3DSSR: 3D Subscene Retrieval Supplemental Materials

Reza Asad Manolis Savva
Simon Fraser University
rasad@sfu.ca msavva@sfu.ca

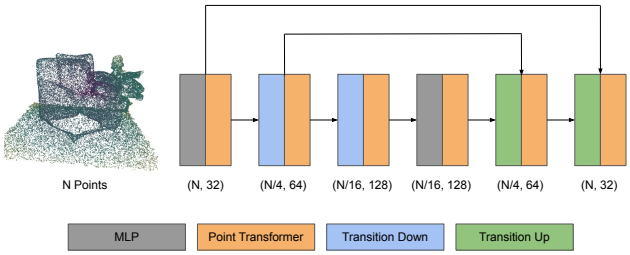

Figure 1. Network architecture for the underlying Point Transformer [11] we use in POINTCROP, CSC [5], and SUPERVISEDTRANSFORMER.

1. Point Transformer Architecture Overview

We use the same underlying Point Transformer [11] architecture for POINTCROP, CSC, and SUPERVISEDTRANSFORMER. Figure 1 shows an overview of the architecture, which is a simplified version of the segmentation architecture proposed by [11]. We first encode the input point cloud using a one-layer MLP with hidden and output dimensions of 32. Next, we apply two blocks, each consisting of a transition down and a point transformer layer. The MLP used in the middle of the network has two layers with hidden and output dimensions of 128. Finally, we apply another two blocks, each consisting of a transition up and a point transformer layer. For the point transformer layers, we always use a dimension of 32. All other parameters in the network (e.g., k used in knn for the transition down module) are the same as in the original paper [11].

2. Implementation Details

For the momentum encoder [4], the centering operation, and the weight decay used in POINTCROP, we follow a cosine scheduler with the same parameters as DINO [1]. We train our model using a learning rate of 0.0005 and the Adam optimizer [6] with default parameters. Before computing the cross entropy loss for POINTCROP, we apply temperature parameters to the teacher and student outputs with the same values as used by DINO [1]. However, unlike DINO [1], for the teacher temperature, we do not use a cosine scheduler or warm-up epochs [9]; i.e., the temperature is simply fixed. For the projection head in POINTCROP [1] we use the output dimension of 2000 (class concepts) without fine-tuning the parameter. To train the SUPERVISEDTRANSFORMER, we use a learning rate of 0.0001 along with the Adam optimizer (default parameters) [6]. For the focal loss [7] in SUPERVISEDTRANSFORMER, we follow the suggestion in the original paper and set the γ parameter to 2. Finally, for all experiments in this paper, we used the categories in Matterport3D [2] based on the 'mpcat40' labels. Note the diversity and the difficulty of this collection compared to the curated 18 'nyu40' labels used in typical 3D object detection pipelines [8, 10]. The categories we have used are: lighting (i.e lamps), sink, appliances, fireplace, shower, blinds, towel, cushion, objects, curtain, chair, furniture, chest of drawers, picture, cabinet, shelving, bathtub, sofa, plant, gym equipment, bed, stool, seating, clothes, toilet, tv monitor, table, mirror.

3. Additional Quantitative Results

3.1. Mean Average Precision Plots

Figure 2 shows the mean average precision (mAP) plots for mAP_{geo} at various IoU thresholds and Chamfer distance (CD) thresholds of 5%, 10%, 20%, and 40%. For both IoU and CD thresholds, we use a range of soft to strict thresholds. Our 3D subscene retrieval model POINTCROPRANK outperforms all models across all threshold values. The results confirm that our model can retrieve 3D subscenes that are more similar to the query subscenes in terms of geometry and object arrangements.

3.2. Replacing IoU with Distance and Angle

The second matching criterion for precision metrics P_{cat} and P_{geo} considers the IoU of each query object and its corresponding candidate in a world coordinate frame. During evaluation, the candidate is rejected as a match if the IoU between the pair is 0. We observe that for some downstream applications, this may be a strict evaluation metric.

