

# Supplementary Document: Extended Differentiable Marching Cubes by Manifold-Preserving Shape Inflation

Kiichi Itoh

itok@den.t.u-tokyo.ac.jp

Tatsuya Yatagawa

tatsy@den.t.u-tokyo.ac.jp

Yutaka Ohtake

ohtake@den.t.u-tokyo.ac.jp

Hiromasa Suzuki

suzuki@den.t.u-tokyo.ac.jp

School of Engineering,

The University of Tokyo,

Tokyo, Japan

## A Networks for Shape-aware Deformation

The network of our E-DMC consists of two components: a differentiable MC module and a shape deformation module. While we borrowed the differentiable MC module from a previous study [23] without any modifications, the shape deformation module based on the NF receives a feature vector representing a shape of a given point cloud. We used Real NVP [8] as an NF in our method whose affine coupling layer is extended to receive the shape feature. The architectures of the shape encoder and conditional affine coupling are illustrated in Fig. A1 and Fig. A2, respectively. In our experiment, the shape encoder processes an input point cloud with 3,000 points to yield a 16-dimensional feature vector.

As shown by Fig. A1, the shape encoder consists of three consecutive set abstraction layers of PointNet++ [41], followed by a single fully connected layer. In contrast, Fig. A2 shows that the conditional affine coupling layer receives a concatenation of two coordinates of 3D points, *e.g.*  $y$  and  $z$ , and the shape feature. Then, its outputs, *i.e.* affine parameters  $\sigma$  and  $\tau$ , transform the remaining coordinate, *e.g.*  $x$  as  $x' = \sigma x + \tau$ . The conditional Real NVP used in our method comprises six conditional affine coupling layers, which transforms each of  $x$ ,  $y$ , and  $z$  twice.

## B Effect of Sample Points and Voxel Size

In the main paper, we tested the proposed method using point clouds with 3,000 points and a grid with  $32^3$  voxels. Here, we evaluate the proposed methods (E-DMC and E-DMC+L) and baseline methods (DMC and DMC+L) using a smaller number of points (300 points) or a smaller voxel grid ( $16^3$  voxels). Tables B1 and B2 show the results of comparison using 300 points and  $16^3$  voxels. Note that the unit of values in these tables is *voxel*, and the values

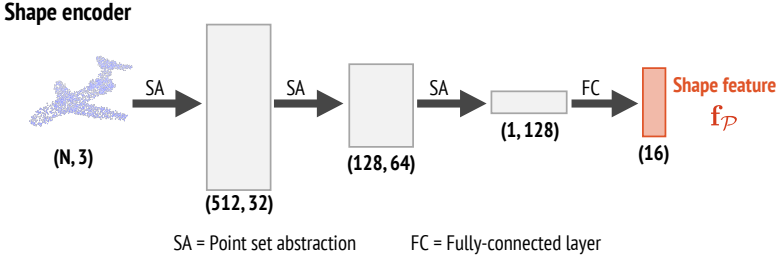


Figure A1: The network architecture of the shape encoder, which consists of three consecutive set abstraction layers of PointNet++ [31], followed by a fully connected layer.

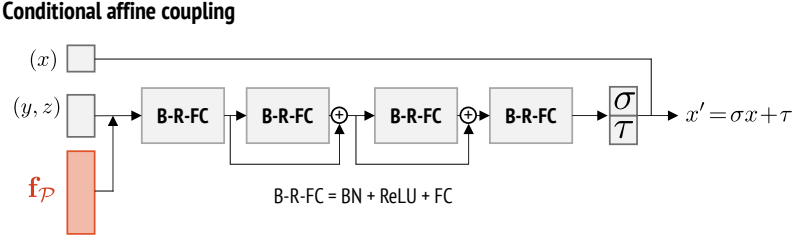


Figure A2: The network architecture of the conditional affine coupling layer. Two coordinates, *e.g.*  $y$  and  $z$ , of 3D points are concatenated with the shape feature  $f_P$ , and are handed to a network with residual blocks to obtain affine parameters  $\sigma$  and  $\tau$ . The remaining coordinate  $x$  is transformed as  $x' = \sigma x + \tau$ .

in Table B2 are twice as small as the values in other tables such as Table 1 for  $32^3$  voxels in the main paper.

