ConDor: Self-Supervised Canonicalization of 3D Pose for Partial Shapes Supplementary Material

Rahul Sajnani[1] Adrien Poulenard[2] Jivitesh Jain[1] Radhika Dua[3]
Leonidas J. Guibas[2] Srinath Sridhar[4]
[1]RRC, IIIT-Hyderabad [2]Stanford University [3]KAIST [4]Brown University

1. Network details

1.1. Architecture

We reuse the classification architecture described in Section 3.1 of [2] as our backbone. The architecture comprises of three equivariant convolution layers followed by a global max-pooling layer, and the remaining layers specialize for classification; we drop these last layers and specialize the network for our tasks instead. The global max-pooling layer of [2] proceeds by first interpreting each point-wise signal as coefficients of spherical functions in the SH basis and performing a discrete inverse spherical harmonics transform to convert them into functions over a discrete sampling of the sphere. For any direction, the resulting signal is then spatially pooled over the shape, resulting in a single function over the sphere sampling (specifically, a single map from the sphere sampling to $\mathbf{R}^C$, where $C = 256$ as we have 256 channels). We then apply point-wise MLPs (with ReLU activations) on this sphere map and convert it back to TFN-like features via forward spherical harmonics transform (SHT) [2].

Spherical Harmonic Coefficients: In order to predict the coefficients $F(X)$ of the invariant embedding $H(X)$, we apply a $[128, 64]$-MLP whose last layer is linear and convert to types $\ell \in [\![0, 3]\!]$ via SHT.

Rotation-Invariant Point Cloud: We obtain our 3D invariant point cloud X^c by applying a linear layer to $H(X)$.

Rotation-Equivariant Frame: To predict E, we apply a $[64, 3]$-MLP whose last layer has a linear activation. We then extract type 1 features with SHT, giving us a collection of 3 equivariant 3D vectors.

Segmentation: To predict the segmentation we apply a point-wise $[256, 128, 10]$-MLP whose last layer is soft-max to get the segmentation masks S described in Section 2.

1.2. Training Details

Cropping operator $\mathcal{O}$: We introduce synthetic occlusion in our training setting by slicing full shapes using the cropping operator $\mathcal{O}$. To perform a crop, we uniformly sample a direction v on the unit sphere and remove the top $K/2$ points in the shape that have the highest value of $x^T v$ for $x \in X$. Additionally, we train our model on the ShapeNet-COCO dataset [3, 5] which has pre-determined occlusion due to camera motion, as seen in Figure 2. In order to pre-process this data for training, we aggregate the parts in the canonical NOCS space of every sequence to obtain the full shape and perform a nearest neighbor search in the NOCS space to find correspondences between the full and partial shape.

Hyper-parameters: During training, we use a batch size of 16 in every step for all our models. We set an L^1 kernel regularizer at every layer of the network with weight 0.1. We weigh the loss functions by their effect on reducing the Canonical Shape loss $\mathcal{L}_{canon}$. The loss functions are weighed as: $\mathcal{L}_{canon}$ (2), $\mathcal{L}_{rest}$ (1), $\mathcal{L}_{ortho}$ (1), $\mathcal{L}_{sep}$ (0.8), and $\mathcal{L}_{amod}$ (1).

2. Unsupervised Co-segmentation

2.1. Predicting parts

We predict the part segments $S \in \mathbb{R}^{K \times C}$ wherein C are the number of parts. We use the rotation-invariant embedding $H^\ell(X)$ with all the types $0 \leq \ell \leq 3$ to predict the segmentation S. We define the following notation for normalized parts $A(X)$ and part centroids $\theta(X)$ similar to [6].:

$$\begin{aligned} S(X) &:= \text{Softmax}[\text{MLP}(H(X))] \\ A_{ij}(X) &:= \frac{S_{ij}(X)}{\sum_i S_{ij}(X)} \\ \theta_j(X) &:= \sum_i A_{ij}(X) X_{i,:} \end{aligned} \tag{1}$$

2.2. Loss functions

We use part segmentation to enforce semantic consistency between full and partial shapes. We borrow the localization loss ($\mathcal{L}_{localization}$) and equilibrium loss ($\mathcal{L}_{equilibrium}$) from [6] for the full shape to evenly spread

part segmentation across the shape. Additionally, we employ the following losses.