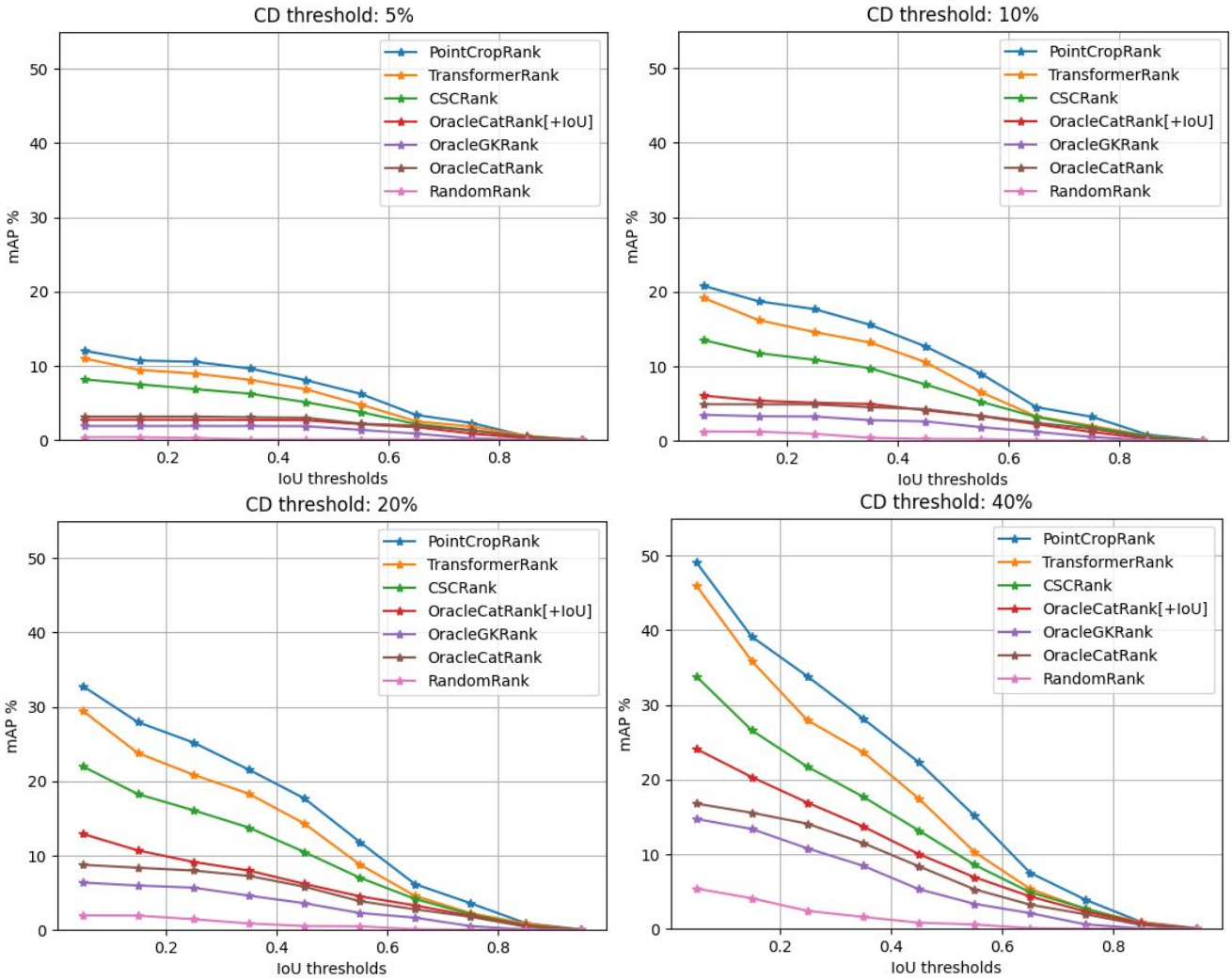

Figure 2. 3D subscene retrieval on 50 test set queries. We evaluate the retrieved results using different methods based on geometric and object arrangement similarity with the query subscene. More concretely, we use mAP_{geo} and compare models at various IoU and Chamfer Distance (CD) thresholds. The top row plots correspond to CD thresholds of 5% and 10% (left to right). The bottom row shows plots for CD thresholds of 20% and 40%. Our model POINTCROPRANK (blue) outperforms all models including the supervised TRANSFORMERRANK (orange) across all thresholds.

