Matryoshka Networks: Predicting 3D Geometry via Nested Shape Layers
– Supplementary Material –

Stephan R. Richter[1,2*] Stefan Roth[1]
[1]TU Darmstadt [2] Intel Labs

We here give details on the network architectures and present additional visual results.

A. Network Architecture

Our networks consist of multiple residual modules (*c.f.* [12]) as shown in Fig. 9. Across all modules, we keep the kernel size k and the stride s constant as depicted. For all convolutions with a kernel size $k > 1$, we use reflective padding of size 1. Altering the spatial resolution and number of feature channels requires special handling of the identity pathway of the residual modules. For down-sampling (Fig. 9, middle), we simply drop every other pixel and initialize the added feature channels as zero using zero-padding. For up-sampling (Fig. 9, right), we use nearest neighbor interpolation and a 1×1 convolution to project the feature dimension. We experimented with more sophisticated up- and down-sampling alternatives, but found no significant benefits.

In all experiments with images as inputs, processing in our networks begins with a feature generation module, which produces an initial representation with f_{in} feature channels. This module is equivalent to the residual module operating at constant resolution (Fig. 9, left), but with the first rectified linear unit and identity pathway removed. Each module is only parametrized by the number of feature channels added during down-sampling $\Delta f_{\downarrow}$ or removed during up-sampling $\Delta f_{\uparrow}$, and we pair each up-sampling and down-sampling module with a subsequent module of same resolution to form one residual block. Thus, we specify network architectures by a desired number of initial features $\hat{f}_{in}$, output features $\hat{f}_{out}$, features at the bottleneck f_{inner}, number of desired down-sampling blocks d in the decoder, and residual blocks at the bottleneck b. We match the number of down-sampling blocks in the encoder with the number of up-sampling blocks in the decoder. We set it to 3 for an output resolution $s_{out} = 32$ and increase it by 1 for every doubling of s_{out}. If input and output resolutions are different, we add $d_i = \log_2 s_{in} - \log_2 s_{out}$ down-sampling blocks or $d_o = \log_2 s_{out} - \log_2 s_{in}$ up-sampling blocks accordingly. For all networks, we scale input images to powers of 2. We compute the number of feature channels to add for each down-sampling block as

$$\Delta f_{\downarrow} = \left\lfloor \frac{f_{inner} - \hat{f}_{in}}{d_i + d} \right\rfloor, \tag{10}$$

and adjust the number of initially generated features as

$$f_{in} = \hat{f}_{in} + (f_{inner} - \hat{f}_{in}) \bmod \Delta f_{\downarrow} \tag{11}$$

to obtain integral numbers for the number of feature channels. Analogously, we compute the number of feature channels added per up-sampling block as

$$\Delta f_{\uparrow} = \left\lfloor \frac{f_{inner} - \hat{f}_{out}}{d_o + d} \right\rfloor. \tag{12}$$

To obtain predictions with the desired number of output channels (equaling s_{out} for voxel tube networks and the number of shape layers $\times 6$ for Matryoshka networks) we simply add a 2D convolution with kernel size 1 as final layer to our networks. We summarize the architectures and training schedules used in the individual experiments in Tab. 7.

For a batch size of 128, we start with a learning rate of 0.001 and reduce it by a factor of 10 after *drop* epochs. For any different batch size, we scale the learning rate accordingly. All models were trained on a single GPU.

For the ablation studies, we used a voxel tube network as summarized in the penultimate row of Tab. 7. Since the networks for the shape-from-silhouette task and the ablation studies were trained on renderings of smaller resolution and on a smaller number of categories, we roughly halved the number of feature channels at the bottleneck (setting it to 257 for an integer $\Delta f_{\uparrow}$).

For the study on network architectures, we refer to the voxel tube network as described above as ResNet-based network. We remove the identity pathways from all residual modules to obtain an Encoder/decoder network, and add skip connections between layers of same spatial resolution to obtain a U-Net, *c.f.* [23]. To adapt the number of feature channels for the skip connections, we use a

*This work was carried out while at TU Darmstadt.

