

LOGAN: Unpaired Shape Transform in Latent Overcomplete Space

Kangxue Yin¹, Zhiqin Chen¹, Hui Huang², Daniel Cohen-Or³, and Hao Zhang¹

¹Simon Fraser University ²Shenzhen University ³Tel Aviv University

A. Supplementary material

A.1. Details of network architecture

In this section, we provide details of the autoencoder, discriminator, and translator networks in LOGAN.

A set abstraction layer of PointNet++ is denoted as $SA(M, r, N, [l_1, \dots, l_d])$ where M is the number of local patches, r is the radius of balls that bound the patches, N is the number of sample points selected in each patch, $[l_1, \dots, l_d]$ are the widths of fully-connected layers used in local PointNet.

A fully connected layer is denoted as $FC(l)$ where l is its width. A dropout layer is denoted as $DROPOUT(p)$ where p is the probability that each element is dropped. We use $RELU$, BN and $MAXPOOL$ to represent ReLU activation layer, batch normalization layer and max pooling layer, respectively.

The architectures of our autoencoder, translator and discriminator are shown in Figure 1.

A.2. Visualization of latent space

As a supplement to the visualization of latent spaces in the paper, we show in Figure 2 the joint embedding of latent sub-vectors constructed by different autoencoders for *chair-table*. Same with the paper, the embedding is generated by t-SNE with hamming distance after discretize the values into integers.

A.3. Additional Chair-Table Transfer Results

We show more examples of *chair* \rightarrow *table* and *table* \rightarrow *chair* transfer in Figure 3 and Figure 4, respectively.

A.4. Additional Transfer Results on the datasets of P2P-NET

We show more examples of our method for non-trivial transformation task on the datasets of P2P-NET, including *skeleton* \rightarrow *shape*(Figure 5, Figure 6), *scan* \rightarrow *shape*(Figure 7, Figure 8), *cross-sectional profiles* \rightarrow *shape*(Figure 9, Figure 10).

A.5. Additional Font Style/Content Transfer Results

We show more examples of $G \rightarrow R$, $R \rightarrow G$, $thinG/R \rightarrow thickG/R$ and $thickG/R \rightarrow thinG/R$ transfer in Figure 11, Figure 12, Figure 13 and Figure 14, respectively. Our method outperforms other methods in $G-R$ translation tasks and achieves competitive results in $thinG/R-thickG/R$ translation.

A.6. Additional Armchair-Armless Chair Transfer Results

We show more examples of *armchair* \rightarrow *armless chair* and *armless chair* \rightarrow *armchair* transfer in Figure 15 and Figure 16, respectively.

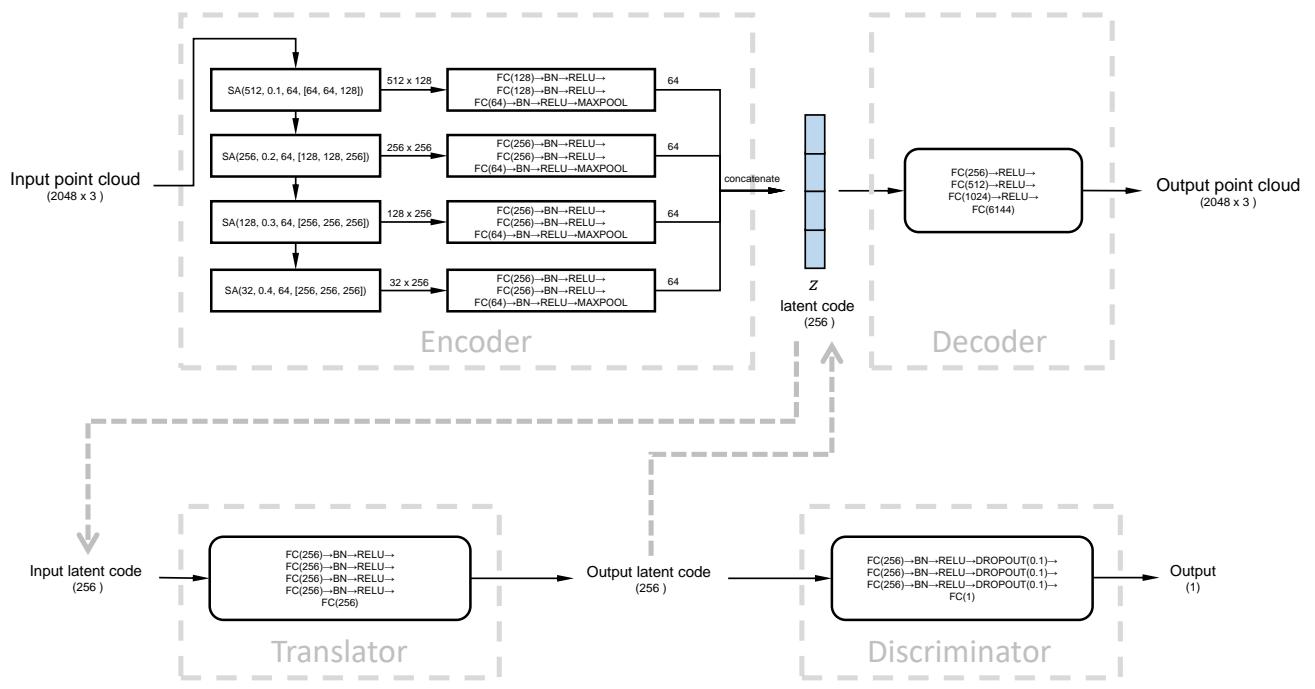


Figure 1. The architectures of our autoencoder, translator and discriminator.

