# Supplementary Materials of Characteristic-preserving Latent Space for Unpaired Cross-domain Translation of 3D Point Clouds

Jia-Wen Zheng, Jhen-Yung Hsu, Chih-Chia Li, and I-Chen Lin, *Member, IEEE*

## 1 Choice of the Distance

Our autoencoder network has to compute the cost or distance between two point clouds. Both Chamfer distance (CD) [3] and Earth Mover's Distance (EMD) [7] are popularly used methods.

The chamfer distance calculates the squared distance between each point in one point cloud to its closest point in the other point cloud, chamfer distance is defined as:

$$\begin{aligned} d_{CD}(P_1, P_2) = \sum_{p_1 \in P_1} \underset{p_2 \in P_2}{min} \; ||p_1 - p_2||_2^2 \\ + \sum_{p_2 \in P_2} \underset{p_1 \in P_1}{min} \; ||p_1 - p_2||_2^2. \end{aligned} \tag{1}$$

Chamfer distance is more efficient for training, but in our experiments, we found that point clouds trained by the chamfer distance suffer from an uneven distribution problem. The points easily gather at the flat parts. Hence, as described in the main paper, we employed the EMD distance in our loss. EMD distance guarantees one-to-one mapping when the number of two point clouds is identical. It makes the generated points more uniformly distributed.

## 2 Detailed Description about the Tree-based Generator

In our generator, to synthesize point clouds with detailed geometric structure, we considered upsampling methods instead of taking numerous fully connected layers directly. We took the tree-structure generator proposed in treeGAN [8] as the backbone. The tree structure ensures that the nodes with the same ancestor have similar geometry structure. We used seven layers of tree-based graph convolution (TreeGCN) [8] to generate our point clouds.

TreeGCN is composed of an activation function $\sigma$, a network for loop term $F_K$, a linear mapping for ancestor term $U$, and a bias $b$, and it is defined as:

$$p_i^{l+1} = \sigma(F_K^l(p_i^l) + \sum_{q_j \in A(p_i^l)} U_j^l q_j + b^l), \tag{2}$$

where $p_i^l$ is the $i$-th node of the tree structure in $l$-th layer, $q_j$ is the $j$-th ancestor of $p_i^l$, and $A(p_i^l)$ is the set of all ancestors of $p_i^l$. As shown in Fig. 1, the nodes with same ancestors can preserve similar feature by the ancestor term. The loop term makes the nodes with same ancestors have diversity. By this structure, we can generate point clouds with detailed geometric structure.

We regarded our latent code $Z$ as the root of our tree-structure generator and presented it as $p_1^0$. The set of nodes $\{p_i^L \; \forall i \in [1, 2, \ldots, n]\}$ in the final layer $L$ is our output point cloud with $n$ nodes. To upsample the number of nodes in each layer, we took the branch matrix $V$ to generate child nodes from a single node. Taking the $i$-th node $p_i^l$ in $l$ layer for an example, we defined the branch as:

$$p^{l+1} = \{V_{ij}^{l+1} \cdot p_i^l \; \forall j \in [1, 2, \cdots, d_l]\}, \tag{3}$$

where $d_l$ is the upsampling degree in $l$ layer. We adopted the degrees $\{2, 2, 2, 2, 2, 4, 16\}$ for seven layers, respectively, and it generated 2048 nodes in the final layer.

Inspired by [5], we took advantage of AdaIN [4] to enhance the shape information in every layer except for the final layer. For $l$-th layer, we mapped our latent code $z$ into a vector containing scale coefficient $\sigma^l$ and translation coefficients $\mu^l$ via a shape mapping function which is { FC-256 $\rightarrow$ leakyReLU $\rightarrow$ FC-256 $\rightarrow$ leakyReLU $\rightarrow$ FC }. Hence, in $l$-th layer, we obtained the nodes $\tilde{p}^l$ as:

$$\tilde{p}^l = \frac{p^l - \mu(p^l)}{\sigma(p^l)} \cdot \sigma^l + \mu^l, \tag{4}$$

where $\mu(p^l)$ and $\sigma(p^l)$ are mean and variance of $p^l$. As shown in the following experiments, with the AdaIN layers, we can preserve more shape details.

- *J.-W. Zheng, J.-Y. Hsu, C.-C. Li, and I.-C. Lin are with College of Computer Science, National Yang Ming Chiao Tung University, Taiwan.*

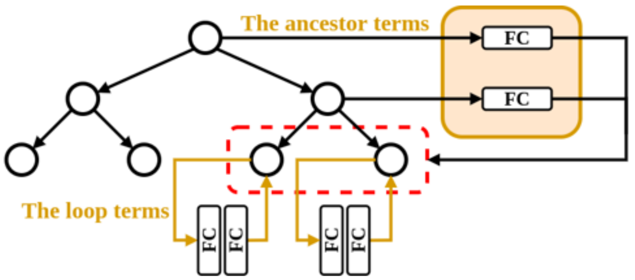

Fig. 1: The concept about treeGCN layers. Taking the nodes in the second layer as an example, the nodes with the same ancestors, e.g. the nodes in the red circle, learn the same features from the ancestor term. The loop term enhances the difference between the sibling nodes.

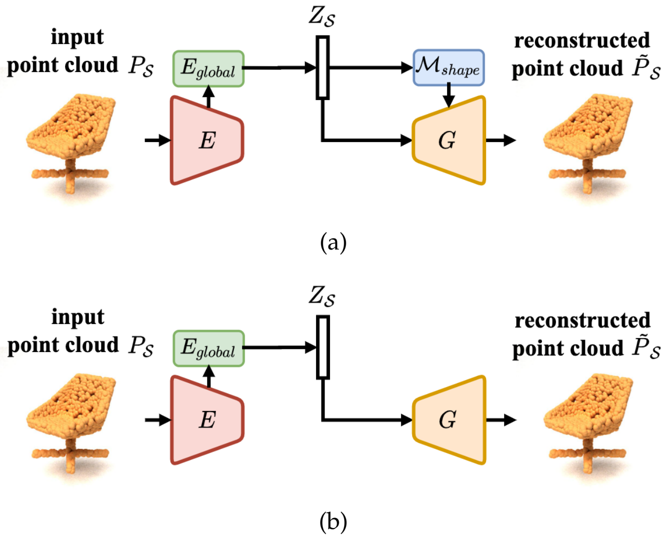

Fig. 2: (a) The overview architecture of our autoencoder. (b) The overview architecture of the autoencoder in *model A* for ablation study, in which we removed our shape mapping function $\mathcal{M}_{shape}$.

## 3 Additional Qualitative Ablation Studies

### 3.1 AdaIN and Multi-scale translator

We constructed two additional network structures, *model A* and *model B*, to investigate the effectiveness of our shape mapping function $\mathcal{M}_{shape}$ (for AdaIN) and multi-scale translator $\{T^1_{scale}, T^2_{scale}, T^3_{scale}, T^4_{scale}\}$.

The *model A* involved two modifications. While fixing the encoder in our framework, we first removed all $\mathcal{M}_{shape}$ of our generator as depicted in Fig. 2b. In the second modification, we replaced the proposed multi-scale translators, where four sub-translators are applied, with a global translator as depicted in Fig. 3b. Based on *model A*, we put $\mathcal{M}_{shape}$ back to corresponding layers of our generator as the *model B*.

As shown in Fig. 4, there are several vague parts and noisy points in the reconstructed and the transferred point clouds of *model A*. In *model B*, the $\mathcal{M}_{shape}$ enhances the generator to synthesize more clear shape characteristics than results of *model A*. However, the transferred point clouds of *model B* miss certain distinct shape characteristics, such as the four branches of the table base. With our complete architecture, where multi-scale translators are applied, the transferred shape characteristics are more distinct and closer to the originals than those by global translators in *model B*.

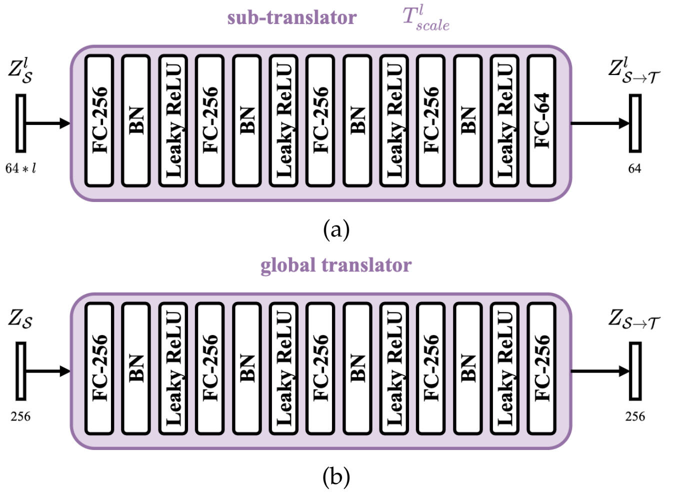

Fig. 3: Comparison of the translator architecture. (a) Our sub-translator. Four sub-translators are applied to transfer latent codes of four scales. (This sub-figure is identical to Fig.6(b) in the main manuscript.) (b) Global translator used in *model A* and *model B* for ablation study.