Part Distribution loss: We compute the two-way Chamfer distance (CD) between the part centroids and the input shape. In practice, this helps to distribute parts more evenly across the shape.

$$\mathcal{L}_{dist} = \text{CD}\left(X, \theta(X)\right) \tag{2}$$

Part Restriction loss: The parts discovered by the network for the partial shape should be congruent to the parts discovered by the network for the full shape. We penalize the part prediction for corresponding parts by minimizing the negative Cosine Similarity (CS) for our capsule predictions.

$$\mathcal{L}_{rest(part)} = -\frac{2}{K}\sum_{i\in\mathcal{S}} \mathbf{CS}(S(\mathcal{O}(X))_{i,:}, \mathcal{O}(S(X))_{i,:}) \tag{3}$$

Part Directional loss: To avoid part centers of the visible parts of a shape from deviating from the part centers of the full shape, we use a soft loss to ensure that the directional vector between part centers are consistent between the full and partial shape. $\text{dir}(\theta(X))$ computes the vector directions between every ${}^{C}C_2$ centroid pairs for C part centroids.

$$\mathcal{L}_{direc} = -\frac{1}{{}^{C}C_2}\sum_{i\in\mathcal{S}} \mathbf{CS}(\text{dir}(\theta(\mathcal{O}(X_i))), \text{dir}(\mathcal{O}(\theta(X_i)))) \tag{4}$$

3. Registration

Table 1. **Registration** – Distance in terms of root mean-square error (RMSE) and Chamfer distance between registered and ground-truth points on the ShapeNet (core) dataset for full shapes only.

	RMSE↓			Chamfer (CD)↓		
Method	**Airplane**	**Chair**	**Multi**	**Airplane**	**Chair**	**Multi**
PCA	0.616	0.695	0.715	0.050	0.097	0.054
Deep Closest Points [7]	0.318	0.160	0.131	-	-	-
Deep GMR [9]	0.079	0.082	0.077	-	-	-
CaCa [6]	**0.024**	**0.027**	**0.070**	**0.009**	0.026	0.040
Compass [4]	0.361	0.369	0.487	0.061	0.079	0.051
Ours (F)	0.254	0.314	0.496	0.015	0.026	0.040
Ours (F + P)	0.201	0.280	0.404	0.014	**0.023**	**0.033**

We note in Table 1 that our method does not perform well in this task as we predict a frame $E \in O(3)$ which can have reflection symmetries, we observe symmetries such as left-right reflection for planes. Symmetries cause high RMSE error because points are matched with their image under symmetry which are often very distant. However, when using Chamfer Distance metric which is symmetry agnostic our registration error decreases by an order of magnitude achieving competitive results on this benchmark. We also note that Ours(F+P) noticeably decreases RMSE compared to Ours(F) as during training the frame consistency is enforced between the full shape and a randomly rotated partial by the $\mathcal{L}_{rest}$ loss.

4. Ablations

We now provide detailed ablations to justify the following key design choices: the effect of increasing amounts of occlusion/partiality, and loss functions.

Degree of Occlusion/Partiality: We examine the ability of our model to handle varying amounts of occlusion/partiality for the car category in Table 2. Our occlusion function, $\mathcal{O}$, occludes shapes to only keep a fraction of the original shape between 25% and 75% (*i.e.*, 75% is more occluded than 25%). We observe that our method performs optimally over all metrics when trained at 50% occlusion.

