Supplementary material

Unsupervised Detection of Distinctive Regions on 3D Shapes

by

Xianzhi Li, Lequan Yu, Chi-Wing Fu,
Daniel Cohen-Or, and Pheng-Ann Heng

published in ACM TOG

Overview

This supplementary document contains the following parts.

- In Part A, we compare the distinctive regions detected by our framework with different clustering methods.
- In Part B, we explore the effect of the number of clusters (C) on distinctiveness detection.
- In Part C, we compare the distinctive regions detected with different per-point distinctiveness extraction methods.
- In Part D, we compare the distinctive regions detected with different backbones for learning the per-point local features.
- In Part E, we show more examples in the collected airplane datasets.
- In Part F, we provide details on our user studies.
- In Part G, we show additional results on distinctiveness-guided shape retrieval.
- In Part H, we show additional results on distinctiveness-guided sampling.
- In Part I, we show additional results on view selection of 3D scenes.
- In Part J, we discuss the hyper-parameters in our network.
- In Part K, we show additional distinctiveness detection results based on random sampling.
- In Part L, we show additional distinctiveness detection results to demonstrate the effect of using different training sets.
- In Part M, we show additional non-extreme distinctive regions detected by our method.
- In Part N, we present the hierarchical analysis on our method.

A. Effects of Different Clustering Methods

As introduced in the main paper, we propose an unsupervised framework for distinctive region detection on 3D shapes, where we iteratively re-cluster the global per-shape features during the network training process. In our current implementation, we adopt the spectral clustering algorithm [7]. In this part, we explored two other clustering algorithms, *i.e.*, k-means clustering [1] and Ward's hierarchical clustering [3]; see Figure 1 below for the visual comparison between the distinctive regions detected by our method with different clustering algorithms. Note that all the network models were trained using the ModelNet40 training split dataset.

From the results we can see that, the detected distinctive regions are similar among the three clustering methods for most of the shapes, except that the k-means clustering method produces worse results on the Table and Car models. This demonstrates that the per-shape features learnt by our framework are well discriminative, thus the clustering performance is less affected by variations in the clustering methods.

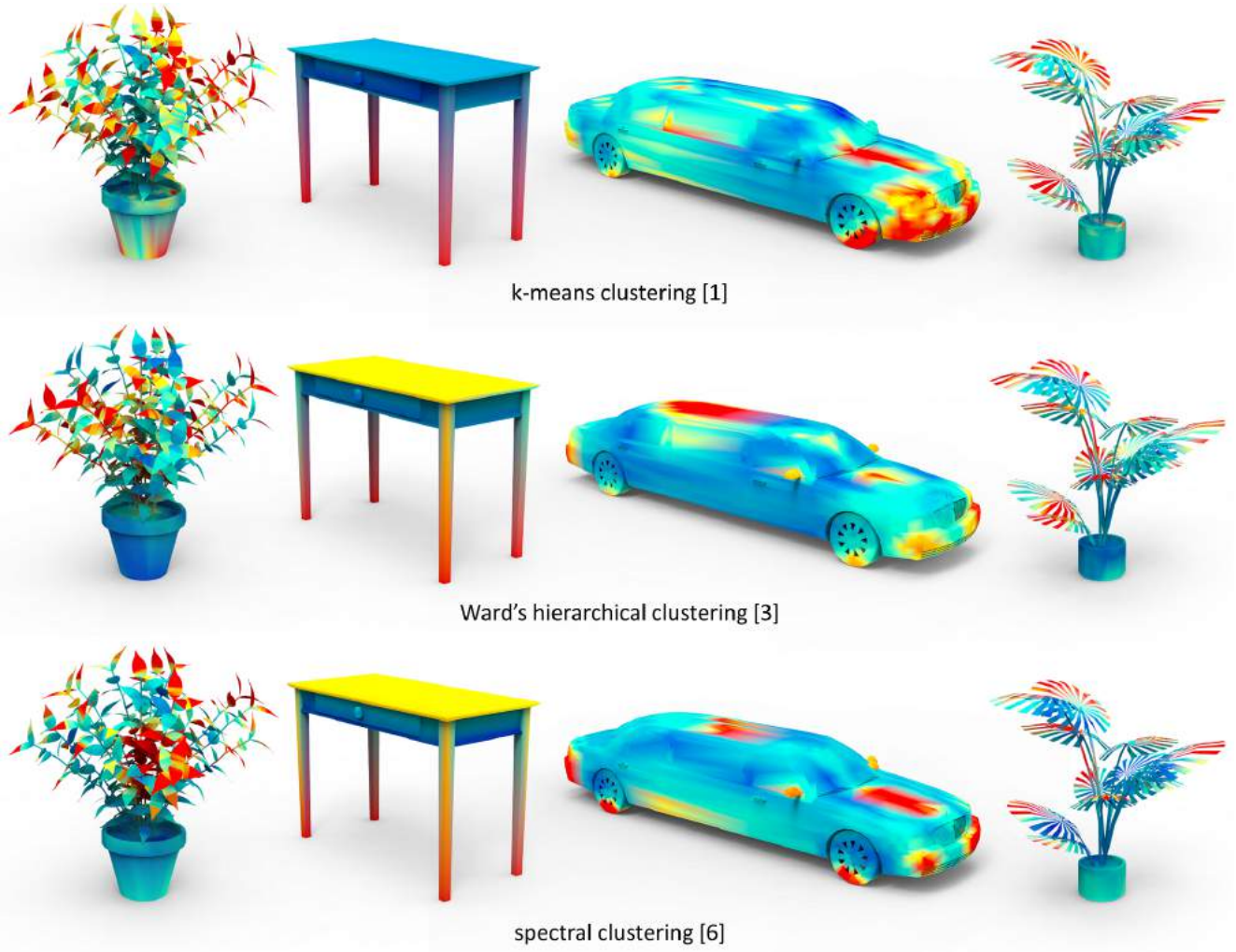

Figure 1. Distinctive regions detected by our framework with three different clustering methods.

B. Effect of the Number of Clusters (C)

Our network can be trained with different number of clusters C. To explore the effect of C on distinctiveness detection, we trained our network on a set of around 3.5k Car objects from the ShapeNet dataset with different C values.