Method	AUC[dist+CD]	AUC[angle+CD]	AUC[dist+angle+CD]	AUC[dist+angle+Cat+CD]
ORACLECATRANK	5.88	5.75	4.88	4.88
ORACLECATRANK[+IOU]	8.58	8.99	8.07	8.07
ORACLEGKRANK	5.52	5.22	4.03	4.03
TRANSFORMERRANK	16.14	16.03	14.88	9.46
RANDOMRANK	2.49	1.87	1.60	0.16
CSCRANK	14.73	14.71	13.32	8.52
POINTCROPRANK	**22.53**	**23.05**	**20.98**	**11.52**

Table 1. Comparing 3DSSR models using various matching criteria. The top group shows models that use oracle categories or are supervised with category labels. The bottom group shows the self-supervised models along with a random baseline. Our model POINTCROPRANK outperforms all models using various metric combinations.

Method	AUC[dist+CD]	AUC[angle+CD]	AUC[dist+angle+CD]	AUC[dist+angle+Cat+CD]
TRANSFORMERRANK	20.34	19.91	18.50	**12.44**
CSCRANK	14.89	14.71	13.76	8.56
POINTCROPRANK	**21.01**	**20.74**	**19.26**	10.79

Table 2. Training on ScanNet, evaluating on Matterport3D. Our POINTCROPRANK outperforms CSCRANK across all metrics and is competitive with TRANSFORMERRANK. Note that, unlike the bottom group, TRANSFORMERRANK utilizes category-label supervision during training.

Method	mAP[CD]	mAP[$CD + Cat$]
ORACLECATRANK	7.81	7.81
TRANSFORMERRANK	40.48	27.17
RANDOMRANK	1.96	0.39
CSCRANK	25.88	15.07
POINTCROPRANK	**64.92**	**32.49**

Table 3. Single 3D object retrieval results.

Therefore, we suggest two additional metrics: **radial distance (dist)** and **angular difference (angle)** for each corresponding query and target object, normalized by the query object's radius and 90 degrees respectively. Note that the radius here is computed relative to the centroid of the anchor objects (i.e., the origin of the world coordinate frame). Table 1 reports the Area Under the Curve (AUC) for mean average precision (mAP) across 50 test queries at various distance and angle thresholds. All experiments use a Chamfer Distance (CD) threshold of 10%. POINTCROPRANK outperforms all models including ones that have access to oracle categories.

3.3. Generalization Across Different Datasets

We evaluate the generalization capabilities of the self-supervised models, POINTCROPRANK and CSCRANK across another dataset. To do so, we train POINTCROPRANK and CSCRANK on ScanNet [3] (no category label supervision) and evaluate the trained models on the 50 Matterport3D [2] test queries. We follow the same procedure for TRANSFORMERRANK except that this model benefits from category-label supervision on ScanNet [3]. To perform training, we prepare ScanNet [3] exactly the same way we prepared Matterport3d [2] and do not change any hyperparameters for any of the models. Table 2 shows results using various matching criteria. We observe that our model POINTCROPRANK outperforms CSCRANK across all metrics and is very competitive to the supervised TRANSFORMERRANK.

3.4. Evaluation on 3D Object Retrieval

In the main paper, we showed that better classification accuracy for a point cloud encoder does not necessarily imply better 3D subscene retrieval using that encoder. Consider a user querying a single 3D object (e.g. 'chair'). Retrieving objects from the database by relying on categories alone is problematic because there are many types of chairs (e.g. living room chairs, dining table chairs, swivel chairs, etc). Our downstream applications benefit from retrieving geometrically similar objects within a category. To make this clearer, Table 3 shows single 3D object retrieval results (top 10) with Chamfer distance (CD) alone and CD together with category labels (taking the mean over 50 test queries). The results suggest our POINTCROP outperforms all models across the two metrics.