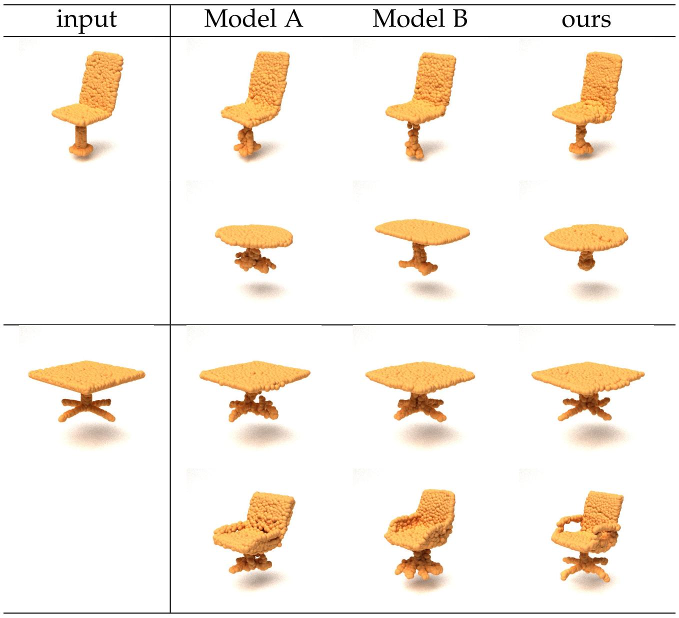

Fig. 4: Results of reconstruction (the first row and the third row) and translation (the second row and the fourth row) by different architectures for ablation study. The upper and lower halves show the results of chair-to-table transfer and table-to-chair transfer, respectively.

### 3.2 Training Strategies

We also trained our framework with different strategies to compare our alternate training to the separate training

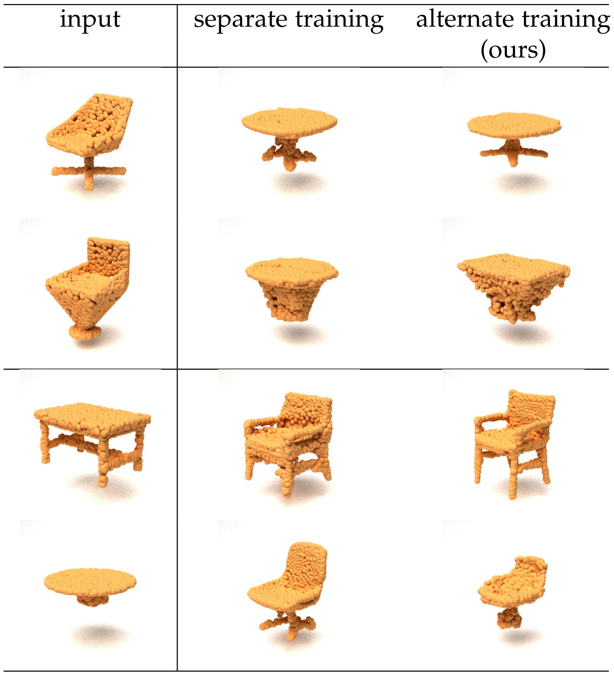

Fig. 5: Transferred results by our model trained with different training strategies. The upper and lower halves show results about chair-to-table transfer and table-to-chair transfer, respectively.

used in LOGAN [9]. In LOGAN, they first trained the autoencoder to produce its latent space and then trained the translators in the fixed latent space. However, the results of separate training highly depended on the diversity of shape characteristics in the latent space. As shown in Fig. 5, the latent space learned from the autoencoder with the separate training is not able to make the following translators preserve the detailed shape characteristics such as the inverted pyramid shape of the chair base and the single short leg of the table.

### 3.3 Test on *Paired Arm-and-armless Chairs* Dataset

As mentioned in the main manuscript, to objectively assess the unpaired translation methods, we constructed the *Paired Arm-and-armless Chairs* test dataset. This small test dataset is composed of thirty armchair data from ShapeNet and thirty corresponding armless chairs, which are crafted manually and unseen in ShapeNet. When we train a framework with the training split (1710 armchairs and 2875 armless chairs), and translate an armchair to an armless chair, the manually crafted armless version can become the pseudo ground truth, and vice versa.

Fig. 6 compares the results of our translation framework with different configurations of modules, Ours(MVS) and LOGAN [9]. Fig. 7 compares the results of our translation framework with different configurations of loss functions, Ours(MVS) and LOGAN [9].

| | similarity(%) | fidelity(%) |
|---|---|---|
| ours | **50.97** | **40.41** |
| LOGAN [9] | 29.72 | 25.00 |
| both are plausible | 7.50 | 22.91 |
| both are not plausible | 11.80 | 11.66 |

TABLE 1: User evaluation about preferences of transferred point clouds by the proposed method versus those by LOGAN [9] in terms of style *similarity* to the input and point cloud *fidelity* for the target domain, respectively.

| | *chair→table→chair* | | *table→chair→table* | |
|---|---|---|---|---|
| Model/Distance | CD | EMD | CD | EMD |
| Ours | 3.81 | 9.51 | 4.63 | 9.98 |
| OurCB | 1.88 | 7.01 | **1.89** | 7.26 |
| LOGAN [9] | 3.30 | 8.54 | 2.76 | 7.43 |
| UNIST [2] | **1.54** | **4.61** | 2.04 | **4.50** |

TABLE 2: Statistics on the cycle-reconstruction error of the proposed framework (Ours), LOGAN [9], and UNIST [2] on the chair-table dataset. OurCB (cycle-loss-boosted) denotes our framework trained with a double weight on cycle-consistency loss. The reported CD (chamfer distance) scores are multiplied by $10^3$ and EMD (earth mover's distance) scores are multiplied by $10^2$.

## 4 Additional Experiment Results

### 4.1 User Evaluation

We also compared our approach to LOGAN [9] through user evaluation about the transferred point clouds. We randomly selected and generated thirty sets of results about chair-to-table transfer and thirty sets of results about table-to-chair transfer.

Fifty-eight volunteers participated in the evaluation. thirty-two of them are familiar with 3D models from movies or games, and seventeen of them are experienced in 3D modeling. After looking around the input point clouds and the transferred results by two methods, volunteers then voted for their preferred results in terms of style *similarity* to the input and point cloud *fidelity* for the target domain.

The percentages of similarity and fidelity are listed in Table 1. More volunteers preferred our results in both similarity and fidelity. We also performed t-test: paired two sample for means (one-tail) on the votes (hypothesis $H_0$: LOGAN is better and $H_1$: the proposed is better). The p-values for similarity and fidelity are $0$ and $2 \times 10^{-7} (<.05)$. The advantages of the proposed work are significant.

### 4.2 Argument about Cycle-reconstruction Error for Evaluating Translation

It is a challenging task to objectively evaluate the translation quality with a large dataset without ground truth. One possible thought is to evaluate the reconstruction error between a source and its cycle-transferred model. For example, for chair→table translation, evaluating the distance between a source chair and the chair generated by chair→table and then table→chair translation. Hence, we conducted an experiment to discuss the use of cycle-reconstruction error for evaluating translation.

As shown in Table 2, we evaluated the cycle-reconstruction error of results by the proposed framework, LOGAN [9] and UNIST [2]. Our average cycle-reconstruction error is inferior to those of LOGAN and

