CompoNet: Learning to Generate the Unseen by Part Synthesis and Composition
Supplementary Material

1. Supplementary Material

1.1. Network Architectures

Part synthesis The architectures of the part synthesis generative autoencoders, for both 3D and 2D cases, are listed in Table 1 and Table 2 respectively. We used the following standard hyper-parameters to train the 3D (2D) model: Adam optimizer, $\beta_1 = 0.9(0.5)$, $\beta_2 = 0.999$, learning rate $= 0.001(2e^{-4})$, batch size $= 64$.

Operation	Kernel	Strides	Feature maps	Act. func.
Encode x: 400x3 point cloud $\rightarrow$ 64-dim feature vector f_x				
1D conv.	1x64	1	400x64	Relu
1D conv.	1x64	1	400x64	Relu
1D conv.	1x64	1	400x64	Relu
1D conv.	1x128	1	400x128	Relu
1D conv.	1x64	1	400x64	Relu
Max Pooling	–	–	128	–
Decode f_x: 64-dim feature vector $\rightarrow$ 400x3 point cloud x'				
Linear	–	–	256	Relu
Linear	–	–	256	Relu
Linear	–	–	400x3	–

Table 1: Layers of the 3D part generative AE shown in order from input to output.

Operation	Kernel	Strides	Feature maps	Act. func.
Encode x: 64x64x1 input shape $\rightarrow$ 10-dim feature vector f_x				
Conv.	5x5x8	2x2	32x32x8	l-Relu
Conv.	5x5x16	2x2	16x16x16	l-Relu
Conv.	5x5x32	2x2	8x8x32	l-Relu
Conv.	5x5x64	2x2	4x4x64	l-Relu
2xLinear	–	–	10	–
Decode f_x: 10-dim feature vector $\rightarrow$ 64x64x1 shape x				
Linear	–	–	1024 (=4x4x64)	Relu
Trans. conv.	5x5x32	2x2	8x8x32	Relu
Trans. conv.	5x5x16	2x2	16x16x16	Relu
Trans. conv.	5x5x8	2x2	32x32x8	Relu
Trans. conv.	5x5x1	2x2	64x64x1	Sigmoid

Table 2: Layers of the 2D part VAE shown in order from input to output. After the last conv. layer we have two parallel linear layers for computing the mean and std of the VAE.

Parts composition The architectures of the parts composition units are listed in Table 3 and Table 4, for the 3D and 2D cases respectively. We used the following standard hyper-parameters to train the 3D and 2D models: Adam optimizer, $\beta_1 = 0.9$, $\beta_2 = 0.999$, learning rate $= 0.001$, batch size $= 64$.

Operation	Feature maps	Act. func.
Comp. net x: 64xC+16 feature vector $\rightarrow$ 6XC comp. vector f_x		
Linear	256	Relu
Linear	128	Relu
Linear	6xC	–

Table 3: Layers of the 3D parts composition network shown in order from input to output. Where C is the number of parts. The input vector is a concatenation of C feature vectors of size 64 with a noise vector of size 16.

Operation	Feature maps	Act. func.
Comp. net x: 10xC+8 feature vector $\rightarrow$ 4XC comp. vector f_x		
Linear	128	Relu
Linear	128	Relu
Linear	4xC	–

Table 4: Layers of the 2D parts composition network shown in order from input to output. Where C is the number of parts. The input vector is a concatenation of C feature vectors of size 10 with a noise vector of size 8.

1.2. More results

In this section, we present additional results of CompoNet for the three categories. For the Chair and Airplane categories we have randomly sampled 80 shapes from the $5,000$ we have generated for the quantitative metrics; see Figure 1 and Figure 2 respectively. For the Vases, since the output is smaller, we sampled 300 shapes from the $1,024$ we generated for the quantitative metrics; see Figure 3. Furthermore, we present additional interpolation results from CompoNet on the 3D categories; see Figure 4 and Figure 6 for linear interpolations, Figure 5 and Figure 7 for part-by-part interpolations.

1.3. Comparison to baseline

In this section, we present randomly picked generated results from CompoNet and the baseline on the Chair and

Airplane categories. For each generated shape, we present its three nearest neighbors based on the Chamfer distance. We present the results side by side to emphasize the power of CompoNet in generating novel shapes in comparison to the baseline; see Figure 8 for chairs and Figure 9 for airplanes.

