Deformation-Aware 3D Model Embedding and Retrieval — Supplementary Material

Mikaela Angelina Uy[1], Jingwei Huang[1], Minhyuk Sung[2], Tolga Birdal[1], and Leonidas Guibas[1]

[1] Stanford University [2] Adobe Research

S.1 Details of Deformation Function $\mathcal{D}$

We opt to use a simple deformation function for $\mathcal{D}$ in our experiments, which is designed to preserve local rigidity similarly with ARAP [4] but much simpler yet effective in practice. Specifically, given a source mesh $\mathbf{s} = (\mathcal{V} \in \{\mathbb{R}^3\}_{1\cdots N}, \mathcal{E} \in \mathcal{V}^2)$, where $\mathcal{V}$ and $\mathcal{E}$ denote the collections of vertices and edges, and a target $\mathbf{t}$ represented as an unsigned distance function $f_{\mathbf{t}}$, we define our deformation function $\mathcal{D}(\mathbf{s};\mathbf{t})$ as follows:

$$\mathcal{D}(\mathbf{s};\mathbf{t}) = \left(\underset{\widehat{\mathcal{V}}}{\operatorname{argmin}} \left\{ \sum_{\hat{v_i} \in \widehat{\mathcal{V}}} f_{\mathbf{t}}(\hat{v_i}) + \lambda \sum_{(i,j)\in\mathcal{E}} ||(\hat{v_i} - \hat{v_j}) - (v_i - v_j)||^2 \right\}, \mathcal{E} \right) \quad \text{(S1)}$$

where v_i and $\hat{v_i}$ are the given and optimized positions of i-th vertex. The first term represents the fitting loss that pushes the deformed source shape $\mathcal{D}(\mathbf{s};\mathbf{t})$ to be close to $\mathbf{t}$, and the second term is the rigidity regularization loss that penalizes for the length changes of each edge in $\mathcal{E}$. In our implementation, we solve the minimization in Eq. S1 using Ceres solver [1] by initializing the vertex coordinates with the source mesh and defining the unsigned distance function with 100^3 voxel grids. We set $\lambda = 1$ in all our experiments, as we found that it well-preserves the CAD model features including sharp edges and corners for most of the 3D models we used.

While we convert the CAD models to simplified watertight meshes in Sec. 4 in the paper to efficiently deform and preserve the connectivity across the connected components in the CAD model, the ARAP deformation can also be directly applied to the surface of the CAD model with a simple preprocessing. We found that remeshing each connected component and linking the components each other with additional edges based on the proximity can also give a very similar result in the deformation with that of using the converted watertight meshes. This way can maintain the original CAD model structure with its accompanied meta information. Fig. S1 shows the difference between the converted watertight mesh deformation to the direct CAD model deformation, which are almost indistinguishable. All figures of the qualitative evaluation results in our main paper are rendered with the results of the direct CAD model deformation.

S.2 Image-to-CAD

To show the flexibility of our approach, we now extend it to the application of image-to-CAD generation. Given an image of a 3D model, we first use

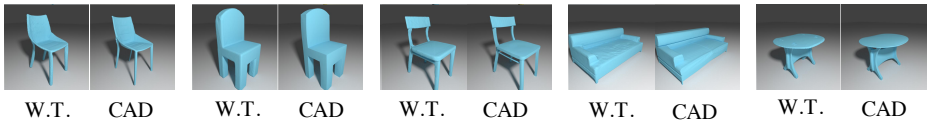

Fig. S1. Deformation results with the converted watertight meshes and the raw CAD models.