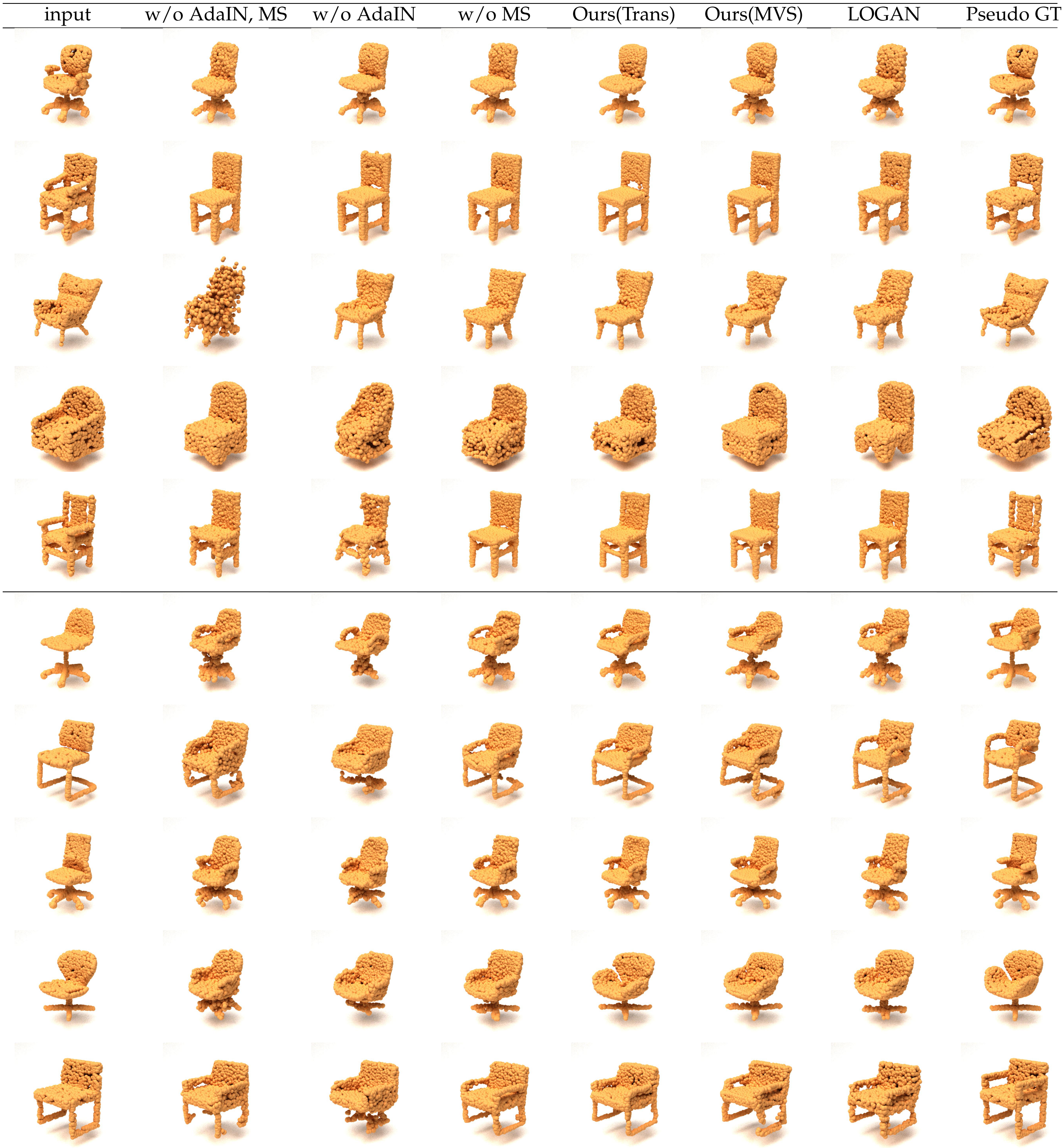

Fig. 6: Qualitative comparison among our models with different configurations of modules and LOGAN [9]. These translation methods were trained with the armchair and armless chair training split and tested on the *Paired Arm-and-Armless Chairs* dataset. The first five rows are the results for armchair → armless chair, while the last five rows are the results for armless chair → armchair. The first and last columns are the input and pseudo ground truth pairs. The fifth column represents the results of our full model, while the 2-4th columns represent the ablation of AdaIn and Multi-Scale (MS) modules used in the generator and encoder. The sixth column shows the results by our mean-vector-shift (MVS) operation, where the model was trained on the chair-table datasets. The seventh column represents the results of LOGAN.

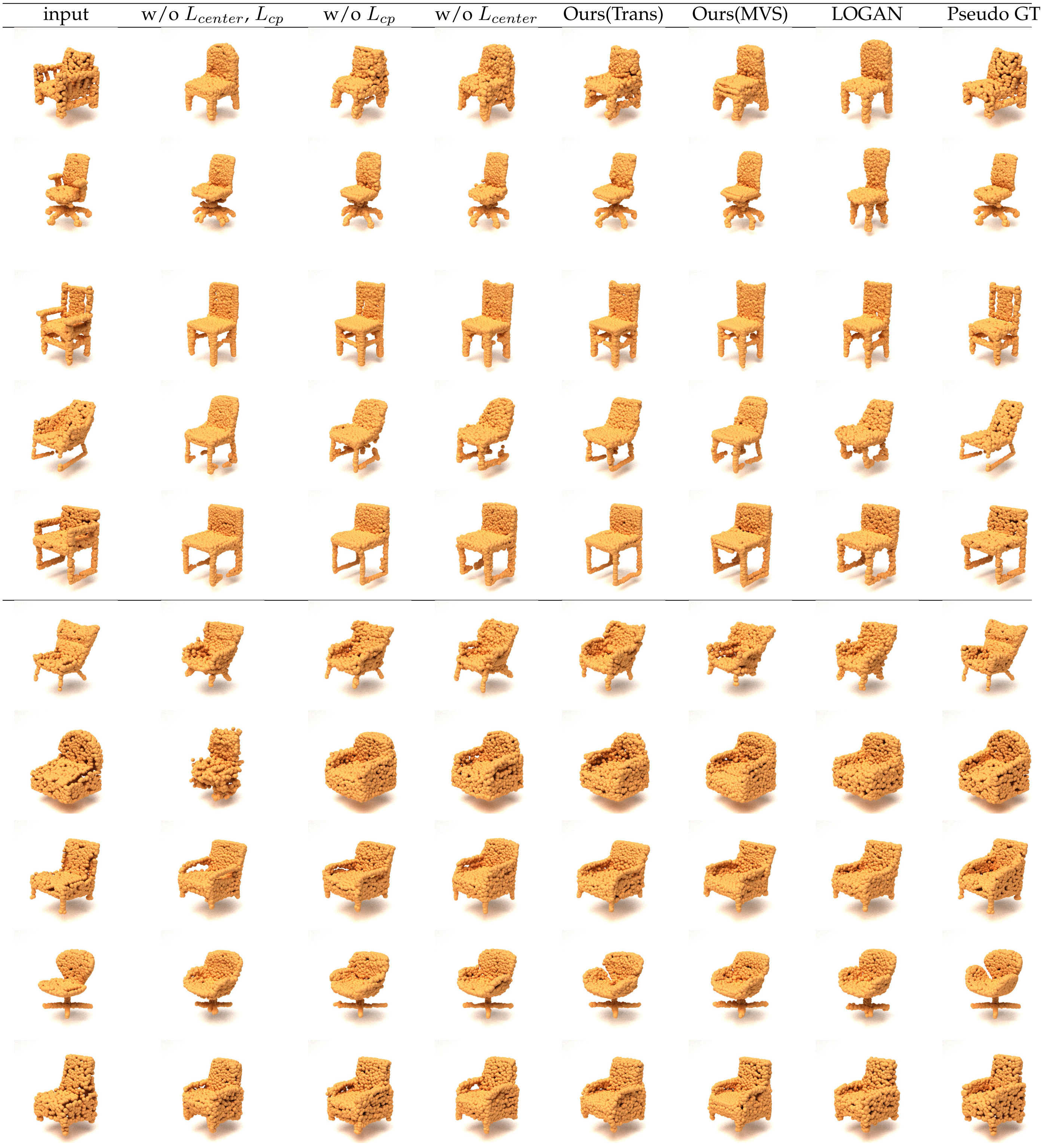

Fig. 7: Qualitative comparison among our models with different configurations of loss functions, Ours(MVS) and LOGAN [9]. The translation methods (except MVS) were trained with the armchair and armless chair training split and tested on the *Paired Arm-and-Armless Chairs* dataset. The first five rows are the results for armchair → armless chair, while the last five rows are the results for armless chair → armchair. The first and last columns are the input and pseudo ground truth pairs. The fifth column represents the results of our full model, while the 2-4th columns represent the ablation of $L_{center}$ and $L_{cp}$. The sixth column shows the results by our mean-vector-shift (MVS) operation, where the model was trained on the chair-table datasets. The seventh column represents the results of LOGAN.

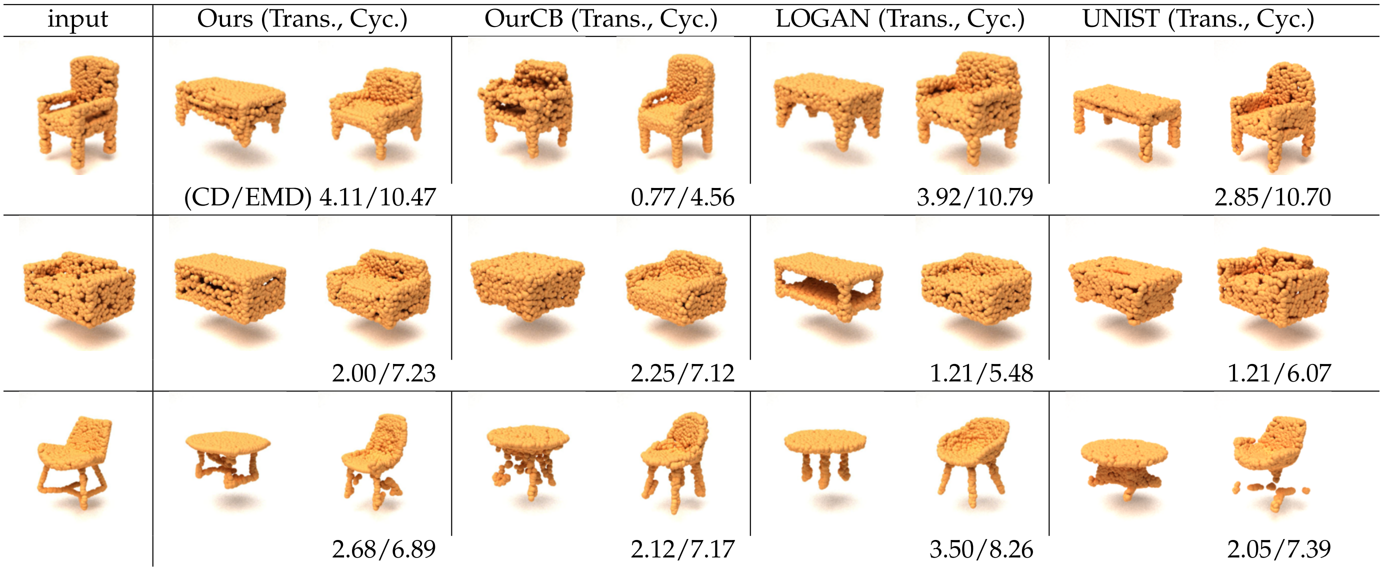

Fig. 8: Examples of chair→table translation (Trans.) by the proposed framework (Ours), LOGAN [9], and UNIST [2] and their corresponding chair→table→chair cycle-reconstructed models (Cyc.) and cycle-reconstruction error in terms of (CD/EMD). OurCB (cycle-loss boosted) denotes our framework trained with a double weight on cycle-consistency loss. The reported CD (chamfer distance) scores are multiplied by $10^3$ and EMD (earth mover's distance) scores are multiplied by $10^2$. We found that the cycle-reconstruction error may not directly accord with the translation quality.