Figure 1: Chair gallery. We present 80 randomly sampled chairs from the $5,000$ which were generated by CompoNet.

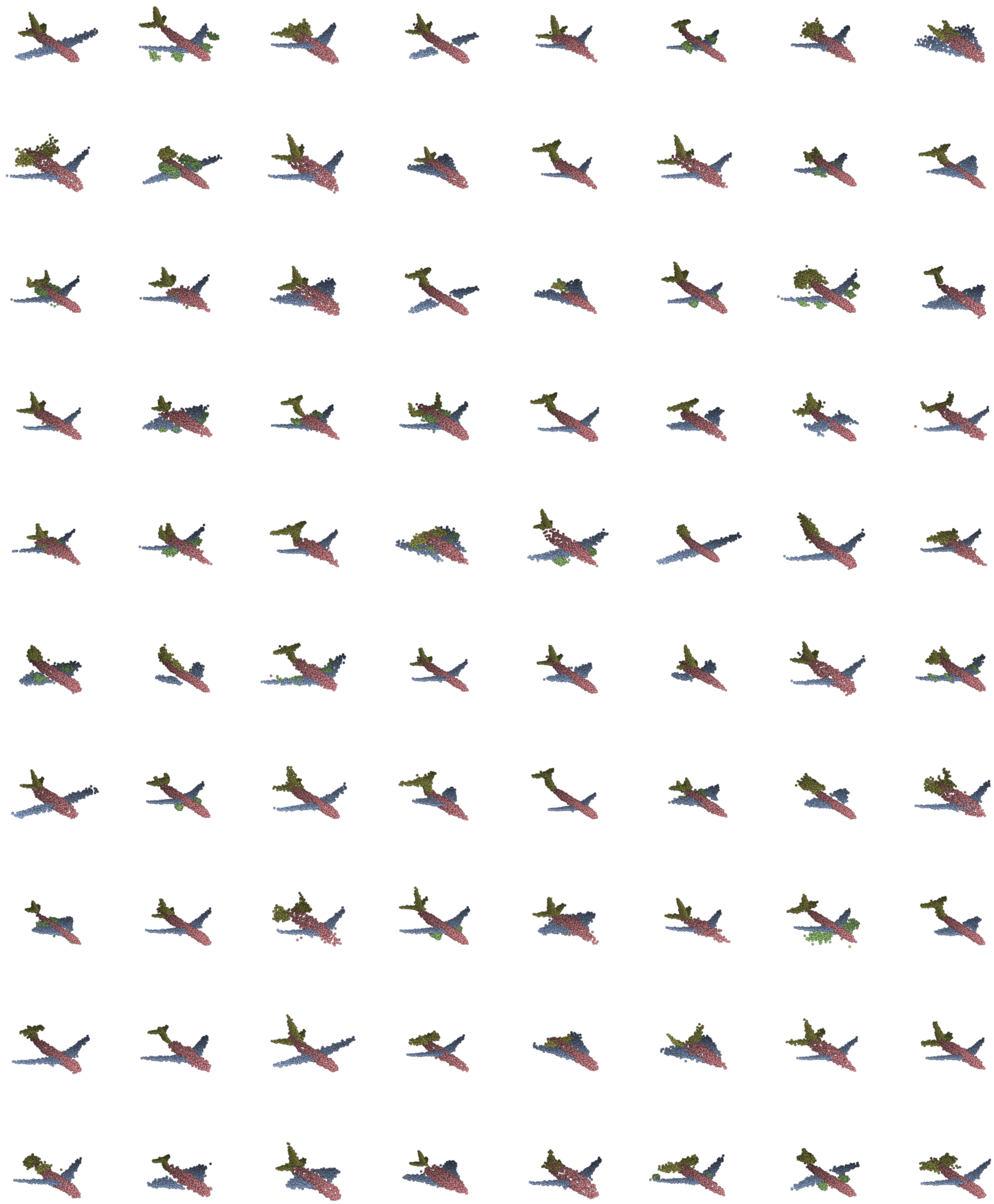

Figure 2: Airplane gallery. We present 80 randomly sampled airplanes from the $5,000$ which were generated by CompoNet.