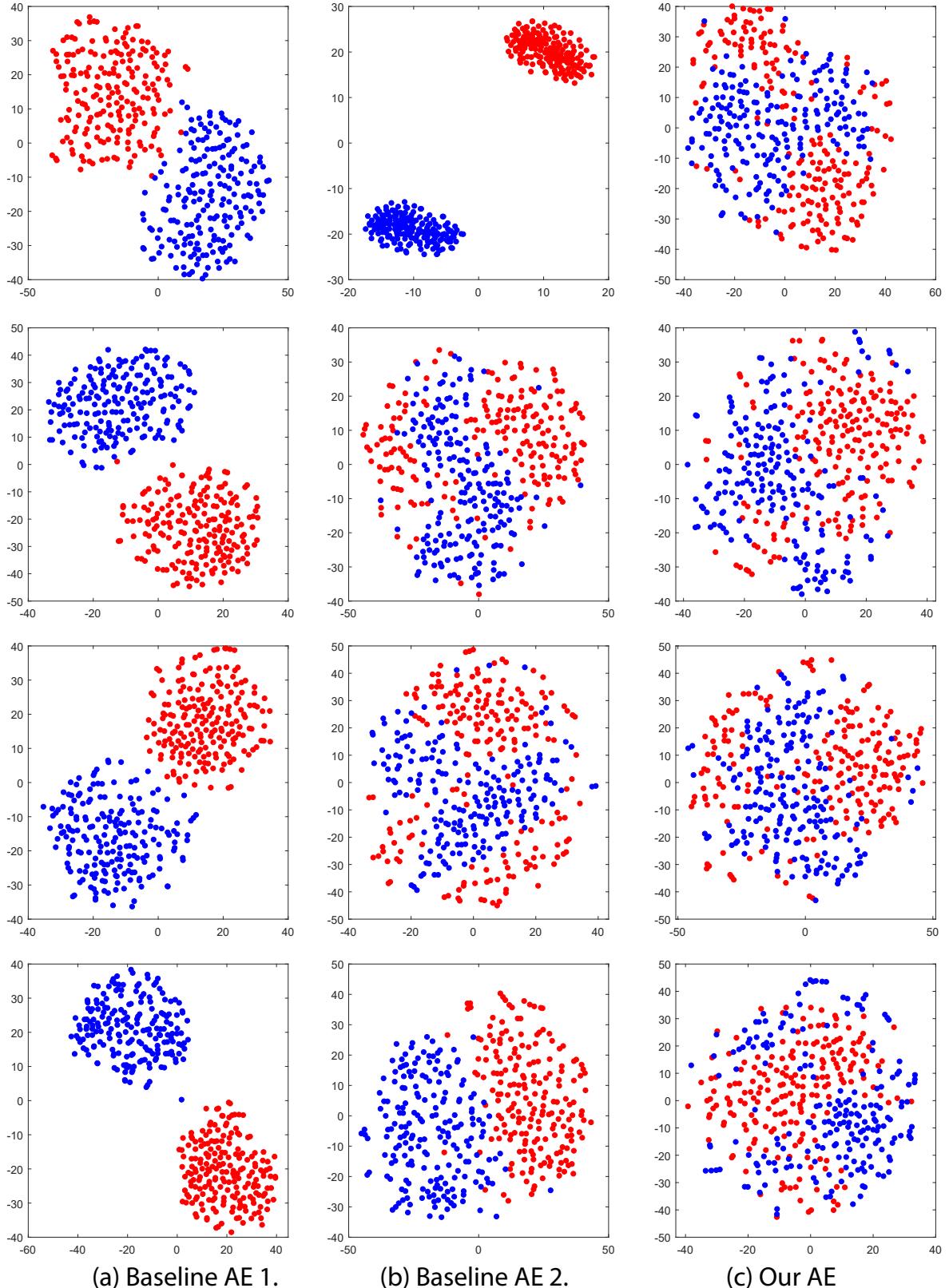


Figure 2. Joint embedding of the latent sub-vectors generated by our autoencoder(c) for chairs (red) and tables (blue). From top down, we show embedding result for z_1, z_2, z_3, z_4 . For comparison, we also provide the embedding result for the corresponding sub-vectors in the codes generated by the two baseline autoencoders(a,b) described in Section 4.1 of the paper.



Figure 3. Results for *chair* → *table*.