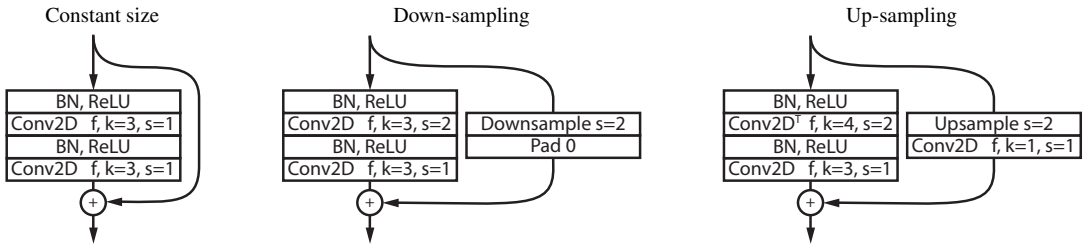

Figure 9: **Residual modules.** Each residual module consists of Batch normalization (BN) and Rectified Linear Unit (ReLU) layers followed by a 2D convolution, except for the up-sampling module where we replace the first convolution with a transposed convolution. The number of feature channels is denoted as f. Moreover, k denotes the filter size and s the stride.

Network	s_{in}	s_{out}	batch size	epochs	drop	$\hat{f}_{in}$	d	f_{inner}	b	$\hat{f}_{out}$
ShapeNet-all										
Voxel tube	128	32	128	45	15	8	3	512	1	32
Matryoshka	128	32	128	40	20	8	3	512	2	128
High resolution										
Matryoshka	128	32	128	30	20	8	3	512	0	128
Matryoshka	128	64	128	30	20	8	4	512	3	128
Matryoshka	128	128	32	30	20	8	5	512	1	128
Matryoshka	128	256	8	30	20	8	6	512	0	128
Shape from Silhouette										
Matryoshka	64	32	128	40	15	8	3	257	2	32
Shape from ID										
Matryoshka	2	512	4	28K	12K	–	8	1	0	196
Ablation studies										
Voxel tube	64	32	128	40	15	8	3	257	2	32
Shape from Similarity										
Matryoshka	1	128	8	60	25	–	7	2424	0	128

Table 7: **Network architectures for individual experiments.** See text for a description of the network parameters.

1×1 2D convolution akin to the identity path of the up-sampling module in Fig. 9. The DenseNet-inspired version (*c.f.* [14]) of our voxel-tube network consists of 7 dense blocks (B), 2 up-transitions (U) and 3 down-transitions (D) arranged as BDBDBDBBUBUB. Each dense block contains 2 dense layers with an expansion factor of 16. For the down-transitions we halve the spatial resolution while keeping the number of feature channels constant. For the up-transitions we double the spatial resolution and halve the number of feature channels.

B. More Results

Shape from silhouette. We investigate the performance of our Matryoshka network on the task of reconstructing a 3D shape from a single silhouette image. To that end, we reconstruct the shapes of the 3 categories with the most examples (chair, car, table) from ShapeNet-core with the dataset split and shapes from Choy *et al*. [5]. We obtain silhouettes from the alpha-channels of the renderings of Choy *et al*. As can be seen in Tab. 8, the network performs much better on cars than on tables or chairs. This can be attributed to the approximately convex shape of cars, which makes their silhouette a very effective cue for the overall shape. Compared to the easier setting of reconstructing shapes from a color image, the network performs remarkably well. Note, however, that the network for predicting shapes from color images was trained in a category-agnostic way, making the prediction considerably harder.

Category	car	chair	table	mean
Shape from silhouette	86.7	53.2	58.8	66.2

Table 8: **Shape from silhouette** on ShapeNet-core.

Real-world images. To assess the performance of our proposed network for real world examples, we tested it on images from the Stanford Products Dataset [37] (chairs) and the web (cars). Qualitative examples are shown in Fig. 10 (chairs) and Fig. 11 (cars). In both cases, we trained a category-specific Matryoshka network to predict 3D shapes at 128^3 resolution from a single image. For predicting chairs, we took the renderings of Choy *et al*. [5] and created ground truth shapes of higher resolution from the corresponding ShapeNet [4] models using *binvox* [36]. Since most images of cars found on the web are recorded from different camera positions than the renderings of Choy *et al*., we re-rendered the car shapes from ShapeNet with random camera positions (focal length $\in [40\text{mm}, 90\text{mm})$, azimuth $\in [0°, 360°)$, elevation $\in [0°, 25°]$) and environment maps collected from the web[1,2]. We find that Matryoshka networks generalize well to real-world imagery even when only trained on synthetic images. They are able to reconstruct thin structures (*e.g*., the legs of the right-most chair in Fig. 10) and a wide variety of shapes (both Figs. 10 and 11).

