Supplementary Materials for Revisiting Point Cloud Classification: A New Benchmark Dataset and Classification Model on Real-World Data

Mikaela Angelina Uy[1] Quang-Hieu Pham[2] Binh-Son Hua[3] Duc Thanh Nguyen[4] Sai-Kit Yeung[1]

[1]Hong Kong University of Science and Technology [2]Singapore University of Technology and Design
[3]The University of Tokyo [4]Deakin University

Abstract

In this document, we first detail experiments in generalizing between CAD models and real world scans, i.e., training on ModelNet40 and testing on our dataset, and vice versa. A class mapping between ModelNet40 and our dataset is also provided for this experiment. Second, experiments with the use of background points and other train/test splits are also shown with per-class accuracies plot for the tested methods. We also justify the design choice in our Background-Aware Classification Network in the main paper and compare it with a naive approach. We also discuss more about our data collection. Finally, we provide renderings that show representative objects of the hardest variant of our dataset in color, greyscale, and part annotated point clouds.

1. Generalization between CADs and Scans

We explore the training on CADs and testing on scans and vice versa in order to understand how well the generalization is. Table 1 shows the 11-class mapping between ModelNet40 and our dataset which we use for the experiments. Some classes are named differently while some others have a many-to-one mapping on our data and ModelNet40.

	ObjectNet classes	ModelNet40 classes
1	cabinet	dresser, wardrobe
2	chair	bench, chair, stool
3	desk	desk
4	display	monitor
5	door	door
6	shelf	bookshelf
7	table	table
8	bed	bed
9	sink	sink
10	sofa	sofa
11	toilet	toilet

Table 1. 11 common classes between ObjectNet and ModelNet40.

	OBJ_BG	PB_T25	PB_T25_R	PB_T50_R	PB_T50_RS
3DmFV [1]	24	19.9	17.1	15	16.4
PointNet [3]	41.1	30.1	26.6	20.9	23.2
SpiderCNN [6]	42.1	26.8	23.2	20.2	22.2
PointNet++ [4]	37.7	28.2	25.8	21.9	22.9
DGCNN [5]	46.7	33.3	29.9	25.5	27.2
PointCNN [2]	29.5	21.9	20.6	18.3	19.2

Table 2. Train on CADs, test on scans: Overall accuracy in % on our dataset when training was done on ModelNet40. Background points are present in this additional experiment unlike the table in the main paper. It can be seen that the generalization of training on CAD models is even worse when tested on objects in context, where background points are present.

	w/o BG			w/ BG		
	Comb.	Scan	CAD	Comb.	Scan	CAD
3DmFV [1]	79.6	73.2	95.1	76.2	68.4	95.1
PointNet [3]	82.6	78.5	92.5	78.1	72.4	91.9
SpiderCNN [6]	83.5	79.8	92.4	82.3	78.1	92.5
PointNet++ [4]	85.7	82.7	92.9	85.7	82.9	92.4
DGCNN [5]	87	84	94.3	85.8	83	92.8
PointCNN [2]	86.3	83.1	94	85.5	81.8	94.8

Table 3. Train on combined data, test on both CADs and scans: Overall accuracy in % when trained on the two combined dataset consisting on CAD and real data. First header row specifies whether samples from PB_T50_RS contained background, while second header row specifies the test set on the corresponding reported performances. A gap between CAD and scan performance is evident.

1.1. Training on CADs

We show the per class accuracies when training all methods on ModelNet40 and testing on our various dataset variants. Results are shown in Figures 1, 2, 3, 4 and 5, where background points are removed in all the variants. Additionally, we also test on the various dataset variants with the presence of background. This experiment illustrates the

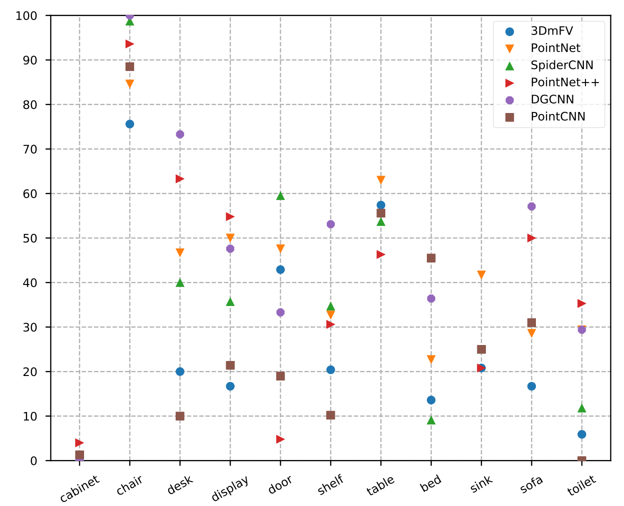

Figure 1. Per class accuracy results on OBJ_ONLY when trained on ModelNet40.

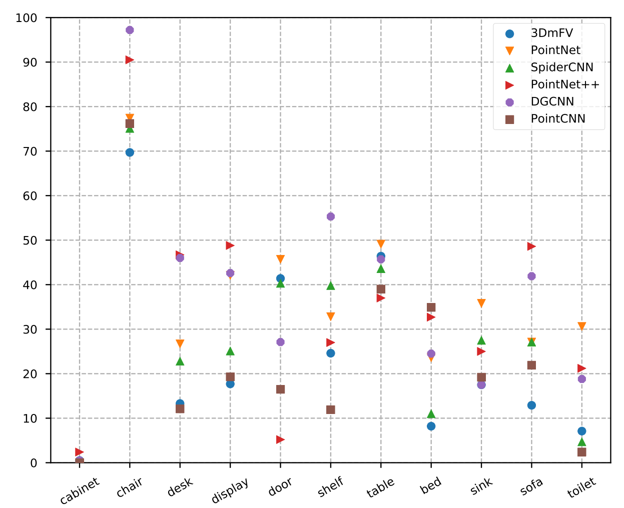

Figure 2. Per class accuracy results on PB_T25 with no background when trained on ModelNet40.

performance when tested on real settings where objects appear in context. Results are shown in Table 2, and it can be seen that the generalization is even worse when background points are not removed from object scans.

1.2. Training on Scans

Figures 6 and 7 show per class accuracy results when trained on our hardest variant PB_T50_RS, with and without background points.

1.3. Combined Training (CADs + Scans)

We also train on two combined datasets consisting of the training samples from ModelNet40 and our hardest variant PB_T50_RS with and without background points. We only

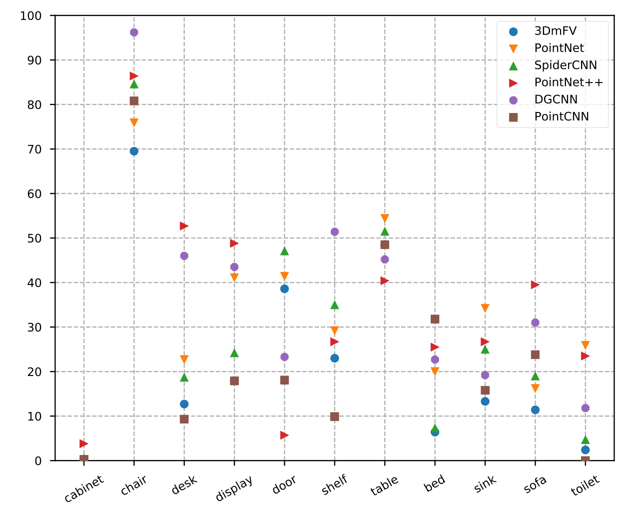

Figure 3. Per class accuracy results on PB_T25_R with no background when trained on ModelNet40.

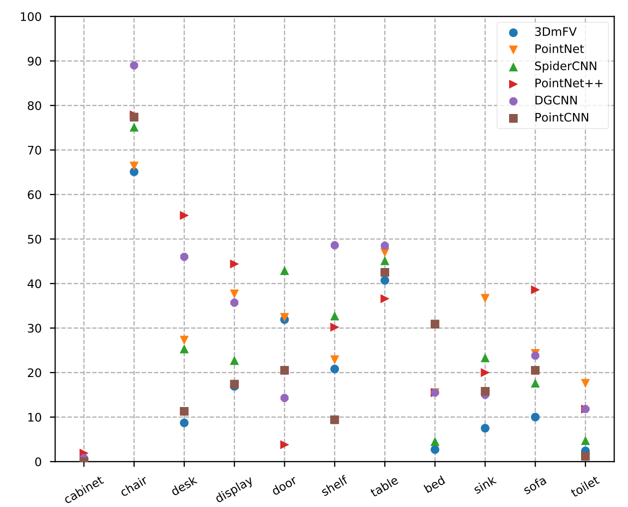

Figure 4. Per class accuracy results on PB_T50_R with no background when trained on ModelNet40.

consider the samples from the 11 common classes between the two datasets. We then evaluate the different methods performances on the combined dataset and separately on each of the CAD and real test sets. Results are shown in Table 3. It can be seen that even when training on both CAD and real data, there is a significant performance gap when testing on CAD and real data.

2. Per-class Accuracies

We provide a more detailed breakdown of the performances of the different methods on our dataset variants on the hardest data split as shown in Table 4 of our main paper. Figures 8, 9, 10, 11, 12 and 13 show the per-class accuracies

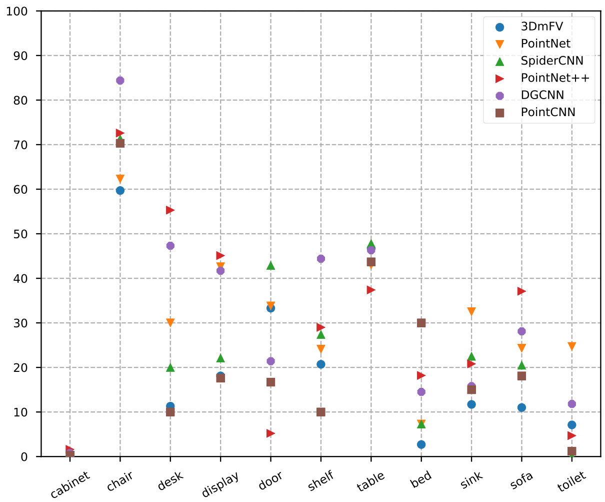

Figure 5. Per class accuracy results on PB_T50_RS with no background when trained on ModelNet40.

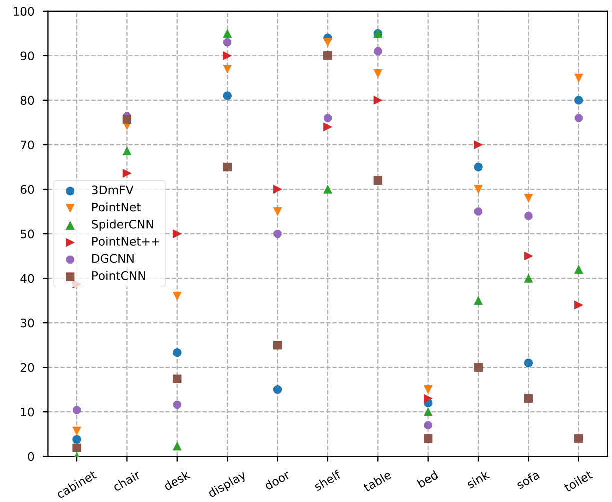

Figure 7. Per class accuracy results on ModelNet when trained on PB_T50_RS with no background.

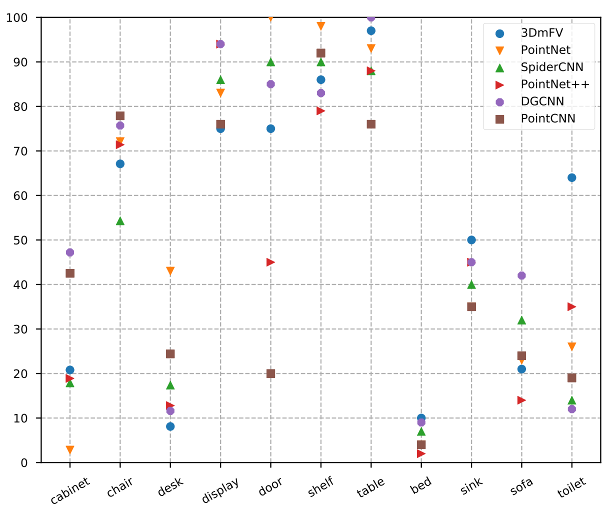

Figure 6. Per class accuracy results on ModelNet when trained on PB_T50_RS with background.

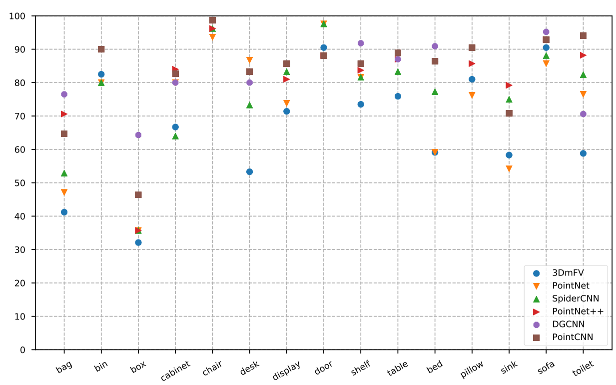

Figure 8. Per class accuracy results on OBJ_ONLY.

