Joint Learning of 3D Shape Retrieval and Deformation

Mikaela Angelina Uy[1] Vladimir G. Kim[2] Minhyuk Sung[3] Noam Aigerman[2]
Siddhartha Chaudhuri[2,4] Leonidas Guibas[1]
[1]Stanford University [2]Adobe Research [3]KAIST [4]IIT Bombay

We provide additional implementation details (Section S.1), and additional quantitative evaluations (Section S.2.1) and qualitative results (Section S.2.2).

S.1. Implementation Details

Inner Deformation Optimization. We provide additional details for the inner deformation optimization step, as described in Section 3.1 of the main paper.

We initialize the inner deformation optimization with the parameters predicted by our deformation network. We propagate gradients directly to the parameters by minimizing the mean chamfer loss of the batch. We use the SGD optimizer with a learning rate of 0.05, and we terminate upon convergence (i.e., when the maximum loss change in a pair in the batch is less than 10^{-6} or it has reach the maximum number of iterations $= 2000$).

Structure-Aware Neural Deformation. We provide additional details for our structure-aware neural deformation as described in Section 3.2 of the main paper.

Our structure-aware neural deformation module predicts the deformation parameter offset from the default parameters of each source model. Specifically for a specific source-target pair, given network prediction p and default source parameter $\bar{p}$, our output parameters to obtain the deformed source model is given by $(\bar{p} + \alpha * p$, where $\alpha = 0.1)$ in all our experiments.

We also add the symmetry loss to supervise the training of our structure-aware neural deformation. Note that all the source shapes in our databases have global reflective symmetry, and have been pre-aligned so that yz-plane aligns with the symmetry axis. Given the output deformed source shape, represented as a sampled point cloud O, for target point cloud T of given target $\mathbf{t}$, we reflect each point O about the yz-plane to obtain reflected point cloud O', then the symmetry loss is given by

$$\mathcal{L}_{\text{symm}} = \mathcal{L}_{\text{CD}}(O, O'),$$

where $\mathcal{L}_{\text{CD}}$ is the chamfer distance. Then the loss we use to train our deformation module is given by

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{def}} + \mathcal{L}_{\text{symm}},$$

where $\mathcal{L}_{\text{def}}$ is defined in Equation 4 in the main paper.

Connectivity constraint. We provide the details on how we obtain our connectivity constraint as described in Section 3.2 of the main paper.

We precompute the constraint projection matrix for each source $\mathbf{s} \in \mathbf{S}$ in an automatic pre-processing step, where we first identify contacts based on the distance between the closest pairs of keypoints between pairs of parts $(\mathbf{s}^i_{\mathcal{D}}, \mathbf{s}^j_{\mathcal{D}})$. Parts $\mathbf{s}^i_{\mathcal{D}}$ and $\mathbf{s}^j_{\mathcal{D}}$ are deemed connected if the closest part of keypoints falls below a threshold $\tau = 0.05$. Part keypoints is the set of face centers, edge midpoints, and corners of each part's axis-aligned bounding box. We then define contacts as the midpoint of the closest pair of keypoints of two connected parts, and obtain 3 linear constraints (one for each axis) for each pair of connected parts that enforces the contact point to maintain connectivity during deformation. We obtain a number of linear constraints from the collection of contacts that results in a different number of linear constraints for each source model. We concatenate all the linear constraints and represent these with constraint matrix $B_{\mathbf{s}}$ for source model $\mathbf{s}$. Let $Q_{\mathbf{s}}$ be the nullspace, *i.e.* columns representing the nullspace basis vectors, of $B_{\mathbf{s}}$ computed via SVD, then the constraint projection matrix of $\mathbf{s}$ is given by $Q_{\mathbf{s}}Q_{\mathbf{s}}^T$.

Training details and training time. We alternately update the retrieval module and the deformation module at each iteration during our training procedure, and train for 300 epochs. To speedup training, we cache the distances to the sources for each target and update this cache every 5 epochs. We use a batch size of 16 targets in each iteration, the SGD optimizer with learning rate of 0.001, momentum of 0.9 and weight decay of 0.0005. For the inner deformation optimization, also use the SGD optimizer with a learning rate of 0.05 until the termination criteria is reached, which is when the fitting loss decreases by less than 10^{-5} or the maximum number of 5000 iterations is reached.

