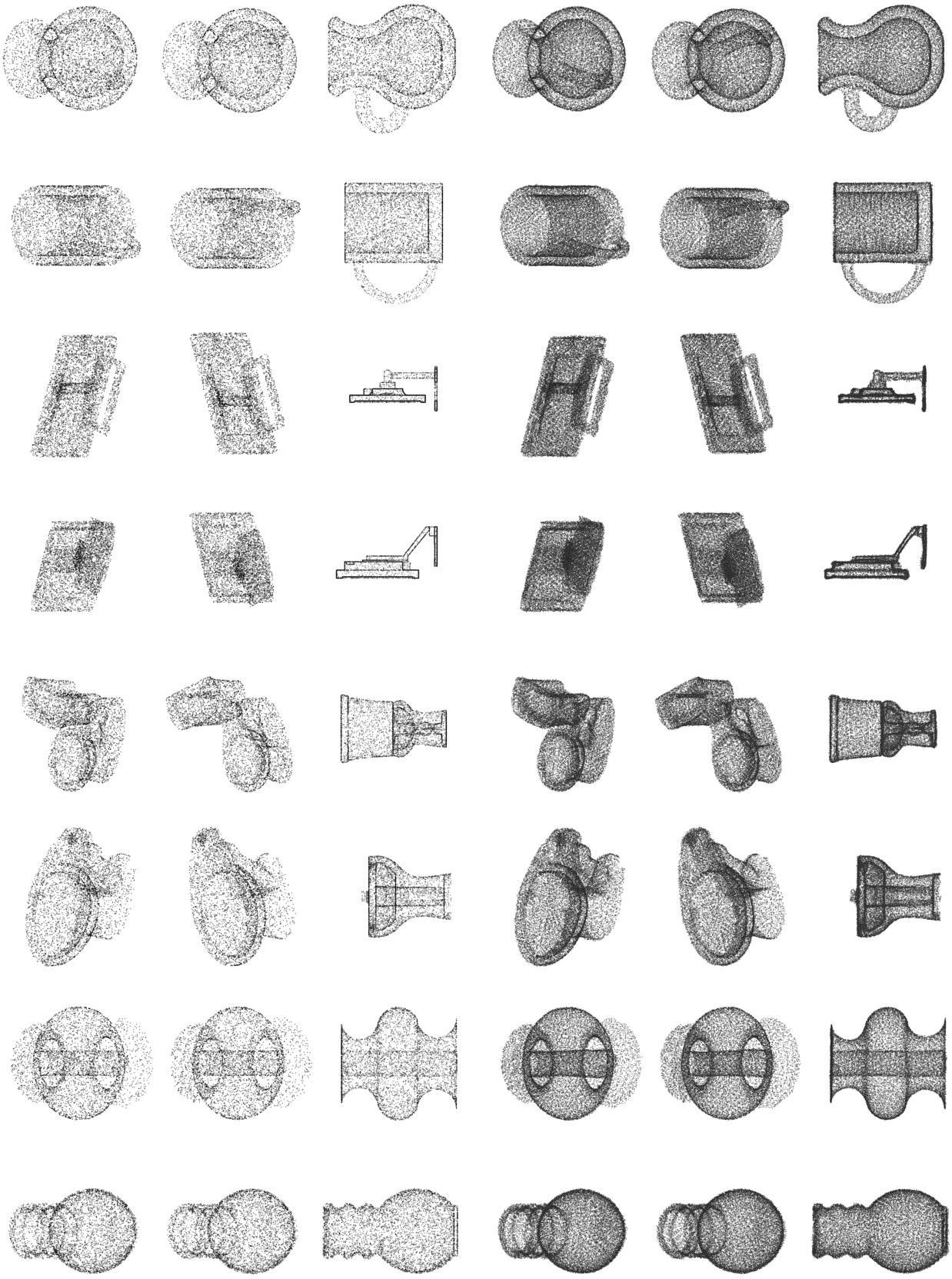

Figure 7. Visual comparison of input and output point clouds on objects from ModelNet40.