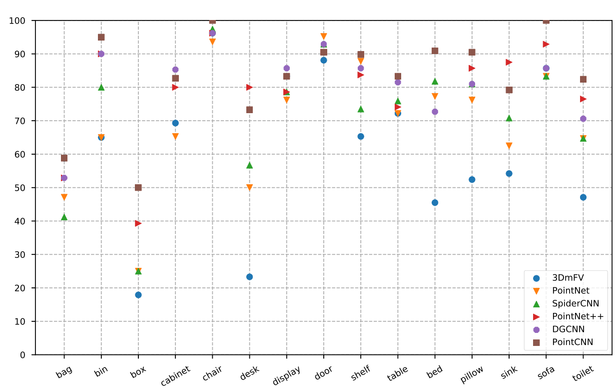

Figure 9. Per class accuracy results on OBJ_BG.

of the different methods on our various dataset variants.

3. Detailed Quantitative Results

Here we evaluate the influence of background points, a unique factor that differentiates real-world scans from CADs, to the classification performance. We also show the performance on some other train/test splits in our dataset.

3.1. Background Points

We additionally ran experiments on all our dataset variants *with* and *without* background points on our hardest data split. Table 4 shows that the presence of background points introduces noise to the learning of the networks. It is also observed that as the perturbations become more aggressive, the bigger the difference between the accuracies of the cases *with* and *without* background points become. This is consistent with the goal of making harder perturbations of objects

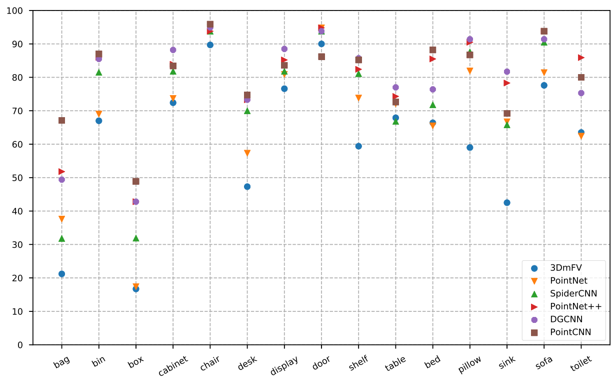

Figure 10. Per class accuracy results on PB_T25.

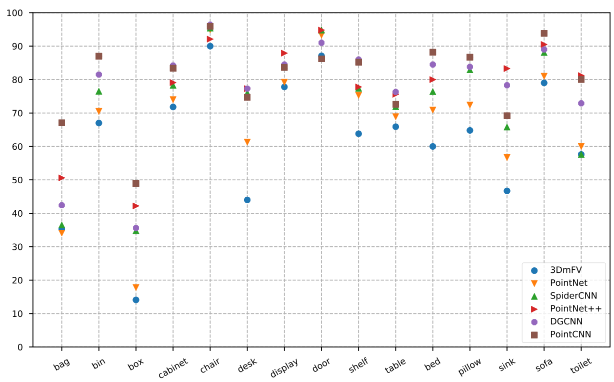

Figure 11. Per class accuracy results on PB_T25_R.

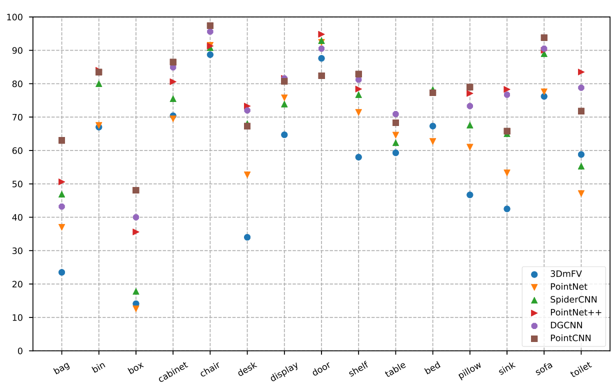

Figure 12. Per class accuracy results on PB_T50_R.

in scene contexts.

3.2. Train/Test Splits

We include results from four other training/test splits on our ObjectNet dataset as shown in Tables 6, 7, 8 and 9. Moreover, we also include the results of these additional splits for the background points study as shown in Table 10, and this again shows that in our hardest variant, background points introduce noise to the network learning as depicted by

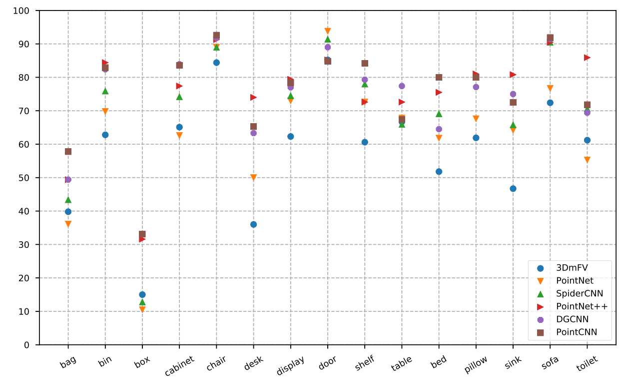

Figure 13. Per class accuracy results on PB_T50_RS.

the lower performances of the w/ BG cases in the different splits.

4. Two-branch Network on Real-world Data

We implement and run a two branch network of PointNet[3], PointNet++[4] and PointCNN[2] to test whether jointly segmenting and classifying improves the classification performance on real-world point cloud scans. Table 5 show that learning with a two-branch network improves the classification performance on all our perturbation variants. This experiment also shows that our Background-aware Classification Network achieves state-of-the-art performance, and the vector concatenation aids in network learning compared to a standard two-branch segmentation and classification network.

5. Data Collection

Perturbations are our keys to scale up our data and at the same time, make our scans closer to those found in real-world applications. Here we discuss the detailed steps on how the perturbations were done.

Translation Perturbation. We randomly translated bounding boxes' centers. This perturbation is to produce partial objects (as translated boxes may split objects) and introduce cases where objects are not perfectly centered. Specifically, we define the translation as a random percentage of the bounding box dimensions, mathematically as $\text{rand}([1 - x\%, 1 + x\%]) \cdot size$ where x is the maximum translation percentage and $size$ the bounding box dimensions. Translation perturbation is denoted by suffix *_$\mathbf{T}x$.

Rotation Perturbation. We also randomly rotated bounding boxes to generate different horizontal orientations of objects together with more background variety. Here, we limit ourselves to rotation about the gravity axis as so far most of the object classification techniques for point cloud are not designed to be rotation invariant. In other words, we

	PB_T25		PB_T25_R		PB_T50_R		PB_T50_RS	
	w/o BG	w/ BG	w/o BG	w/ BG	w/o BG	w/ BG	w/o BG	w/ BG
PointNet [3]	78.4	73.5	77.7	72.7	76	68.2	74.4	68.2
PointNet++ [4]	83.5	82.7	81.8	81.4	80.5	79.1	80.2	77.9
PointCNN [2]	83.9	83.6	82.9	82.5	81.3	78.5	80.8	78.5

Table 4. Overall accuracy in % when training on our different perturbation variants with and without background (BG) points. These consistently show that performances are higher in our perturbed dataset variants without the presence of background.

	PB_T25		PB_T25_R		PB_T50_R		PB_T50_RS	
	Vanilla	Two-branch	Vanilla	Two-branch	Vanilla	Two-branch	Vanilla	Two-branch
PointNet [3]	73.5	74.7	72.7	74.5	68.2	68.7	68.2	67.7
PointNet++ [4]	82.7	84.9	81.4	83.1	79.1	79.2	77.9	79.4
PointCNN [2]	83.6	83.8	82.5	82.4	78.5	80.8	78.5	79.2

Table 5. Two-branch vs. vanilla: overall accuracy in % when training on our different perturbation variants with and without background (BG) points. These results show that a two-branch network outperforms the vanilla classification architecture when classifying real-world data. However, our proposed Background-aware Classification Network in the main paper still achieves a higher accuracy of 80.2% on the hardest variant PB_T50_RS.

assume that the floor plane can be reliably determined before an object is classified. We denote the rotation perturbation with suffix *_**R**.

Scale Perturbation. We scaled bounding boxes proportionally to their original size in each axis dimension. Scaling may make objects incomplete and add more background. This perturbation is inspired by inaccurate region proposals in object detection pipelines. In particular, the new size of a bounding box is given by $\mathrm{rand}([1-y\%, 1+z\%]) \cdot size$. We applied scale perturbation in all variants with $z = 25\%$. The suffix ***S** denotes applying an additional "shrinking" scale perturbation with $y = 25\%$.

6. Object Visualizations

6.1. The Hardest Variant PB_T50_RS

Here we visualize objects in our hardest perturbation variant, PB_T50_RS. The objects are grouped based on the 15 categories in the dataset. The visualization start from page 7.

6.2. Object Parts

Here we visualize example object parts in our dataset. The objects are selected and grouped from the 15 categories in the dataset. To the best of our knowledge, this is the first dataset with part annotations on real-world data. The visualization is from page 22.

References

[1] Y. Ben-Shabat, M. Lindenbaum, and A. Fischer. 3dmfv: Three-dimensional point cloud classification in real-time using convolutional neural networks. *IEEE Robotics and Automation Letters*, 2018. 1, 6

[2] Y. Li, R. Bu, M. Sun, and B. Chen. Pointcnn: Convolution on x-transformed points. *Advances in Neural Information Processing Systems*, 2018. 1, 4, 5, 6

[3] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. *CVPR*, 2017. 1, 4, 5, 6

[4] C. R. Qi, L. Yi, H. Su, and L. J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *NIPS*, 2017. 1, 4, 5, 6

[5] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon. Dynamic graph cnn for learning on point clouds. *arXiv preprint arXiv:1801.07829*, 2018. 1, 6

[6] Y. Xu, T. Fan, M. Xu, L. Zeng, and Y. Qiao. Spidercnn: Deep learning on point sets with parameterized convolutional filters. In *ECCV*, 2018. 1

	OBJ_ONLY	OBJ_BG	PB_T25	PB_T25_R	PB_T50_R	PB_T50_RS
3DmFV [1]	77.7	76.8	74.1	71.4	67.6	67.4
PointNet [3]	79.8	79.4	80.4	76.7	72.5	71.4
PointNet++ [4]	85.5	87.8	87.4	84.6	82.6	82
DGCNN [5]	86.2	87.3	86.5	84.4	81	81.7
PointCNN [2]	86.3	89.9	87.2	84.4	83.6	82.5

Table 6. Overall accuracy in % on our real dataset for data split 1.

	OBJ_ONLY	OBJ_BG	PB_T25	PB_T25_R	PB_T50_R	PB_T50_RS
3DmFV [1]	76.3	71.6	69.7	67.4	63.5	64.3
PointNet [3]	79	75.9	74.5	73.2	69.3	67.8
PointNet++ [4]	83.5	83.8	85.4	82.8	80.9	78.7
DGCNN [5]	85.3	86.9	85.7	83.8	80.9	81
PointCNN [2]	88.3	89.9	86.4	85.1	82.4	81.7

Table 7. Overall accuracy in % on our real dataset for data split 2.

	OBJ_ONLY	OBJ_BG	PB_T25	PB_T25_R	PB_T50_R	PB_T50_RS
3DmFV [1]	70.5	69.8	70.3	67.4	64.7	64.9
PointNet [3]	77.3	75	77.6	75	69.6	68.1
PointNet++ [4]	81.2	83.6	85	84.7	82.5	81
DGCNN [5]	83.4	84	84.5	83.7	80.8	80.2
PointCNN [2]	85.3	88.4	85.8	83.5	82.2	80.8

Table 8. Overall accuracy in % on our real dataset for data split 3.

	OBJ_ONLY	OBJ_BG	PB_T25	PB_T25_R	PB_T50_R	PB_T50_RS
3DmFV [1]	75.9	73.3	70.7	68	65.4	65.2
PointNet [3]	80.9	76.6	76	73.4	68.7	68.7
PointNet++ [4]	85.8	84.9	85.6	83.9	80.1	80.6
DGCNN [5]	87.2	86.8	86.5	83.7	80.3	80.8
PointCNN [2]	87.2	88.6	86.3	85.3	81.5	82.7

Table 9. Overall accuracy in % on our real dataset for data split 4.

	Split 1		Split 2		Split 3		Split 4	
	w/o BG	w/ BG	w/o BG	w/ BG	w/o BG	w/ BG	w/o BG	w/ BG
3DmFV [1]	73.6	67.4	74.2	64.3	70.3	64.9	73.6	65.2
PointNet [3]	78.4	71.4	77.3	67.8	74.6	68.1	76.3	68.7
PointNet++ [4]	82	82	80.9	78.7	80.5	81	83.3	80.6
DGCNN [5]	83.1	81.7	83.7	81	81.2	80.2	84.4	80.8
PointCNN [2]	83.7	82.5	83.4	81.7	81.2	80.8	83.7	82.7

Table 10. Overall accuracy in % when training and testing on our hardest variant PB_T50_RS, with and without background (BG) points using the other train/test splits.