Figure 2 below shows the distinctive regions detected by some of these trained networks, where we can see that the network tends to highlight the shape contours for small C and focuses more on the details for large C. For example, when $C = 2$, the network tends to highlight the front, tail, rearview mirror, and tire of the car, which describe the contour of a car. However, when $C = 20$, the network only detects some fine-details as distinctive regions, *i.e.*, only the front of the car or only the top of the car. Noting that our network is trained for a shape clustering task, the value of C leads to a contour-to-detail (or coarse-to-fine) detection of distinctive regions, since a rough clustering requires to focus only on the shape contour, while to sort the shapes into more clusters requires more attention to the local details. Figure 3 on next page shows another sets of examples, which exhibit a similar contour-to-detail (or coarse-to-fine) detection of distinctive regions.

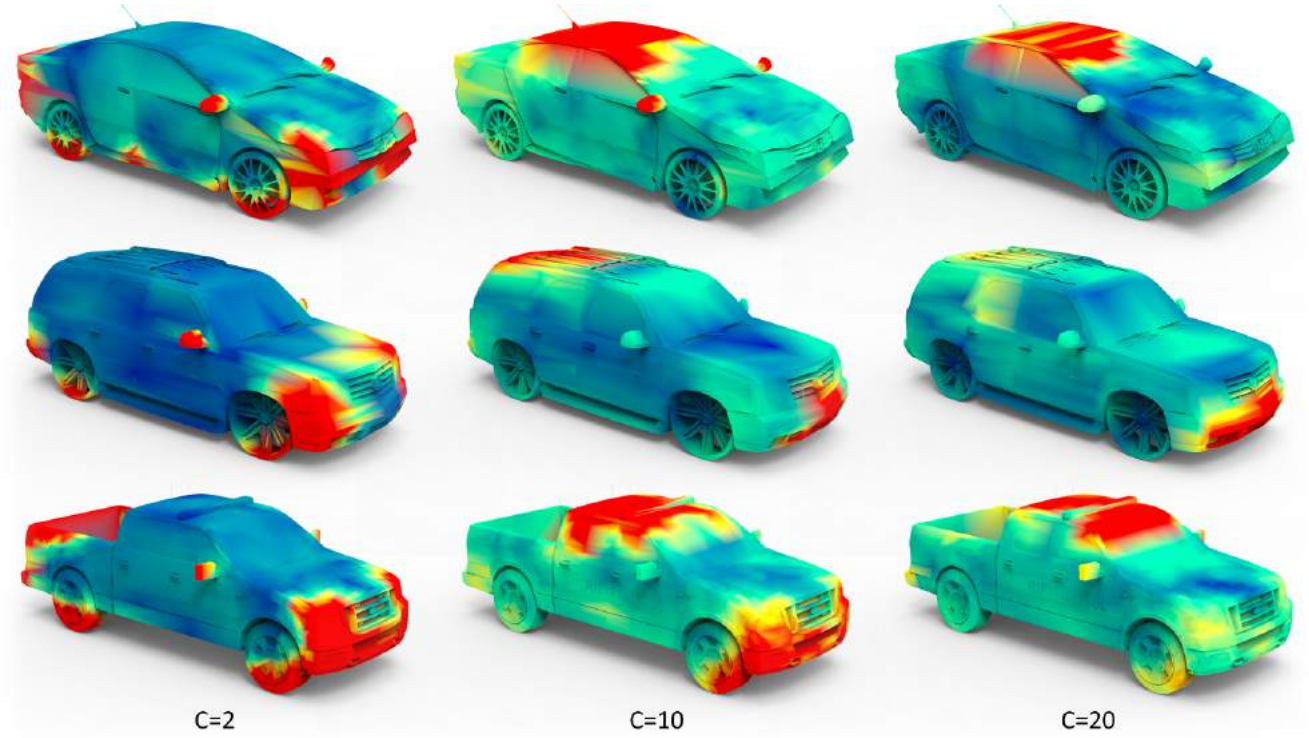

Figure 2. Distinctive regions detected by our network when trained with different number of clusters (C), from left to right.