For our joint training module, we first train our Structure-Aware neural deformation module until convergence on random pairs, and also train our retrieval module on random

	Chair	Table	Cabinet
DAR+DF (No Conn.)	1.107	1.728	1.480
Uniform Sampling (No Conn.)	1.129	1.655	1.358
Ours (No Conn.)	**0.757**	**0.708**	**0.846**

Table S1. Our approach compared to the baselines in the setup with no connectivity constraint.

pairs to initialize our joint training optimization scheme. Also note that when training image-based ResNet encoder for the retrieval and deformation modules, we warm-start with weights that are pre-trained on ImageNet, and only train the fourth block and the final fully-connected layers.

Training takes 18 and 40 hours on point clouds and images, respectively, for the chair class. With the inner loop direct optimization, the corresponding training time for chairs takes 3 days for both the point cloud and image experiments as the inner optimization dominates the runtime.

S.2. Additional Results

S.2.1 Additional Quantitative Evaluations

No connectivity constraint ablation. We also test our joint training scheme in the setting where the source database models do not have connectivity constraints. In this set-up we do not use the constraint projection matrix. Table S1 shows that even in the set-up with no connectivity, our approach achieves the best results in all three object classes.

Retrieval-and-deformation results for different retrieved sources. We further evaluate how well our method works with other than top-1 retrieved source. In particular, we plot the mean chamfer distance for the k^{th} retrieved source, for $k = 1, 2, 3, 4, 5$.

For image-to-mesh experiment, we show the result in Figure S1, which complements Table 1 of the main paper. For points-to-mesh experiment, we show the result in Figure S2, which complements Table 3 of the main paper. Note that in both cases the chamfer distance for up to top-5 retrieved results is consistently lower than the baselines.

Retrieval module evaluation. We further evaluate the retrieval modules of our joint approach compared to the baselines. To evaluate the retrieval module, we report both *ranking evaluation* and *recall* similar to the metrics used in [6].

One challenge in defining an evaluation metric is that we do not know which source model should be used for each target. Thus, to create the ground truth we use *oracle retrieval*, where we use the each method's deformation module to deform each source to the target, and assume that if we sort the sources by the chamfer distance, it will give us the desired ground truth ordering for the retrieval.

Ranking evaluation reports the average rank of the top-1 retrieved model with respect to this ground truth. We report the metrics for image-to-mesh (Table S2) and points-to-mesh (Table S3) experiments, across all categories, and see consistent improvement with respect to the baselines.

	Chair	Table	Cabinet
DAR+DF	23.98	59.51	19.50
Uniform Sampling	20.88	53.01	23.39
Ours	**15.35**	**22.19**	21.70
Ours w/ IDO	21.94	**36.92**	**16.89**

Table S2. **Ranking evaluation for retrieval.** Comparing our method using the ranking evaluation metric on image-to-mesh benchmark. Numbers show the average rank of the retrieved model. (Lower is better)

	Chair	Table	Cabinet
DAR+DF	13.88	76.25	20.20
Uniform Sampling	18.27	72.44	23.44
Ours	**6.37**	**6.97**	**17.91**
Ours w/ IDO	**6.62**	**18.03**	**18.22**

Table S3. **Ranking evaluation for retrieval.** Comparing our method using the ranking evaluation metric on points-to-mesh benchmark. Numbers show the average rank of the retrieved model. (Lower is better)

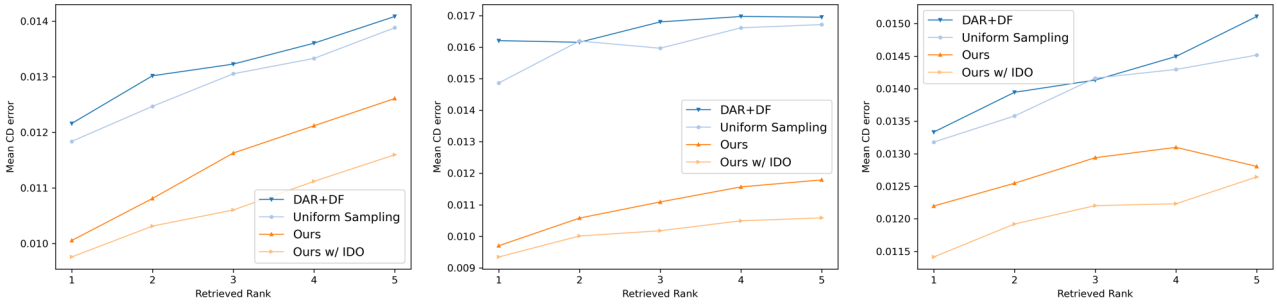
	Chair		Table		Cabinet	
	recall@1	recall@5	recall@1	recall@5	recall@1	recall@5
DAR+DF	37.53	74.65	14.55	43.46	22.37	57.89
Uniform Sampling	38.94	75.56	21.90	54.79	21.05	53.81
Ours	**53.60**	**81.03**	**53.81**	**82.93**	**30.70**	**61.40**
Ours w/ IDO	**45.65**	**77.30**	**35.83**	**69.35**	**35.96**	**65.79**