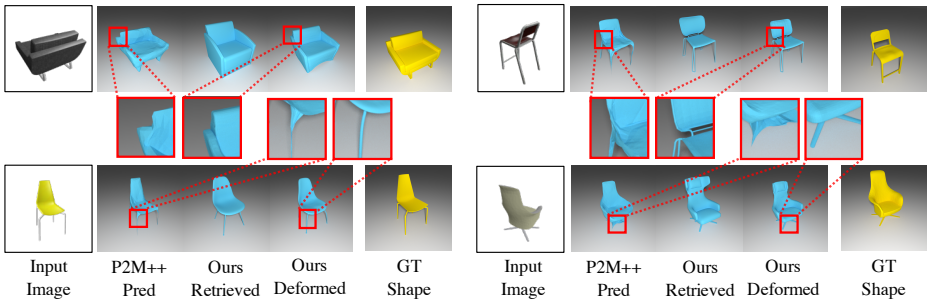

Fig. S2. Qualitative results to show the feasibility of our approach for the Image-to-CAD application. We show one of three input viewpoints used by Pixel2Mesh++ [5] to produce their coarse mesh. We use this to retrieve a CAD model, which is then deformed to fit the coarse mesh. Rigidity constraints ensure the quality of our output as shown. See Fig. S6 for more results.

Pixel2Mesh++ [5], a state-of-the-art image-to-mesh network, to generate an initial coarse mesh. We then use its output to retrieve a CAD model using the proposed *Ours-Reg* trained on ShapeNet [2] and deform it to fit the coarse mesh. Fig. S2 shows that our approach is able to output models without artifacts produced by direct generation networks *i.e.* in this case Pixel2Mesh++. It is clearly shown that our output has sharper edges and preserves thin structures such as the legs of the chairs in Fig. S2. Note that our retrieval solely relies upon the mesh prediction of Pixel2Mesh++ [5] and we warp the retrieved model towards the output of this mesh prediction network without the knowledge of the input images.

S.3 Additional Baselines

We further compare our method with additional baselines. *CD-Margin* is the case of using the margin loss in Sec. 3.2 in the paper for training, but employing d^m (Chamfer distance) to define the relationships among the shapes (instead of the fitting gap e_D^m (Eq. 1 in the paper) and using Euclidean distance in the embedding space (fixing $\mathcal{G}(\cdot)$ in Eq. 2 in the paper to identity). Given this, we introduce three more baselines:

– *CD-Reg*: The network is trained with the regression loss in Sec. 3.3 in the paper, but using d^m and identity $\mathcal{G}(\cdot)$ to define shape relationships and embedding distance.

Table S1. Additional baseline results that show the fitting gap for the top-1 retrieval of the different object classes in three additional set-ups.The fitting gap $e^m_{\mathcal{D}}(\mathbf{s},\mathbf{t})$ multiplied by $1e^{-2}$ are reported.

	Method	Table		Chair		Sofa		Car	
		Top-1	Top-3	Top-1	Top-3	Top-1	Top-3	Top-1	Top-3
Mean $d^m(\mathbf{s},\mathbf{t})$	CD-Margin	**4.875**	**3.449**	**4.750**	**3.518**	**3.087**	4.151	**2.525**	**1.905**
	CD-Reg	9.457	5.828	9.127	5.980	7.095	4.547	2.658	1.947
	Symm-Margin	5.939	3.887	5.533	3.857	4.709	**3.301**	2.958	2.137
	Symm-Reg	6.517	4.025	9.824	6.579	7.667	4.990	2.989	2.218
	Ours-Margin	6.227	4.026	5.664	3.889	4.825	3.400	2.962	2.142
	Ours-Reg	5.955	3.979	5.751	3.981	5.091	3.628	3.119	2.263
Mean $e^m_{\mathcal{D}}(\mathbf{s},\mathbf{t})$	CD-Margin	2.362	1.373	2.134	1.242	1.587	0.909	1.249	0.773
	CD-Reg	5.086	2.736	4.166	2.310	3.186	1.498	1.327	0.778
	Symm-Margin	2.183	1.267	1.946	1.169	1.497	0.855	1.261	0.743
	Symm-Reg	2.500	1.334	4.349	2.591	3.313	1.639	1.157	0.695
	Ours-Margin	2.127	1.251	1.915	1.144	1.420	0.835	1.226	0.747
	Ours-Reg	**1.969**	**1.129**	**1.752**	**1.054**	**1.338**	**0.788**	**1.112**	**0.681**

Table S2. Additional baseline results for ranking evaluations with 150 models per query. The models are randomly selected and sorted by $e^m_{\mathcal{D}}(\mathbf{s},\mathbf{t})$ (the query is not included). All results are for the top-1 retrieval results of each method. The numbers multiplied by $1e^{-2}$ are reported.