Test partiality	**Degree of partiality during training**			
	75%	50%	25%	[25%, 75%]
Ground Truth Consistency (GC)↓				
75%	0.0451	**0.0438**	0.1420	0.0681
50%	0.0375	0.0356	0.0504	**0.0296**
25%	0.0388	0.0301	**0.0241**	0.0299
[25%, 75%]	**0.0438**	0.0553	0.0894	0.0558
Instance-Level Consistency (IC)↓				
75%	0.0728	**0.0719**	0.1542	0.0797
50%	0.0452	**0.0349**	0.0526	0.0380
25%	0.0456	0.0333	**0.0221**	0.0334
[25%, 75%]	**0.0719**	0.0792	0.1049	0.0804
Category-Level Consistency (CC)↓				
75%	0.0914	**0.0895**	0.1702	0.0966
50%	0.0652	0.0632	0.0731	**0.0617**
25%	0.0657	0.0608	**0.0582**	0.0606
[25%, 75%]	**0.0895**	0.0985	0.1216	0.0982
Average	0.0594	**0.0580**	0.0886	0.0610

Table 2. **Degree of partiality** - Partiality introduced during training (vertical) is evaluated on the canonicalization metrics with different fraction of partiality (horizontal). [25%, 75%] indicates that degrees of partiality between 25% and 75% are randomly introduced in the shapes. Our model trained with partiality 50% performs better on average over all the canonicalization metrics. [*Note: 75% is more occluded than 25%.*]

Loss Functions: We evaluate our F+P model on both full and partial shapes trained with all losses, without the separation loss $\mathcal{L}_{sep}$, and without the restriction loss $\mathcal{L}_{rest}$. From Table 3, we observe that using restriction loss $\mathcal{L}_{rest}$ helps in canonicalization of both full and partial shapes in categories *plane*, *table*, and *chair*. However, separation loss, $\mathcal{L}_{sep}$, helps in *plane*, *table* but not in *chair*. Since, both losses help in most of the categories, we utilize them for training our final model.

Effect of introducing occlusion on full shapes: We evaluate the canonicalization of full shapes using our network

Category →	Plane			Table			Chair			Average		
Metric ↓	Ours	w/o sep	w/o rest	Ours	w/o sep	w/o rest	Ours	w/o sep	w/o rest	Ours	w/o sep	w/o rest
GC (full)	**0.0286**	0.0321	0.0303	0.0738	**0.0641**	0.0729	0.0509	**0.0430**	0.0532	0.0511	**0.0464**	0.0521
IC (full)	**0.0144**	0.0187	0.0169	**0.0361**	0.0612	0.0411	0.0235	**0.0224**	0.0245	**0.0247**	0.0341	0.0275
CC (full)	**0.0679**	0.0697	0.0683	**0.1432**	0.1510	0.1434	0.1145	0.1150	**0.1143**	**0.1085**	0.1119	0.1087
GC (partial)	0.0360	0.0389	**0.0332**	0.0662	**0.0523**	0.0683	0.0780	**0.0681**	0.0850	0.0601	**0.0531**	0.0622
IC (partial)	**0.0265**	0.0324	0.0479	**0.0739**	0.0791	0.0805	0.0622	**0.0537**	0.0841	**0.0542**	0.0551	0.0708
CC (partial)	**0.0713**	0.0733	0.0765	**0.1579**	0.1590	0.1598	0.1270	**0.1250**	0.1377	**0.1187**	0.1191	0.1247
Average	**0.0408**	0.0442	0.0455	**0.0912**	0.0945	0.0943	0.0760	**0.0712**	0.0831	**0.0696**	0.0700	0.0743

Table 3. Ablation study to investigate the effect of different loss functions. "w/o sep" and "w/o rest" denote training without separation and without restriction loss, respectively.

trained on full and partial shapes. We observe that on average both our models **Ours(F)** and **Ours(F+P)** perform the same on the canonicalization metrics for full shapes. For a few categories such as *lamp, car, chair, watercraft*, introducing partial shapes in the training improves its performance on the canonicalization metrics. Whereas introducing occlusion during training degrades the performance for category *bench*.

5. Applications

5.1. Co-Canonicalization

Commonly used datasets in 3D vision, such as ShapeNet [1], are manually pre-canonicalized, making expansion of such datasets expensive. Since our method performs better than others on canonicalization, we believe that it can be used to extend these datasets by canonicalizing corpora of *in-the-wild* shapes into a common pose. Figure 1 shows the results of our model, trained on the ShapeNet (core) dataset [1], being used to canonicalize shapes from the (uncanonicalized) ModelNet40 dataset [8]. These shapes can now be merged into ShapeNet by applying a single category-wide rotation to match the obtained canonical frame with the existing frame used by ShapeNet, instead of the per-instance rotation that would otherwise be required. Furthermore, these results qualitatively demonstrate the ability of our method to generalize to datasets not seen during training.