Figure 14. Bag

Figure 15. Bed

Figure 16. Bin

Figure 17. Box

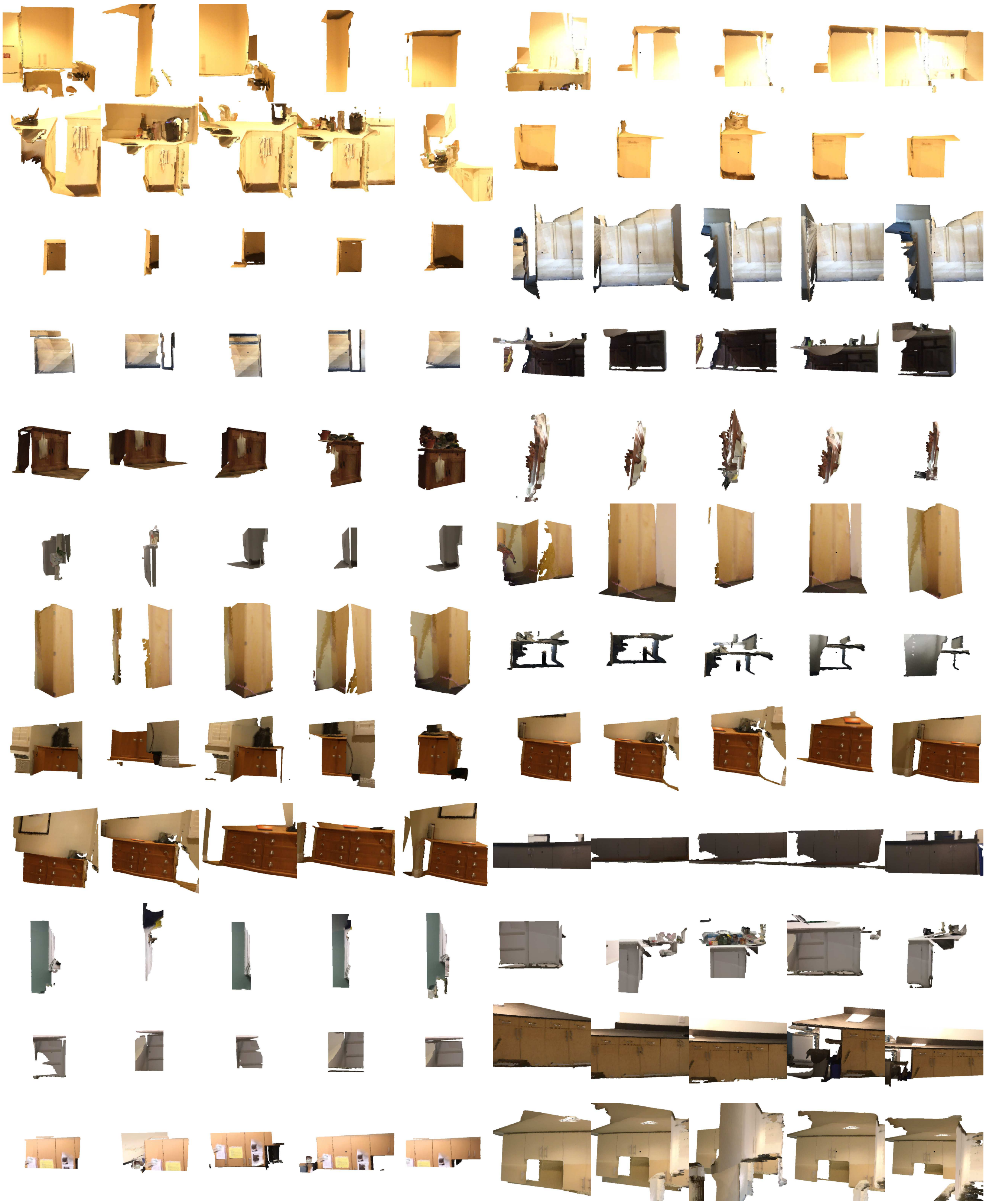

Figure 18. Cabinet

Figure 19. Chair

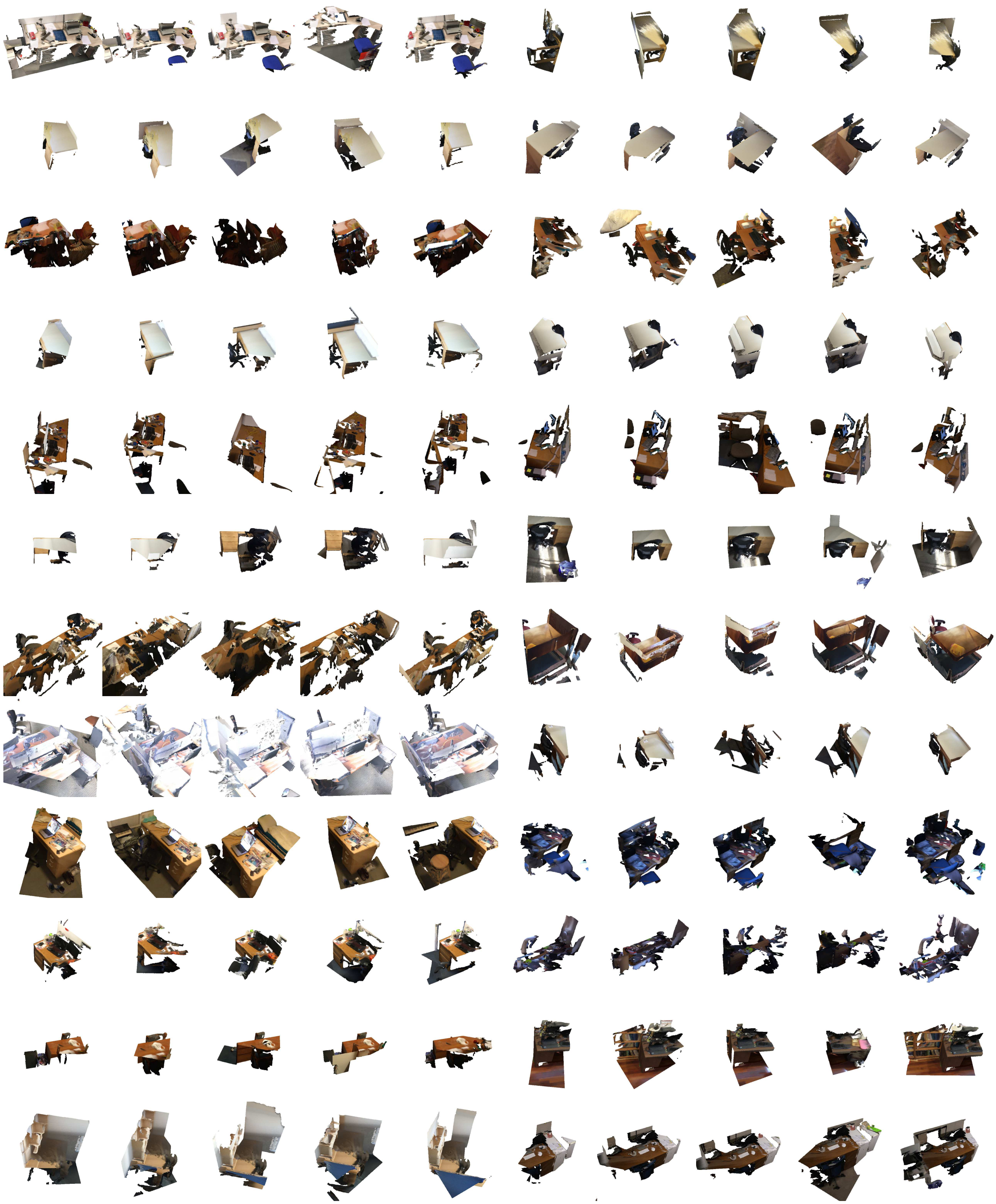

Figure 20. Desk

Figure 21. Display

Figure 22. Door

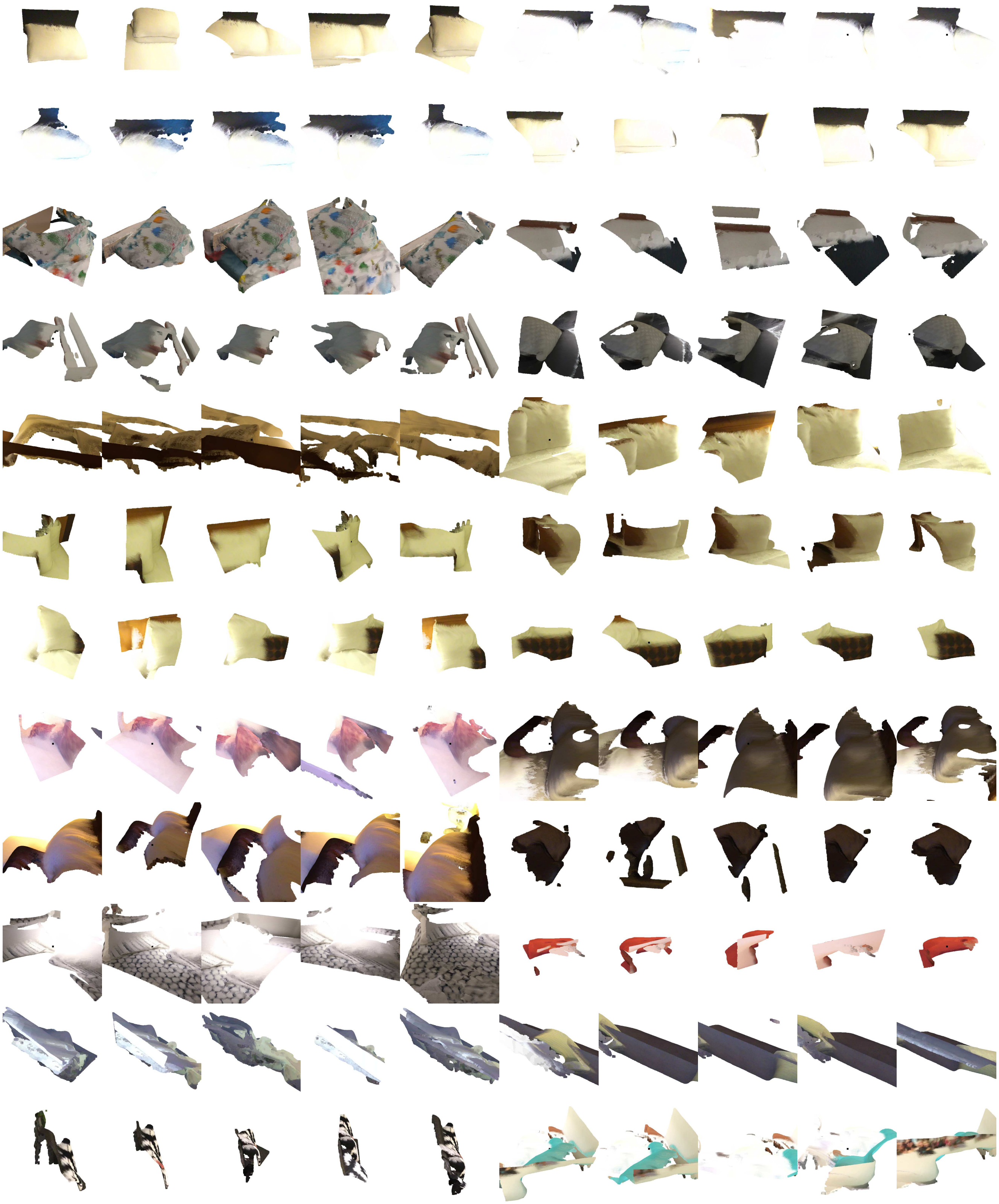