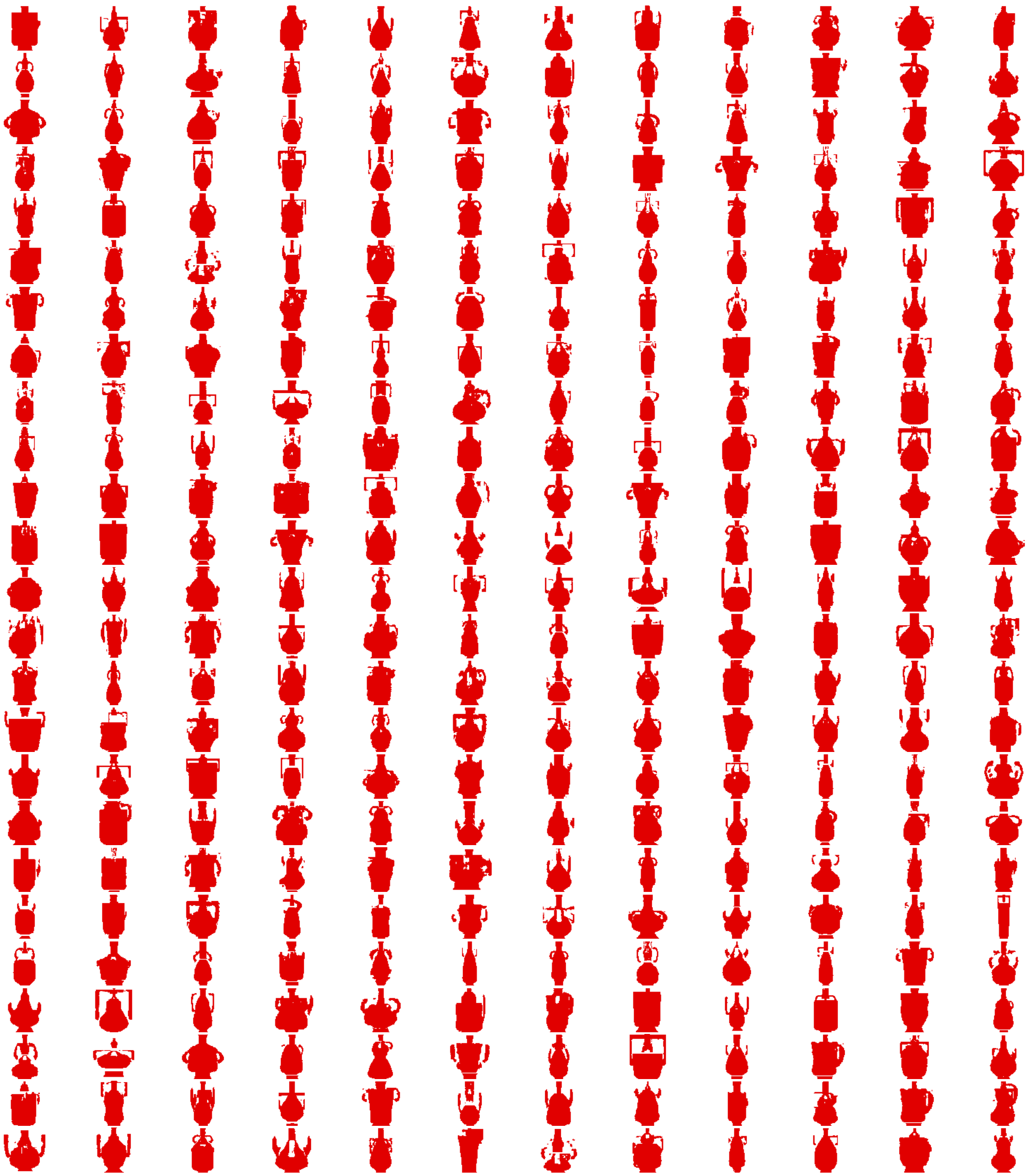

Figure 3: Vases gallery. We present 300 randomly sampled vases from the 1, 024 which were generated by CompoNet.