5.2. Depth Map Canonicalization

Since our method operates on partial shapes, we can canonicalize objects in **depth images**. We use the depth maps from the ShapeNetCOCO dataset, which have predetermined occlusion due to camera motion, and canonicalize partial point clouds. Specifically, we first take depth maps and utilize them to generate groundtruth pointclouds. We then trained and tested our model on it. Figure 2 present examples to demonstrate that our model is capable of canonicalizing depth maps.

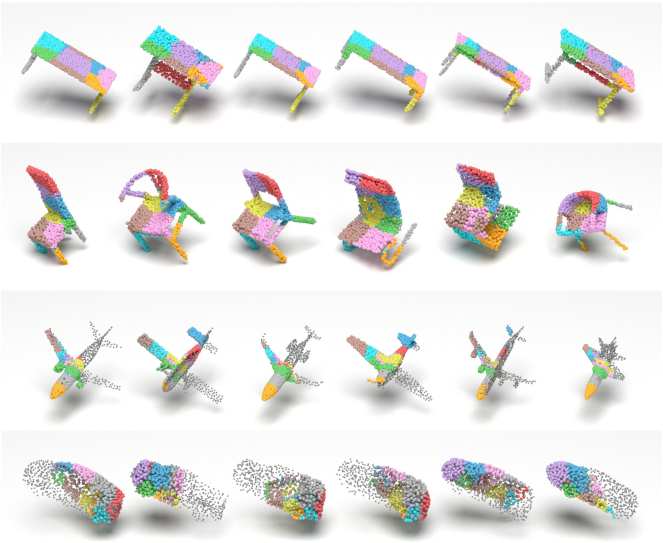

Figure 1. Co-canonicalizing object instances from ModelNet40 using our method trained on ShapeNet (core). (***top***) Canonicalized full shapes. (***bottom***) Canonicalized partial shapes.

5.3. Annotation Transfer

Since a category-level canonical frame is consistent with respect to the geometry and local shape of different object instances of a category, annotations can be transferred across instances that share the same canonical frame. Particularly, we demonstrate the transfer of sparse key-point annotations in Figure 3. We randomly assign labels to a few points of one point cloud in each category, which serves as the source. We then use a remarkably simple transfer function to transfer these labels to points in each target point cloud, making use of the predicted segmentation. To every labeled point in the source point cloud, we obtain a directional vector originating from the centroid of the segment it belongs to. Starting from the corresponding centroid in the target point cloud, we move along this directional vector and then pick the nearest point. While this scheme works

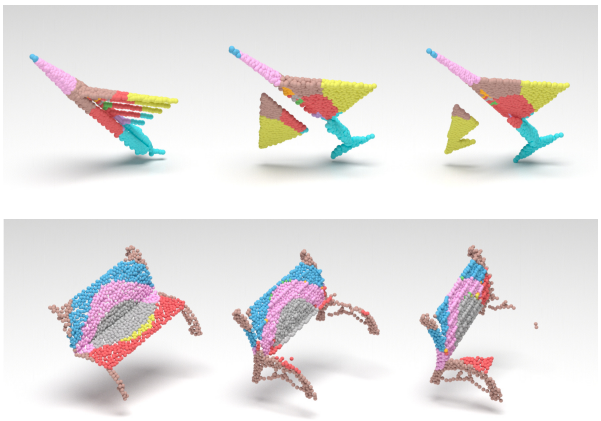

Figure 2. Canonicalizing point clouds obtained from depth maps from the ShapeNetCOCO dataset.

well in our case, more nuanced transfer functions may be required depending on the application.