Method	Table			Chair			Sofa			Car		
	Mean d^m	Mean $e^m_{\mathcal{D}}$	Mean Rank	Mean d^m	Mean $e^m_{\mathcal{D}}$	Mean Rank	Mean d^m	Mean $e^m_{\mathcal{D}}$	Mean Rank	Mean d^m	Mean $e^m_{\mathcal{D}}$	Mean Rank
CD-Margin	**6.77**	3.19	12.55	**6.02**	2.72	13.24	**5.07**	1.93	15.76	**3.02**	1.48	18.94
CD-Reg	10.37	5.42	46.67	9.51	4.31	41.35	7.62	3.32	43.06	3.16	1.45	18.66
Symm-Margin	8.54	2.96	9.70	7.09	2.46	9.31	5.69	1.77	11.04	3.54	1.37	14.83
Symm-Reg	8.72	3.15	12.56	10.37	4.61	46.54	7.62	3.32	43.06	3.16	1.45	18.66
Ours-Margin	8.89	2.88	8.86	7.15	2.37	8.15	5.83	1.67	9.09	3.61	1.34	12.95
Ours-Reg	8.59	**2.71**	**7.05**	7.39	**2.24**	**6.32**	6.23	**1.62**	**7.91**	3.80	**1.24**	**7.80**

– *Symm-Margin*: The relationships among the shapes are defined with the fitting gap e^m_D (Eq. 1 in the paper), but still $\mathcal{G}(\cdot)$ in Eq. 2 in the paper is fixed to identity.
– *Symm-Reg*: The same with *Symm-Margin*, but the network is trained with the regression loss in Sec.3.3.

The quantitative results are reported in Tab. S1 and Tab. S2 (similarly to Tab. 1 and Tab. 2 in the paper). Refer to Sec. 5 for the details of the evaluation metrics. The performance is improved when using the fitting gap e^m_D as the relationships instead of Chamfer distance d^m, as shown in the results of *Symm-Margin* and *Symm-Reg* (compared with the results of *CD-Margin* and *CD-Reg*). However, still the performance of *Symm-Margin* and *Symm-Reg* is inferior to our case (*Ours-Margin* and *Ours-Reg*) using the egocentric distance field to embed the relationships. Also, note that the regression loss provides better performance only when the egocentric distance field is used in the embedding (*Ours-Margin* vs. *Ours-Reg*) but not for the other cases (*CD-Margin* vs. *CD Reg* and *Symm-Margin* vs. *Symm-Reg*).

Table S3. The percentage of recall@1 for different methods. A correct match is defined as the case when the top-1 retrieval is in the top-5 ranks based on $e_D^m(s,t)$.

Method	Table	Chair	Sofa	Car
Ranked-CD	50.50	52.52	46.91	54.26
AE	54.73	54.41	49.76	43.89
CD-Margin	51.04	50.69	44.53	39.20
CD-reg	18.34	17.43	16.64	38.07
Symm-Margin	61.35	61.29	53.25	45.03
Symm-Reg	56.52	14.23	16.80	<u>61.22</u>
Ours-Margin	<u>64.26</u>	<u>65.55</u>	<u>58.32</u>	46.73
Ours-Reg	**70.64**	**73.97**	**65.61**	**67.19**

Table S4. Quantitative comparison of *Ours-Margin* with and without the hard negative mining and *Ours-Reg*, experimented on ShapeNet [2]. The fitting gap $e_D^m(\mathbf{s},\mathbf{t})$ multiplied by $1e^{-2}$ are reported. Bold is the smallest.