As shown by Table B1, the relative performances of the proposed and baseline methods for 300 input points are approximately the same as those 3,000 input points, whereas the absolute performance of each method is two or three times worse. In contrast, when the smaller voxel grid, *i.e.*  $16^3$  voxels, is used, DMC+L without our extension performs the best among the proposed and baseline methods. In practice, more deformation by the NF does not always provide better performance in terms of accuracy and completeness because deformation by the NF can make vertices on the object surface bumpy. When the grid size is small, this negative effect is more than the positive effect from the shape inflation because of little room for shape deformation. Therefore, our extensions do not perform as well as the baseline methods. Even so, the memory consumption with  $32^3$  voxels by our method is only 3 GB, and there is no need to use a voxel grid smaller than that.

300 points	Acc.	Comp.	Overall
DMC	1.266	0.533	0.900
DMC+L	<b>0.287</b>	0.698	0.493
E-DMC	1.204	0.550	0.877
E-DMC+L	0.433	<b>0.471</b>	<b>0.452</b>

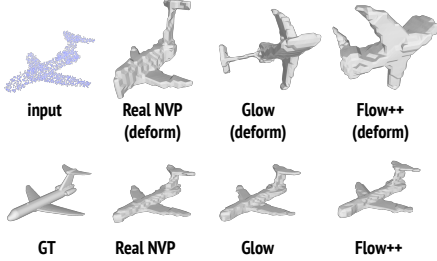
Table B1: Quantitative comparison of the proposed and baseline methods trained and tested using only 300 points.

$16^3$ voxels	Acc.	Comp.	Overall
DMC	0.222	0.199	0.211
DMC+L	<b>0.130</b>	<b>0.122</b>	<b>0.126</b>
E-DMC	0.376	0.250	0.313
E-DMC+L	0.169	0.155	0.162

Table B2: Quantitative comparison of the proposed and baseline methods trained and tested using a  $16^3$  voxel grid.

## C Comparison of Normalizing Flows

Although we used Real NVP [8] to deform the input point cloud in the experiments in the main study, many other choices of the NFs exist for this purpose. Particularly, since Real NVP is a type of NF that uses a bijective coupling layer, we can naturally adopt other NFs using bijective coupling to our method. In this document, we examined two other NFs, *i.e.* Glow [19] and Flow++ [10], to investigate how the type of the NFs affects the results of our method.



	Acc.	Comp.	Overall
Real NVP [8]	0.167	0.190	0.178
Glow [19]	0.163	0.192	0.178
Flow++ [10]	0.169	0.203	0.186

Table C1: Quantitative comparison for E-DMC+L using different NFs. The evaluation scores do not significantly change for the different NFs, whereas those for Flow++ are slightly worse than the others.

Figure C1: Visual comparison of output and deformed meshes for different NFs (*airplane 0674* is used).

Table C1 shows the average scores for accuracy, completeness, and overall scores obtained by different NFs. As shown in this table, the scores are approximately equal for different NFs, whereas those for Flow++ [10] are slightly worse than those for others. The output and deformed meshes are shown in Figs. C1 and C2. As seen in these figures, Glow and Flow++ scale and rotate the input shape while Real NVP only scales the input approximately along each dimension. This rotation arises because of the invertible  $1 \times 1$  convolution [19] considered as a generalised permutation of vector axes.

Although the deformation defined by Flow++ can be more effective due to its high representability, we found that it did not always improve the quality of surface reconstruction. In the piano example shown in Fig. C2, the flexibility of Flow++’s deformation appropriately reproduced a gap between the chair and the body of the piano, while that of Real NVP did not. In contrast, for simpler shapes, *e.g.* in Fig. C1, the boundary shapes given by Flow++ can be jagged, which results in degradation of the evaluation scores, as shown in Table C1.

In summary, Real NVP is a good first choice due to its ease of implementation and computational efficiency, whereas other NFs such as Glow and Flow++ may be more potent for more complicated geometries.