3.5. Considering Rotational Invariance

All 3DSSR models presented in this paper are invariant under translations of 3D subscenes. This is a direct consequence of our retrieval strategy (Sec 4.3 in the main paper). During retrieval, we translate each target and query subscene to a mutual centroid (i.e., the centroid of the anchor objects). However, we do not consider the rotational invariance of the 3D subscenes (around the upward z-axis). To take this into account, we add a grid search module to each 3DSSR model to search over 45 degrees rotations of each subscene. The results in Table 4 indicate that our POINTCROPRANK achieves higher AUC across all metrics. Although a simple grid search can achieve good results, in the future we plan to **learn** subscene representations invariant under rotations.

4. Additional Qualitative Results

In Figure 3 we show two additional test queries comparing our POINTCROPRANK against CSCRANK and the supervised TRANSFORMERRANK. For the first example, we observe that both POINTCROPRANK and CSCRANK find 3 matches at rank 1. However, at rank 2 our model is able to find a complete match while CSCRANK finds no correct matches. For the second example, we observe that our model retrieves a geometrically similar cabinet (in pur-

Method	AUC[dist+CD]	AUC[angle+CD]	AUC[dist+angle+CD]	AUC[dist+angle+Cat+CD]
ORACLECATRANK[+IOU]	8.68	9.09	8.09	8.09
TRANSFORMERRANK	15.49	15.40	14.28	8.81
CSCRANK	15.26	15.40	13.84	8.76
POINTCROPRANK	**21.01**	**21.37**	**19.37**	**10.48**

Table 4. Evaluation with a subscene rotation module. Our POINTCROPRANK achieves higher AUC values across all metrics.

ple) at all ranks. However, the supervised TRANSFORMERRANK does not identify a cabinet at ranks 2 and 3. Furthermore, the tables identified by TRANSFORMERRANK do not seem to be geometrically similar to the query table, except at rank 2.

We show another two test set queries in Figure 4, comparing our model against two models that directly use oracle categories (ORACLEGKRANK and ORACLECATRANK[+IOU]). In the first example, our model identifies a curtain at all ranks. Furthermore, the retrieved table at ranks 1 and 3 from our model seem to be geometrically more similar to the query table. Note that all results from ORACLEGKRANK seem to have arrangements that do not match the query subscene. For the second example, we observe that ORACLECATRANK[+IOU] matches a couch to the query chair at rank 1. This appears to be an annotation error in the original Matterport3D dataset [2]. Furthermore, the lamp and the table from rank 3 of our model are more similar to the query lamp and the query table compared to ORACLECATRANK[+IOU]'s result at rank 3. We observe that ORACLECATRANK[+IOU] retrieves a dining table and a ceiling lamp at rank 3 which is quite different from the query subscene.

References

[1] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9650–9660, 2021.

[2] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d. *International Conference on 3D Vision (3DV)*, 2017.

[3] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017.

[4] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020.

[5] Ji Hou, Benjamin Graham, Matthias Nießner, and Saining Xie. Exploring data-efficient 3d scene understanding with contrastive scene contexts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15587–15597, 2021.

[6] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[7] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

[8] Ze Liu, Zheng Zhang, Yue Cao, Han Hu, and Xin Tong. Group-free 3d object detection via transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2949–2958, 2021.

[9] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

[10] Ishan Misra, Rohit Girdhar, and Armand Joulin. An end-to-end transformer model for 3d object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2906–2917, 2021.

[11] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip H.S. Torr, and Vladlen Koltun. Point transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 16259–16268, October 2021.