Figure 23. Pillow

Figure 24. Shelf

Figure 25. Sink

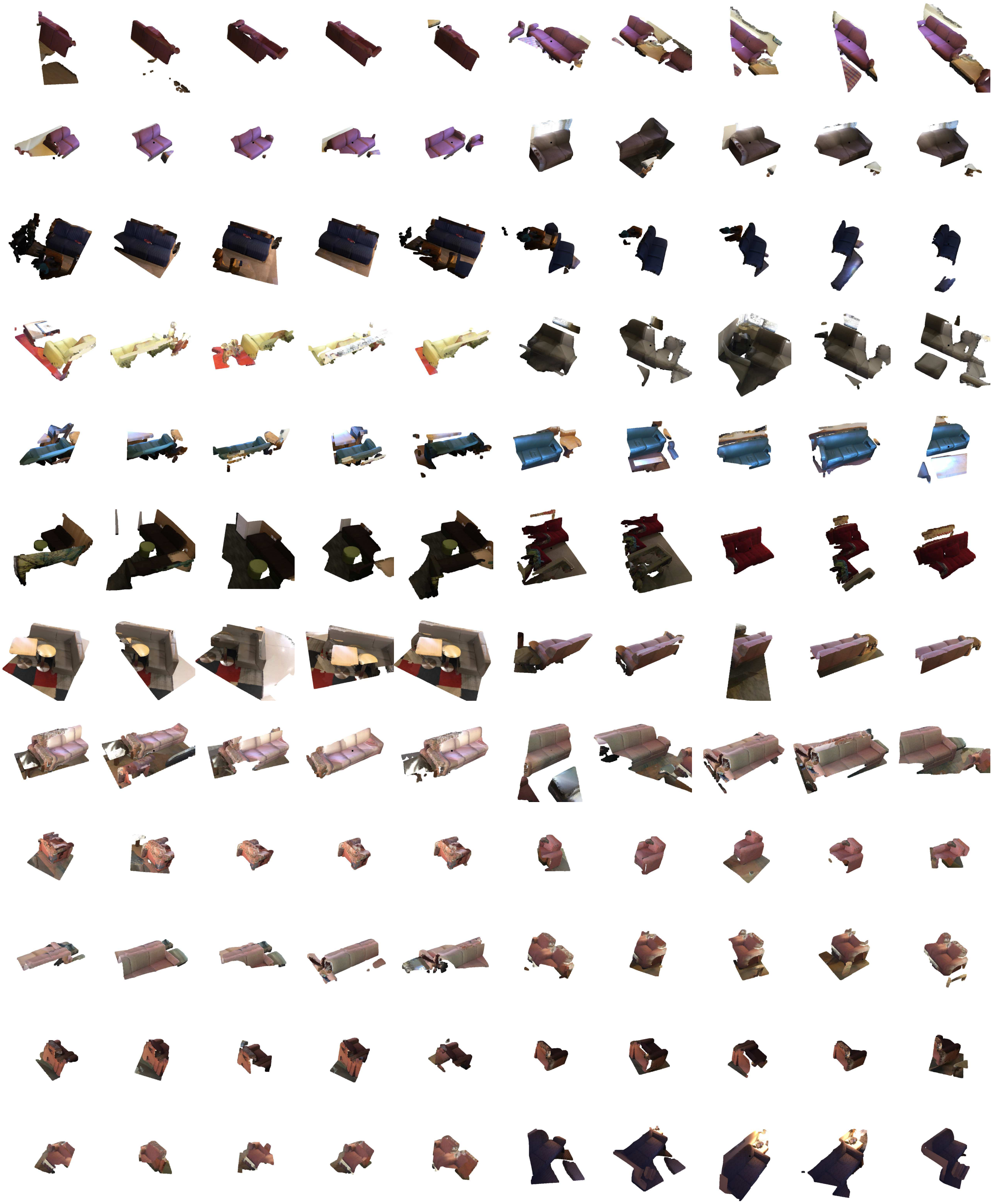

Figure 26. Sofa

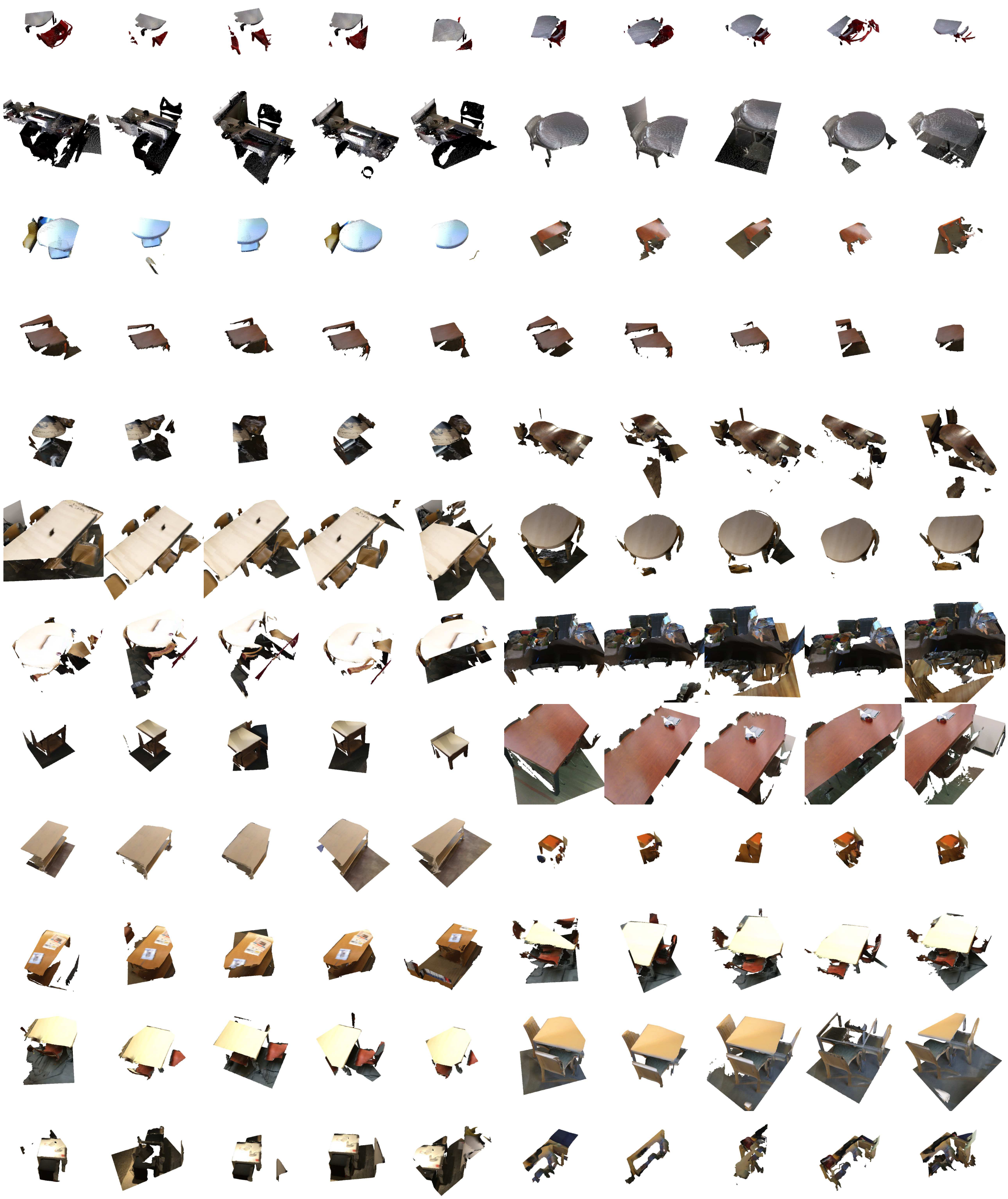

Figure 27. Table

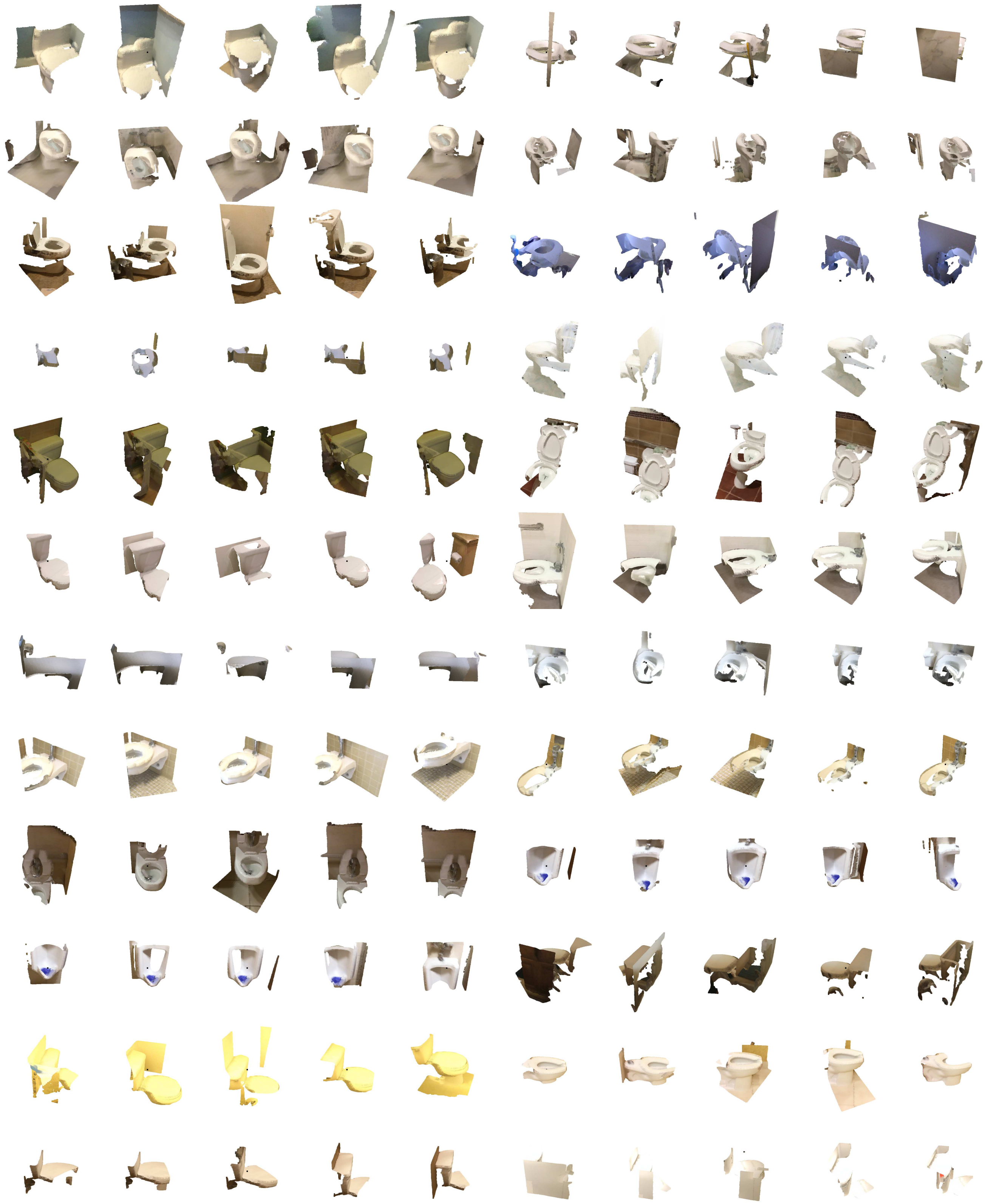

Figure 28. Toilet

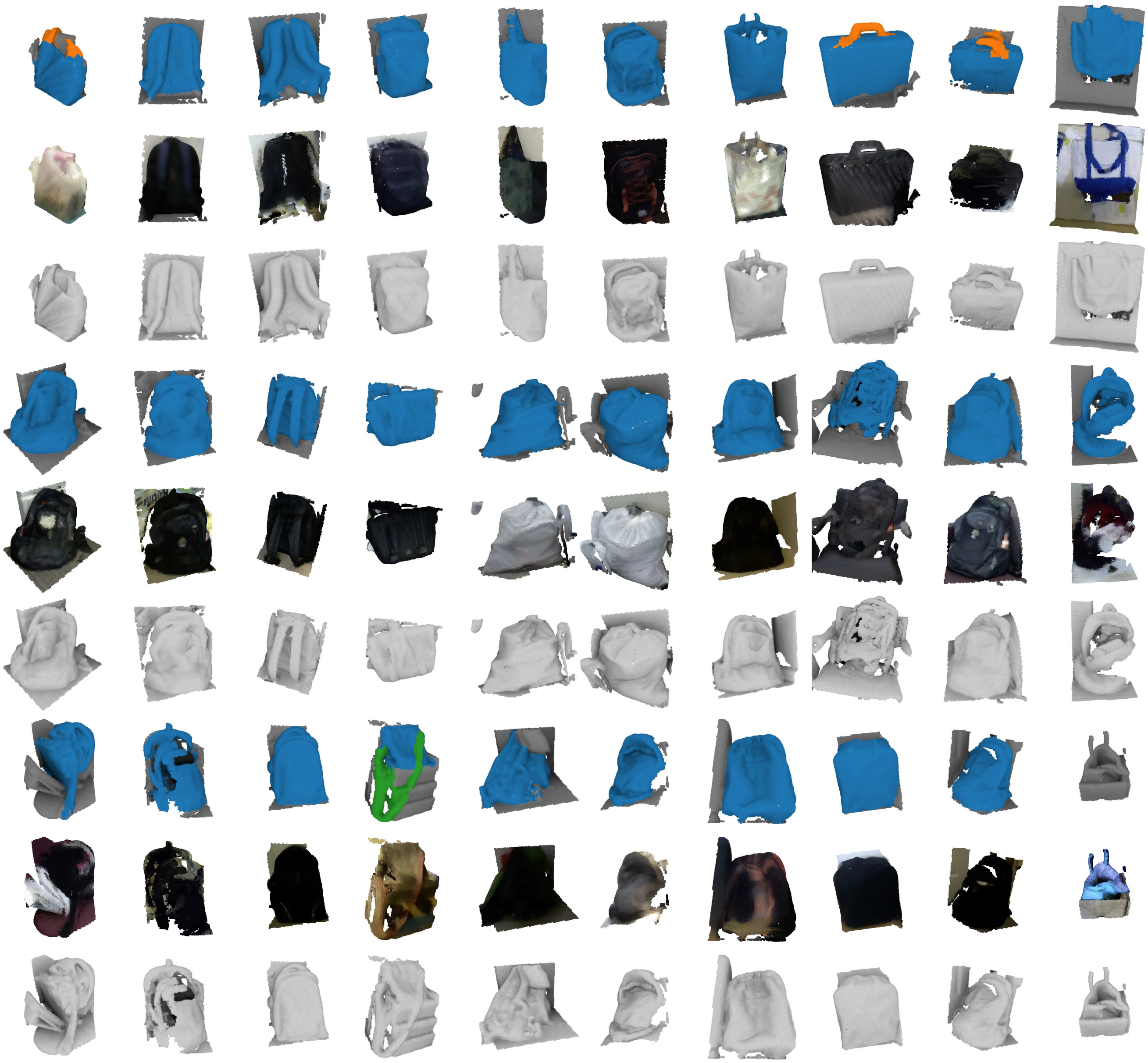