Table S4. **Recall evaluation for retrieval.** Comparing our method using the ranking evaluation metric on image-to-mesh benchmark. Numbers show recall@1 and recall@5. A correct retrieval is when the top-1 and top-5 retrieved models is in the top-5 ranks based on the oracle retrieval. (Higher is better)

We also report the recall of retrieval modules. For recall@N, a correct match is defined as the case where at least one of the top-N retrieved models is in the top-5 ranks based on the oracle retrieval module. We report both recall@1 and recall@5. We report the metrics for image-to-mesh (Table S4) and points-to-mesh (Table S5) experiments, across all categories, and see consistent improvement with respect to the baselines.

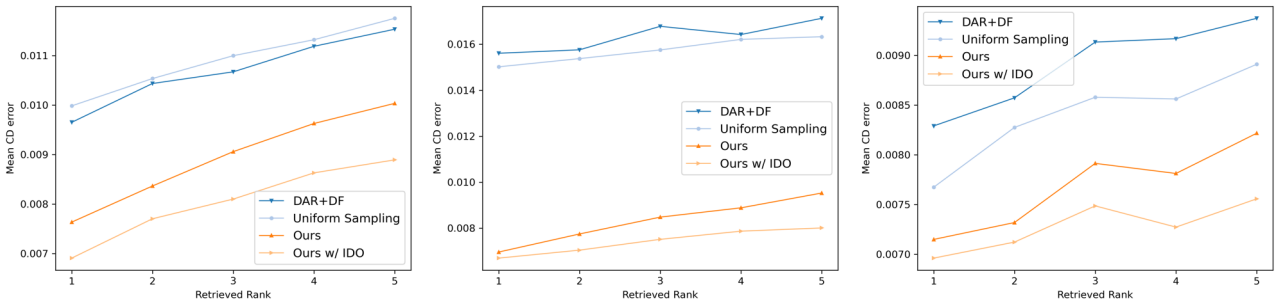
Additional object categories. We ran experiments on additional categories (vases, beds, trash cans), and a combination of categories (chairs+tables+cabinets). As shown in Table S6, we got a comparable performance and improvement over baselines.

Perceptual Metric. We performed a user study comparing our approach to the DAR+DF baseline. We asked 60 participants to pick the better match to input point clouds on 15 randomly selected targets from the test set, where an option of "no difference" can also be selected. Our approach got



(a) Chair (b) Table (c) Cabinet

Figure S1. Quantitative evaluation of Image-to-Mesh.



(a) Chair (b) Table (c) Cabinet

Figure S2. Quantitative evaluation of Points-to-Mesh.

	Chair		Table		Cabinet	
	recall@1	recall@5	recall@1	recall@5	recall@1	recall@5
DAR+DF	61.56	93.54	23.57	54.54	39.83	72.29
Uniform Sampling	53.27	89.98	25.03	59.16	39.83	67.97
Ours	**75.31**	**97.02**	**73.71**	**96.50**	**48.05**	**76.19**
Ours w/ IDO	**76.22**	**96.60**	**55.17**	**89.72**	38.53	**77.06**

Table S5. **Recall evaluation for retrieval.** Comparing our method using the ranking evaluation metric on points-to-mesh benchmark. Numbers show recall@1 and recall@5. A correct retrieval is when the top-1 and top-5 retrieved models is in the top-5 ranks based on the oracle retrieval. (Higher is better)

	Vase	Bed	Trash Can	Combined
DAR+DF	1.538	4.498	0.889	1.968
Uniform Sampling	1.633	4.196	0.886	1.821
Ours	**1.384**	**2.138**	**0.863**	**0.810**

Table S6. **Additional object categories.** Comparing our method to various baselines and ablations on additional object classes and mixture of categories (chamfer distances, $\times 10^{-2}$).

an average score of **8.02**, compared to 3.5 for the baseline and 3.48 abstain votes.