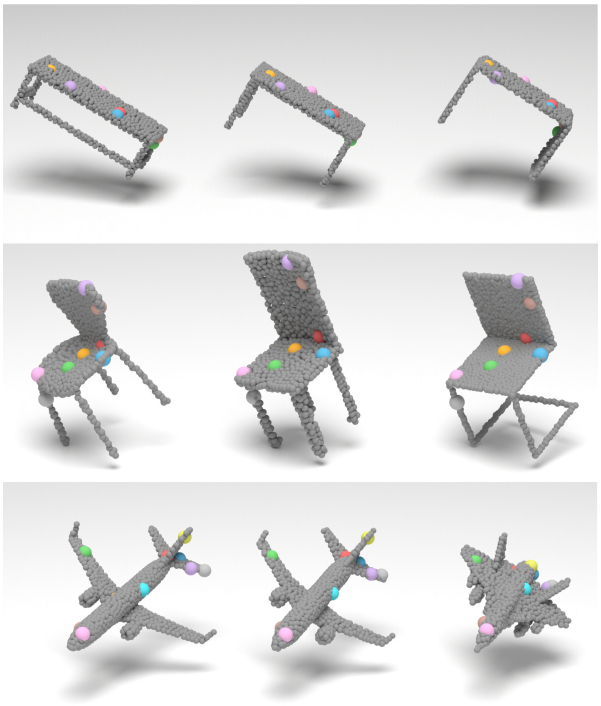

Figure 3. Transferring key-point annotations from one shape to another in the same category. We annotate only the first column of shapes and transfer key-points to all the other columns

6. Proof of Rotation-Invariance Property of our Embedding

Given rotation-equivariant embeddings F^ℓ and Y^ℓ the tensors $H^\ell(X)$ are rotation invariant as:

$$\begin{aligned} H^\ell_{ijk}(R.X) &= \langle F^\ell_{i,:,j}(R.X), Y^\ell_{:,j,k}(R.X)\rangle \\ &= \langle D^\ell(R)F^\ell_{i,:,j}(X), D^\ell(R)Y^\ell_{:,j,k}(X)\rangle \\ &= \langle F^\ell_{i,:,j}(X), Y^\ell_{:,j,k}(X)\rangle = H^\ell_{ijk}(X) \end{aligned}$$

7. Commutative Property of Canonicalization with the Cropping Operator

Canonicalization commutes with the cropping operator $\mathcal{O}$. For a (full) point cloud X and predicted canonicalizing frame $\mathcal{R}(X)$, we prove the commutative property here, we assume X is mean centered for simplification.

$$\begin{aligned} &\widehat{\mathcal{O}[X]}^c + \mathcal{R}(X)\overline{\mathcal{O}[X]} = \mathcal{R}(X)(\widehat{\mathcal{O}[X]} + \overline{\mathcal{O}[X]}) \\ &= \mathcal{R}(X)(\mathcal{O}[X]) = \mathcal{O}[\mathcal{R}(X)X] = \mathcal{O}[X^c] \end{aligned}$$

The above commutative property enables us to a predict a rotation-equivariant translation $\mathcal{T}(\widehat{\mathcal{O}(X)})$ from the mean centered partial shape $\widehat{\mathcal{O}(X)}$ only that aligns the partial shape to its corresponding points in the full shape.

$$\begin{aligned} &\widehat{\mathcal{O}[X]}^c + \mathcal{R}(\widehat{\mathcal{O}[X]})\overline{\mathcal{O}[X]} \simeq \widehat{\mathcal{O}[X]}^c + \mathcal{R}(\widehat{\mathcal{O}[X]})\mathcal{T}(\widehat{\mathcal{O}(X)}) \\ &= \mathcal{O}[X^c] \end{aligned}$$

8. Discussion on Canonicalization Metrics

We complement the discussion of our canonicalization metrics with a few remarks. Our 3 metrics Instance-Level (IC), Category-Level (CC) and Ground Truth (GC) Consistency measure three aspects of canonicalization. The instance-level metric is a measure of the "variance" of the canonical pose under rotation of the input. By definition the canonical pose must be invariant to the input pose. The GC metric provides a way of measuring canonicalization consistency across the entire class of objects by measuring how our canonicalization deviates from a ground truth canonicalization up to a constant rotation. In the absence of a ground truth alignment, we propose the CC metric which compares canonicalization of different shapes within the same class using Chamfer distance (as we don't assume pointwise correspondences between different shapes). The CC metric relies on the assumption that aligned shapes of the same category are similar to each other.