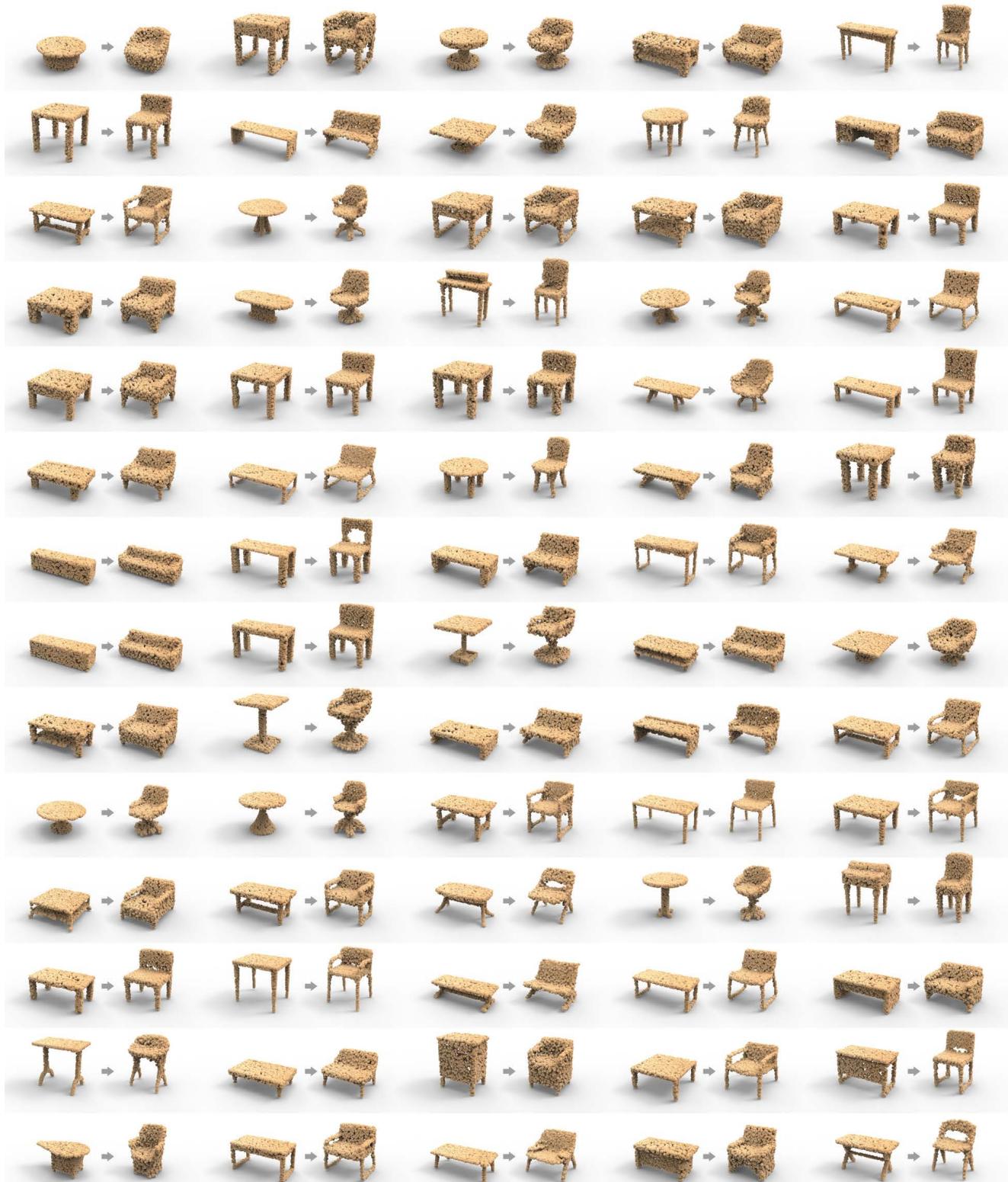


Figure 4. Results for *table* → *chair*.