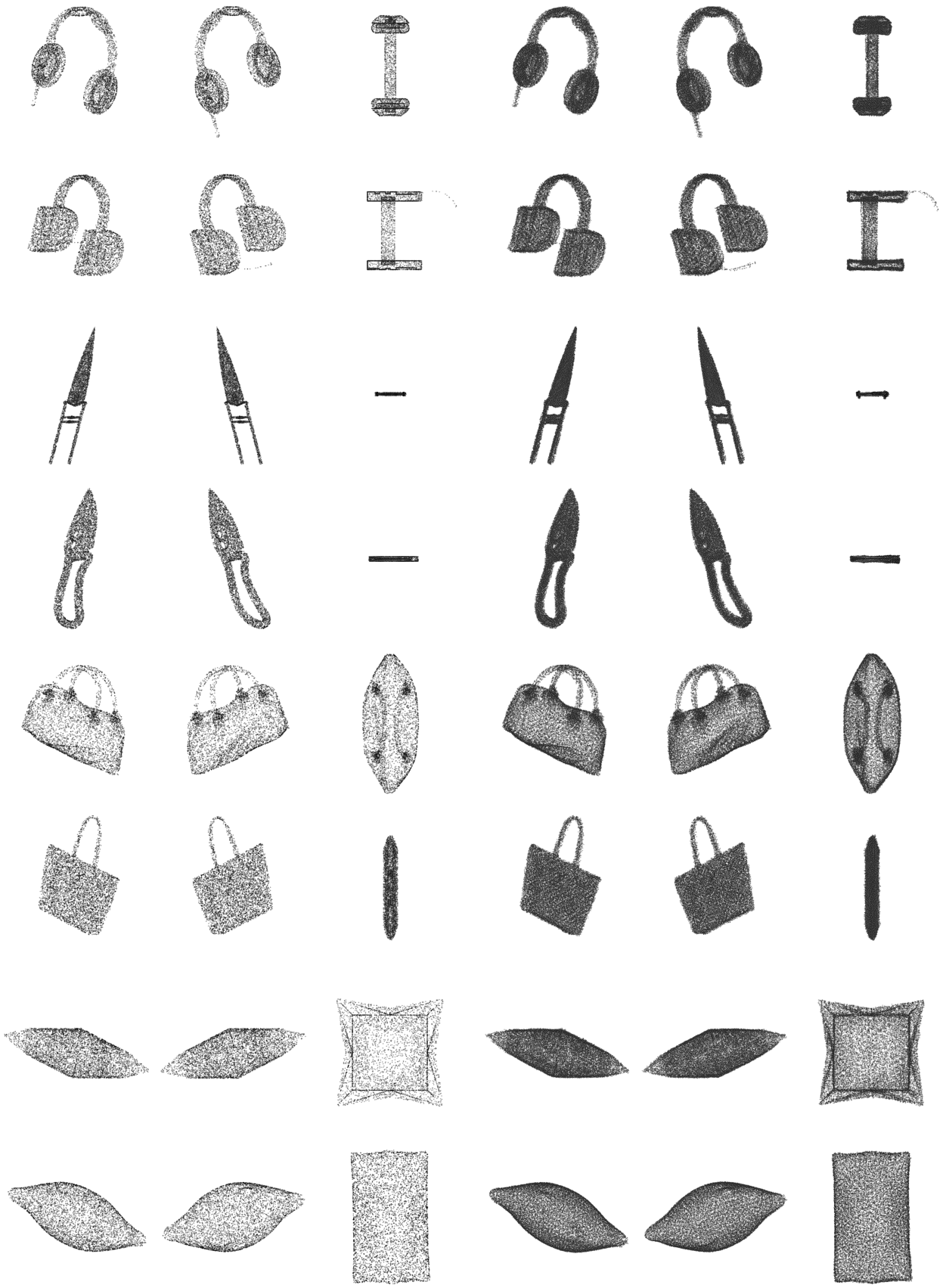

Figure 8. Visual comparison of input and output point clouds on objects from ShapeNet.