We observe in table (1) of our article that some methods have high IC but low GC and vice versa (e.g. CaCa [6] (cabinet), Ours (F + P) speaker). This occurs as we canonicalize based on geometric similarity instead of semantic aspects of the object. The IC and CC metrics measure geometric

properties of the canonicalization while GC measures semantic properties of the canonicalization according to manually aligned shapes.

We build our metrics using the Chamfer distance as it does not assume pointwise correspondences between shapes, this allows measuring the canonicalization quality of symmetric shapes where there may not be a single correct canonical orientation. However, we observe a performance gap with our method when using distances based on pointwise correspondences such as L^2 or root mean square (RMSE) errors as seen in Section 3 of this appendix. We believe our Chamfer distance based metrics are representative of the quality of canonicalization and are consistent with our visual evaluation.

9. Discussion on PCA

PCA Over-Performance on the CC Metric: We note that the competitiveness of PCA is limited to certain experiments for **full shapes** and **multi-category** experiments only. The CC metric compares canonicalized shapes of the same category with possibly different geometry – note that PCA even outperforms ground truth canonicalization for this metric. Thus a method which is optimal for GC metric cannot outperform PCA in CC.

PCA Under-Performance on the IC Metric: The most likely reason why PCA underperforms on the IC metric is because of frame ambiguity. The PCA principal directions are defined up to symmetries of the covariance matrix eigenspaces – the shape does not necessarily share these symmetries. For instance, when eigenvalues are distinct, eigenvectors are defined up to sign, causing random flips over principal directions: *e.g.*, an airplane can be flipped on its back. When two or more eigenvalues are identical, eigenvectors are defined up to rotation, *e.g.*, in chairs, the major component can be from the left leg to the top right corner or bottom right leg to top left corner. Thus, PCA canonicalization of rotated copies of a given shape may not be equal due to symmetries of the shape, resulting in higher Chamfer/IC error.

10. Qualitative Results

We now present more qualitative results in Figure 4, 5 to demonstrate the effectiveness of our method.