Figure 29. Part annotation: Bag

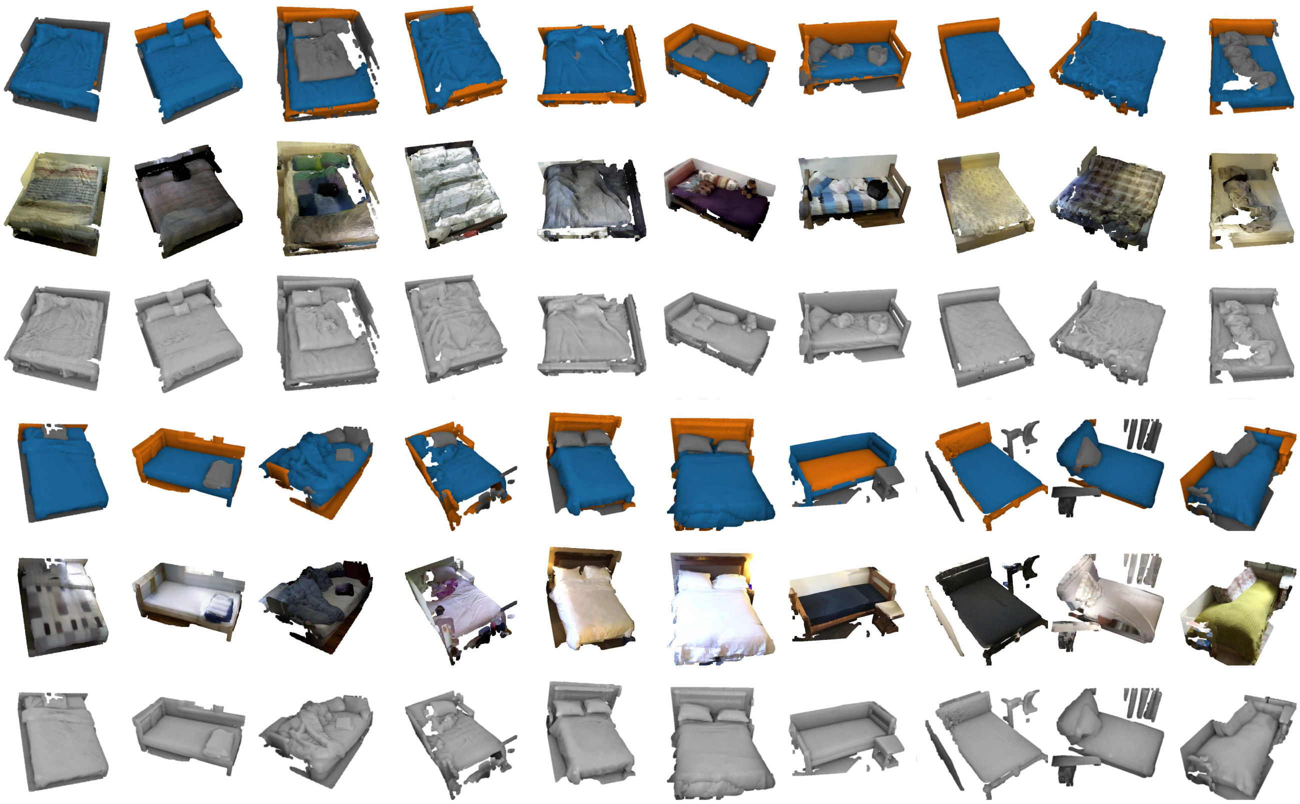
Figure 30. Part annotation: Bed

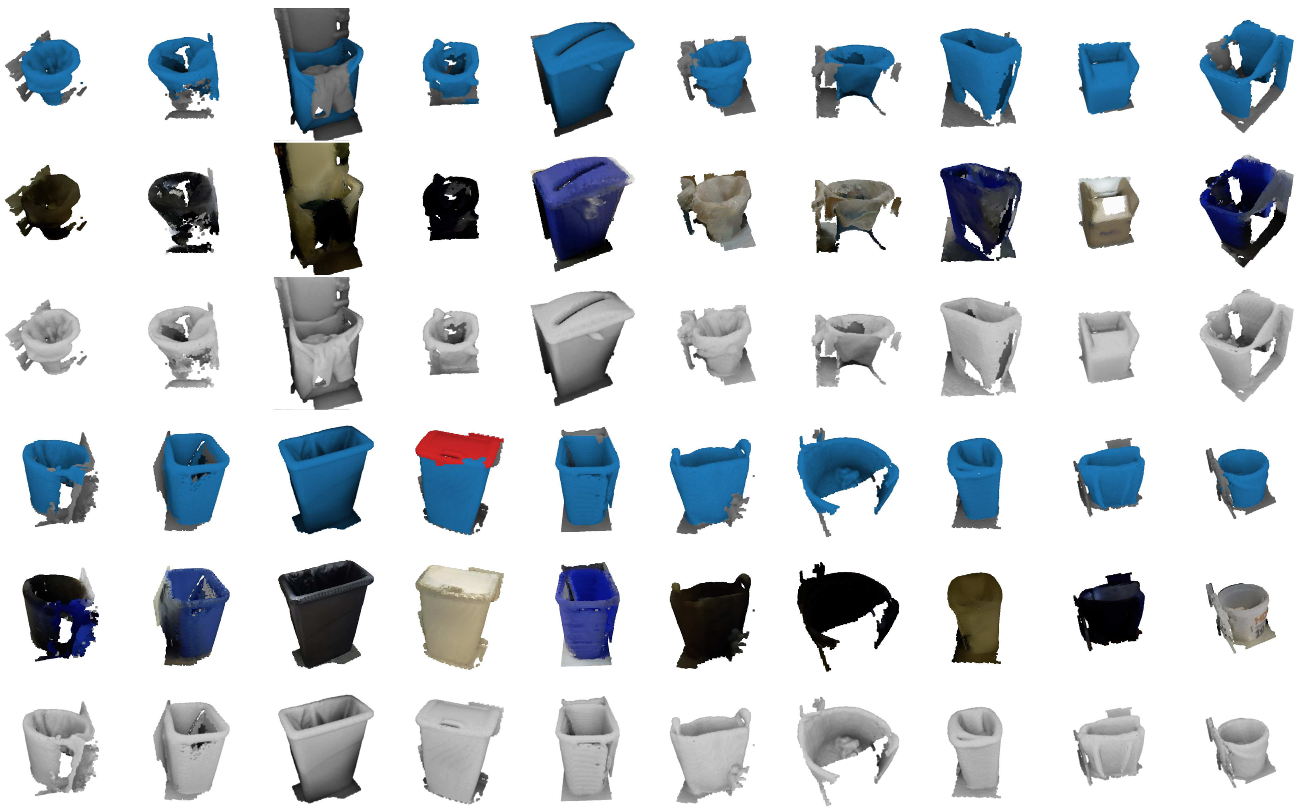

Figure 31. Part annotation: Bin

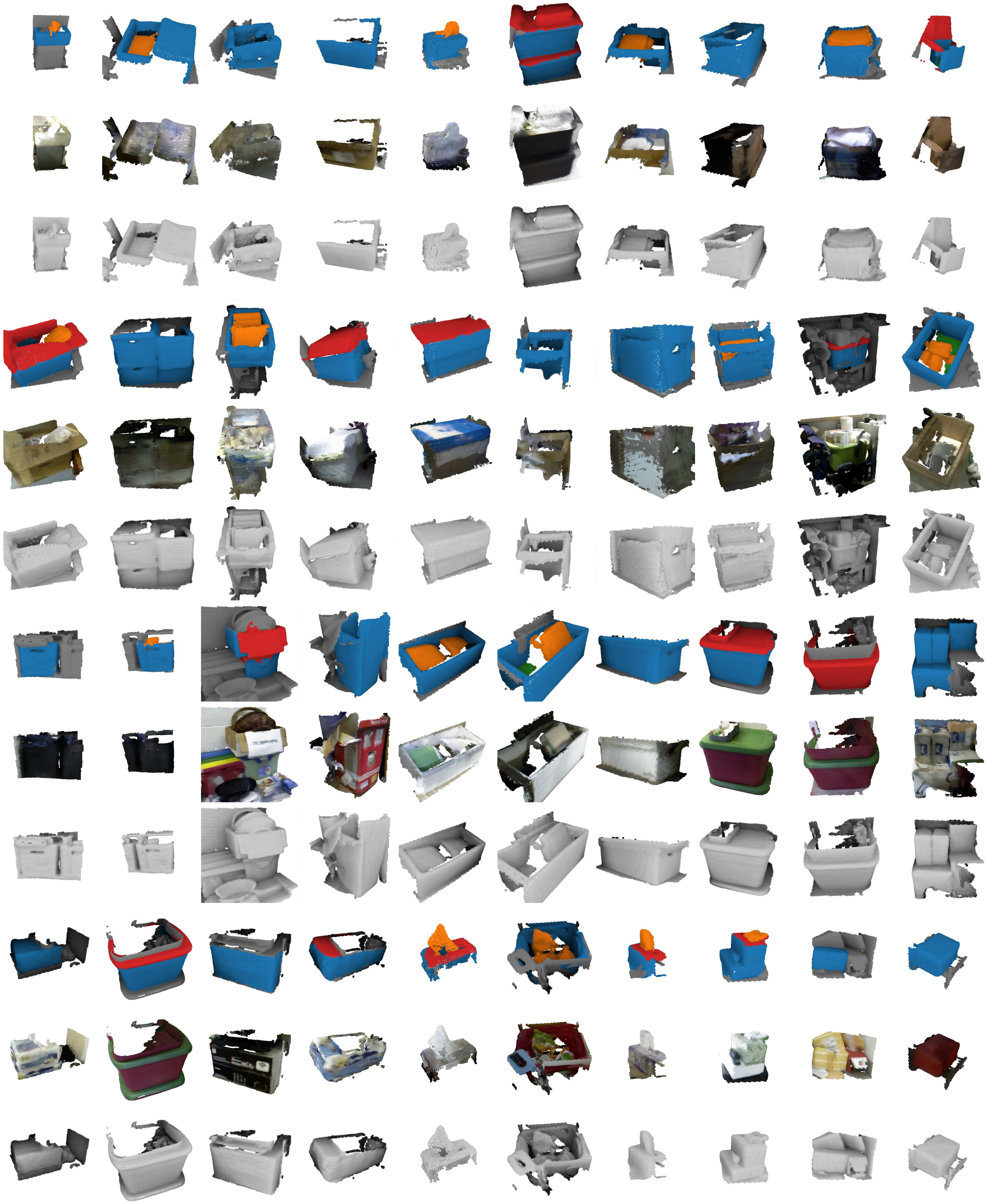

Figure 32. Part annotation: Box