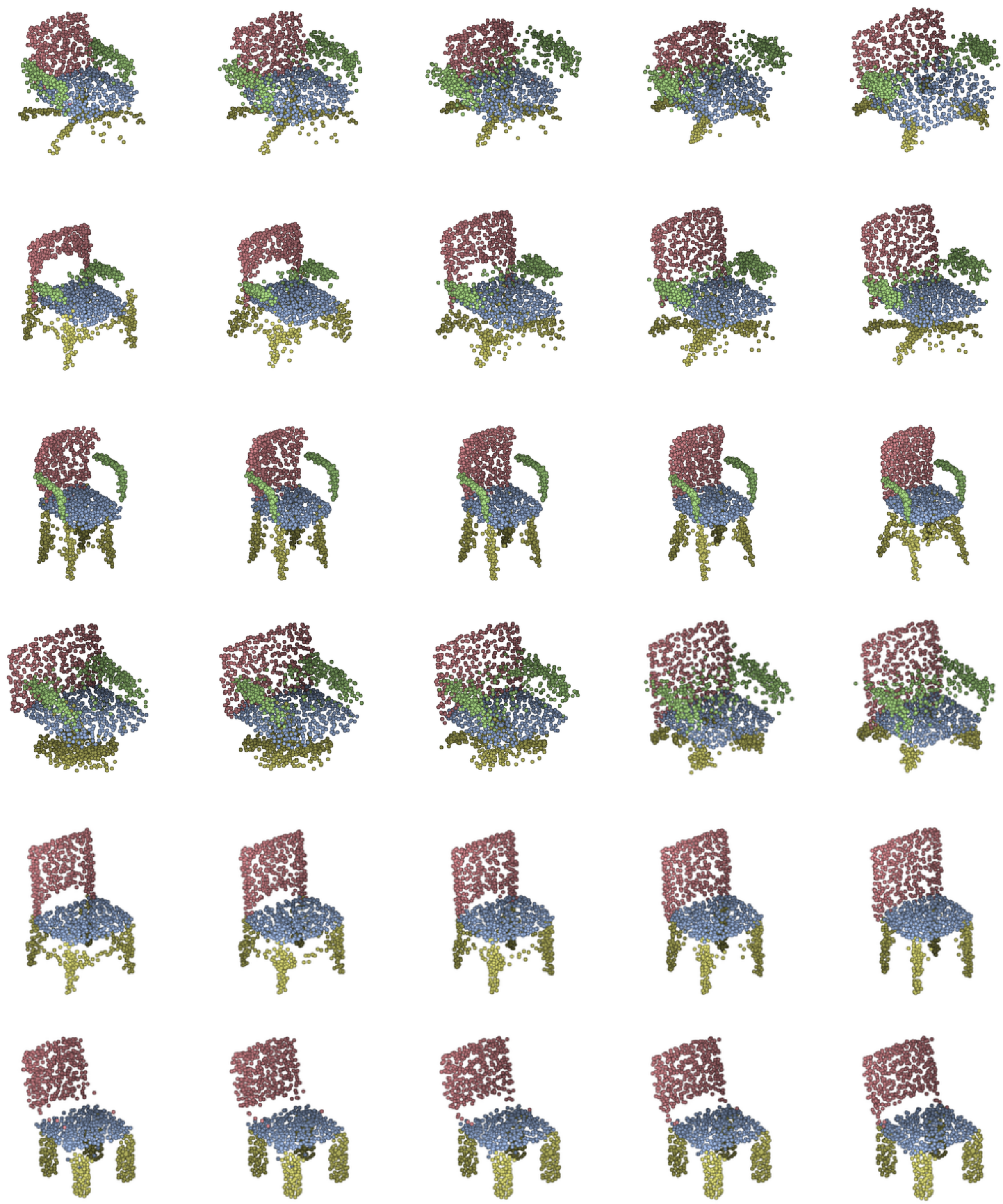

Figure 4: Complete latent space interpolation (parts and composition) for the chair category. The shape's code on the left is linearly interpolated to the shape's code on the right.