## D Ablation Study for E-DMC without linear stretching

In addition to the ablation study for E-DMC+L in the main paper, we conducted another ablation study for E-DMC without linear stretching. The results are shown in Fig. D1 and Table D1. These results show the different behaviours of the proposed components when used with E-DMC rather than E-DMC+L.

We find that two losses, *i.e.* the manifold-preserving loss  $\mathcal{L}_{\text{mp}}$  and the point repulsion loss  $\mathcal{L}_{\text{rep}}$ , work almost equally as they do for E-DMC+L. Thus,  $\mathcal{L}_{\text{mp}}$  works appropriately

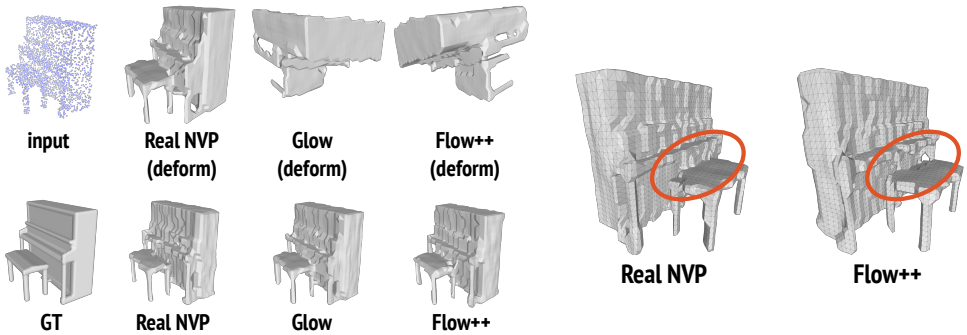


Figure C2: Visual comparison of the output and deformed meshes for different NFs (*piano 0318* is used). As highlighted by red circles in the right images, a complicated deformation represented by Flow++ can have a positive effect to correctly reproduce a space between the chair and body of the piano in this example.

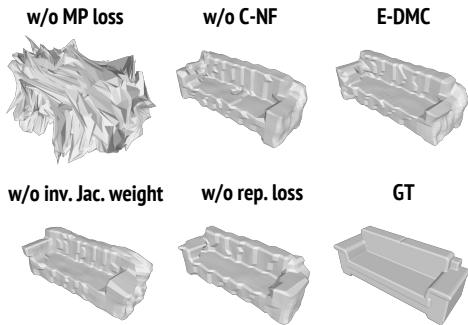


Figure D1: Visual comparison of the output meshes for our E-DMC and those of each proposed component which are ablated (*sofa 0744* are used).

	Acc.	Comp.	Overall
E-DMC	0.454	0.307	0.381
w/o $\mathcal{L}_{mp}$	2.061	0.489	1.275
w/o C-NF	0.384	0.284	0.334
w/o iJW	<b>0.302</b>	0.322	<b>0.312</b>
w/o $\mathcal{L}_{rep}$	0.646	<b>0.252</b>	0.449

Table D1: Quantitative comparison for the ablation study with E-DMC.

to preserve the manifold structure of an input shape, and  $\mathcal{L}_{rep}$  works to improve the surface reconstruction accuracy by expanding a given point cloud. Although the completeness improves when  $\mathcal{L}_{rep}$  is ablated, this is considered because the mesh reproduction by DMC fails when the input point cloud becomes partly sparse after the expansion induced by  $\mathcal{L}_{rep}$ .

In contrast, E-DMC with the C-NF and the inverse Jacobian weight degraded the evaluation scores, which is counterintuitive in terms of the idea that these components would enable Real NVP to more appropriately perform deformation considering the input shape. However, as we observed in the results of Flow++, the complicated deformation does not always improve the evaluation scores. Specifically, complicated deformation can make the reconstructed surfaces more jagged. This problem can be more serious when linear stretching is not used because a deformation to obtain better surface reconstruction becomes more ambiguous, making it more difficult to learn appropriate deformations. On the contrary, when the input shape is linearly stretched to fill the domain of the voxel grid, there remains little room for deformation. In this case, the C-NF and inverse Jacobian weight can fine-tune the deformation, resulting in better accuracy, completeness, and overall scores, as we showed



in the main paper.