Method	Table		Chair		Sofa		Car	
	Top-1	Top-3	Top-1	Top-3	Top-1	Top-3	Top-1	Top-3
Ours-Margin	2.127	1.251	1.915	1.144	1.420	0.835	1.226	0.747
Ours-Margin w/ hardneg	2.090	1.233	1.904	1.131	1.400	0.822	1.220	0.744
Ours-Reg	**1.969**	**1.129**	**1.752**	**1.054**	**1.338**	**0.788**	**1.112**	**0.681**

S.4 Additional Evaluation Metric - Recall

We also report recall of the retrieval results. Since the notion of the *correct* match is not defined in our problem, we compute recall@1 by calculating the proportion of the cases when the top-1 retrieval is in the top-5 ranks based on $e_D^m(s,t)$. The results are reported in Tab. S3. Ours outperforms the baselines with big margins.

S.5 Hard Negative Mining in the Margin-Loss-Based Approach

For our margin-loss-based approach (*Ours-Margin*) described in Sec. 3.2 in the paper, we also tried hard negative mining [3] in the network training. For each query, we generate the set of negative samples $\mathbf{N}'_\mathbf{t}$ with the 8 hardest negatives in $\mathbf{N}_\mathbf{t}$ (the closest to the query by the learned egocentric distance $\delta(\mathbf{t};\mathbf{s})$) and 5 other randomly selected negatives; the additional random negatives are added to avoid overfitting. For training efficiency, instead of forward-propagating the network for each step to compute the egocentric distance $\delta(\mathbf{t};\mathbf{s})$, we cache the latent vectors $\mathcal{F}(\cdot)$ and the distance field (PSD) matrices $\mathcal{G}(\cdot)$ for all the models in the database and update them every 10 epochs. The hard negative mining was tested in the fine-tuning, and the network model was first trained in the normal way (with all randomly selected negatives) for 30 epochs. Tab. S4 shows the quantitative results on ShapeNet [2], indicating that the hard negative mining slightly improves the performance. But, *Ours-Reg* still performs better than *Ours-Margin* in all classes.

Table S5. Qualitative comparisons on ShapeNet Table dataset with the varying dimension of the latent space. The fitting gap $e^m_{\mathcal{D}}(\mathbf{s}, \mathbf{t})$ multiplied by $1e^{-2}$ are reported. Bold is the smallest among the dimensions.

Dimension	AE		CD-Margin		**Ours-Margin**		**Ours-Reg**	
	Top-1	Top-3	Top-1	Top-3	Top-1	Top-3	Top-1	Top-3
$d = 64$	2.481	1.489	2.429	1.418	2.159	**1.229**	1.997	1.153
$d = 128$	**2.325**	1.357	2.369	1.380	2.131	1.243	1.981	1.133
$d = 256$	2.331	**1.334**	2.362	1.373	2.127	1.251	**1.969**	**1.129**
$d = 512$	2.330	1.351	**2.323**	**1.370**	**2.092**	1.235	2.006	1.143

S.6 Analysis of Latent Space Dimension

As mentioned in Sec. 4 in the paper, we demonstrate the effect of varying the dimension of the latent space, both for baselines and our methods. Tab. S5 shows the quantitative results on ShapeNet Table dataset when varying the dimension of the latent space from 64 to 512. While the higher dimensions mostly offer slightly better performance, the difference is marginal, meaning that even the smallest dimension ($d = 64$) has sufficient capacity to encode the asymmetric deformability relationships. Also, regardless of the dimension, our methods consistently outperform the baselines with significant margins.

S.7 More Qualitative Results

In the following figures, we show more qualitative comparisons between our method and baseline methods for the experiments of ShapeNet (Sec. 5.1), Scan-to-CAD (Sec. 5.2) in the paper, and Image-to-CAD (Sec. S.2).