UNIST. When we simply doubled the weight of cycle-consistency error in our framework, the cycle-reconstruction error of such a cycle-loss-boosted version (OurCB) can be comparable to others and even superior in certain items. However, when we inspected the results generated by our cycle-loss-boosted framework and related methods, we found that the superiors in cycle-reconstruction error are not necessary to possess high translation quality.

Fig. 8 shows examples of translation results by our framework, our cycle-loss-boosted framework (OurCB), LOGAN and UNIST and their corresponding cycle-reconstructed models and cycle-reconstruction error. In the first row, OurCB generated the cycle-reconstructed model properly but its translation result is not as appealing as others. In the second row, LOGAN got the lowest cycle-reconstruction error, but it translated a solid and chunky chair into a table with hollow storage space under the table top. In the third row, UNIST got he lowest cycle-reconstruction error in CD but the model translated by our proposed framework keeps the thin and curved chair base.

Since the weight of cycle-consistency loss can influence the learning effort of a model in terms of cycle-reconstruction and the cycle-reconstruction accuracy is not always consistent with translation quality, we argue that the cycle-reconstruction error should not directly be applied as the metric to evaluate the quality of shape translation.

### 4.3 Additional Results of Shape Style Mixing

We can generate point clouds which consist of the coarse and fine shape features from two source point clouds by combining their coarse and fine portions of latent codes, respectively. Additional results of shape style mixing are shown in Fig. 9.

We can also combine coarse and fine features from two different domains, as shown in Fig. 10. Some of the cross-domain style-mixing results are with slightly more noise points compared to those of single-domain style-mixing. We think that is because some features are not perfectly mapped onto the opposite domains.

### 4.4 Additional Results of Shape Editing with Mean-Vector-Shift (MVS) operation

We present additional results of shape editing with MVS operation in Fig. 11. Taking short tables to tall tables as example, the MVS operation is formulated as:

$$Z_{i_{short\to tall\_table}} = Z_{i_{short\_table}} - \bar{Z}_{short\_table} + \bar{Z}_{tall\_table}, \quad (5)$$

where $\bar{Z}_{(.)}$ denotes the mean vector of a given set.

### 4.5 Additional Results of our Autoencoder

We demonstrate additional results of reconstructed chairs in Fig. 12 and results of reconstructed tables in Fig. 13.

### 4.6 Additional Results of Shape Transfer

#### *4.6.1 Chairs and Tables*

We show additional results and comparison about chair-to-table transfer in Fig. 14 and table-to-chair transfer in Fig. 15.

#### *4.6.2 Closest Model Retrieval for the Translated Results*

Fig. 16 shows translation results of chair-to-table and table-to-chair and also their Top-5 closest shapes in the target training dataset. It demonstrates that our proposed network is capable of transferring shapes and not simply retrieving training data from the target domain.

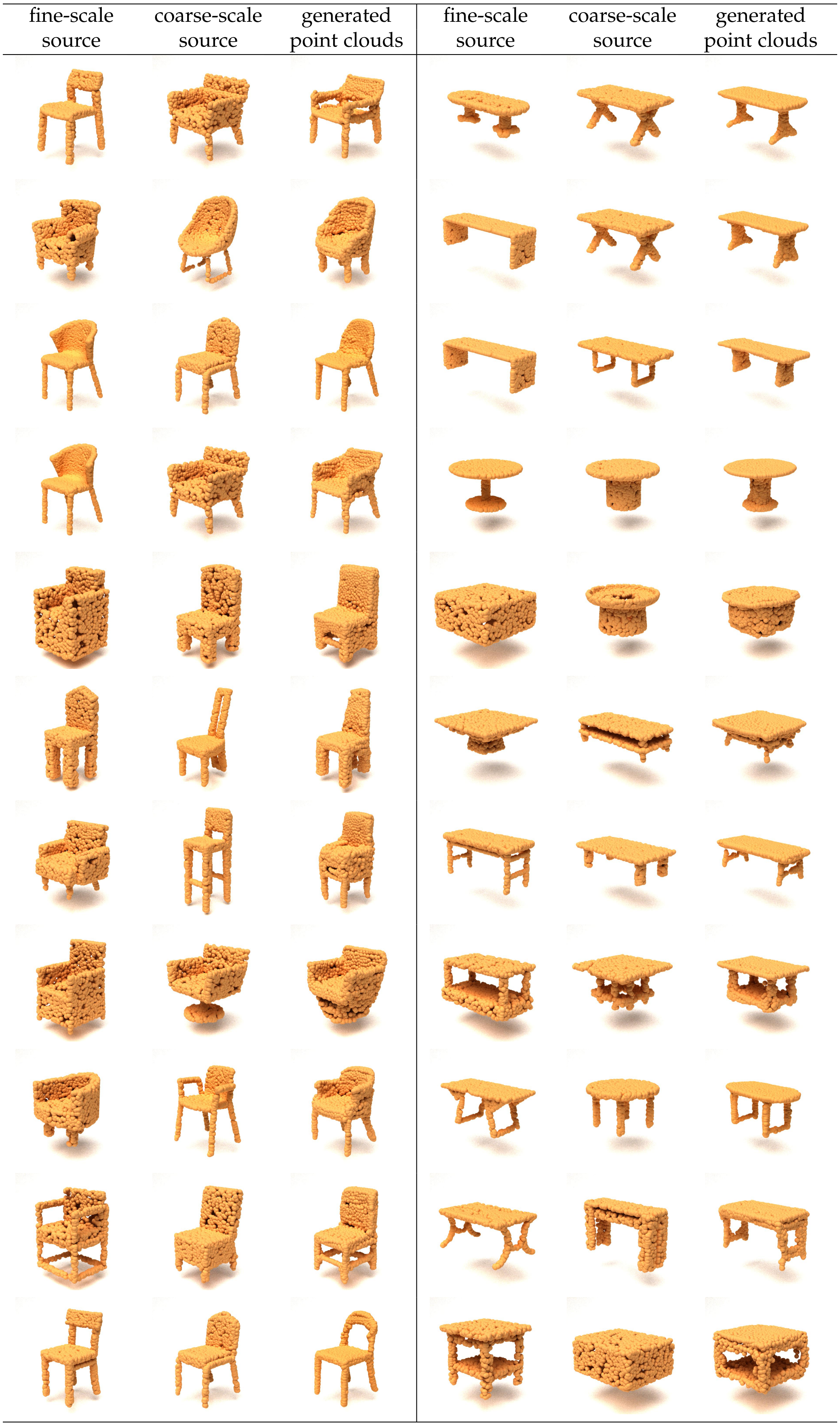

Fig. 9: Our results of shape style mixing. The left part demonstrates the mixing results of chairs. The right part shows the mixing results of tables.