## E Additional results

To demonstrate that our learning-based method appropriately deforms the input point cloud to improve the output meshes, we show the results for every 40 categories of the ModelNet40 dataset. The following figures show output meshes of DMC+L, those of E-DMC+L, deformed meshes of E-DMC+L, and the corresponding ground truth meshes. As shown in these results, our method reproduces holes and hollows better than DMC+L, which we can see in the results of *bookshelf 0574*, *desk 0278*, and *vase 0479*.

In contrast, the DMC layer sometimes generates a mesh with small holes and cracks when the input point cloud is overly deformed by Real NVP. Although we trained the DMC layer using the default parameters shown in the original DMC paper [23] in our experiments, these problems can be alleviated by adjusting the parameters more precisely.

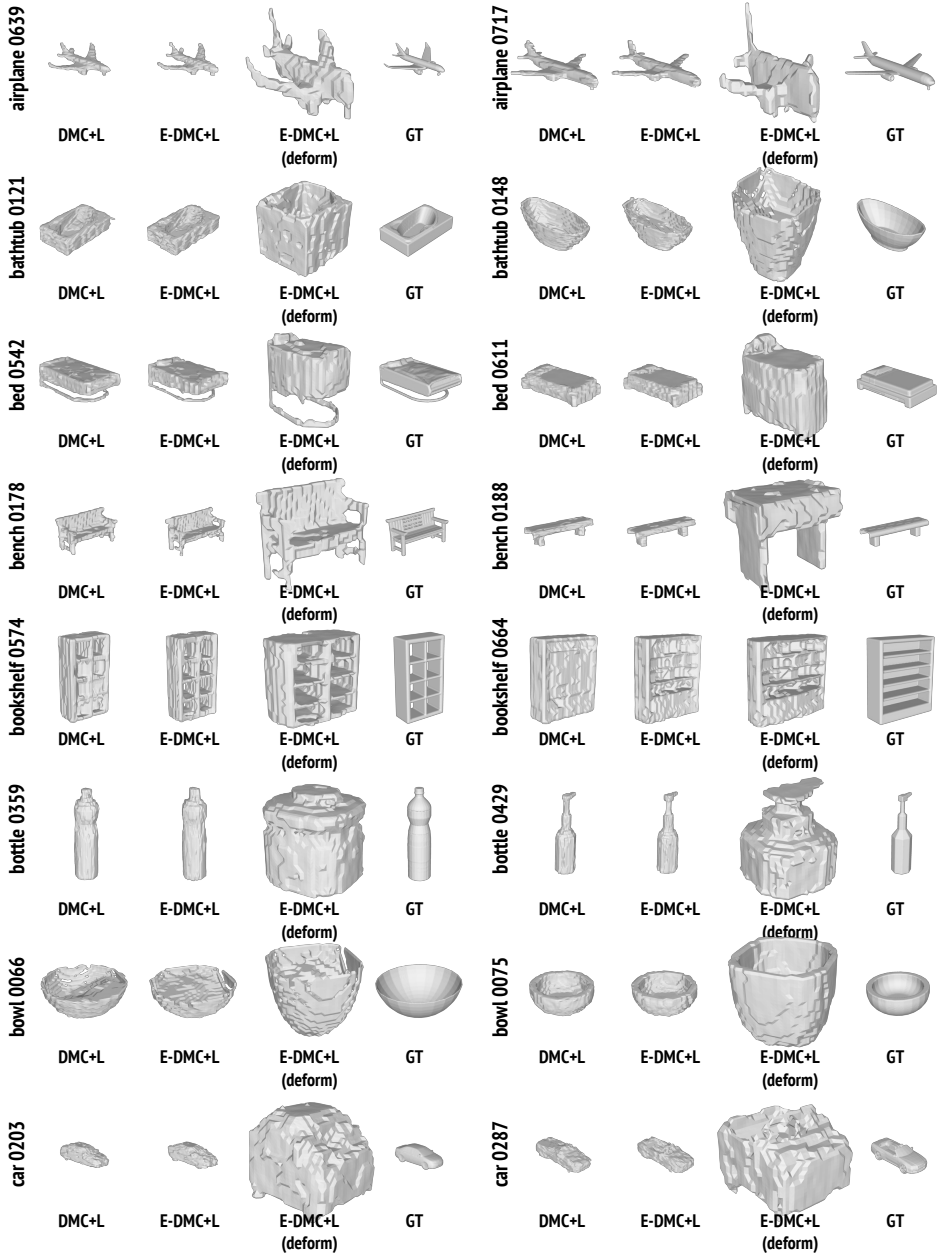


Figure E1: Results for each shape category (airplane, bathub, bed, bench, bookshelf, bottle, bowl, and car).