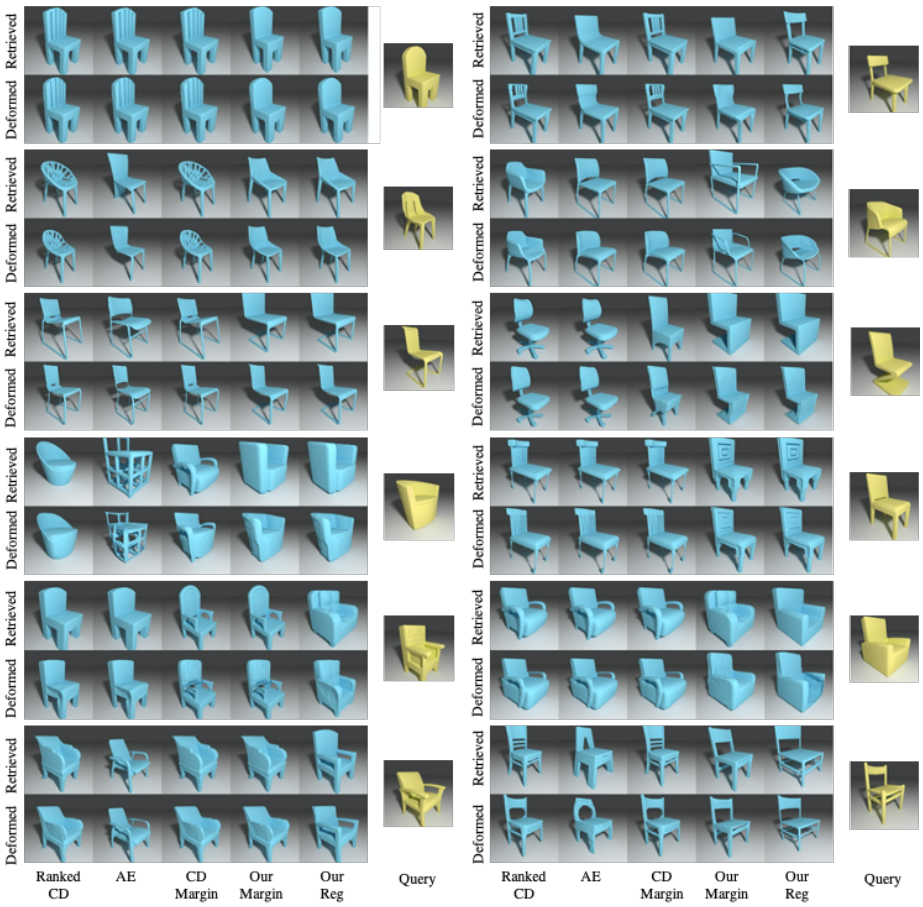

Fig. S3. More qualitative results of ShapeNet experiment (chairs). See Sec. 5.1 in the paper for the details.