Synthetic images. We show more results for predicting 3D shapes of high resolution in Figs. 13 (airplanes), 14 (chairs), and 12 (cars). The input images are renderings from Choy *et al*. [5] and the shapes have been converted to binary voxel grids using *binvox* [36]. The ground truth car shapes have been provided by Tatarchenko *et al*. [27]. Supporting the quantitative results from the main paper, learning to reconstruct 3D shapes at higher resolution produces much more accurate predictions, as can be seen for different resolutions in Fig. 12. Even for highly varied classes such as airplanes or chairs, Matryoshka networks produce high-quality reconstructions. Finally, we show qualitative examples of reconstructed shapes at low resolution from a voxel tube network and a Matryoshka network in Fig. 15. Both networks were trained on 13 categories from ShapeNet-core. Quantitative results for this experiment can be found in Tab. 2 in the main paper.

References

[36] P. Min. binvox. http://www.patrickmin.com/binvox, 2004 - 2017. Accessed: 2017-11-22. 3

[37] H. O. Song, Y. Xiang, S. Jegelka, and S. Savarese. Deep metric learning via lifted structured feature embedding. In *CVPR*, 2016. 3

[1] http://www.hdrlabs.com/sibl/archive.html

[2] https://hdrihaven.com/

Figure 10: **Qualitative results at high resolution** (128^3) **for real-world images of chairs.** For a given input image (top row), our Matryoshka network predicts a 3D shape (bottom row).

Figure 11: **Qualitative results at high resolution** (128^3) **for real-world images of cars.** For a given input image (top row), our Matryoshka network predicts a 3D shape (bottom row).