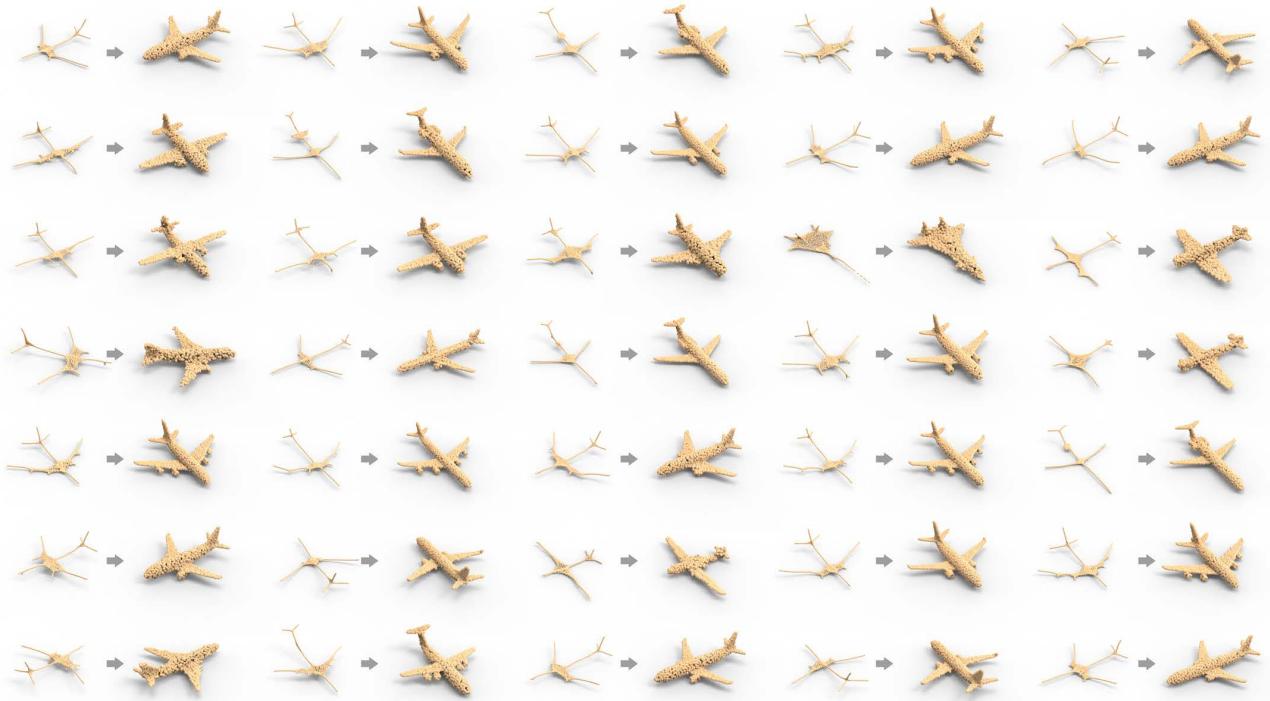


Figure 5. Results for *airplane skeleton* → *airplane*.

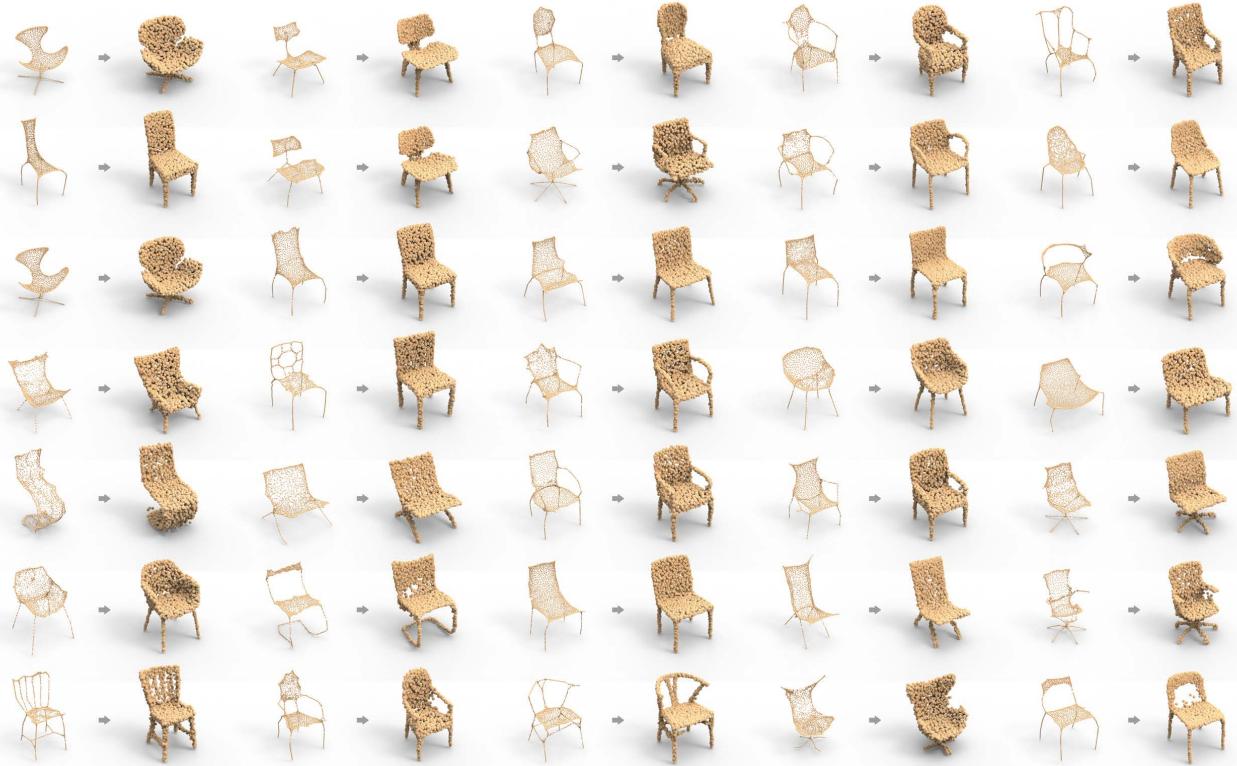


Figure 6. Results for *chair skeleton* → *chair*.