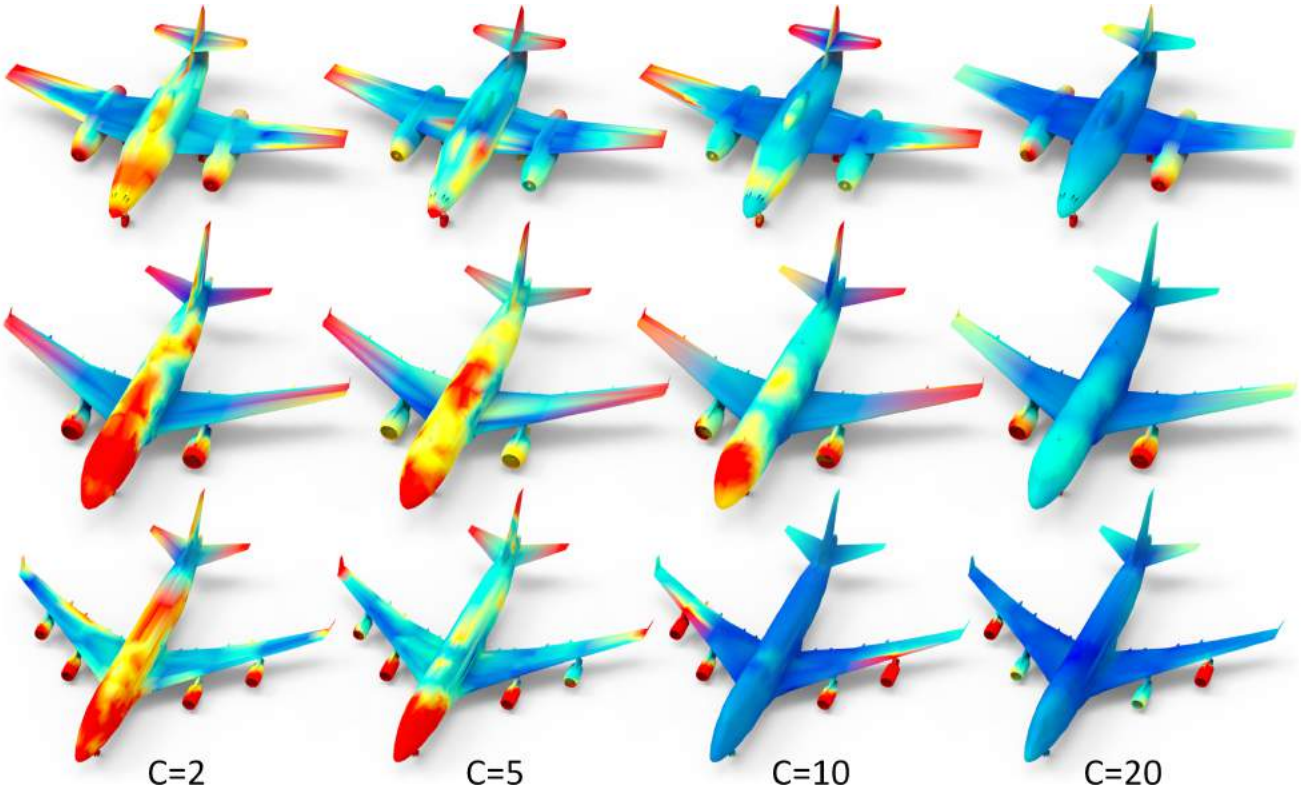

Figure 3. Another example of distinctive regions detected by our network when trained with different number of clusters (C), from left to right.

C. Effects of Different Distinctiveness Extraction Methods

As introduced in Section 3.6 of the main paper, we obtain the per-point distinctiveness $d_{i,j}$ from $\mathbf{f}^r_{i,j}$ of each point $\mathbf{p}_{i,j}$ by taking the maximum value in $\mathbf{f}^r_{i,j}$ for each shape. Here, we explored several other alternatives, including the average of the three largest values, L_2 norm, and the mean.

Figure 4 below shows the effects of different per-point distinctiveness extraction methods. Note that the network was also trained with the ModelNet40 training split dataset. From the results, we can see that the detected distinctive regions by taking the maximum value, the average of the three largest values, and the L_2 norm are very similar, indicating that there are no obvious difference among these choices. However, the detected distinctive regions by the mean operation tend to contain some noise, since such operation considers all the feature values equally.

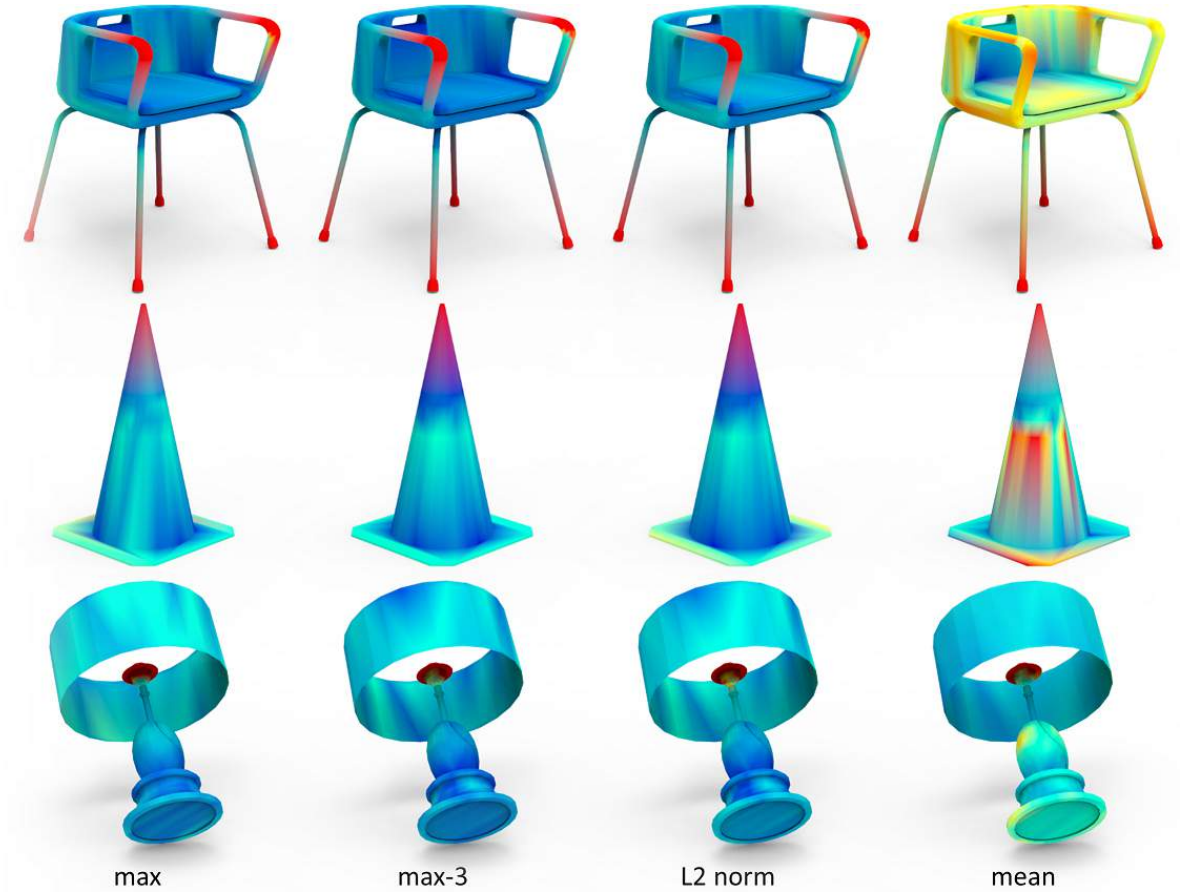

Figure 4. Distinctive regions detected with different per-point distinctiveness extraction methods.

D. Effects of Different Network Backbones

In our proposed distinctive region detection framework, the choice of the network architecture backbone for per-point feature embedding is flexible. In the main paper, we adopt PointCNN [2] as the network backbone. Here, we show the distinctive regions detected by using different network backbones, *i.e.*, PointNet [4] and PointNet++ [5]; see Figure 5 below. From the results, we can see that PointNet++ and PointCNN can produce similar distinctive regions for some highly-discriminative shapes, *e.g.*, the Airplane and the Bottle. However, for some complex shapes like the Person and the Plant, PointCNN can produce more reasonable distinctive results compared with others. This indicates that a more powerful and robust feature embedding network can facilitate more accurate detection results.