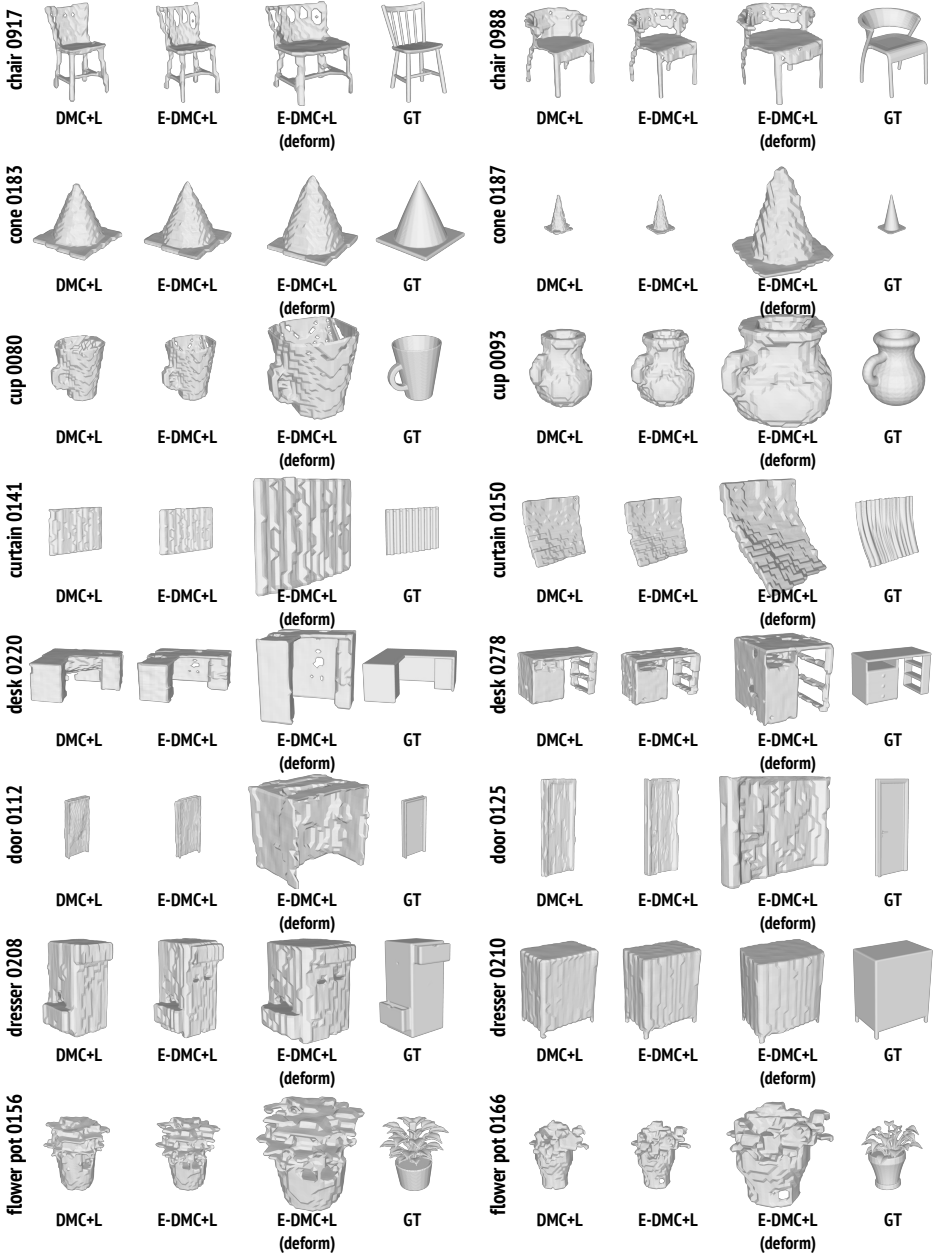


Figure E2: Results for each shape category (chair, cone, cup, curtain, desk, door, dresser, and flower).