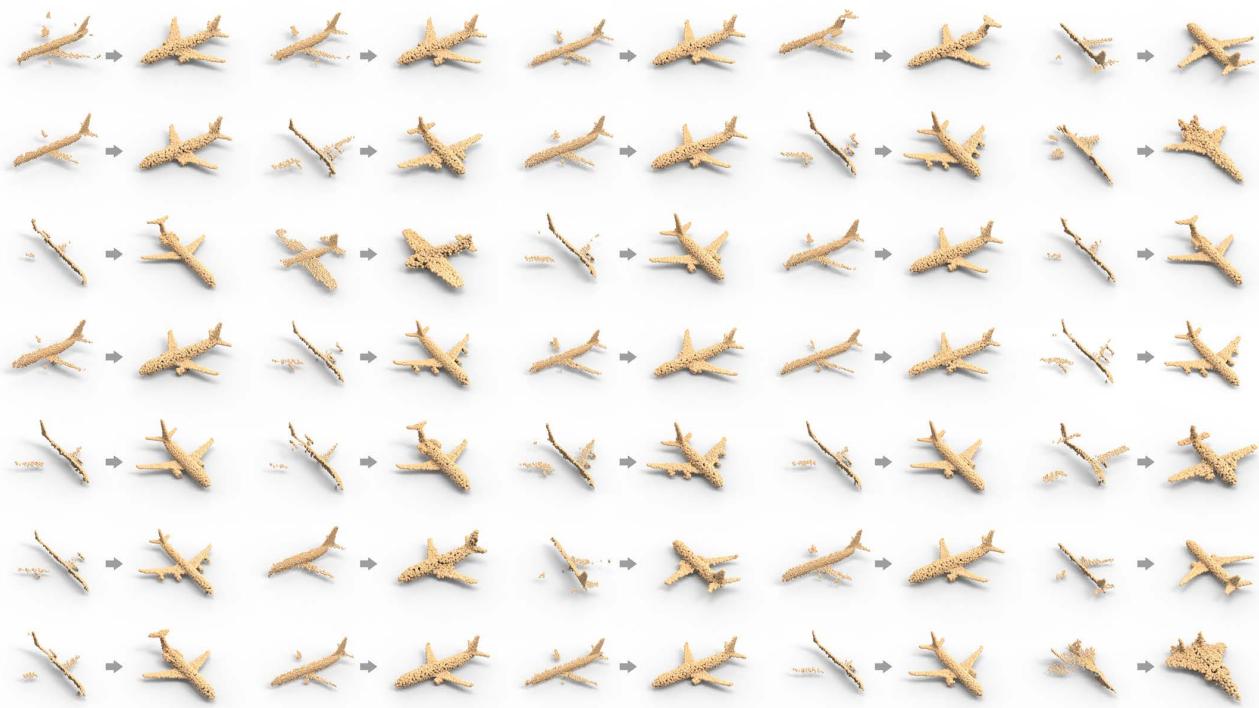


Figure 7. Results for *airplane scan* → *airplane*.