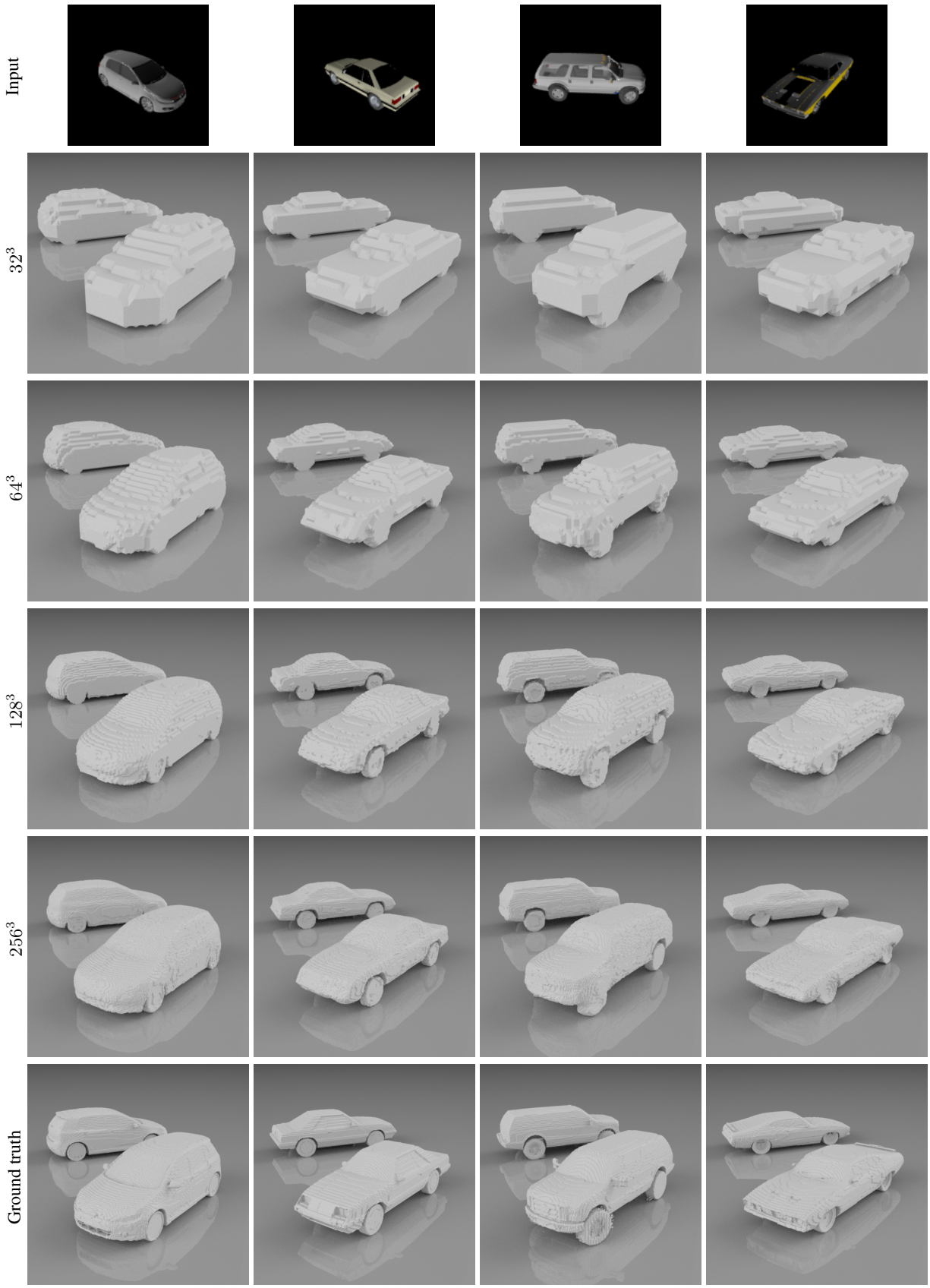

Figure 12: **Qualitative results at varying resolution.** We train Matryoshka networks to reconstruct 3D shapes from a single image rendered from ShapeNet models (top row) for output resolutions 32^3, 64^3, 128^3, and 256^3. The last row shows the ground truth shapes at 256^3.