S.2.2 Additional Qualitative Results.

We provide additional qualitative results using natural images, point cloud scans, and our benchmark as input targets. Note that in all visualizations, we use colors to indicate different segmentations of the source models, where segmentation is essential to the performance of the structure-aware neural deformation module.

Product images targets. Figure S7 shows additional qualitative results of our approach on product images.

Scan2CAD targets. Figure S5 shows additional results of our approach on real scans from the Scan2CAD [1] dataset using the manually segmented PartNet [4] database, while Figure S6 shows the results on real scans using the auto-segmented ComplementMe [5] database.

Image-to-Mesh baseline comparison. Figure S8 shows additional qualitative results on the image-to-mesh set-up that compares our method to the baselines.

Points-to-Mesh baseline comparison. Figure S4 shows additional qualitative results of our joint approach compared to the baselines on the points-to-mesh experiment.

Neural cages. Figure S3 shows additional qualitative results of our joint approach on Neural Cages [7].

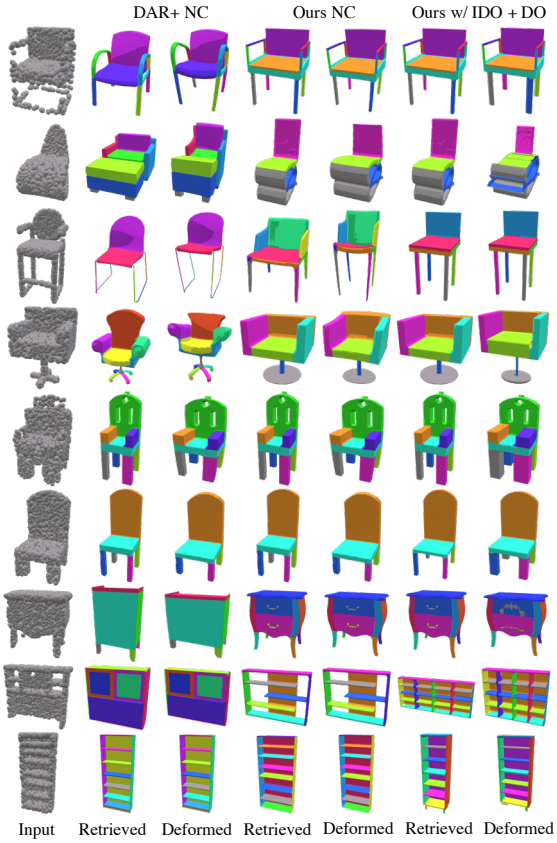


Figure S3. More qualitative results on Neural Cages [7].

Points-to-Mesh ablations qualitative results. Figure S9 shows qualitative results of ablations of our joint approach on the points-to-mesh experiment.

S.3. Discussion on [2]

The differences between our work and with [2] are as follows:

1. **Non-learnable deformations**: The fitting module of [2] is *not learnable*; they directly *optimize* parameters of a handcrafted template to fit to an input point cloud. Thus, one of our key contributions, a *retrieval-aware deformation*, is incompatible with their method.
2. **Infeasibility of image-to-mesh**: Without learnable deformations, their method cannot be used for the main application of our method, image-to-mesh generation.
3. **Manually-designed templates**: Designing templates is a tedious manual task that requires significant expertise. Their method requires users to pre-design a set of templates, hence they only use a small set of 21 templates.
4. **Non-scalable system**: While one could address solving our retrieval problem as a classification problem by treating every source shape as a template, this approach is not scalable. Their method requires a pre-process of matching every template to every input shape for training. Their optimization-based deformation module takes 2-3 mins for a single pair, and thus for all 500 sources and 4000 training targets as in our chair dataset, it would take $\sim$ 8 years. Note that this limitation has been addressed in a recent work of Uy et al. [6] who propose to learn a *deformation-aware retrieval* latent space instead of the non-scalable hard shape-to-template assignment (and we extensively compared to Uy et al. [6]).
5. **Specific to template-based deformations**: Our key contribution, *joint* learning for retrieval and deformation, is not constrained to a specific choice of the deformation module.