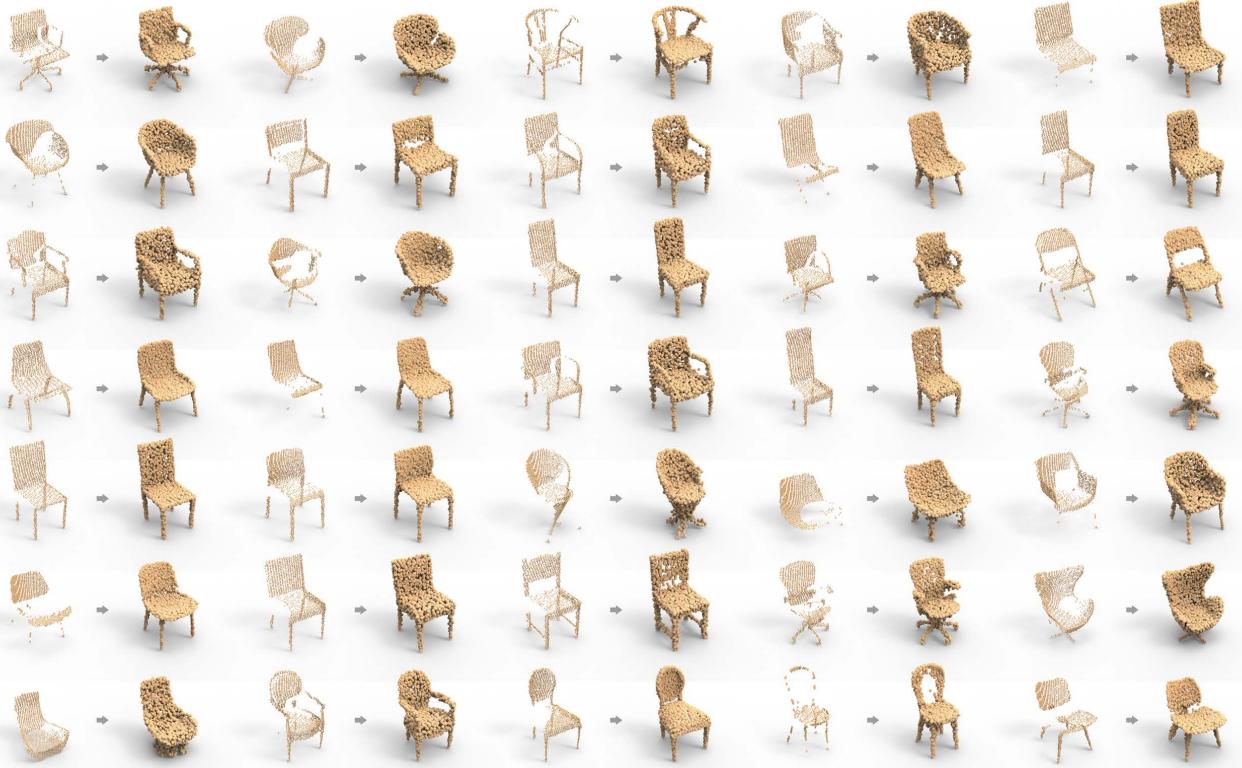


Figure 8. Results for *chair scan* → *chair*.

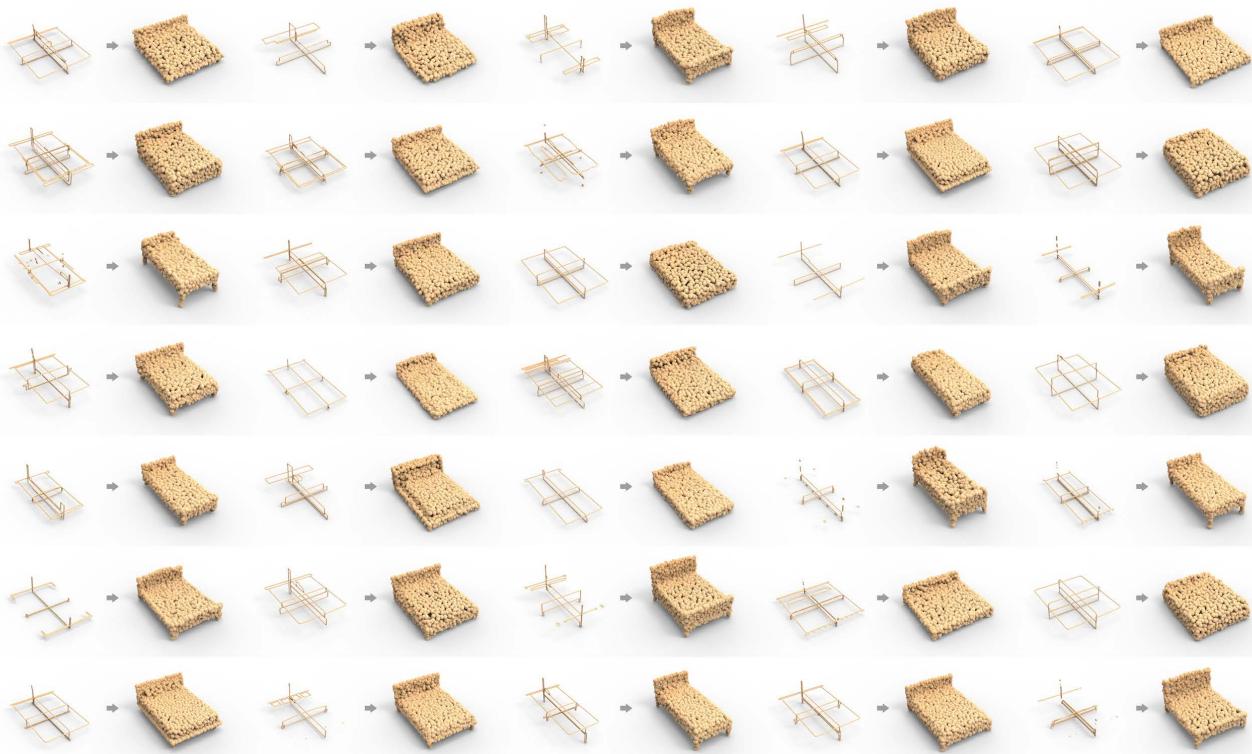


Figure 9. Results for *bed cross-sectional profiles* → *bed*.