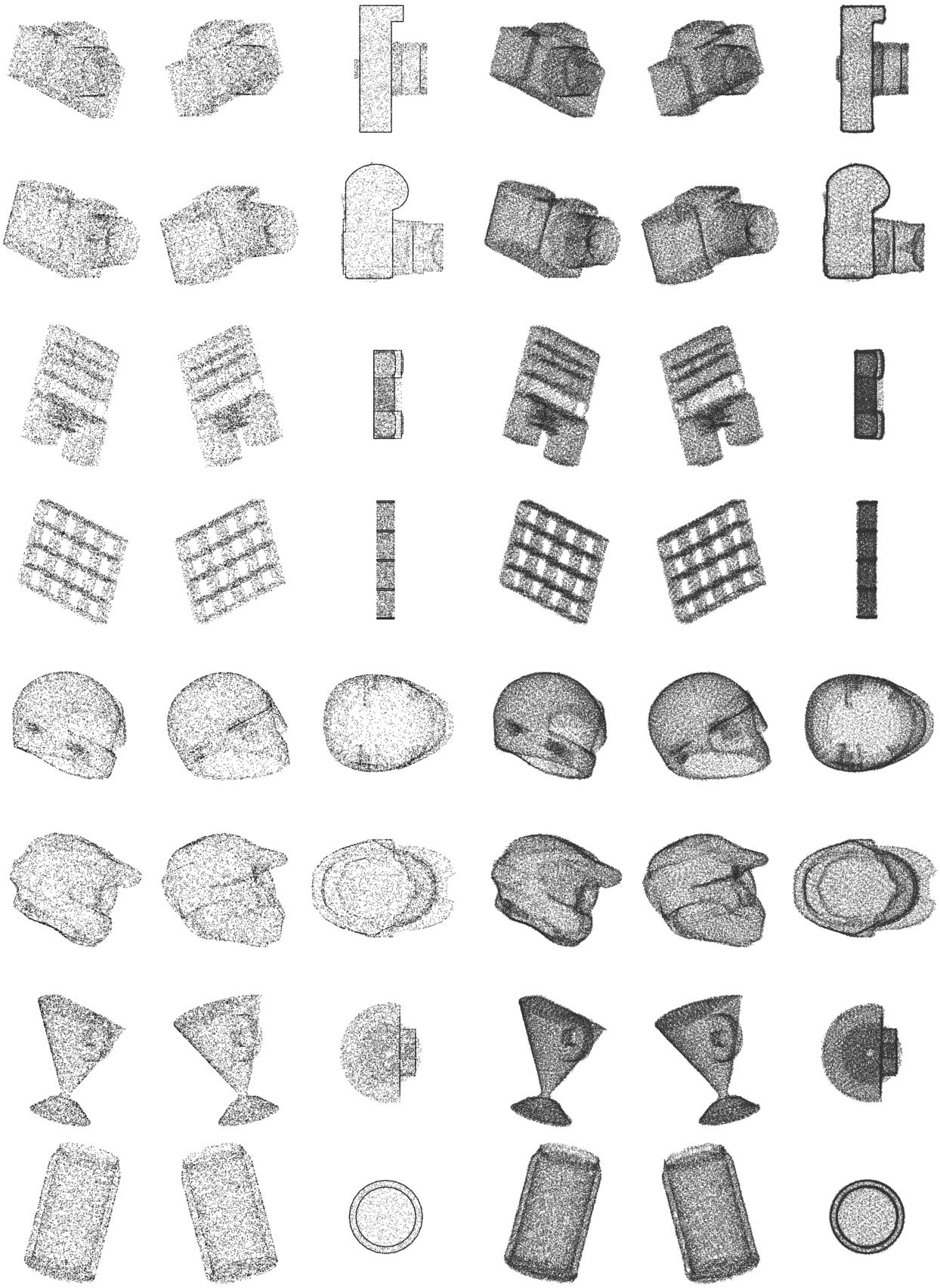

Figure 9. Visual comparison of input and output point clouds on objects from ShapeNet.