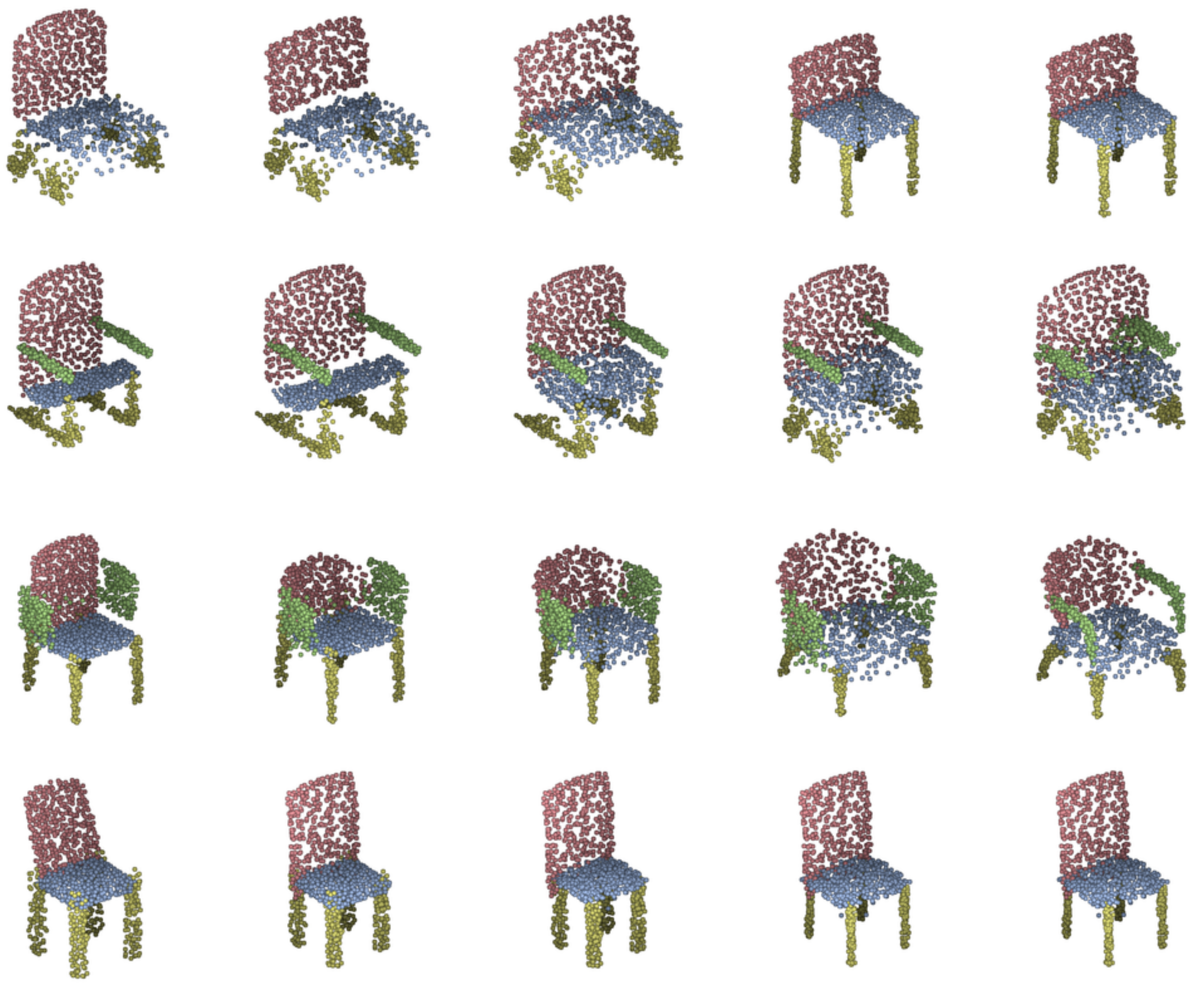

Figure 5: Chairs part-by-part interpolation. In each row we present part-by-part interpolation between two chairs (see left and right objects). The order of replacement is back-seat-legs-armrests. Please note, for certain pairs the arm-rests part does not exist, thus the interpolation of the arm-rest part is identity (no change).