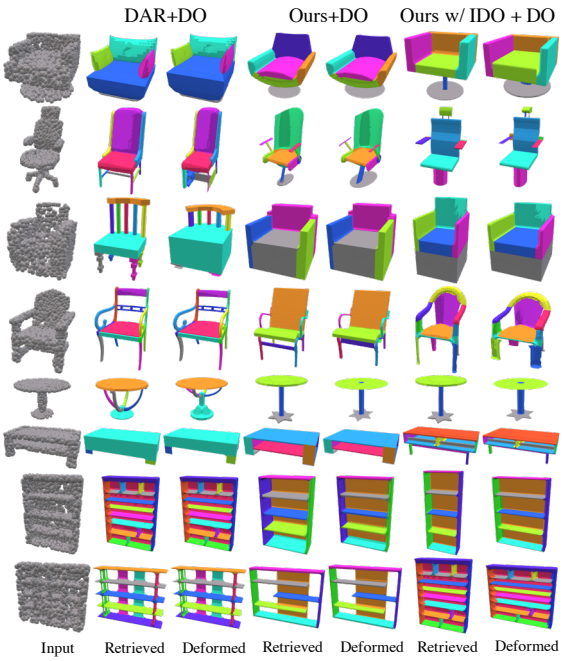


Figure S4. Additional qualitative results on comparisons between our approach and the baselines for the points-to-mesh experiments.

We also remark that, while both ours and their method leverage on part bounding boxes for deformations, neither of these two were the first to use bounding boxes to deform the underlying geometry (e.g., [3]).

References

[1] Armen Avetisyan, Manuel Dahnert, Angela Dai, Manolis Savva, Angel X Chang, and Matthias Nießner. Scan2CAD: Learning cad model alignment in RGB-D scans. In *CVPR*, 2019. 3, 5

[2] V. Ganapathi-Subramanian, O. Diamanti, S. Pirk, Chengcheng Tang, M. Nießner, and L. Guibas. Parsing