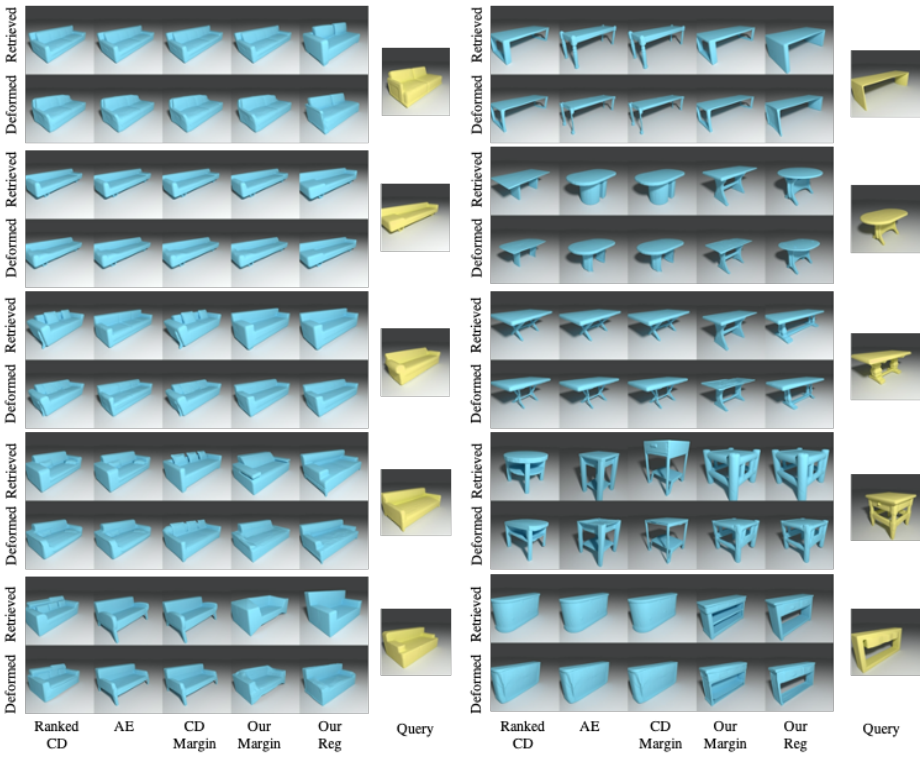

Fig. S4. More qualitative results of ShapeNet experiment (sofas and tables). See Sec. 5.1 in the paper for the details.