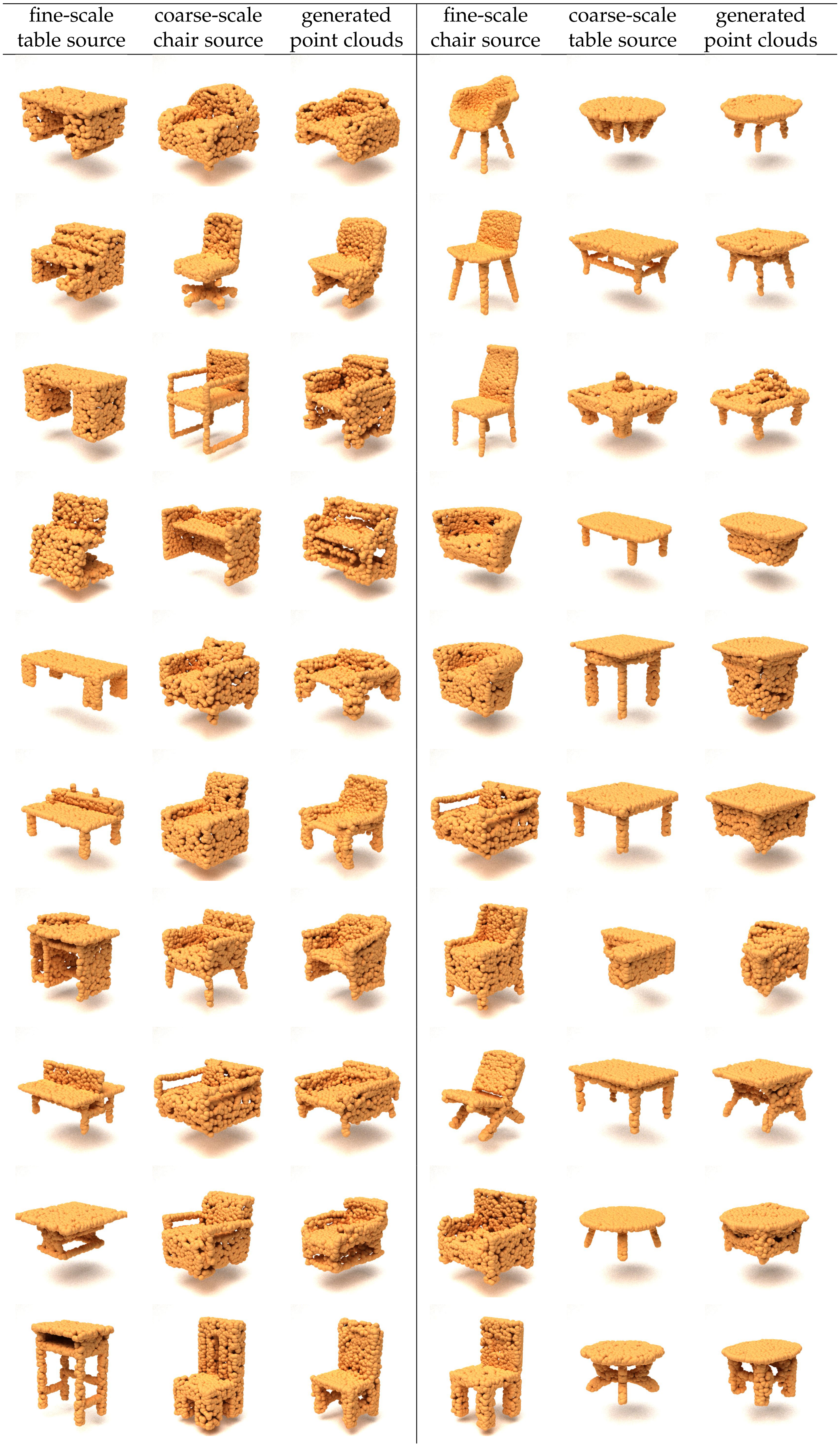

Fig. 10: The results of mixing coarse and fine scale features from two different domains. The left part shows the coarse-scale features from chairs and the fine-scale features from tables, and vice versa for the right part.

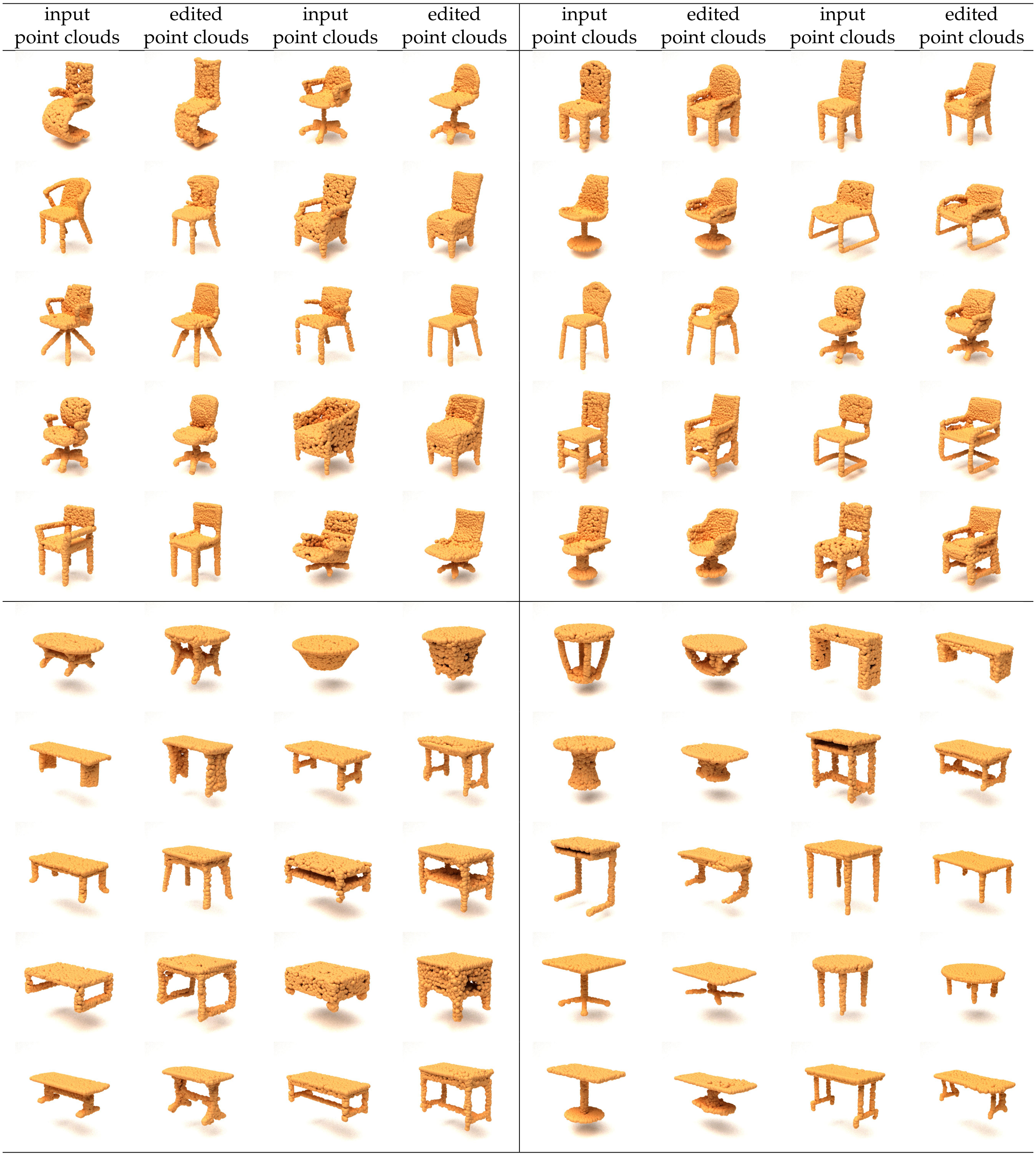

Fig. 11: Our results by MVS operations. The model was trained with the chair and table data. The upper left part shows the MVS results of arm-chair-to-armless-chair. The upper right part shows the MVS results of armless-chair-to-arm-chair. The lower left part shows the MVS results of short-table-to-tall-table. The lower right part shows the MVS results of tall-table-to-short-table.

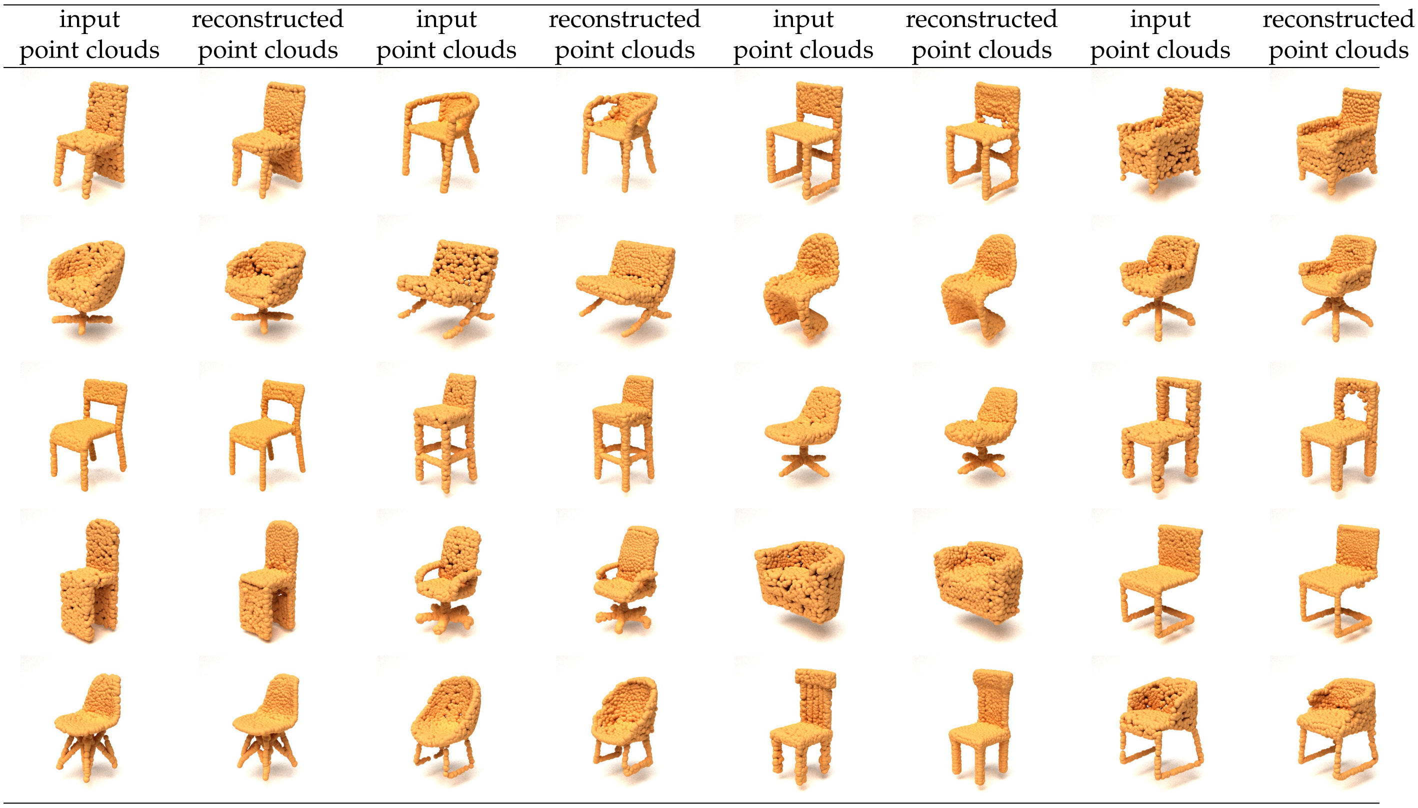
input point clouds
reconstructed point clouds
input point clouds
reconstructed point clouds
input point clouds
reconstructed point clouds
input point clouds
reconstructed point clouds

Fig. 12: The reconstructed results of chairs by the proposed autoencoder.