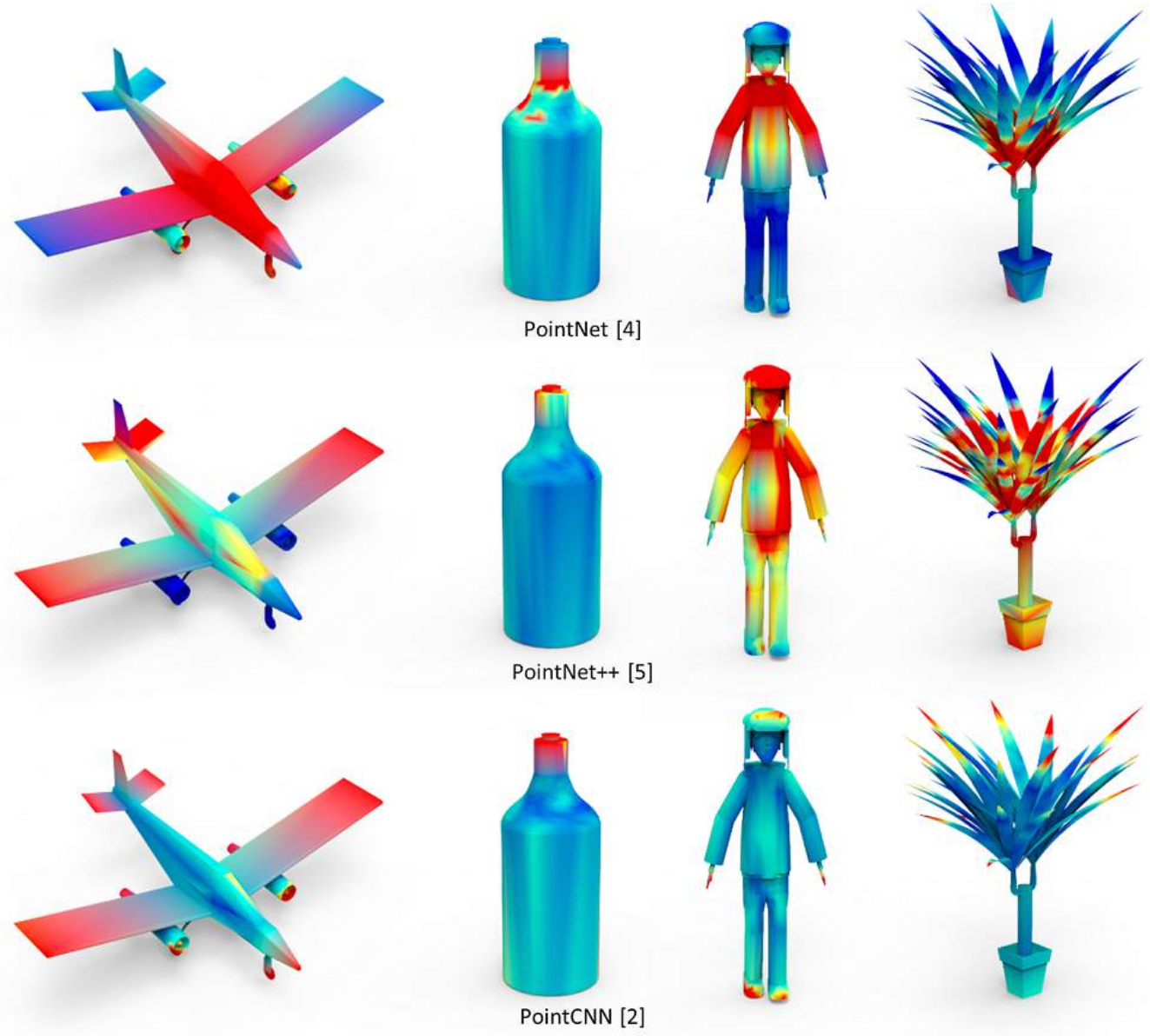

Figure 5. Distinctive regions detected by employing different network backbones for per-point feature embedding.

E. More Examples in Our Collected Airplane Datasets

Figure 6 below shows more examples of our collected airplane datasets, including two-engine airplanes (top row), four-engine airplanes (middle row), and tail-engine airplanes (bottom row). It shows that each kind of collected training dataset contains a wide variety of airplanes.