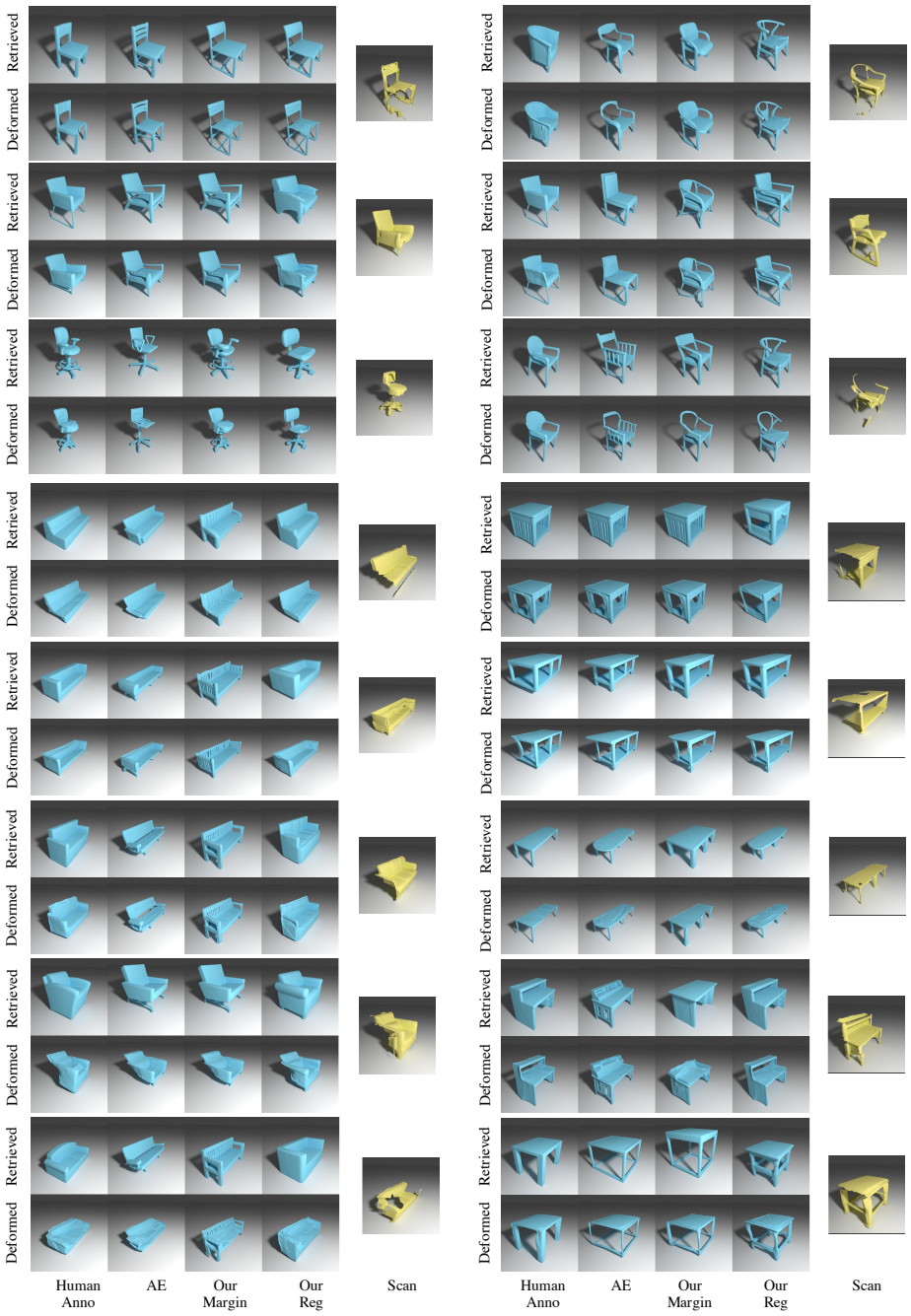

Fig. S5. More qualitative results of Scan-to-CAD experiment (chairs, tables, sofas). See Sec. 5.2 in the paper for the details.

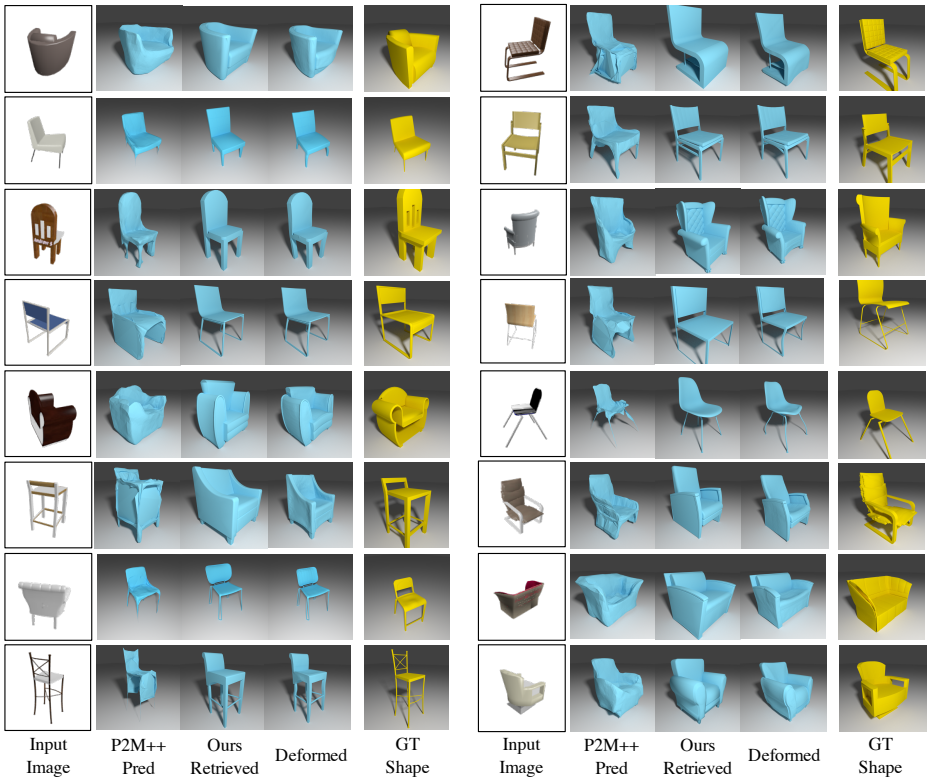

Fig. S6. More qualitative results of Image-to-CAD experiment. See Sec. S.2 for the details.

References

1. Agarwal, S., Mierle, K., Others: Ceres solver. http://ceres-solver.org 1
2. Chang, A.X., Funkhouser, T.A., Guibas, L.J., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., Xiao, J., Yi, L., Yu, F.: Shapenet: An information-rich 3d model repository (2015) 2, 4
3. Schroff, F., Kalenichenko, D., Philbin, J.: FaceNet: A unified embedding for face recognition and clustering. In: CVPR (2015) 4
4. Sorkine, O., Alexa, M.: As-rigid-as-possible surface modeling. In: Eurographics Symposium on Geometry Processing (2007) 1
5. Wen, C., Zhang, Y., Li, Z., Fu, Y.: Pixel2mesh++: Multi-view 3d mesh generation via deformation. In: ICCV (2019) 2