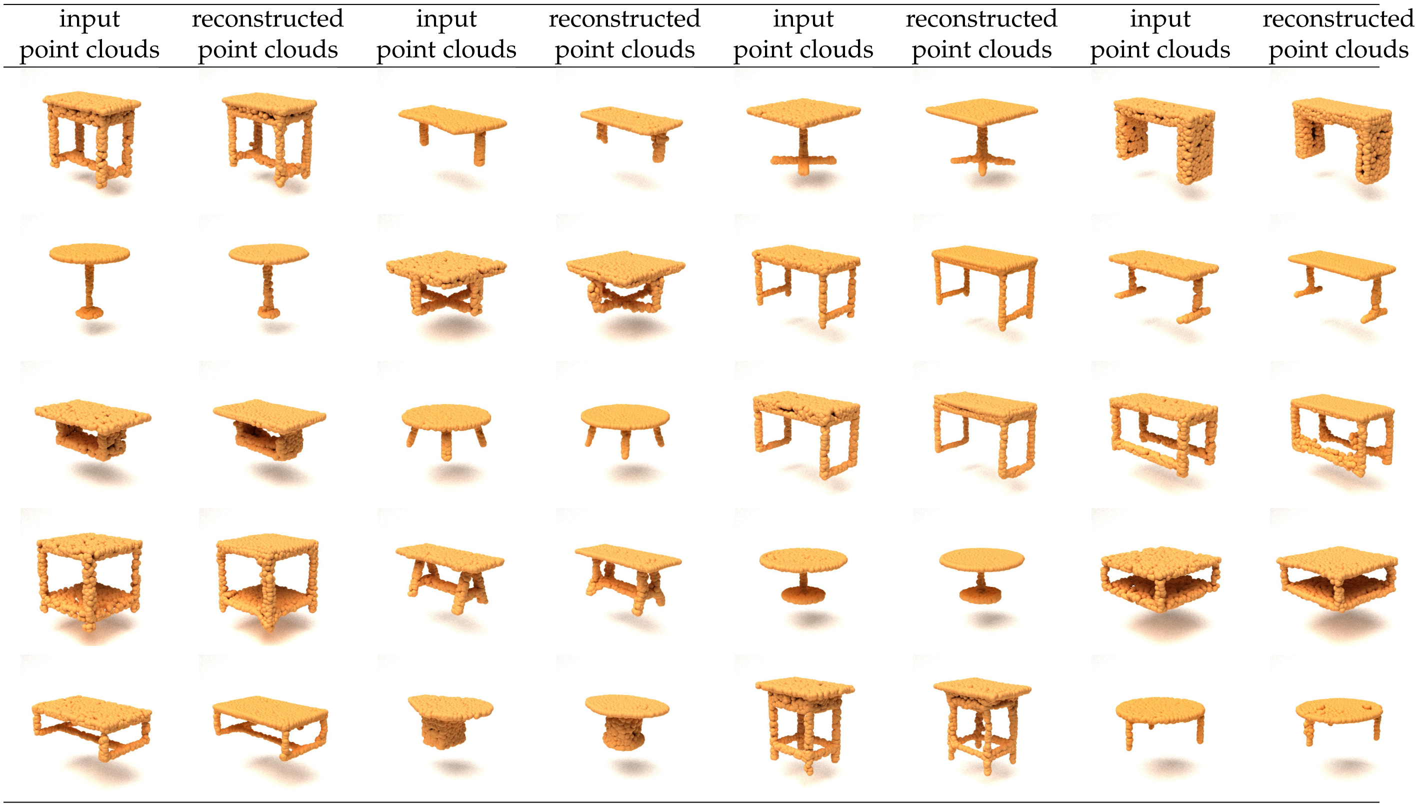
input point clouds
reconstructed point clouds
input point clouds
reconstructed point clouds
input point clouds
reconstructed point clouds
input point clouds
reconstructed point clouds

Fig. 13: The reconstructed results of tables by the proposed autoencoder.

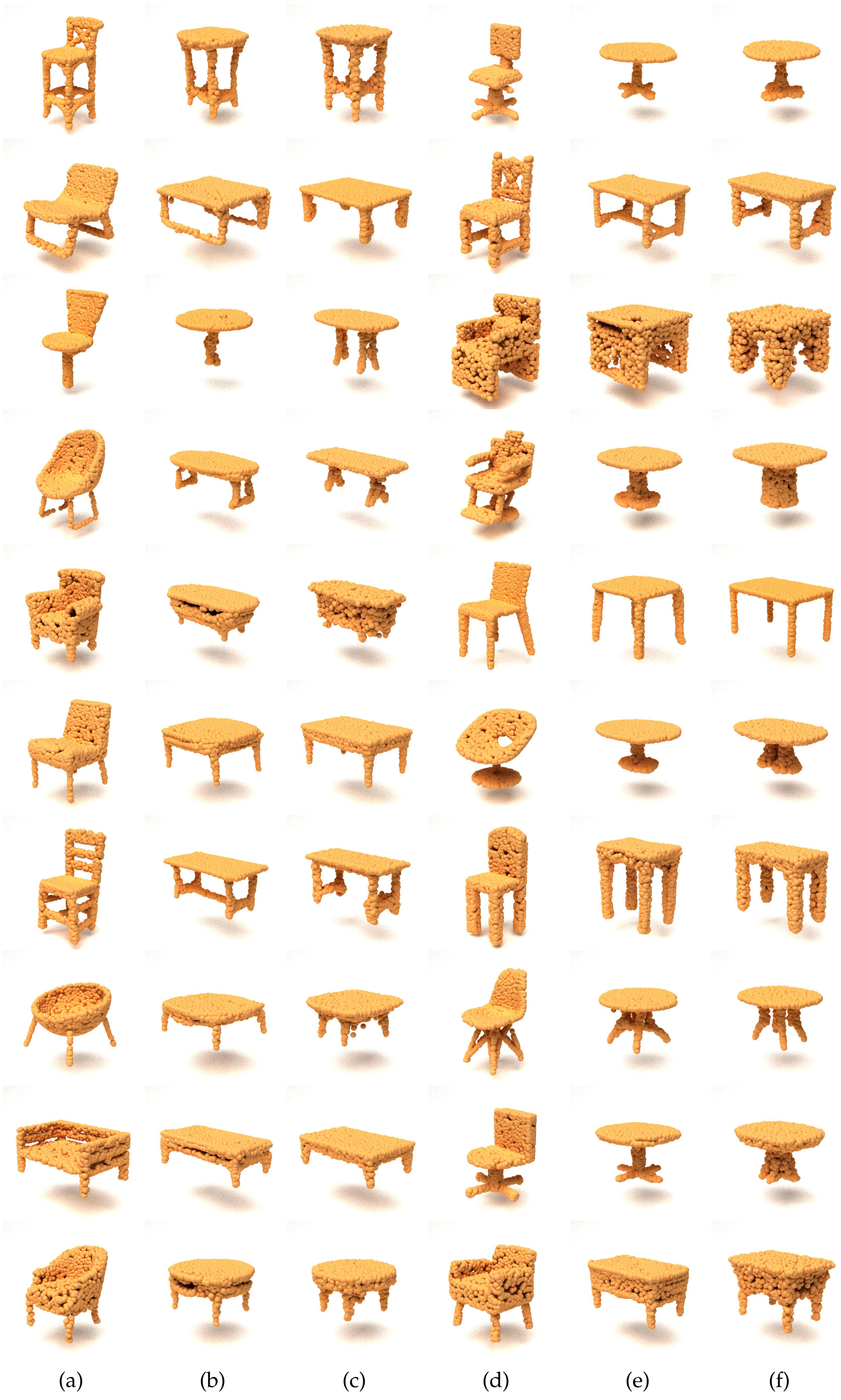

Fig. 14: The results of chair-to-table transfer. (a) Input chairs. (b) Our chair-to-table transferred results from (a). (c) The chair-to-table transferred results of LOGAN [9] from (a). (d) Input chairs. (e) Our chair-to-table transferred results from (d). (f) The chair-to-table transferred results of LOGAN [9] from (d).