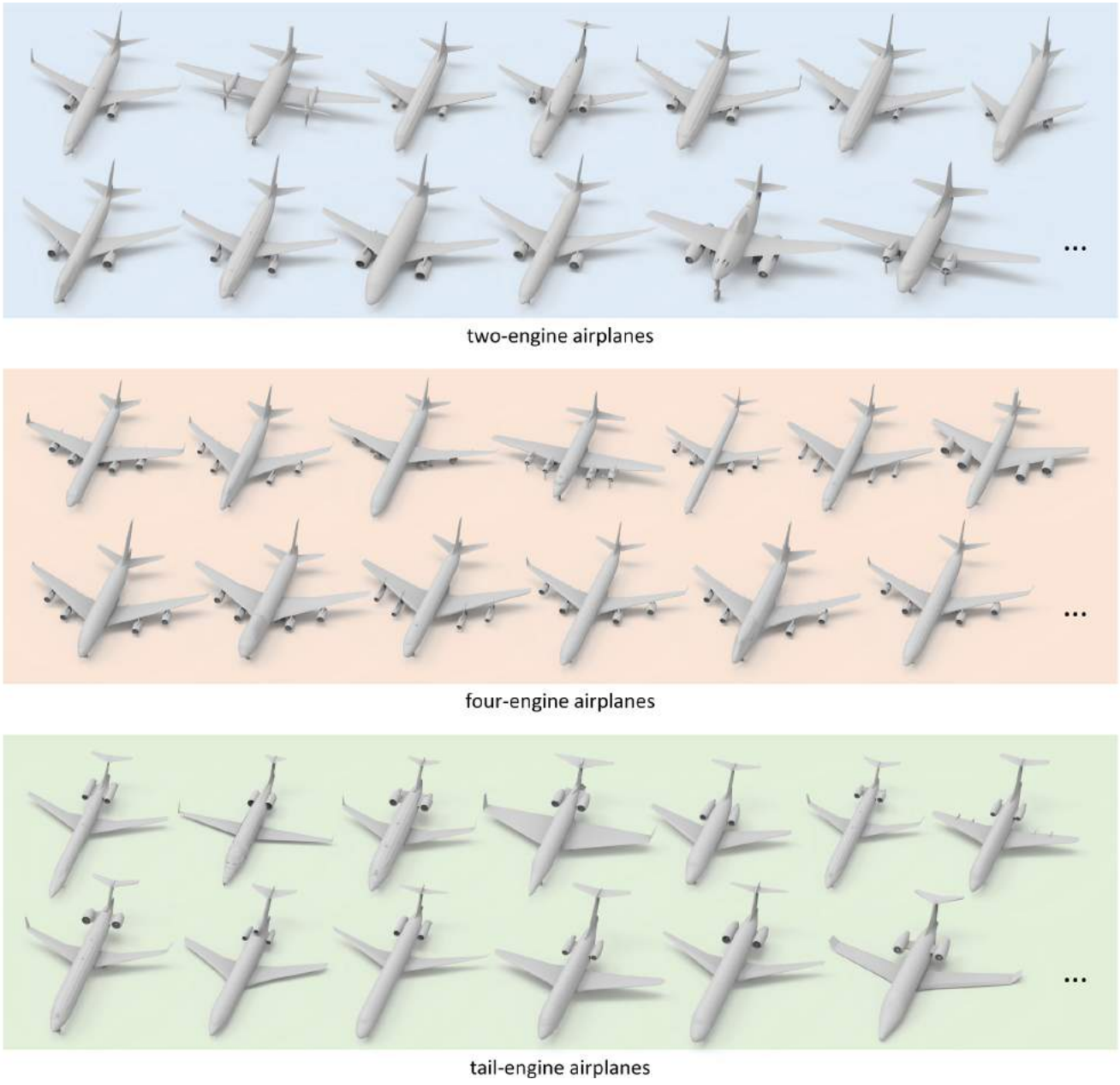

Figure 6. Examples of our collected airplane datasets. From top to bottom: two-engine airplanes, four-engine airplanes, and tail-engine airplanes.

F. Details of User Studies

User interface in our user studies. In Section 4.7 of the main paper, we presented two user studies to obtain a sense of how consistent our results are with humans. Figures 7 & 8 show the user interface screenshots for the intra-class and inter-class studies, respectively. Users can use mouse to freely rotate each shape. In the intra-class user study, we simultaneously showed 32 shapes on the computer screen, a total of 4 rows and 8 columns; see Figure 7, while in the inter-class user study, we simultaneously showed 75 shapes on the computer screen, a total of 5 rows and 15 columns; see Figure 8.

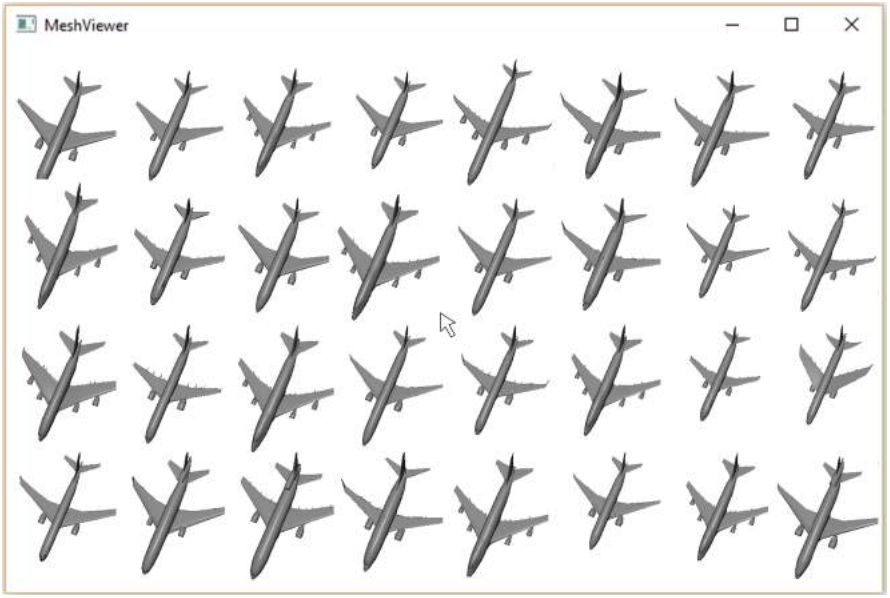

Figure 7. User interface employed in the intra-class study.

Statistical testing on number of participants. To verify that having 30 participants in the inter-class study (see Section 4.7 in our main paper) are sufficiently large to produce stable labeled results, we performed the well-known Student's T-Test as follows. Instead of using the labeled results from all the 30 participants, we randomly select labeled results from 25 out of the 30 participants, and re-calculate the corresponding FNE, FPE, and WME values. By then, we follow the Student's T-Test function in MatLab to check if the calculated FNE, FPE, and WME values with the 25 participants are significantly different from those calculated with all participants. Here, we set the significance level α to be 0.05, and calculate the p values for FNE, FPE, and WME, respectively. Further, to avoid bias, we repeat the above calculation 100 times. In this statistical test, the ranges of the obtained p values of FNE, FPE, and WME over the 100 calculations are [0.81, 1.00], [0,89, 1.00], and [0.70, 0.91], which are all larger than 0.05. Hence, we cannot reject the null hypothesis, i.e., there is no obvious difference between using 25 participants and 30 participants. Hence, 30 participants are sufficient to produce stable labeled results.