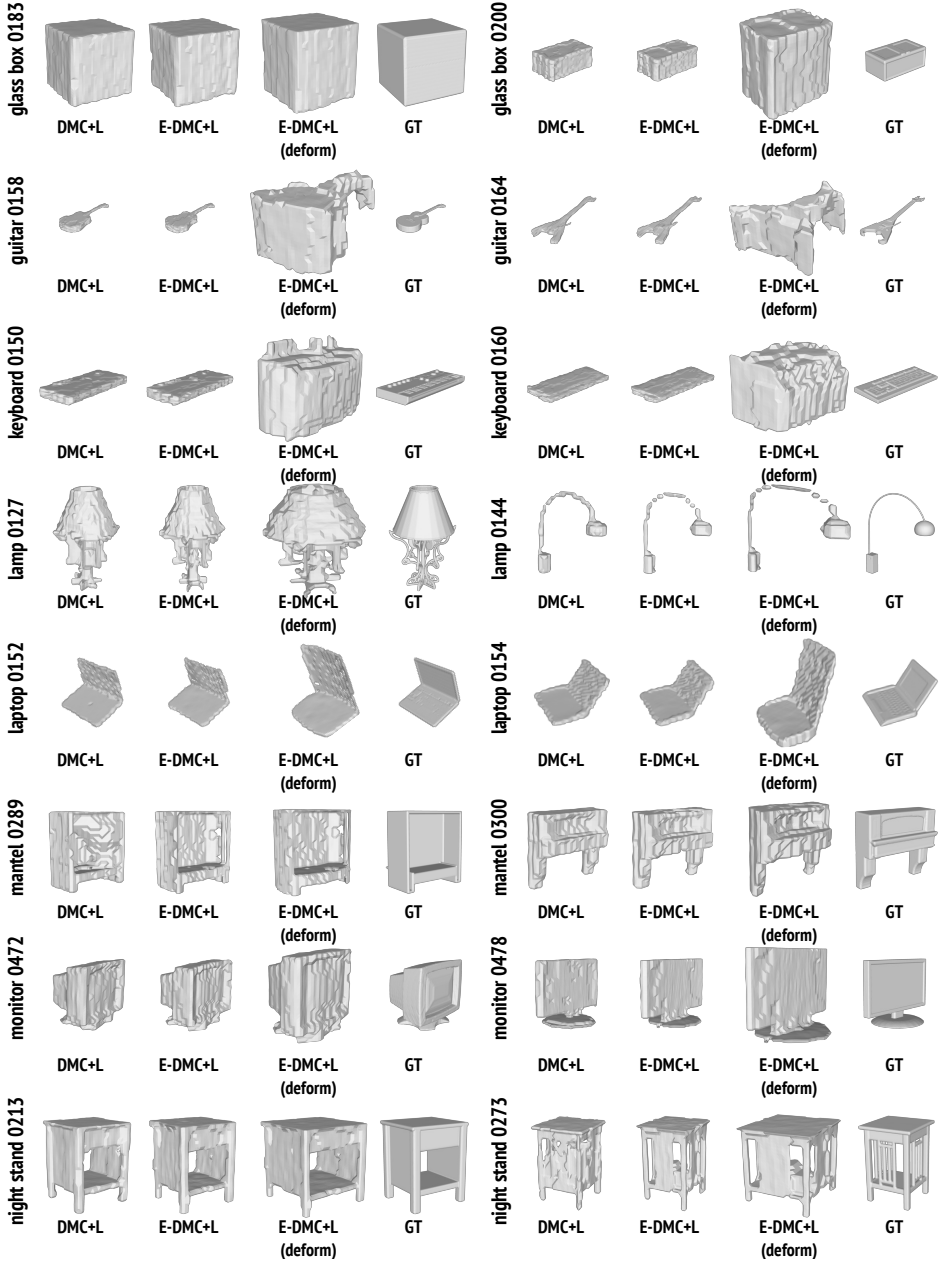


Figure E3: Results for each shape category (glass box, guitar, keyboard, lamp, laptop, mantel, monitor, and night stand).