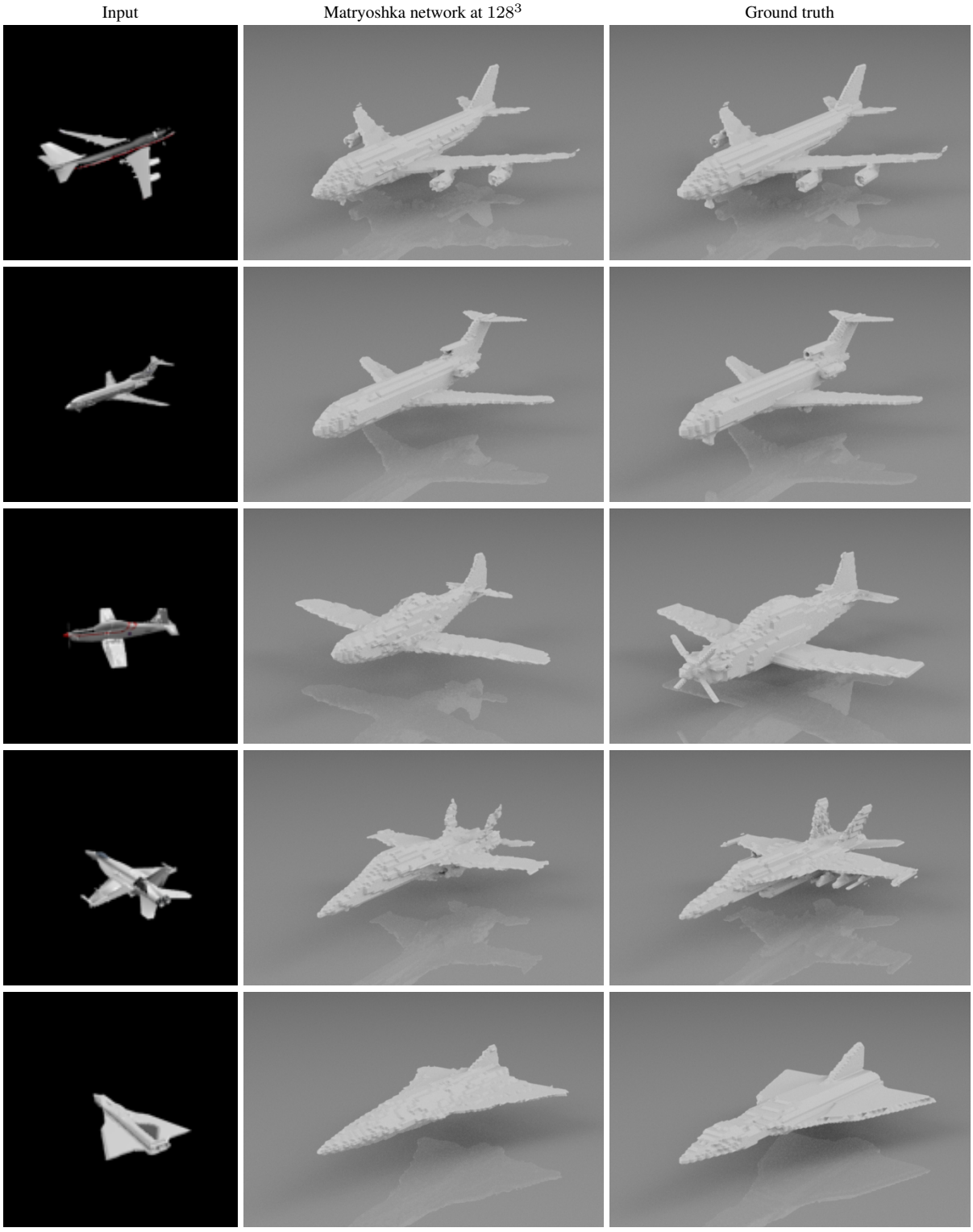

Figure 13: **Qualitative results at high resolution** (128^3) **for airplane images rendered from ShapeNet models.**