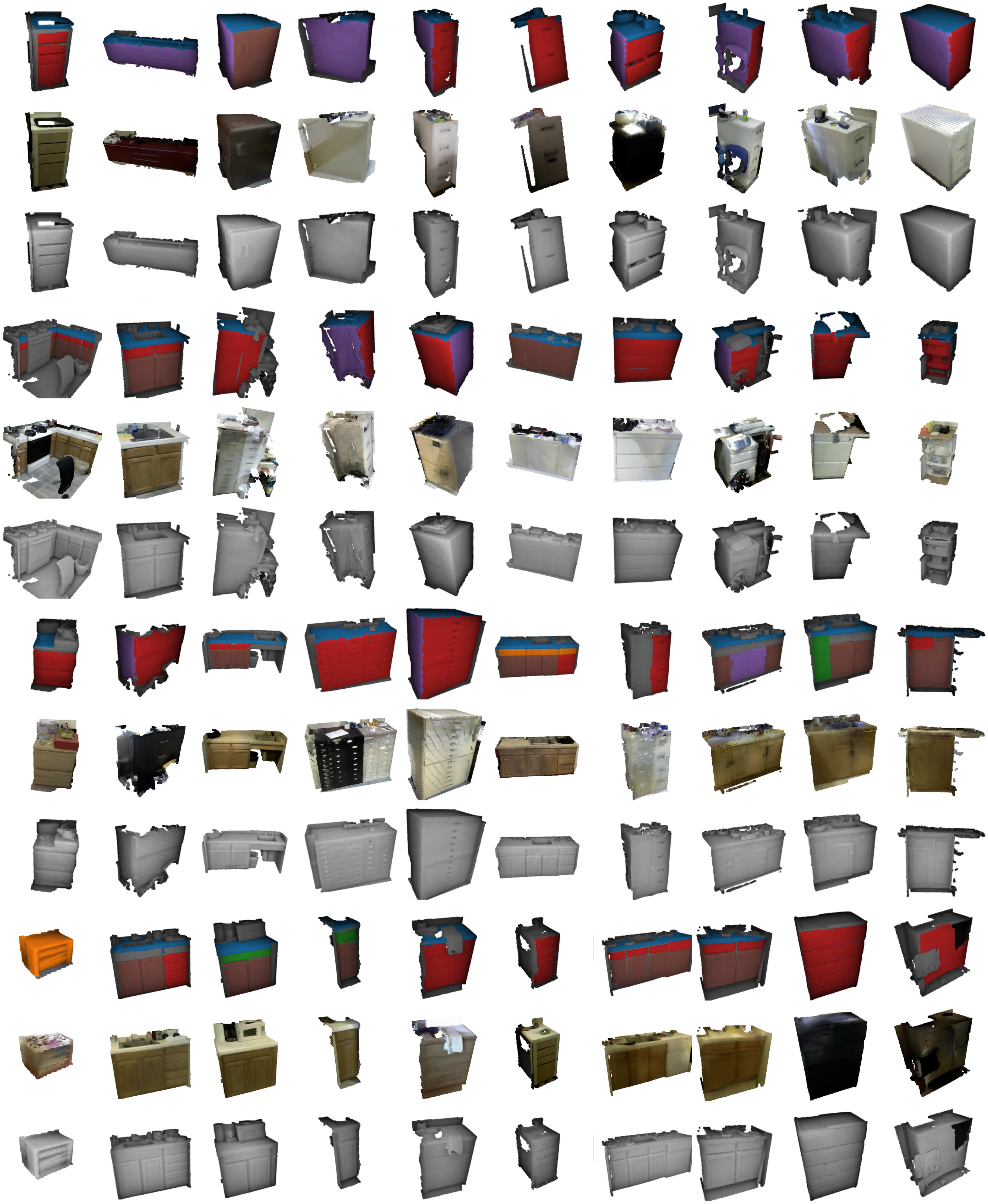

Figure 33. Part annotation: Cabinet

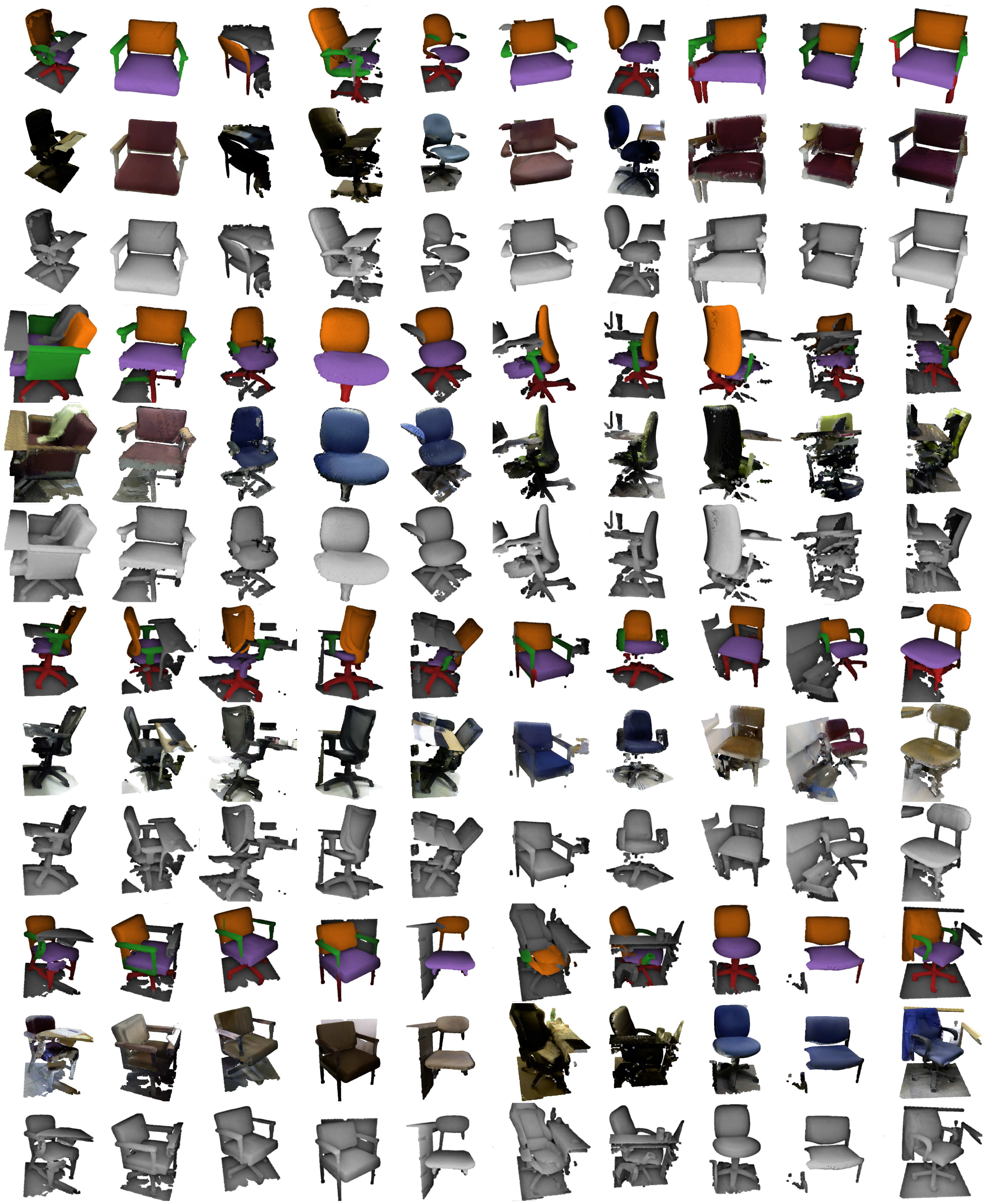

Figure 34. Part annotation: Chair

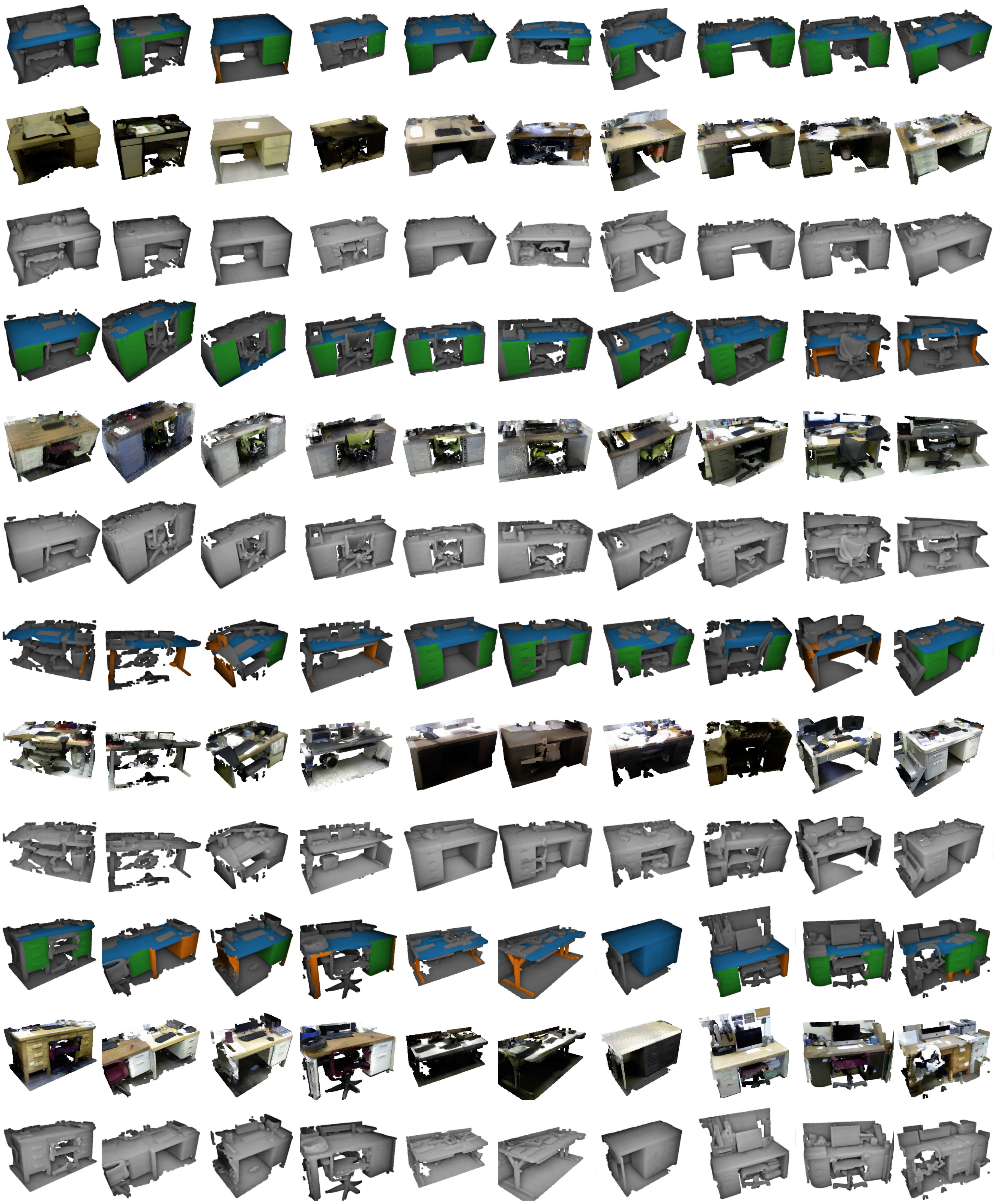

Figure 35. Part annotation: Desk

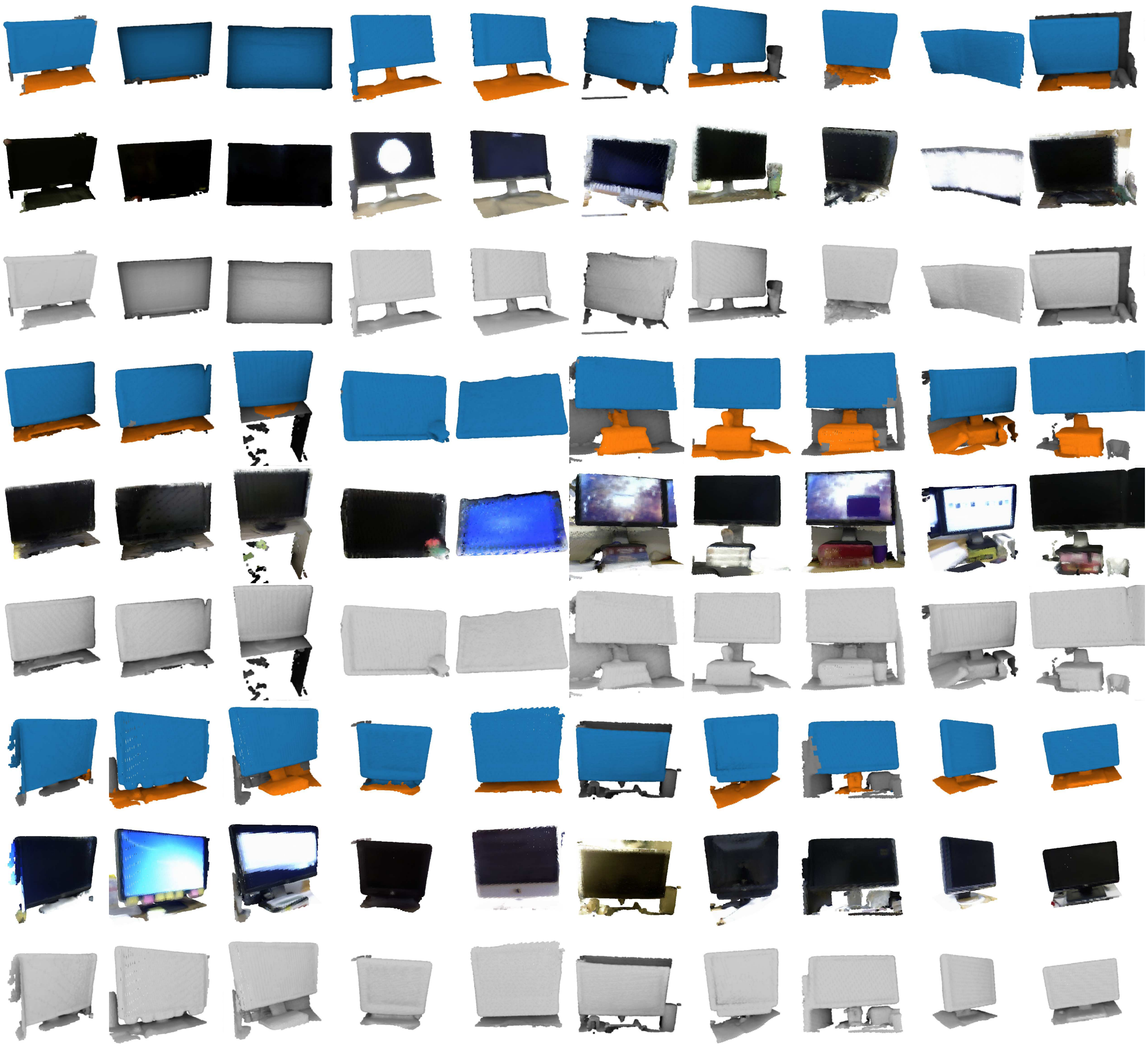

Figure 36. Part annotation: Display