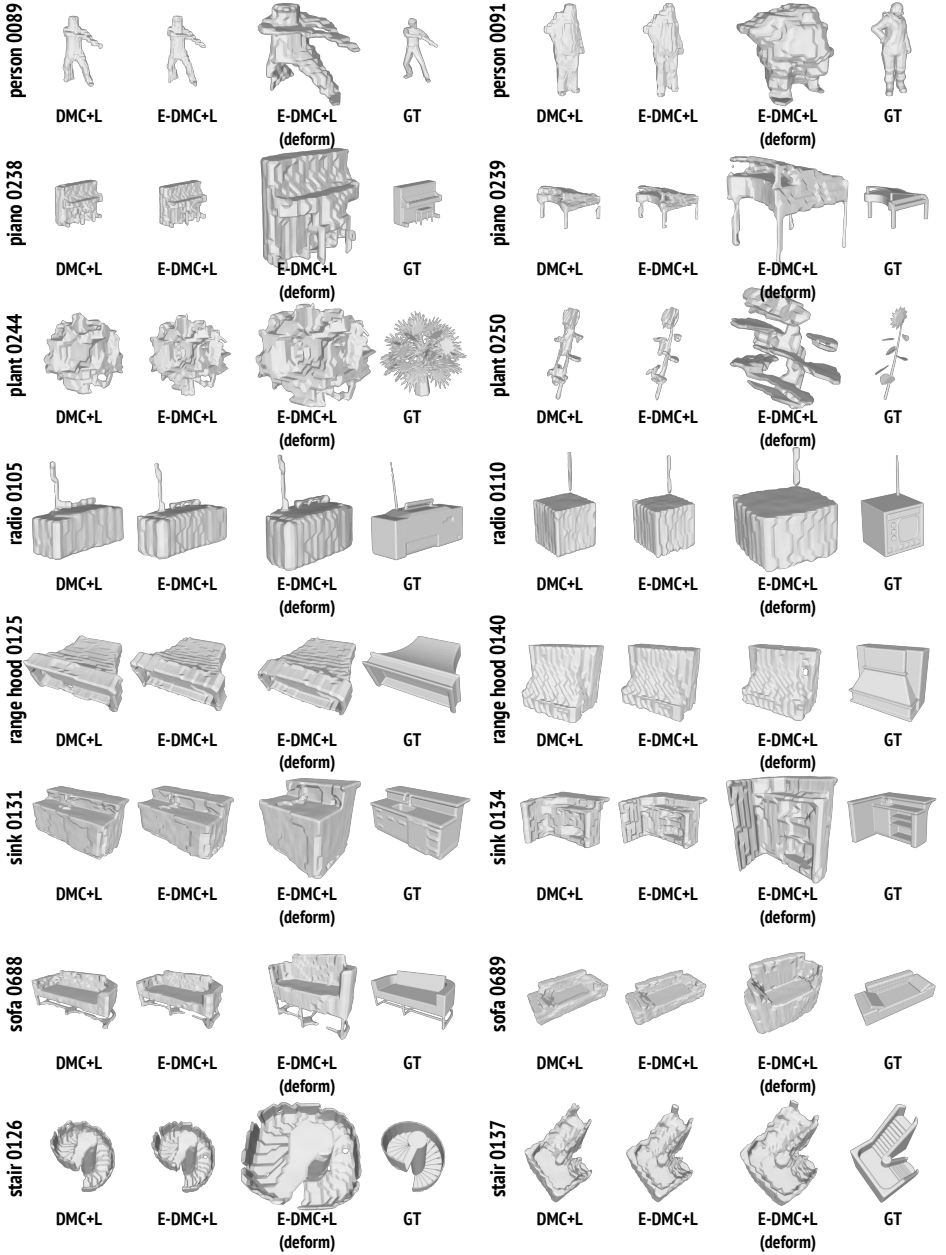


Figure E4: Results for each shape category (person, piano, plant, radio, range hood, sink, sofa, and stairs).