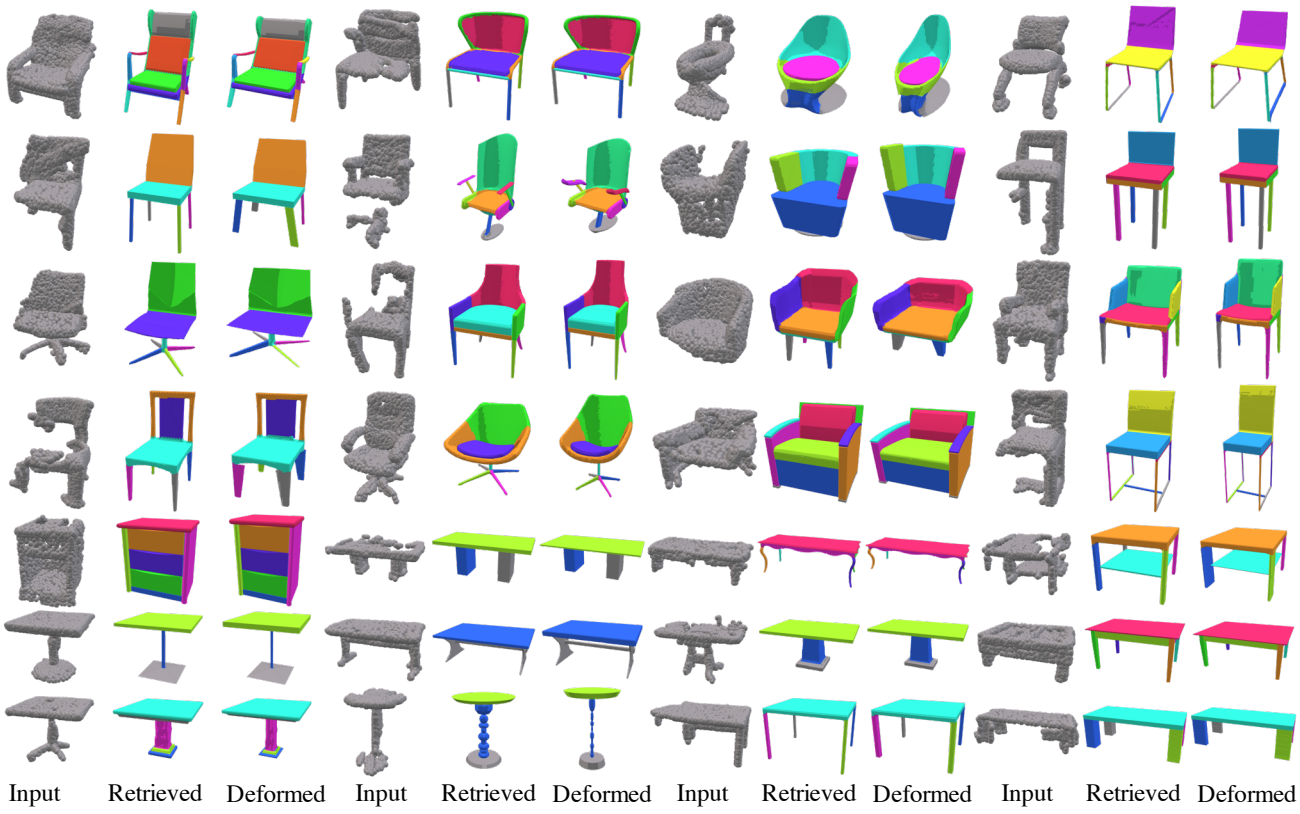


Figure S5. More qualitative results using the Scan2CAD [1] dataset using manually segmented shapes in PartNet [4].

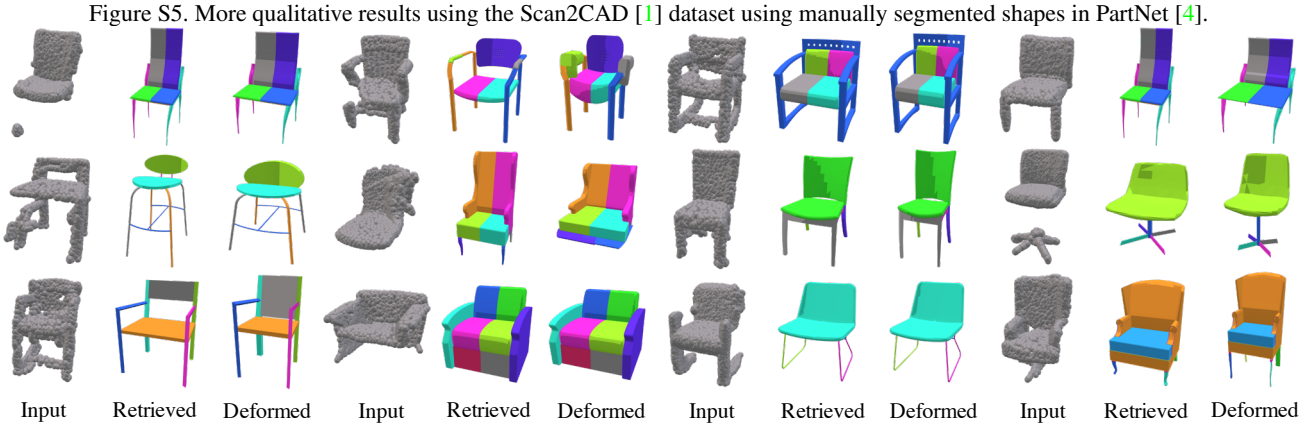


Figure S6. More qualitative results using the Scan2CAD [1] dataset using autosegmented shapes in ComplementMe [5].

geometry using structure-aware shape templates. In *3DV*, 2018. 4

[3] Vladimir G. Kim, Wilmot Li, Niloy J. Mitra, Siddhartha Chaudhuri, Stephen DiVerdi, and Thomas Funkhouser. Learning part-based templates from large collections of 3D shapes. In *SIGGRAPH*, 2013. 4

[4] Kaichun Mo, Shilin Zhu, Angel X. Chang, Li Yi, Subarna Tripathi, Leonidas J. Guibas, and Hao Su. PartNet: A large-scale benchmark for fine-grained and hierarchical part-level 3D object understanding. In *CVPR*, 2019. 3, 5