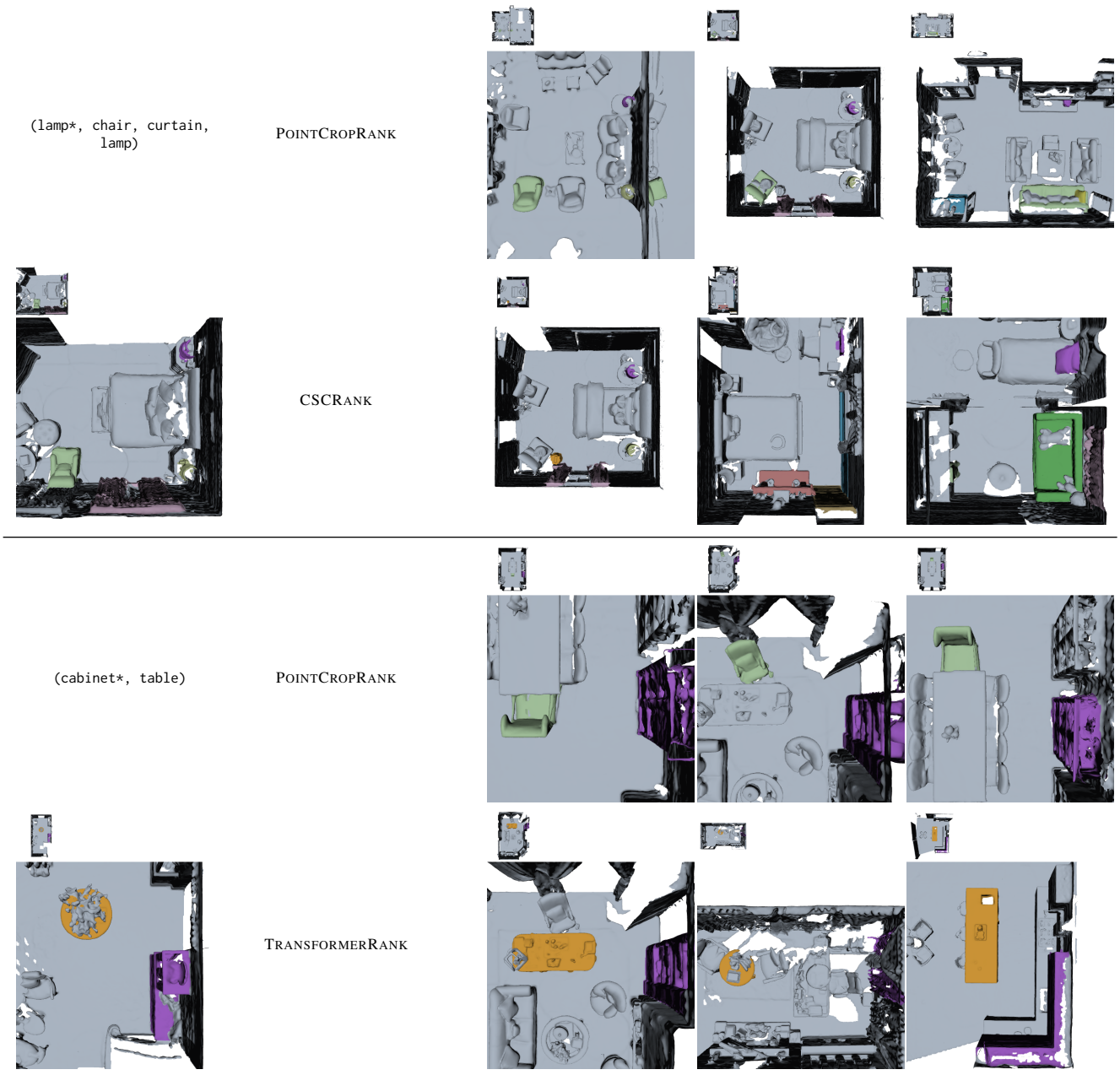

Figure 3. Qualitative results for two additional test set queries comparing our PointCropRank, the self-supervised CSCRank, and the supervised TransformerRank. We observe that in the first example both CSCRank and our model find 3 matches at rank 1. However, at rank 2 our model finds a full match (4 objects) while CSCRank finds no matches. For the second example, we note that only our model is able to retrieve a geometrically similar cabinet (in purple) at all ranks. Here, the supervised TransformerRank does not identify a cabinet at ranks 2 and 3. Moreover, the tables from TransformerRank do not seem to be geometrically similar to the query table, except at rank 2.