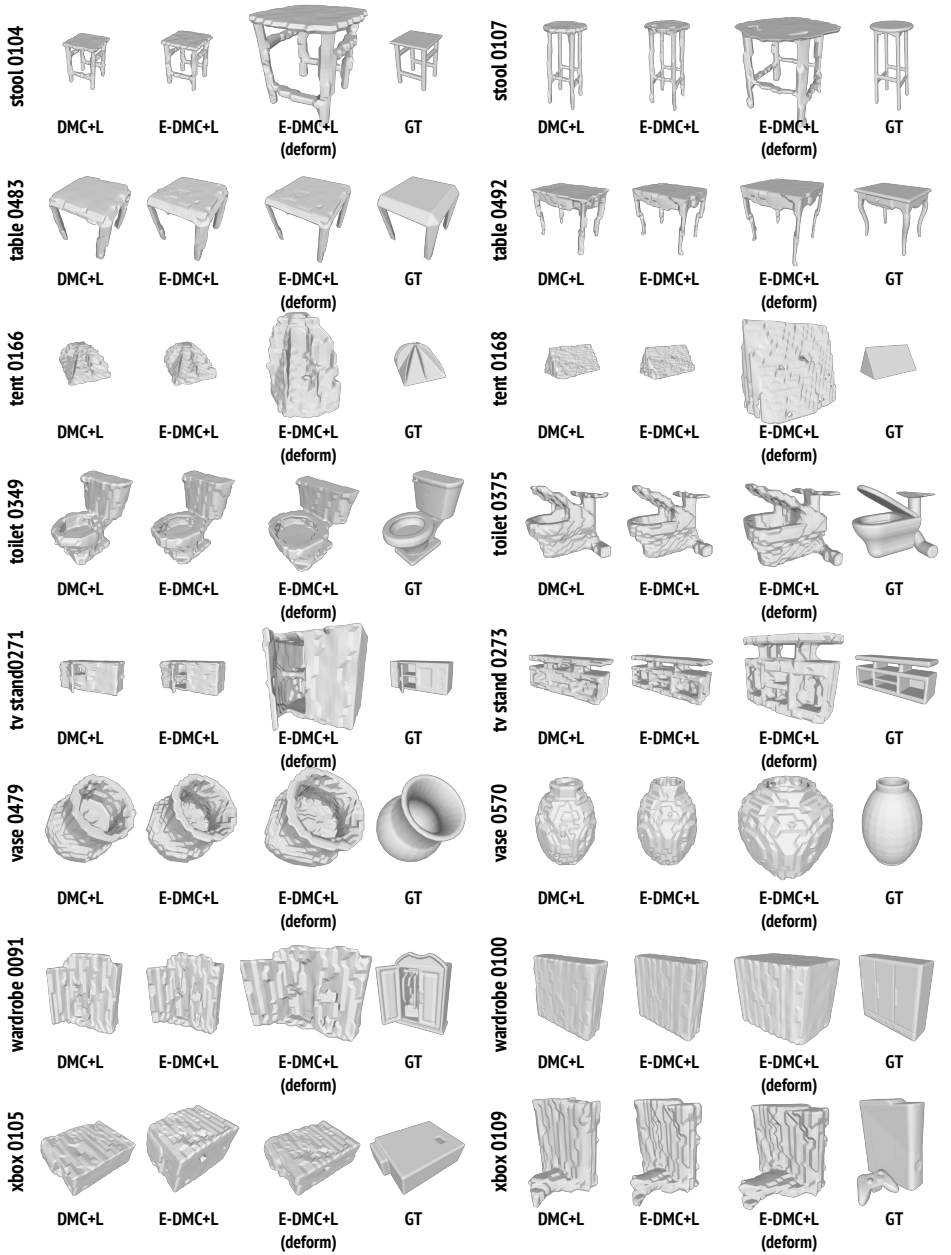


Figure E5: Results for each shape category (stool, table, tent, toilet, tv stand, vase, wardrobe, and Xbox).