[5] Minhyuk Sung, Hao Su, Vladimir G. Kim, Siddhartha Chaudhuri, and Leonidas Guibas. ComplementMe: Weakly-supervised component suggestions for 3D modeling. In *SIGGRAPH Asia*, 2017. 3, 5

[6] Mikaela Angelina Uy, Jingwei Huang, Minhyuk Sung, Tolga Birdal, and Leonidas Guibas. Deformation-Aware 3D model embedding and retrival. In *ECCV*, 2020. 2, 4

[7] Wang Yifan, Noam Aigerman, Vladimir Kim, Siddhartha Chaudhuri, and Olga Sorkine-Hornung. Neural cages for detail-preserving 3D deformations. In *CVPR*, 2020. 3, 4

Figure S7. More qualitative results on product images.

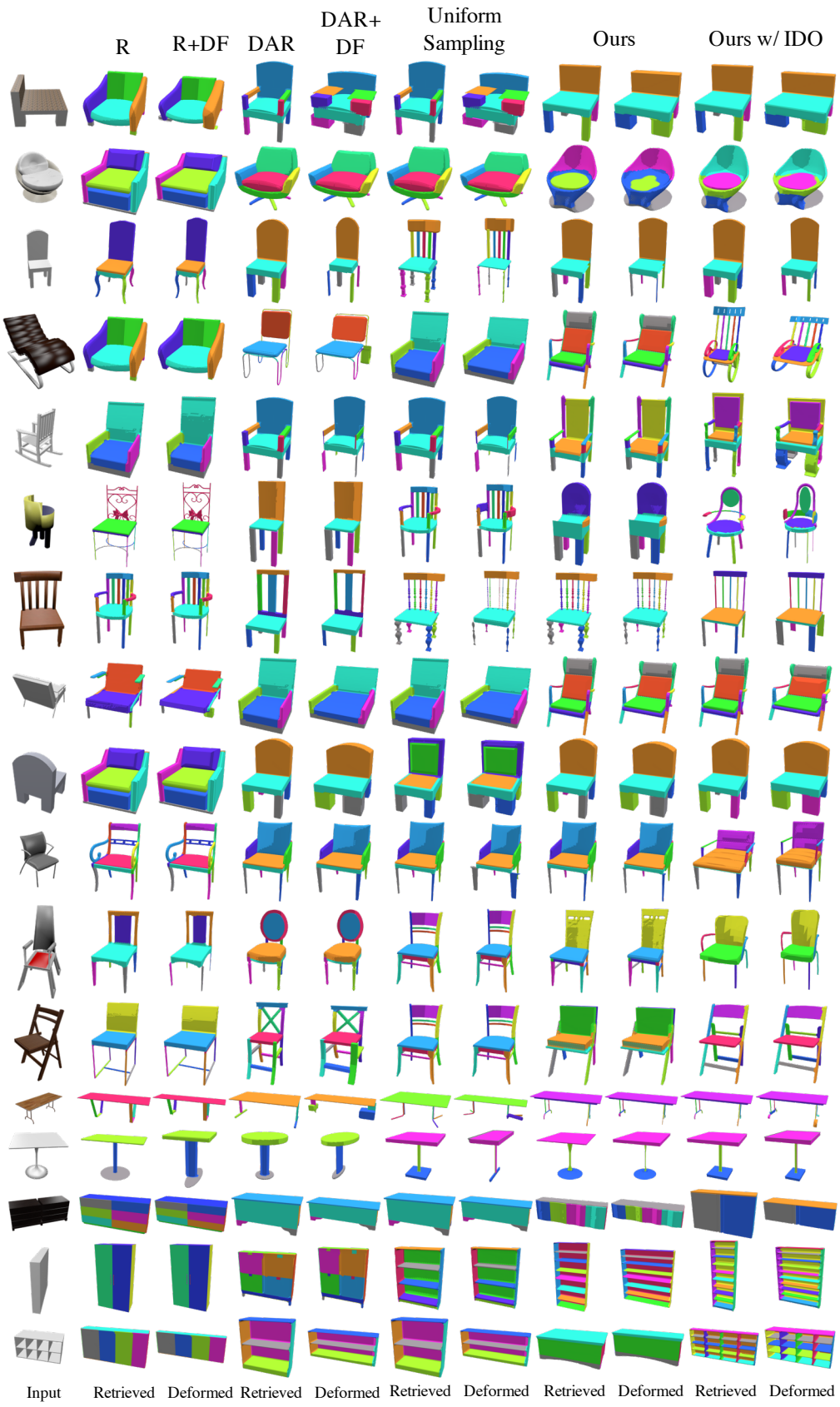


Figure S8. More qualitative results on Image-to-Mesh.