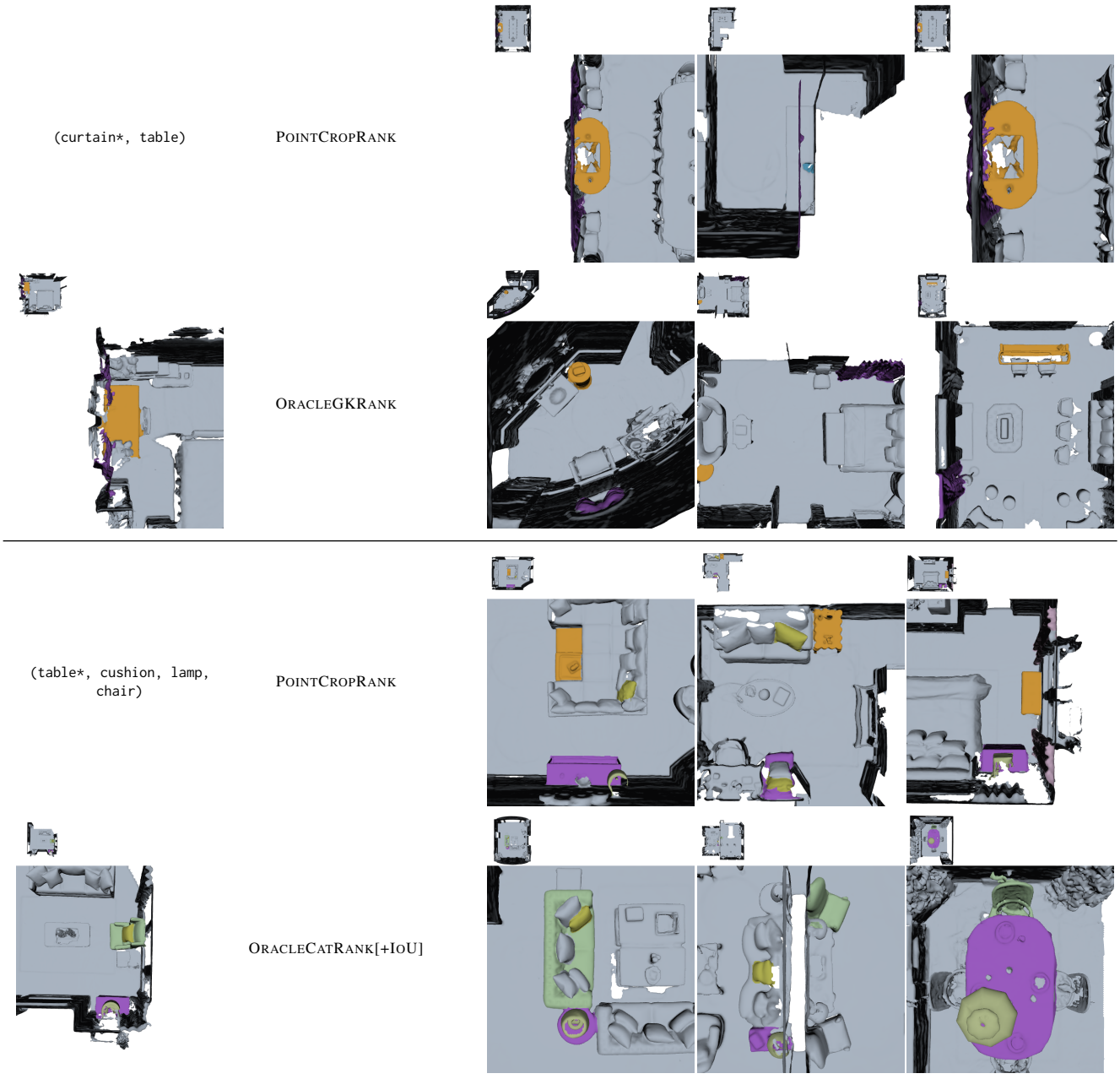

Figure 4. Qualitative results for two test set queries comparing PointCropRank with approaches that directly use oracle object categories. For the first example, we note that our model identifies a curtain at all ranks. Furthermore, the retrieved table at ranks 1 and 3 from ours seem to be geometrically more similar to the query table. For the second example, we observe that OracleCatRank[+IoU] suffers from an annotation error and matches a couch to the query chair at rank 1. Comparing the lamp and the table from rank 3 of our model against OracleCatRank[+IoU], we observe that OracleCatRank[+IoU] retrieves a dining table and a ceiling lamp which is quite different from the query compared to ours.