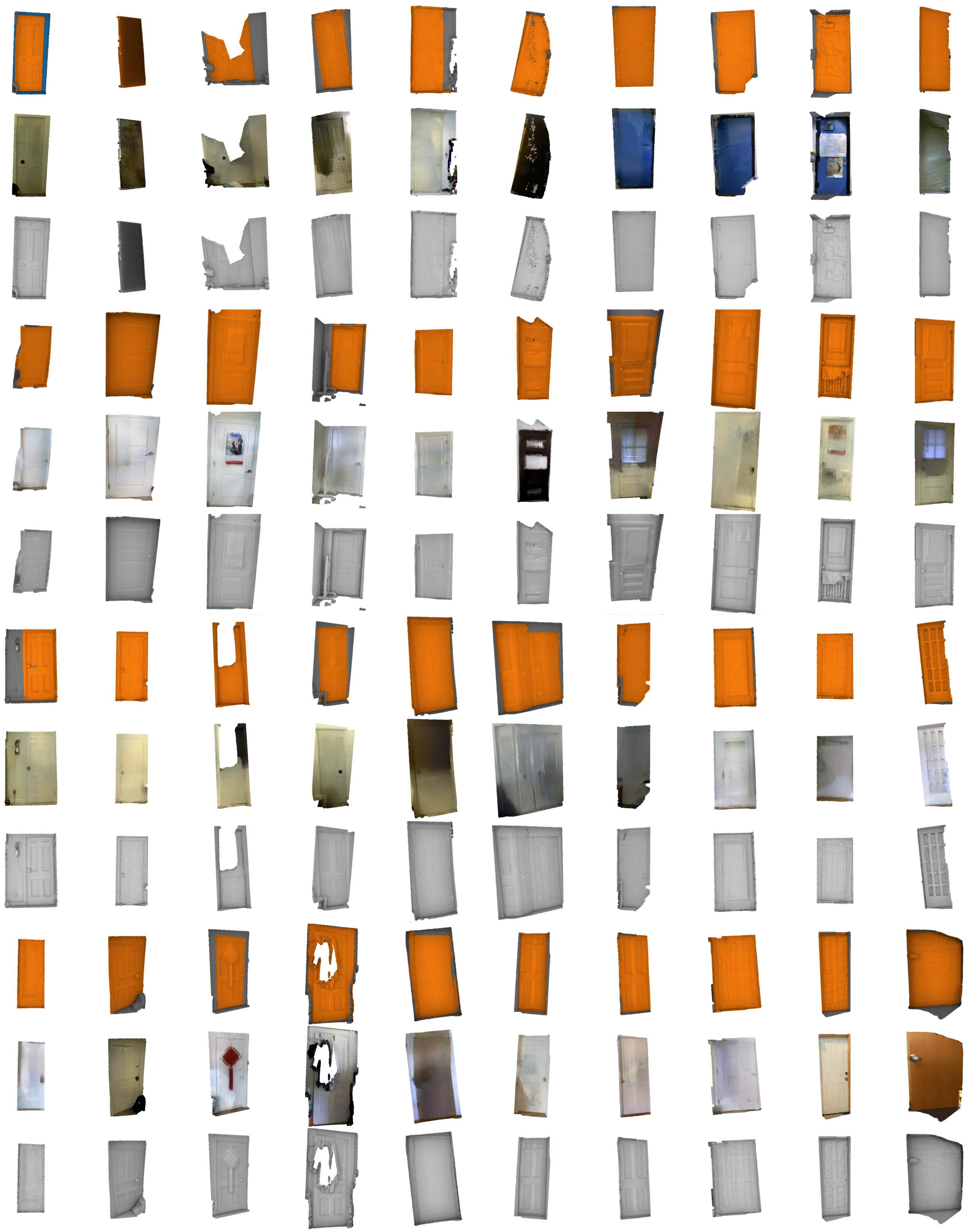

Figure 37. Part annotation: Door

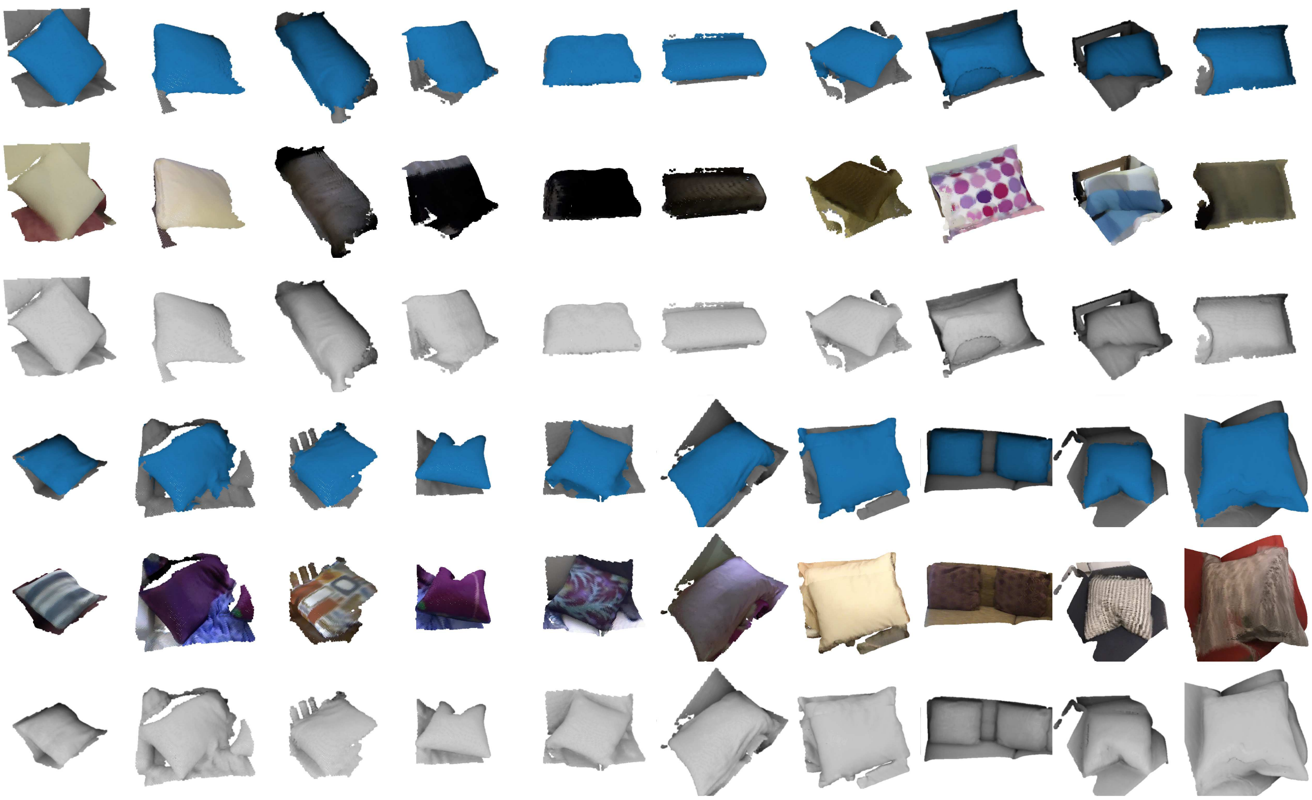

Figure 38. Part annotation: Pillow

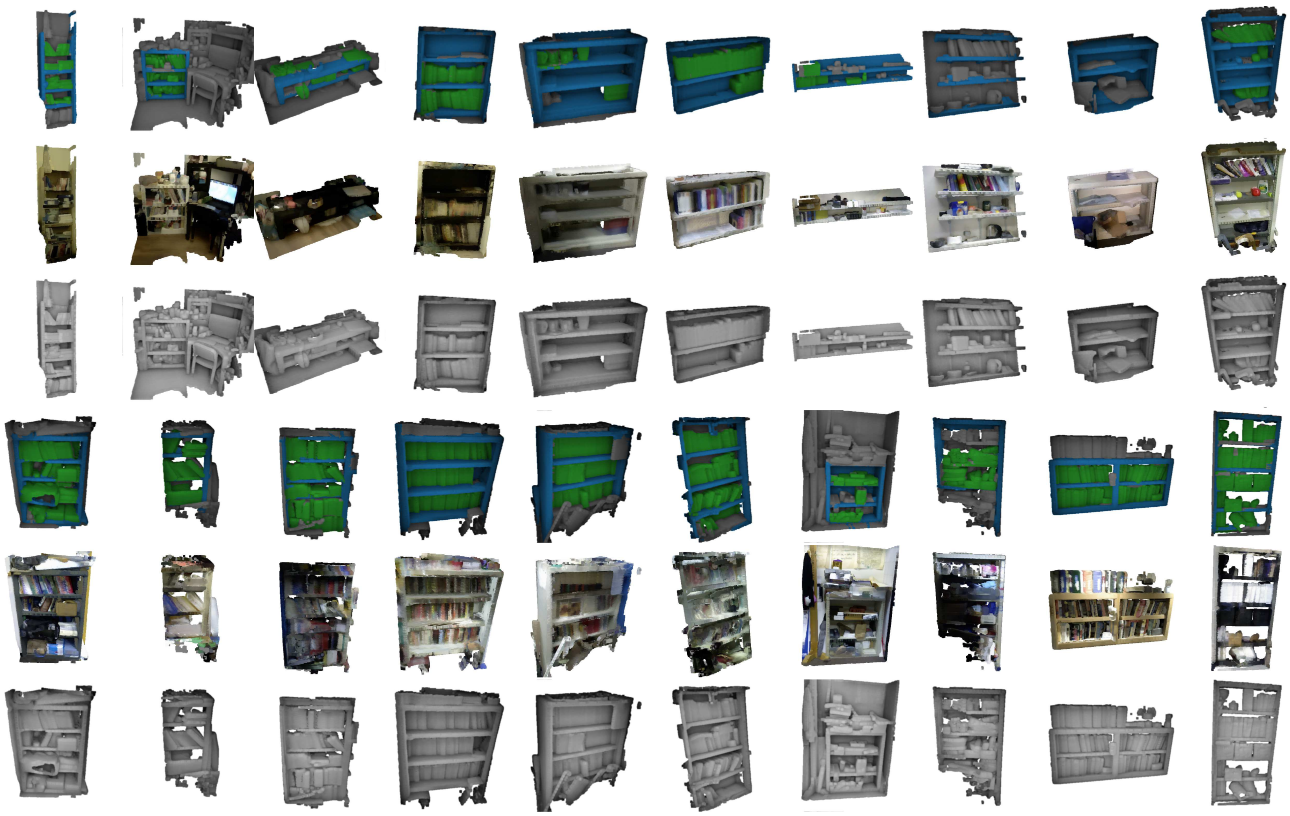

Figure 39. Part annotation: Shelf

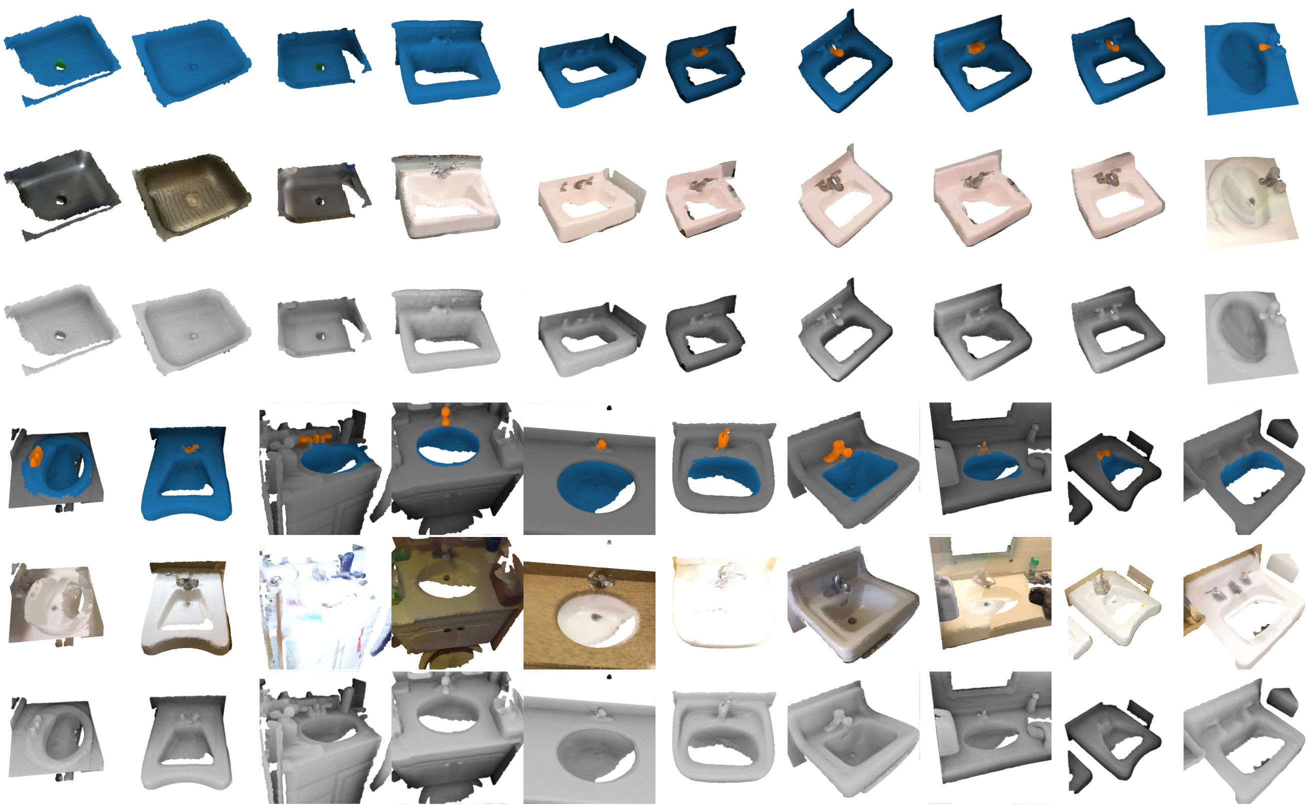