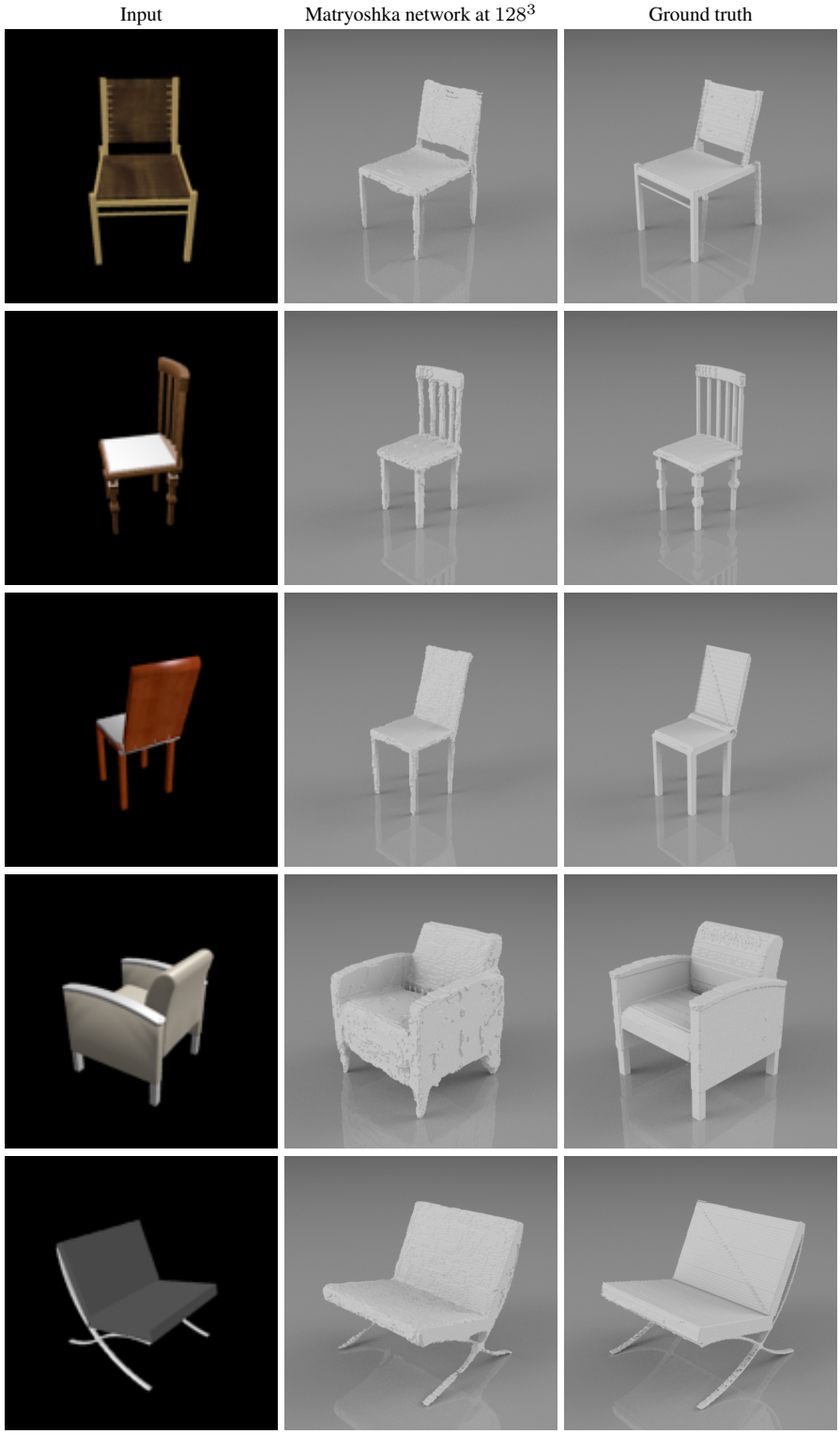

Figure 14: **Qualitative results at high resolution** (128^3) **for chair images rendered from ShapeNet models.**