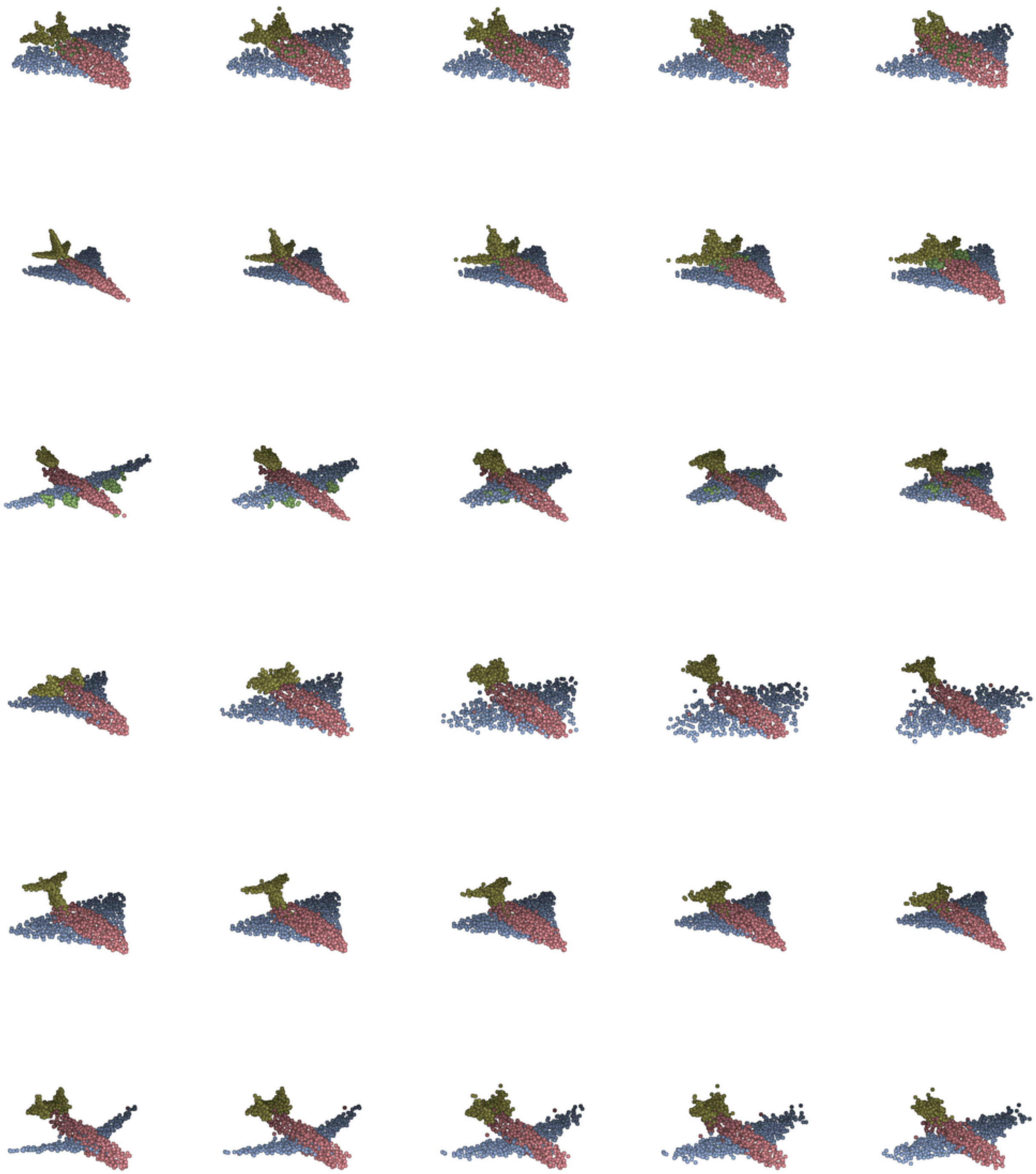

Figure 6: Complete latent space interpolation (parts and composition) for the airplane category. The shape's code on the left is linearly interpolated to the shape's code on the right.

Figure 7: Airplanes part-by-part interpolation. In each row we present part-by-part interpolation between two airplanes (see left and right objects). The order of replacement is body-wings-tail.

Figure 8: Chairs nearest-neighbors. We present random generated results from CompoNet and the baseline, for each result, we present its three nearest-neighbors based on the Chamfer distance. In the left half we show results from CompoNet (in colors), where the left column is generated and the next three columns to the right are its three nearest-neighbors. In the right half we present results from the baseline (gray), where the left column is generated and the next three columns to the right are its three nearest-neighbors.