Figure 40. Part annotation: Sink

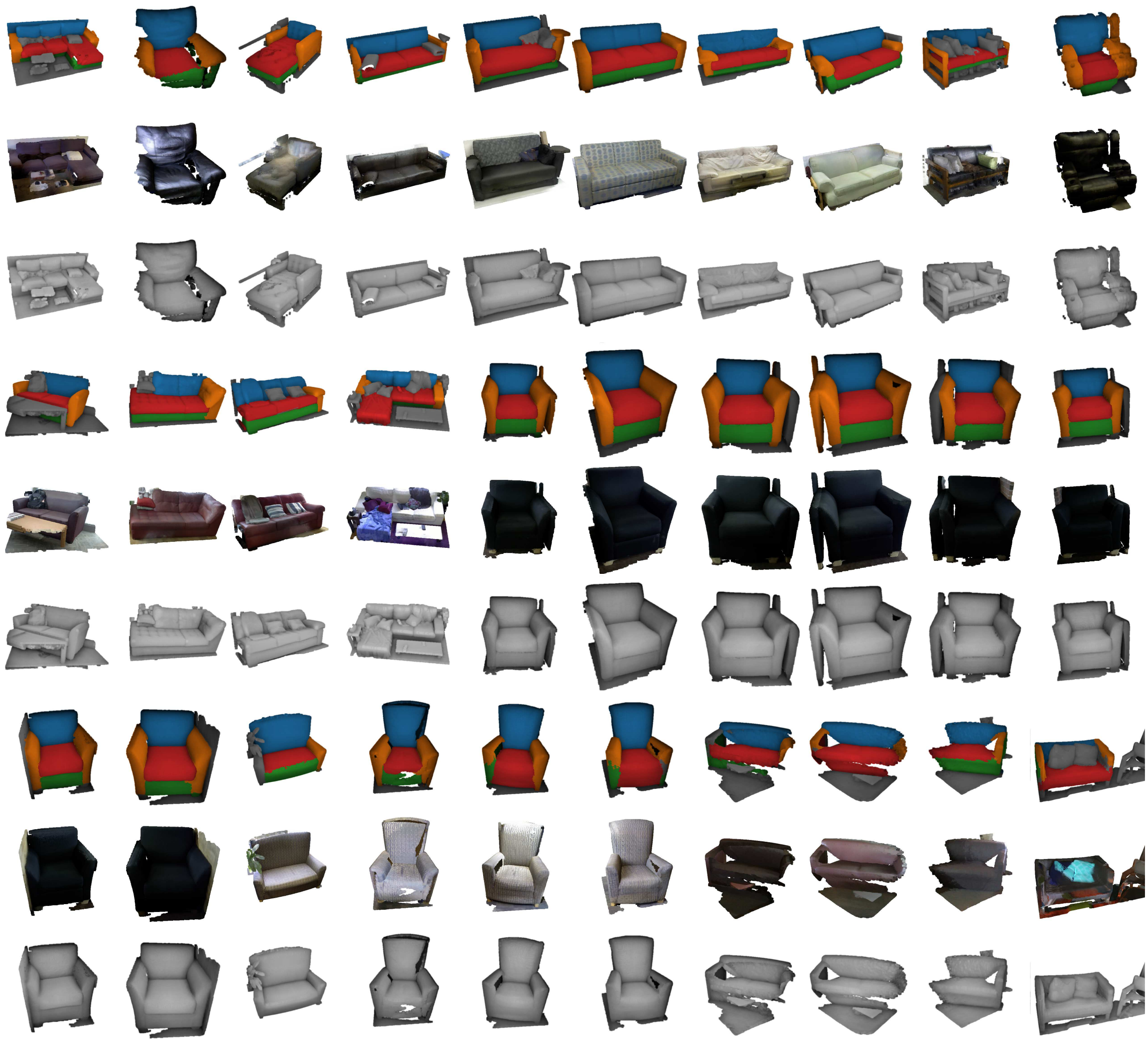

Figure 41. Part annotation: Sofa

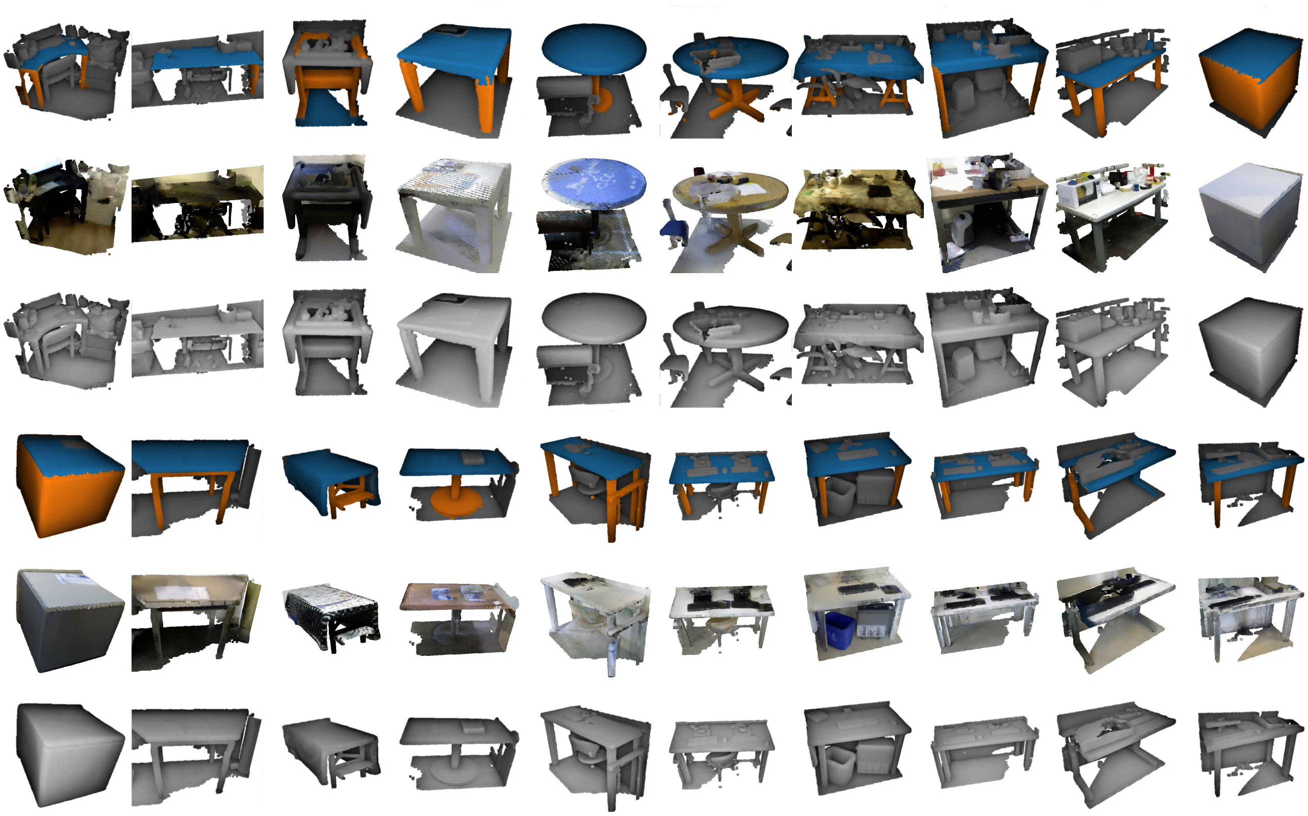

Figure 42. Part annotation: Table

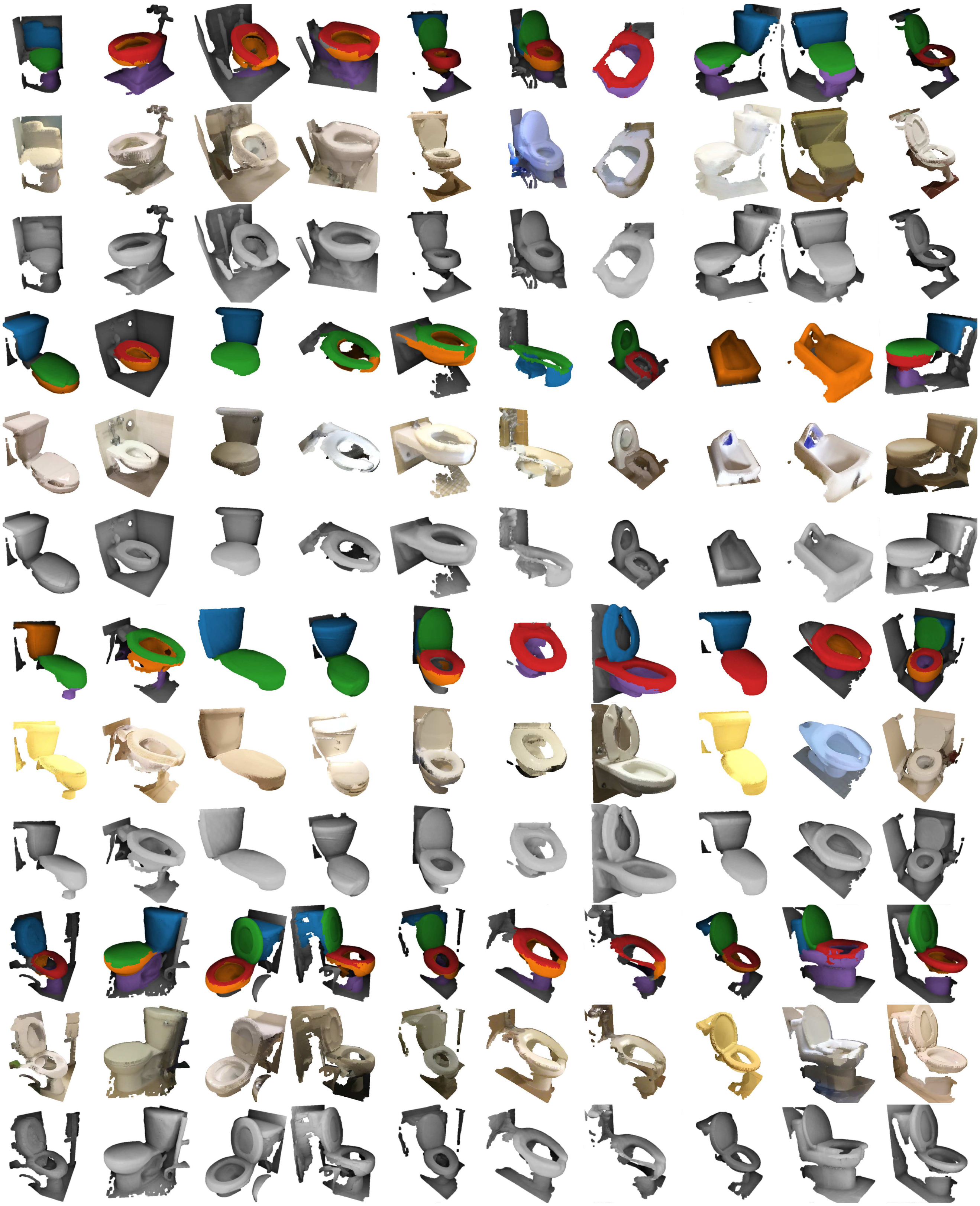

Figure 43. Part annotation: Toilet