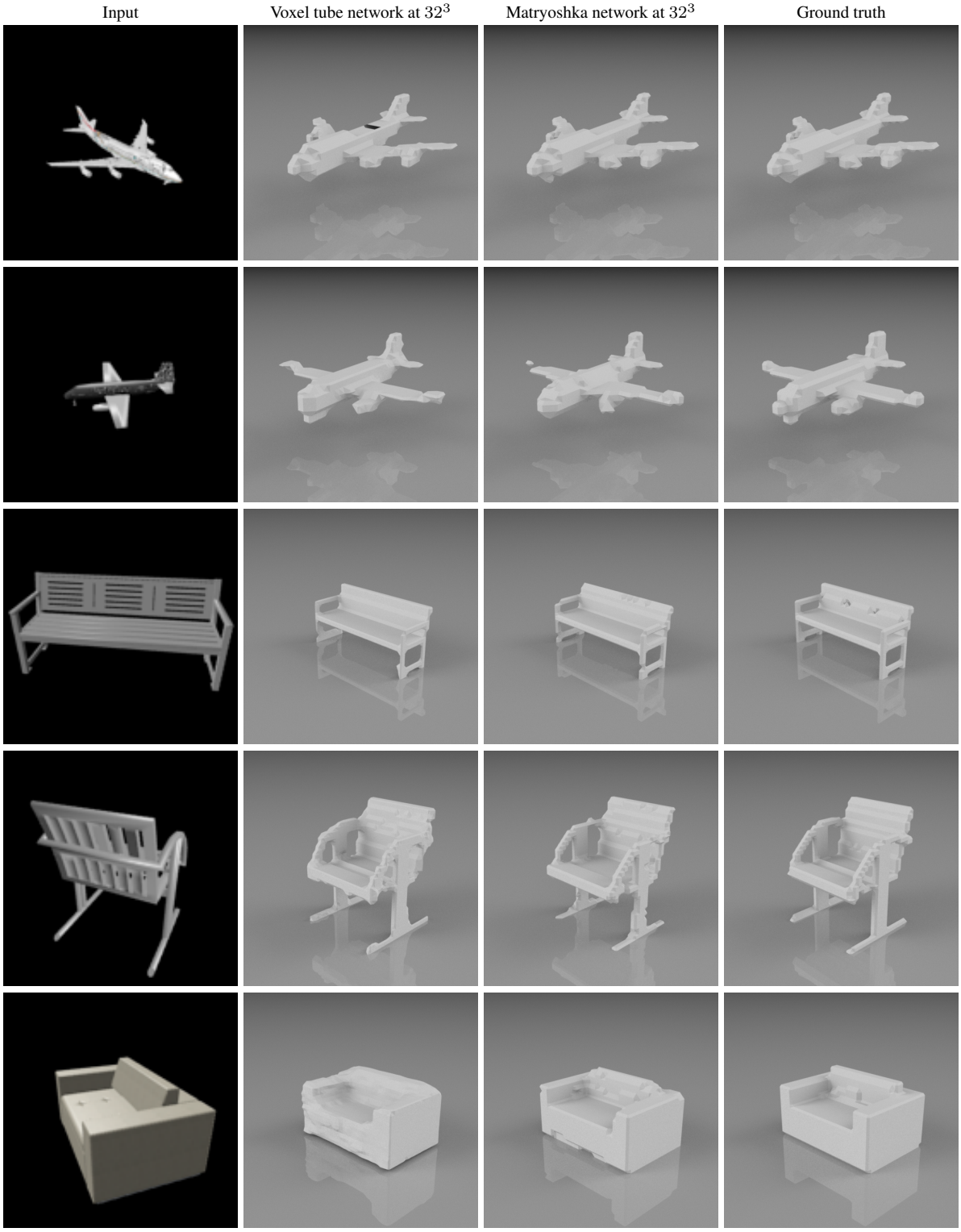

Figure 15: **Qualitative results at low resolution (32^3) for images rendered from ShapeNet models.** For input images (left-most row), we predict 3D shapes using a voxel tube network (2nd column) and a Matryoshka network (3rd column). Ground truth shapes are shown in the right-most column.

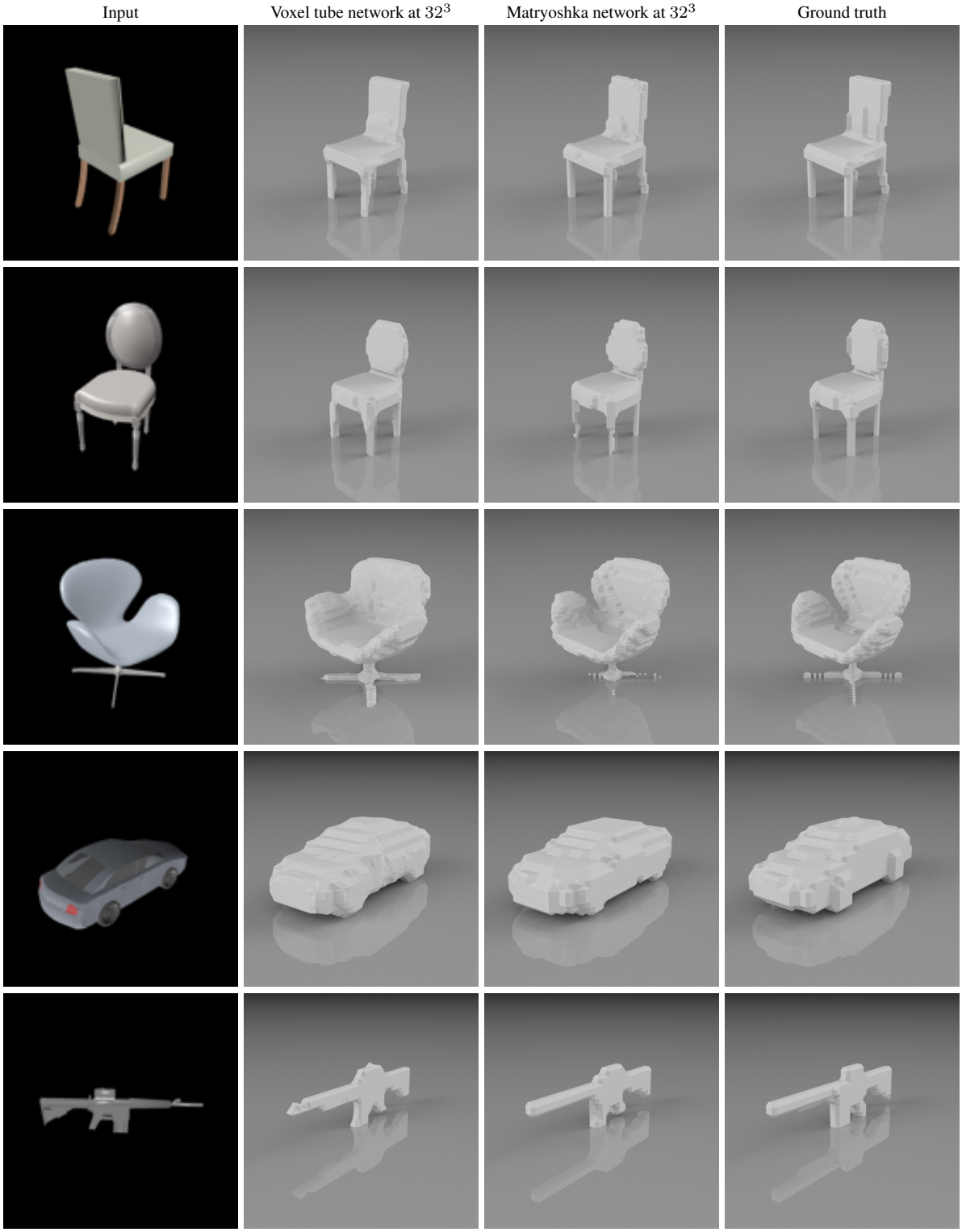

Figure 15: **Qualitative results at low resolution (32^3) for images rendered from ShapeNet models (continued)**.