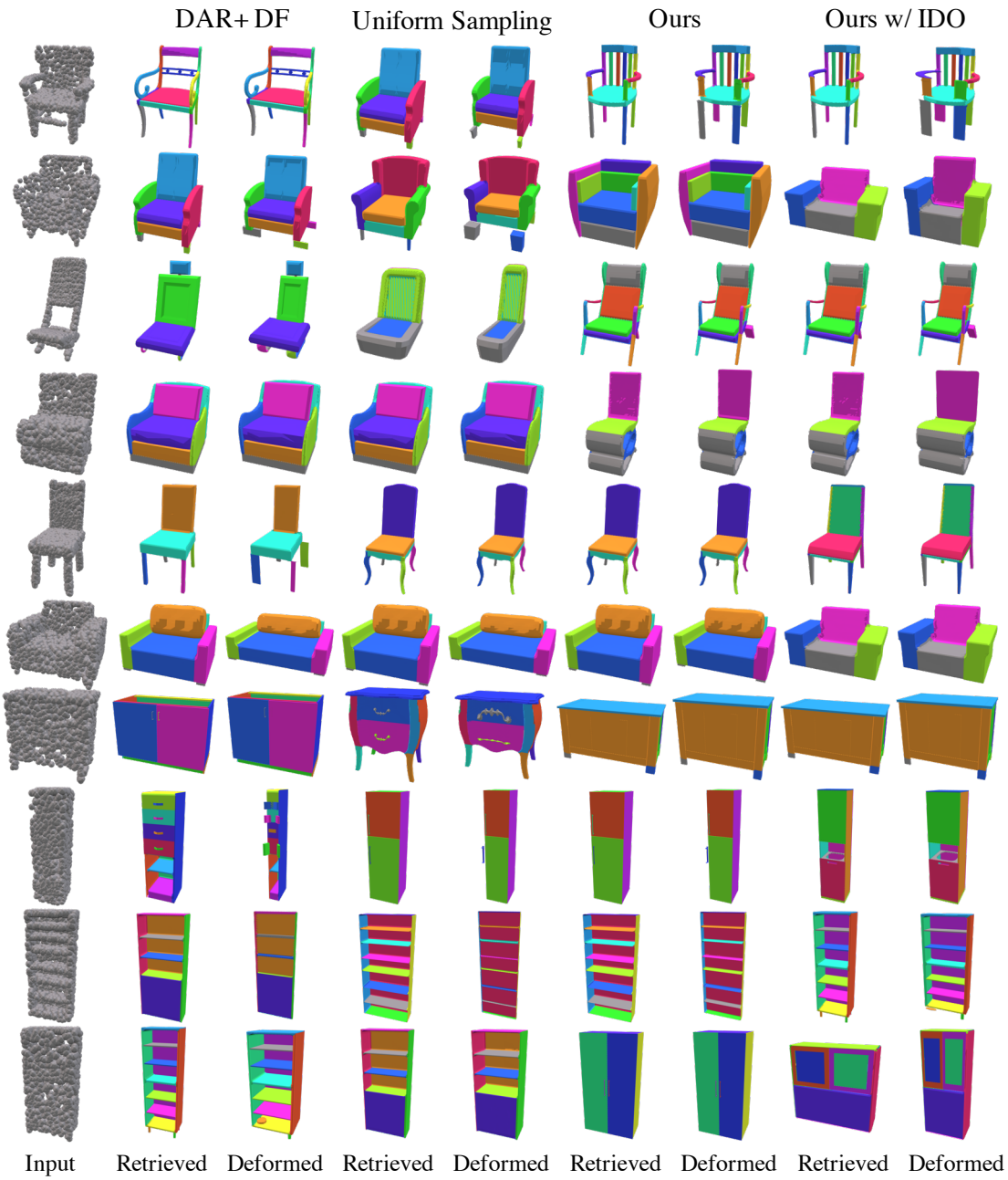


Figure S9. More qualitative results on Points-to-Mesh.