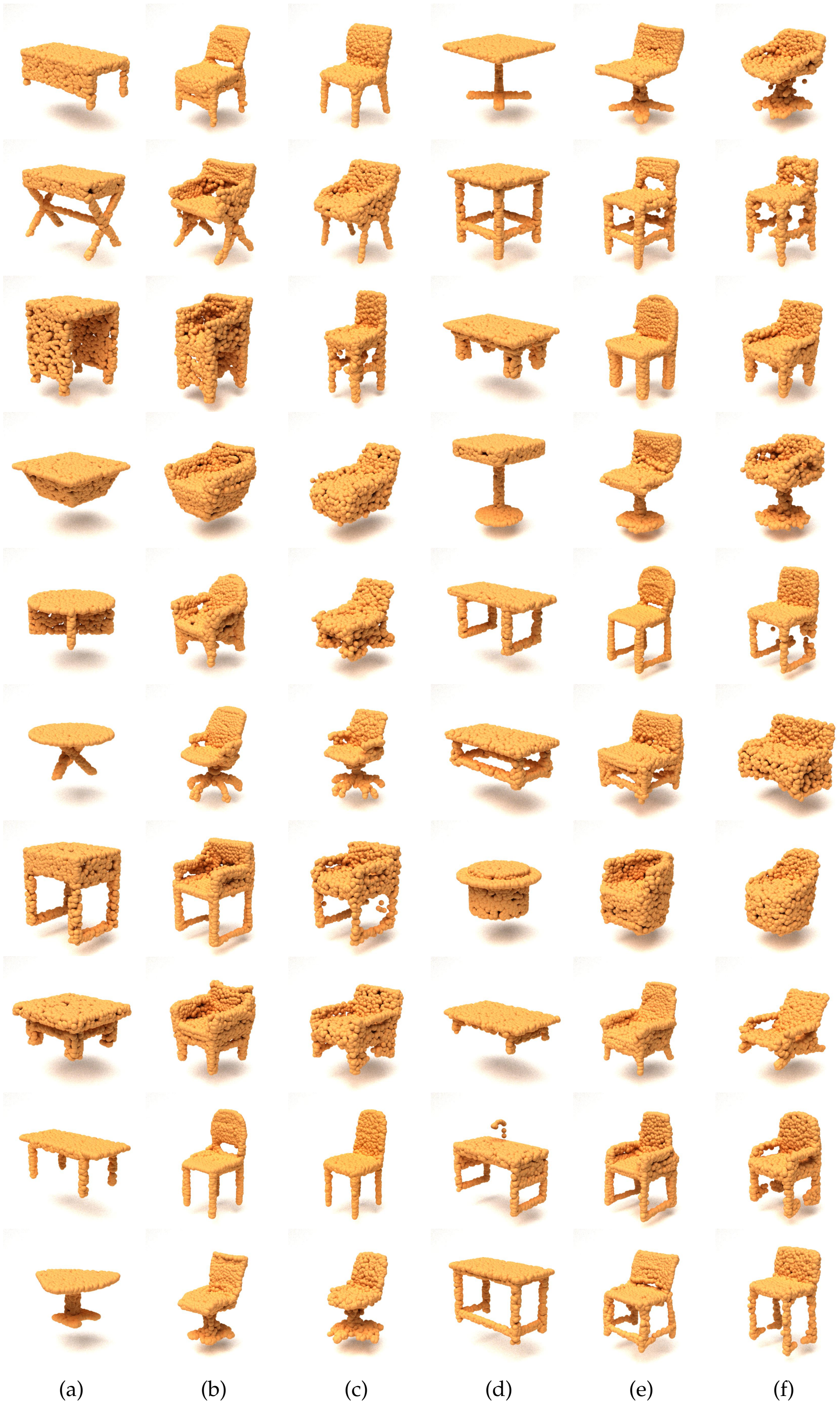

Fig. 15: The results of table-to-chair transfer. (a) Input tables. (b) Our table-to-chair transferred results from (a). (c) The table-to-chair transferred results of LOGAN [9] from (a). (d) Input tables. (e) Our table-to-chair transferred results from (d). (f) The table-to-chair transferred results of LOGAN [9] from (d).

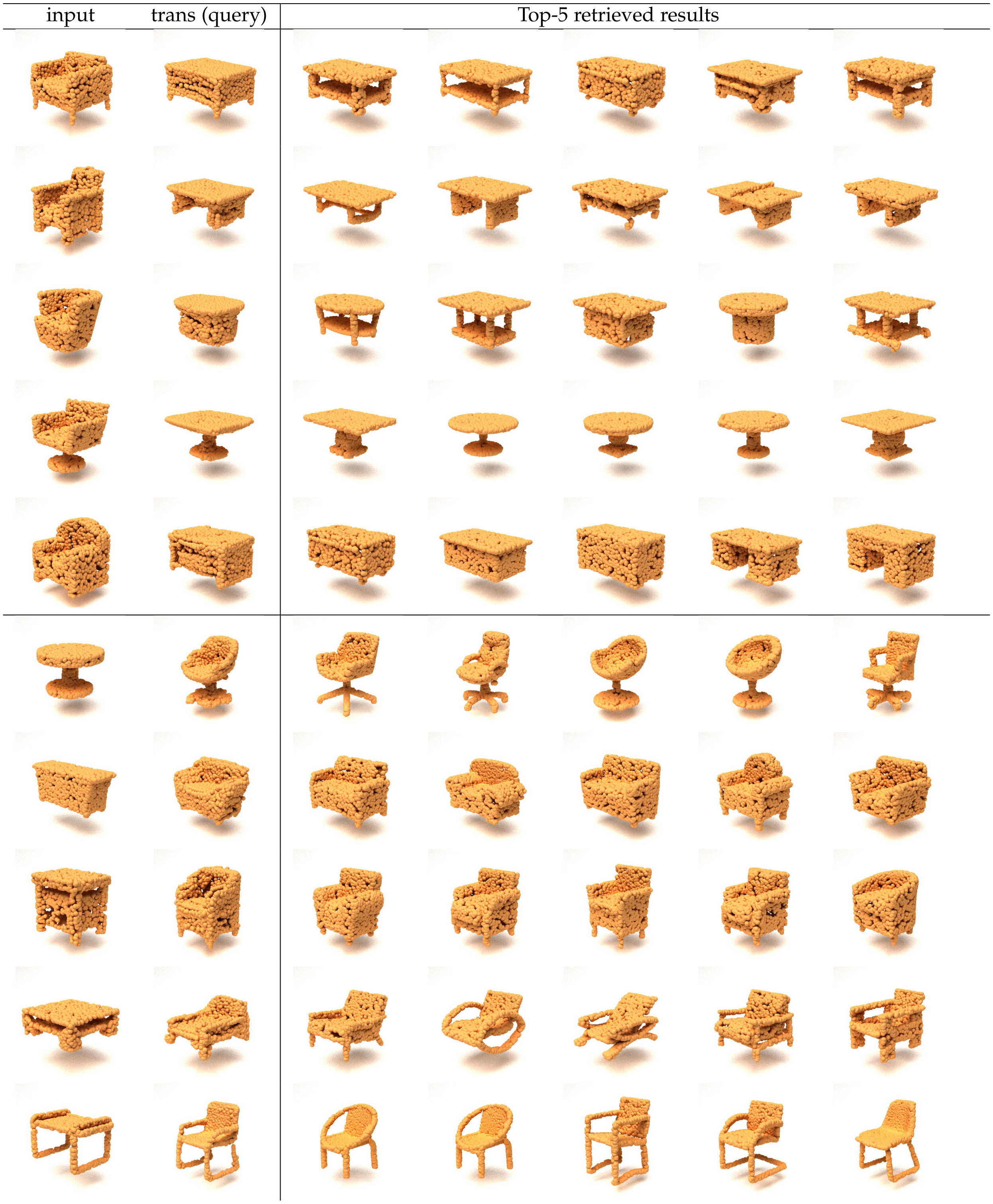

Fig. 16: Top-5 retrieval results from the training data for the chair-table translation results with our model. The inputs and translated results (queries) are shown in the leftmost two columns and the Top-5(EMD) retrieved objects of the corresponding queries are on the right-hand side.

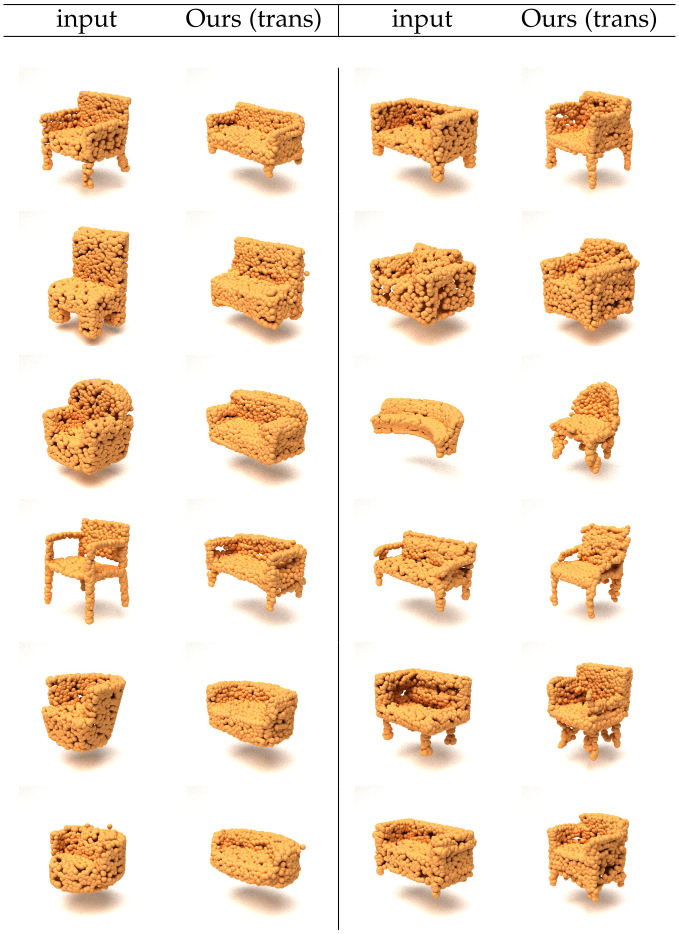

Fig. 17: The results of chair-sofa translation with our model. The left half is the results of chair → sofa, while the right half is the results of sofa → chair.