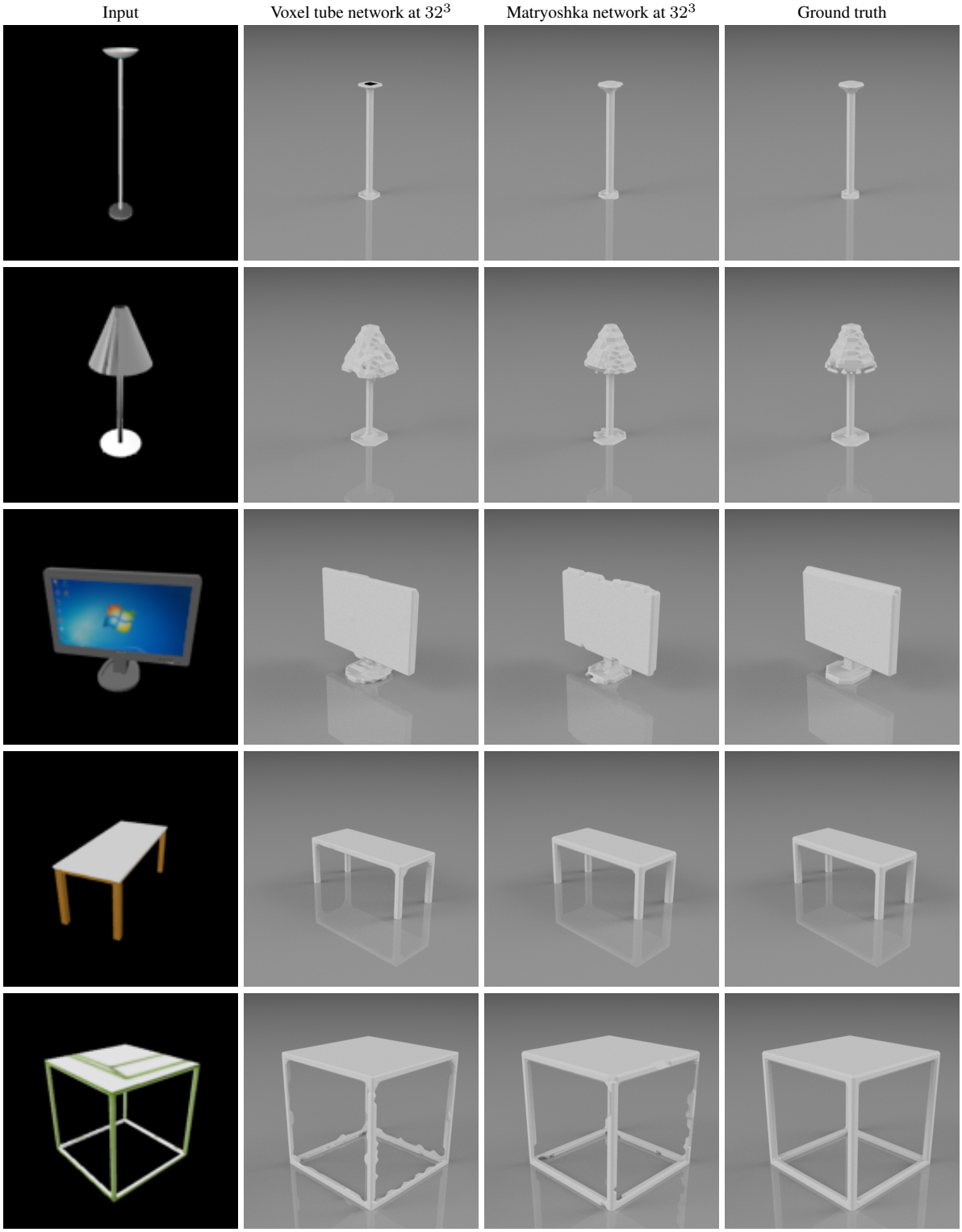

Figure 15: **Qualitative results at low resolution (32^3) for images rendered from ShapeNet models (continued)**.