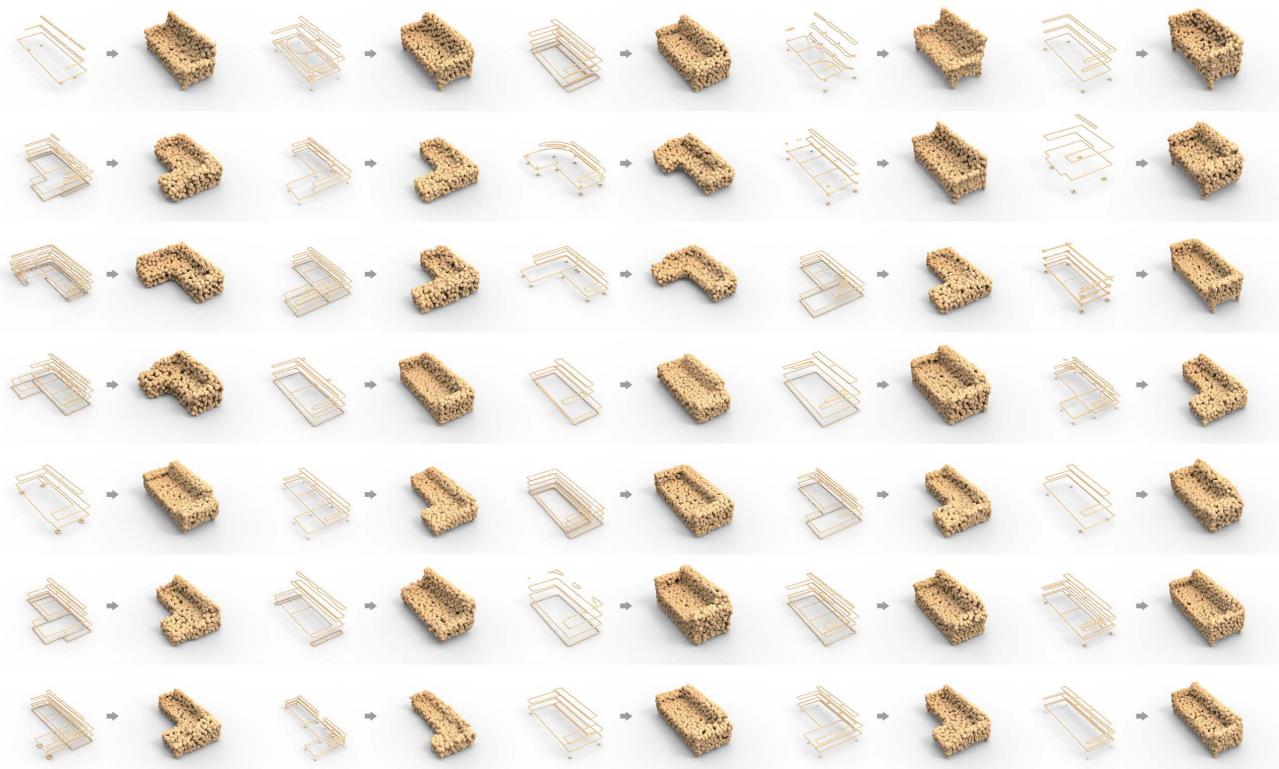


Figure 10. Results for *sofa cross-sectional profiles* → *sofa*.



Figure 11. Qualitative comparisons for $G \rightarrow R$.



Figure 12. Qualitative comparisons for $R \rightarrow G$.



Figure 13. Qualitative comparisons for *thinG/R* \rightarrow *thickG/R*.



Figure 14. Qualitative comparisons for *thickG/R* \rightarrow *thinG/R*.