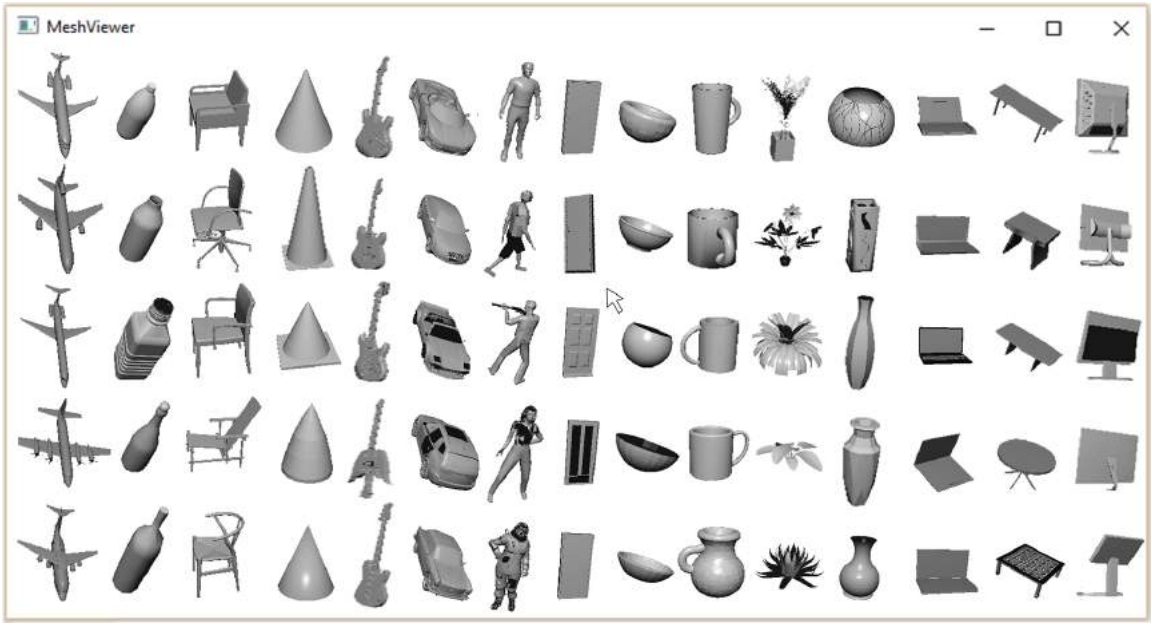

Figure 8. User interface employed in the inter-class study.

G. Additional Results of Distinctiveness-guided Shape Retrieval

Figures 9-12 show more shape retrieval results. In each figure, we retrieved top-five similar shapes by using two traditional methods based on hand-crafted features, *i.e.*, FPFH [6] and SHOT [8], the unsupervised deep-learning based method, *i.e.*, FoldingNet [9], the global features $\mathbf{g}_j$ produced by our network, and also the distinctiveness-guided global features $\mathbf{h}_j$ as the shape descriptor; see main paper for the details. From the results we can see that, using the distinctiveness-guided retrieval method, all the returned shapes have similar local substructures as the query shapes.

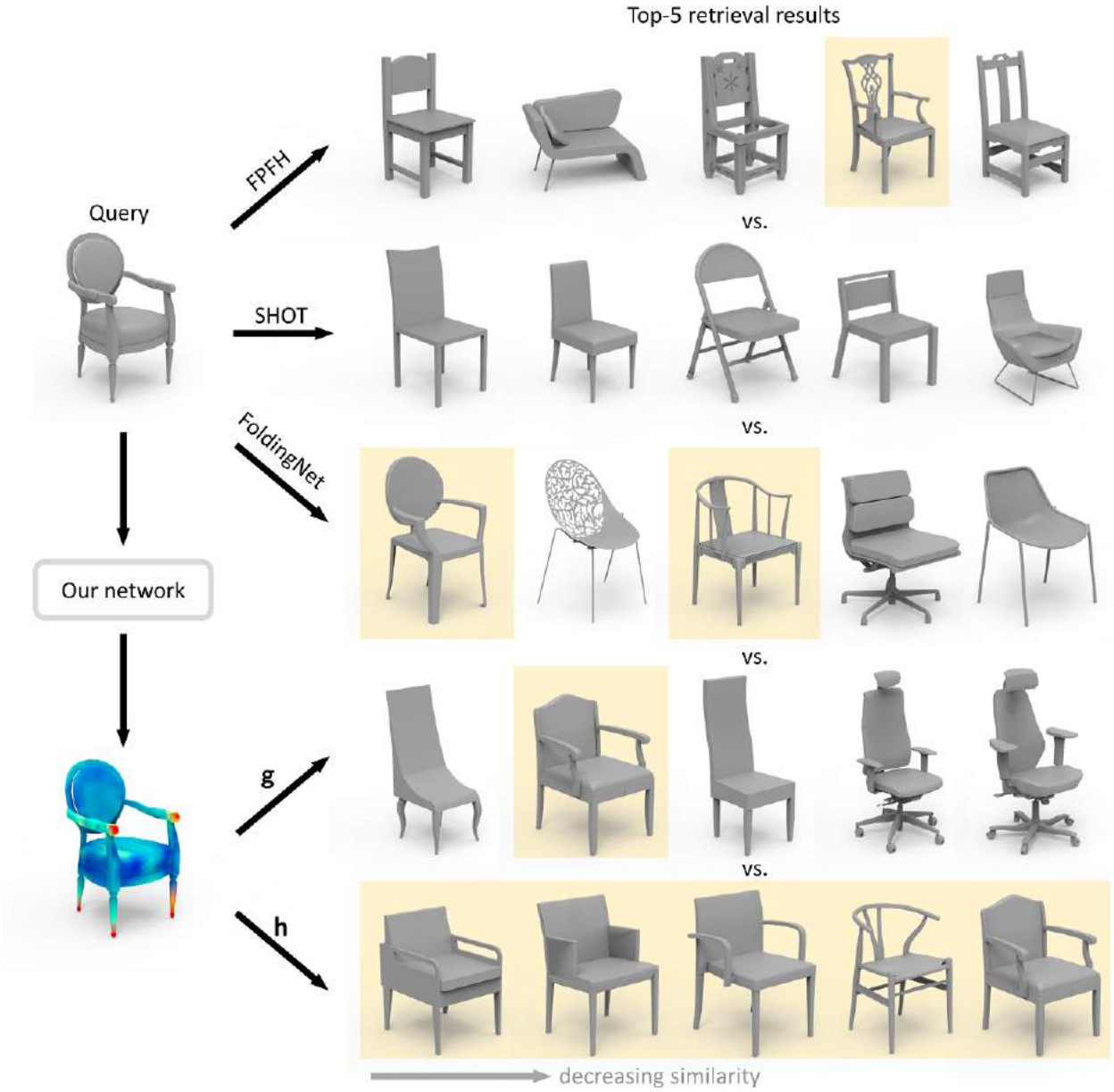

Figure 9. Additional shape retrieval result #1. Retrieved similar chairs are marked in yellow background.