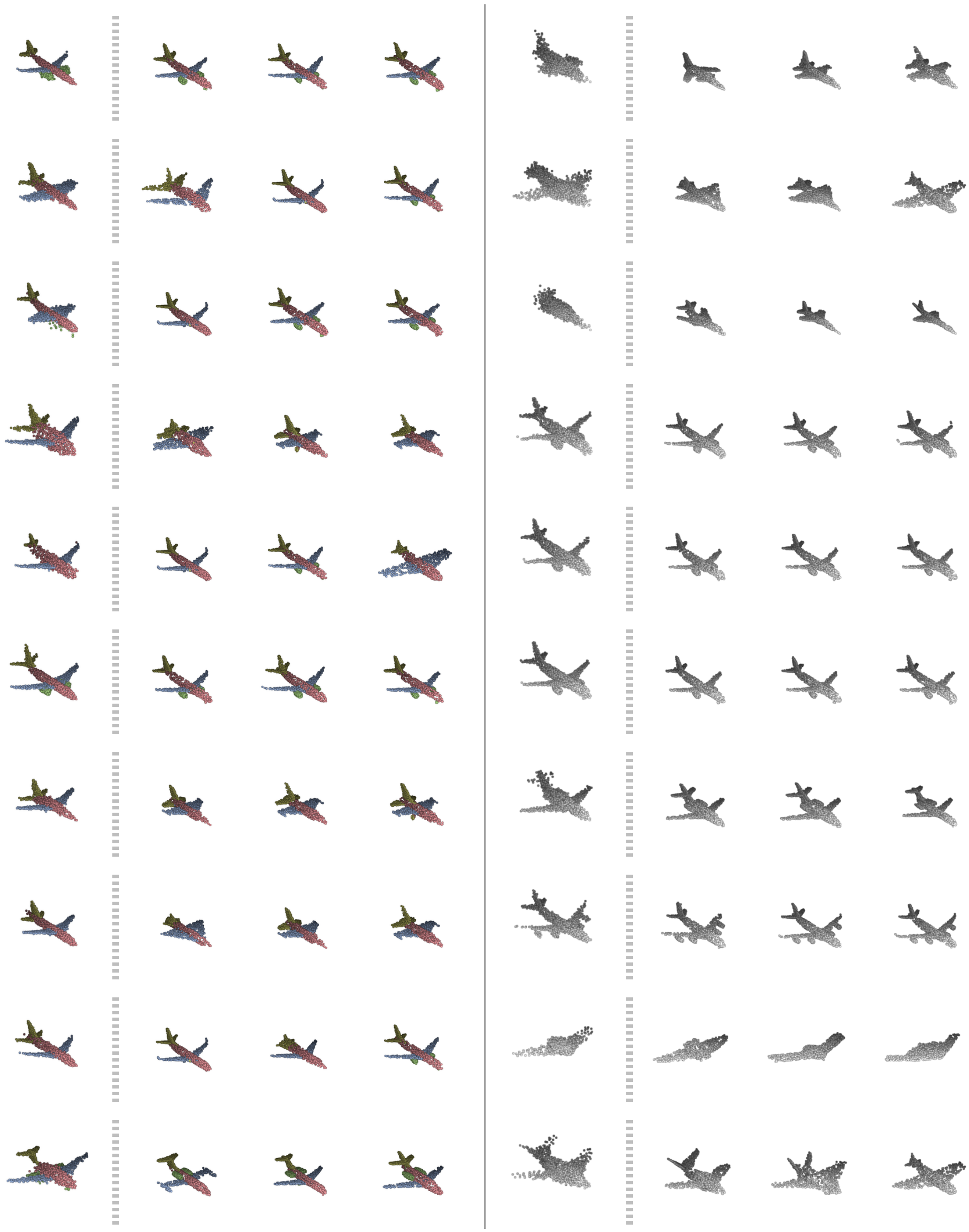

Figure 9: Airplanes nearest-neighbors. We present random generated results from CompoNet and the baseline, for each result, we present its three nearest-neighbors based on the Chamfer distance. In the left half we show results from CompoNet (in colors), where the left column is generated and the next three columns to the right are its three nearest-neighbors. In the right half we present results from the baseline (gray), where the left column is generated and the next three columns to the right are its three nearest-neighbors.