### 4.6.3 Chairs and Sofas

We trained our proposed method on the chair and sofa datasets from ShapeNet Core [1]. The chair dataset has 4768 training shapes and 2010 testing shapes, and the sofa dataset has 3073 training shapes and 100 testing shapes. Our translation results of chair-to-sofa and sofa-to-chair are shown in Fig. 17. As mentioned in the limitation, when some features are rare or difficult to implicitly find the correspondences in the opposite domain, the transferred point clouds can have more noise points than other cases.

### 4.6.4 Cabinets and Chairs

We trained our proposed method on the cabinet and chair datasets from ShapeNet Core [1]. The cabinet dataset has 1472 training shapes and 100 testing shapes, and the chair dataset has 4768 training shapes and 2010 testing shapes. Our translation results of cabinet-to-chair and chair-to-cabinet are shown in Fig. 18. As mentioned in the limitation of the main manuscript, since there are few apparent or semantic correspondences between these two domains, our system mainly transferred the height and width of objects and the round or square contour.

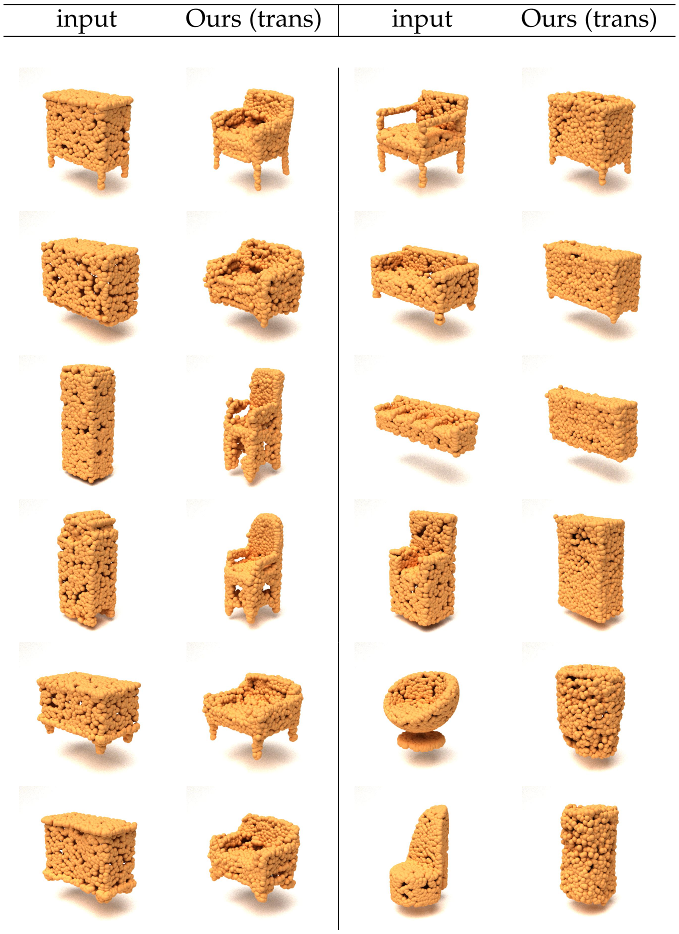

Fig. 18: The results of cabinet-chair translation with our models. The left half is the results of cabinet → chair, while the right half is the results of chair → cabinet.

### 4.6.5 Fit persons and Fat Persons

We trained our framework on the fit person (ID: 50009) and fat person (ID: 50002) data from the MPI DYNA dataset [6]. We uniformly sampled 2048 points as our input. As shown in Fig. 19, we can preserve the poses of persons after fit-person-to-fat-person transfer and fat-person-to-fit-person transfer.

## References

[1] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. In *arXiv preprint arXiv:1512.03012*, 2015.

[2] Qimin Chen, Johannes Merz, Aditya Sanghi, Hooman Shayani, Ali Mahdavi-Amiri, and Hao Zhang. Unist: Unpaired neural implicit shape translation network. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 18593–18601, 2022.

[3] Haoqiang Fan, Hao Su, and Leonidas J Guibas. A point set generation network for 3d object reconstruction from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 605–613, 2017.

[4] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1501–1510, 2017.

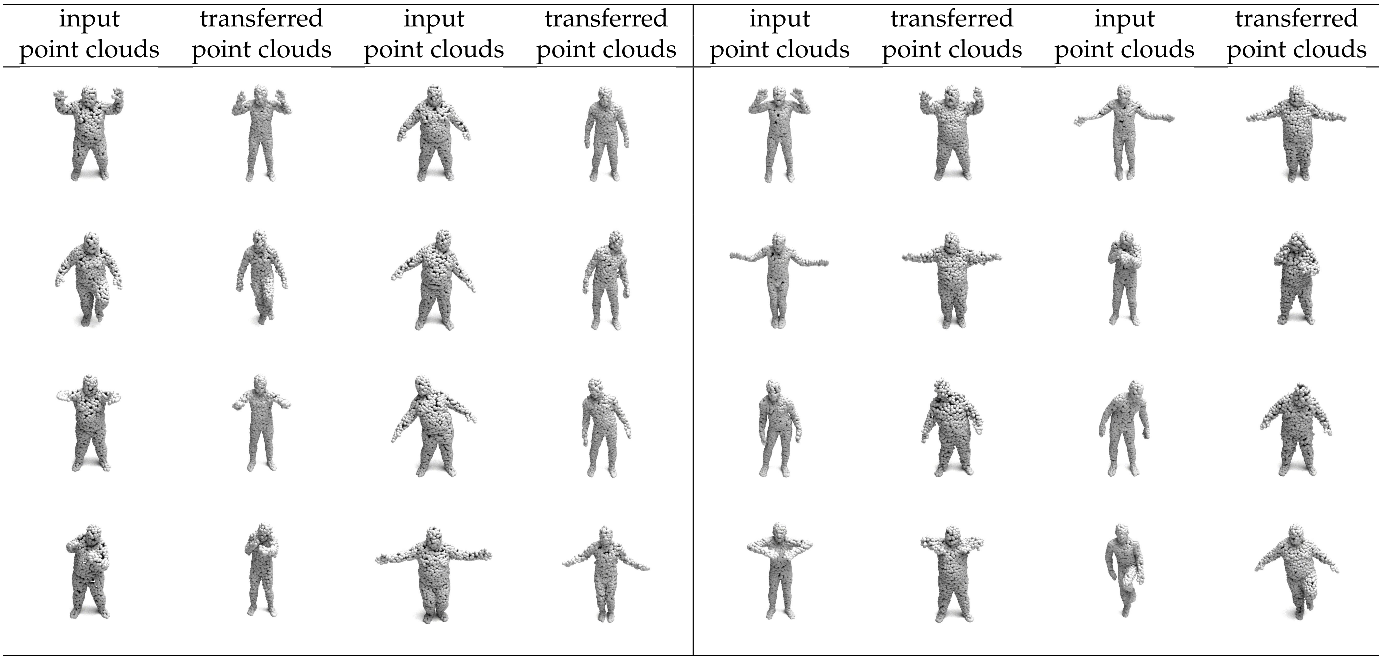

Fig. 19: Our transfer results between fit persons and fat persons. The left part shows the results of fat-person-to-fit-person transfer. The right part shows the results of fit-person-to-fat-person transfer.

[5] Isaak Lim, Moritz Ibing, and Leif Kobbelt. A convolutional decoder for point clouds using adaptive instance normalization. *Computer Graphics Forum*, 38(5):99–108, 2019.

[6] Gerard Pons-Moll, Javier Romero, Naureen Mahmood, and Michael J Black. Dyna: A model of dynamic human shape in motion. *ACM Transactions on Graphics (TOG)*, 34(4):1–14, 2015.

[7] Yossi Rubner, Carlo Tomasi, and Leonidas J. Guibas. The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision*, 40(2):99–121, 2000.

[8] Dong Wook Shu, Sung Woo Park, and Junseok Kwon. 3d point cloud generative adversarial network based on tree structured graph convolutions. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3859–3868, 2019.

[9] Kangxue Yin, Zhiqin Chen, Hui Huang, Daniel Cohen-Or, and Hao Zhang. Logan: unpaired shape transform in latent overcomplete space. *ACM Transactions on Graphics (TOG)*, 38(6):198, 2019.