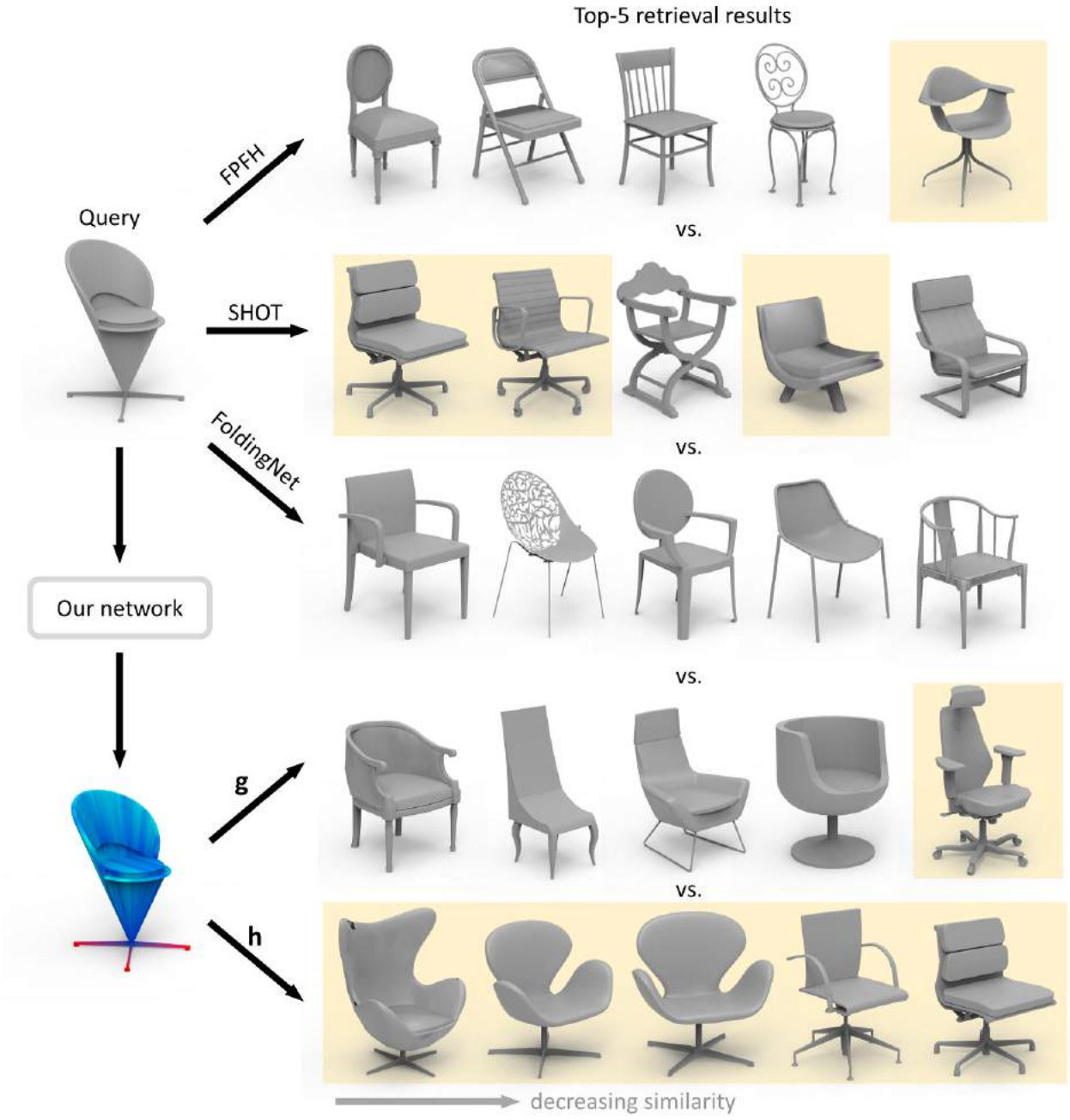

Figure 10. Additional shape retrieval result #2. Retrieved similar chairs are marked in yellow background.