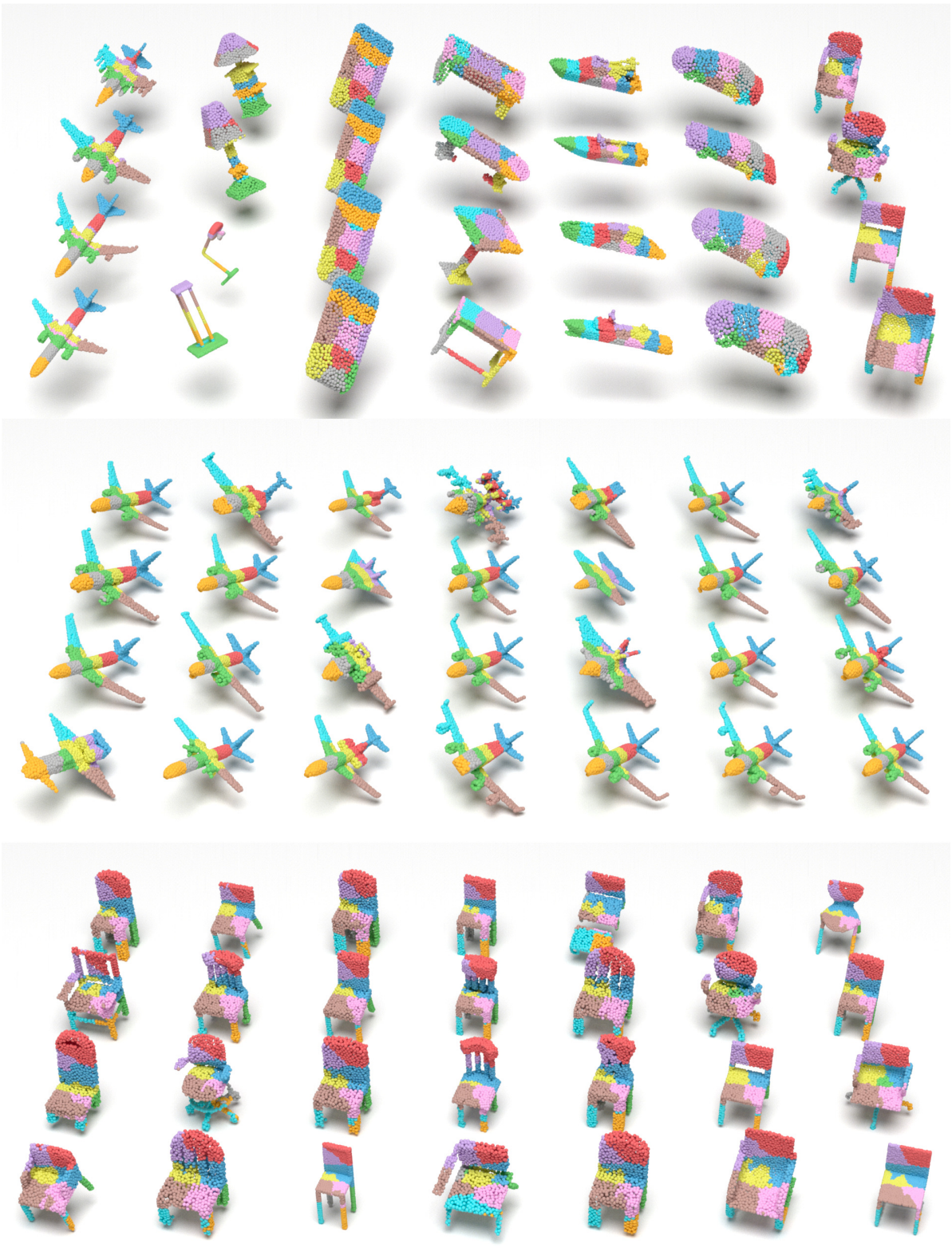

Figure 4. Parking lot for full shape canonicalization for multi-category(***top***), plane (***middle***) and chair (***bottom***).