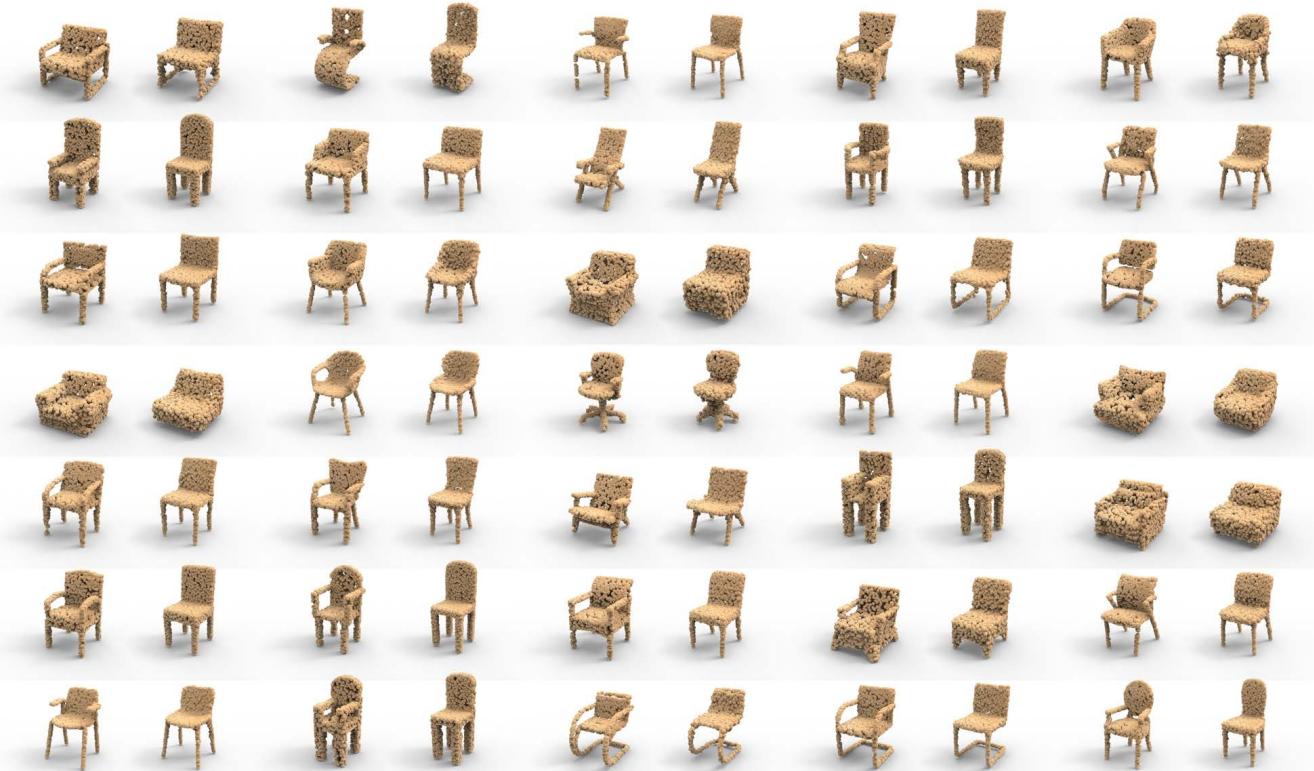


Figure 15. Results for *armchair* \rightarrow *armless chair*.

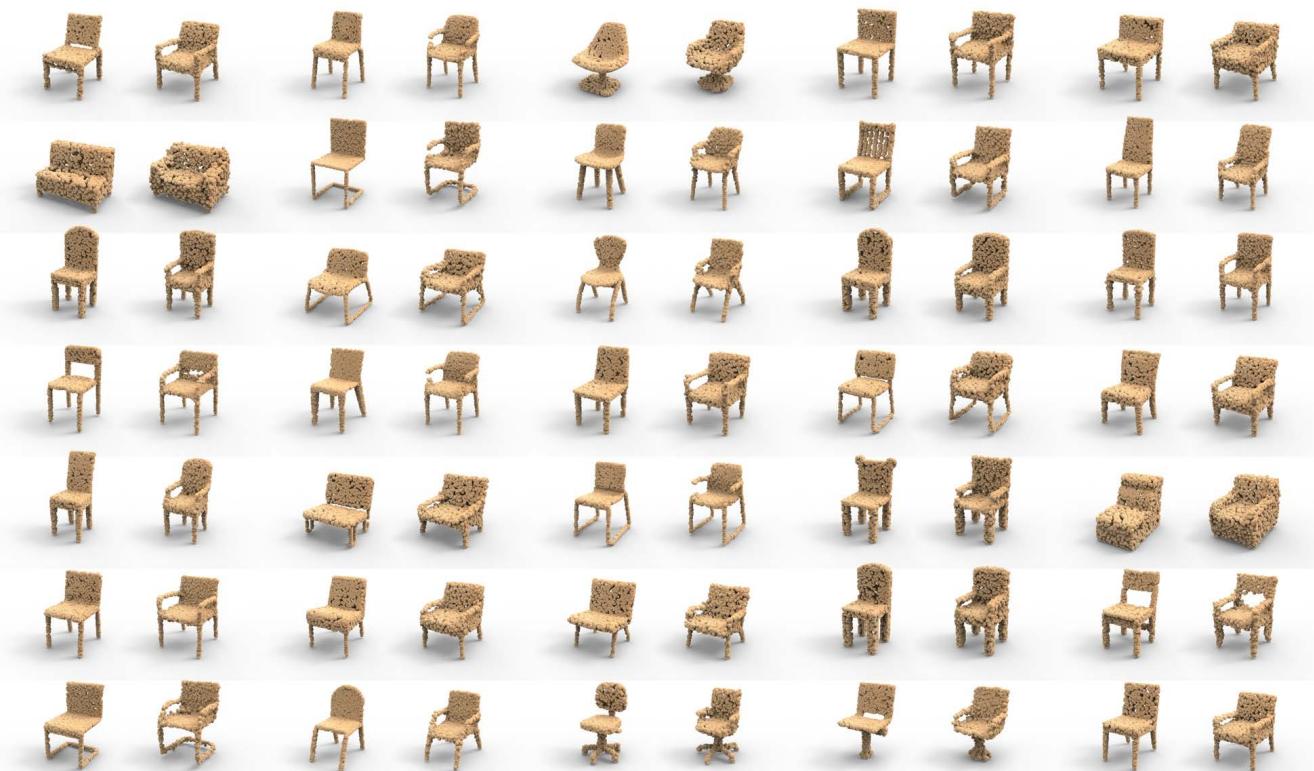


Figure 16. Results for *armless chair* \rightarrow *armchair*.