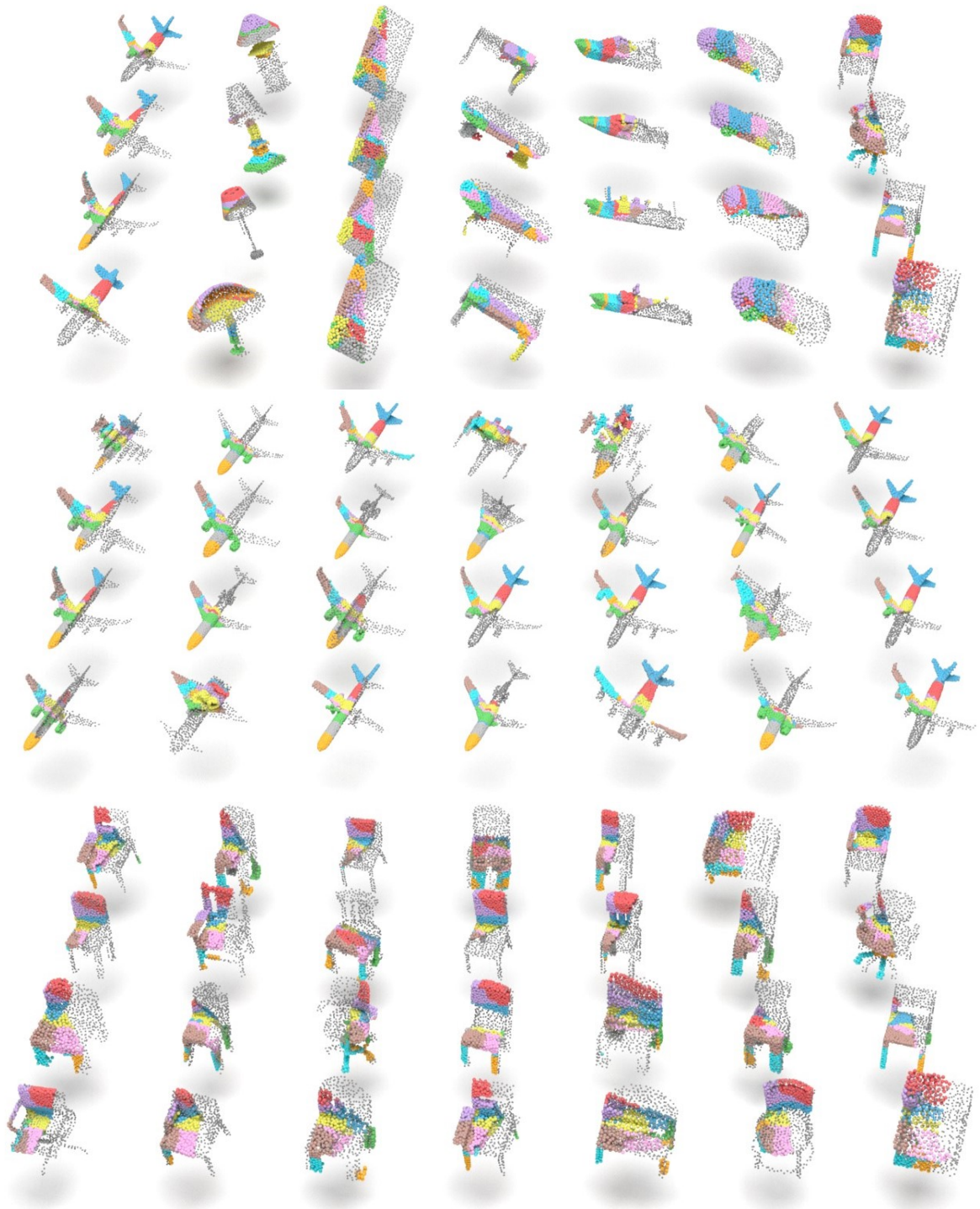

Figure 5. Parking lot for partial shape canonicalization for multi-category(***top***), plane (***middle***) and chair (***bottom***). Note: missing parts only shown for visualization.

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. *arXiv preprint arXiv:1512.03012*, 2015. 3

[2] Adrien Poulenard and Leonidas J Guibas. A functional approach to rotation equivariant non-linearities for tensor field networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13174–13183, 2021. 1

[3] Rahul Sajnani, AadilMehdi Sanchawala, Krishna Murthy Jatavallabhula, Srinath Sridhar, and K Madhava Krishna. Draco: Weakly supervised dense reconstruction and canonicalization of objects. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 10302–10309. IEEE, 2021. 1

[4] Riccardo Spezialetti, Federico Stella, Marlon Marcon, Luciano Silva, Samuele Salti, and Luigi Di Stefano. Learning to orient surfaces by self-supervised spherical cnns. *arXiv preprint arXiv:2011.03298*, 2020. 2

[5] Srinath Sridhar, Davis Rempe, Julien Valentin, Sofien Bouaziz, and Leonidas J Guibas. Multiview aggregation for learning category-specific shape reconstruction. *arXiv preprint arXiv:1907.01085*, 2019. 1

[6] Weiwei Sun, Andrea Tagliasacchi, Boyang Deng, Sara Sabour, Soroosh Yazdani, Geoffrey Hinton, and Kwang Moo Yi. Canonical capsules: Unsupervised capsules in canonical pose. *arXiv preprint arXiv:2012.04718*, 2020. 1, 2, 4

[7] Yue Wang and Justin M Solomon. Deep closest point: Learning representations for point cloud registration. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3523–3532, 2019. 2

[8] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proc. CVPR*, pages 1912–1920, 2015. 3

[9] Wentao Yuan, Benjamin Eckart, Kihwan Kim, Varun Jampani, Dieter Fox, and Jan Kautz. Deepgmr: Learning latent gaussian mixture models for registration. In *European Conference on Computer Vision*, pages 733–750. Springer, 2020. 2