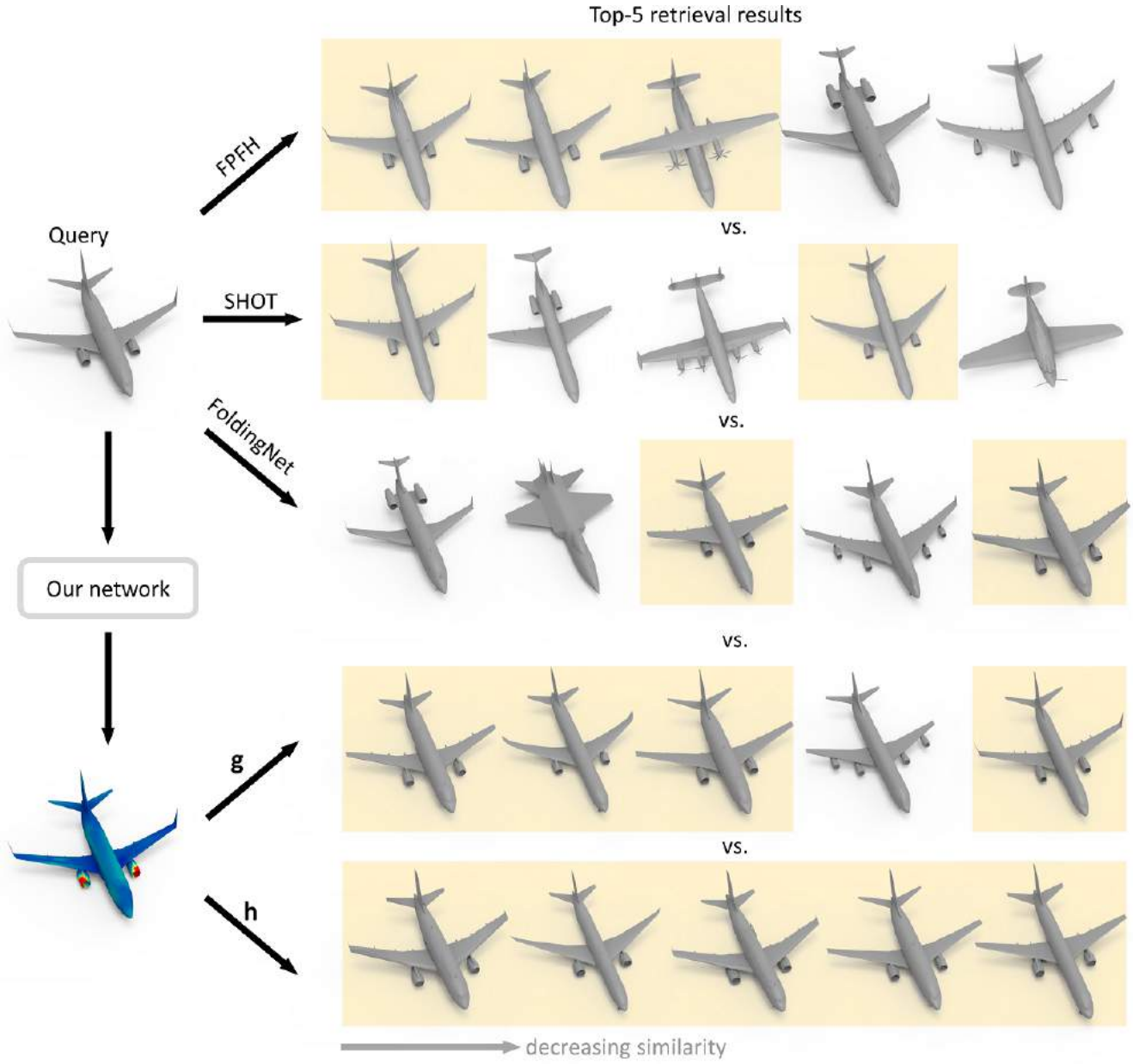

Figure 11. Additional shape retrieval result #3. Retrieved similar airplanes are marked in yellow background.

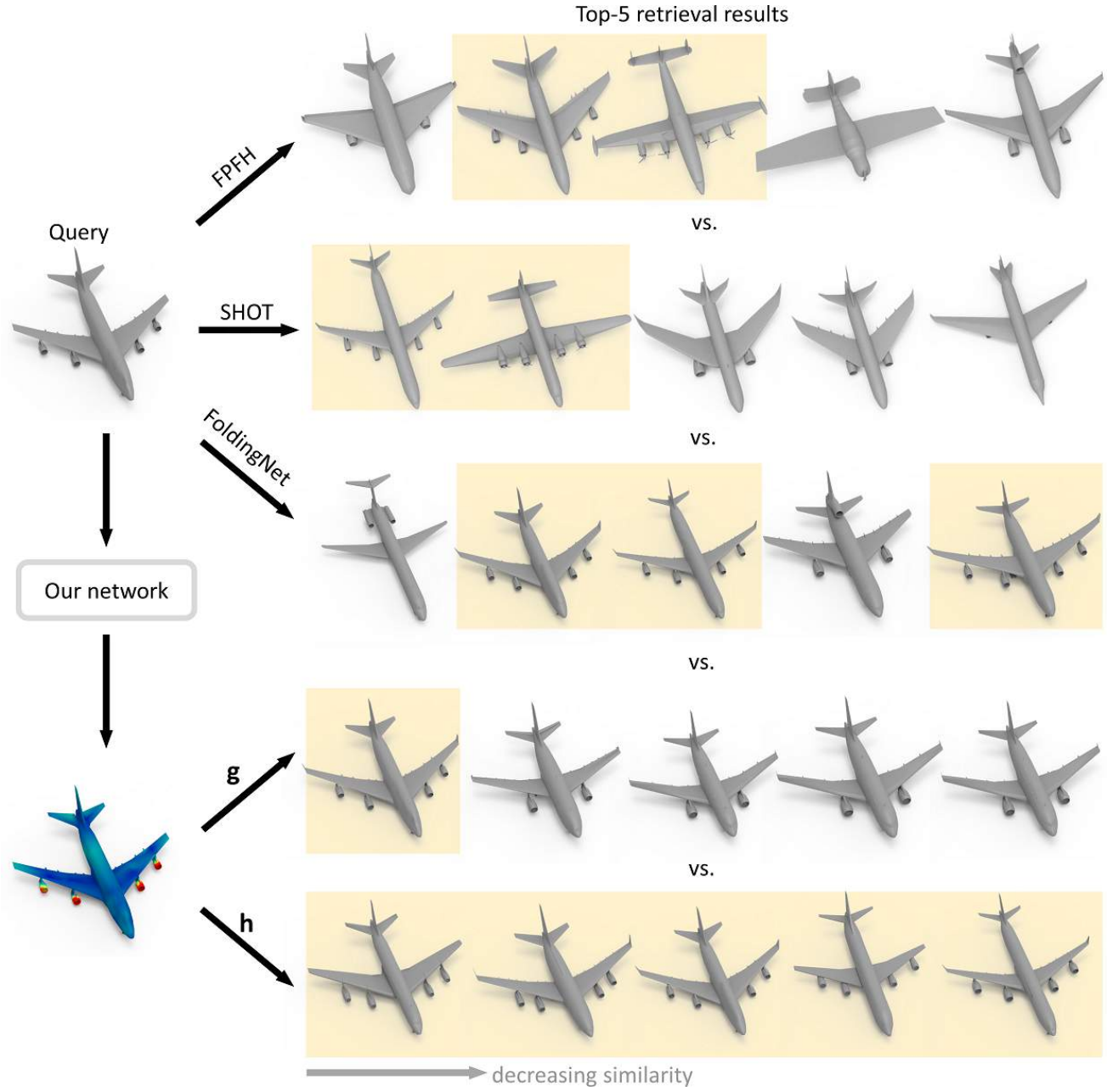

Figure 12. Additional shape retrieval result #4. Retrieved similar airplanes are marked in yellow background.

H. Additional Results of Distinctiveness-guided Sampling

Figure 13 shows more sampling results on a Chair model and on an Airplane model, produced using conventional Poisson disk sampling and our adaptive Poisson disk sampling guided by the network-predicted distinctiveness values; see the main paper for the details. From the results, we can see that distinctiveness-guided sampling arranges more points in high distinctive regions.

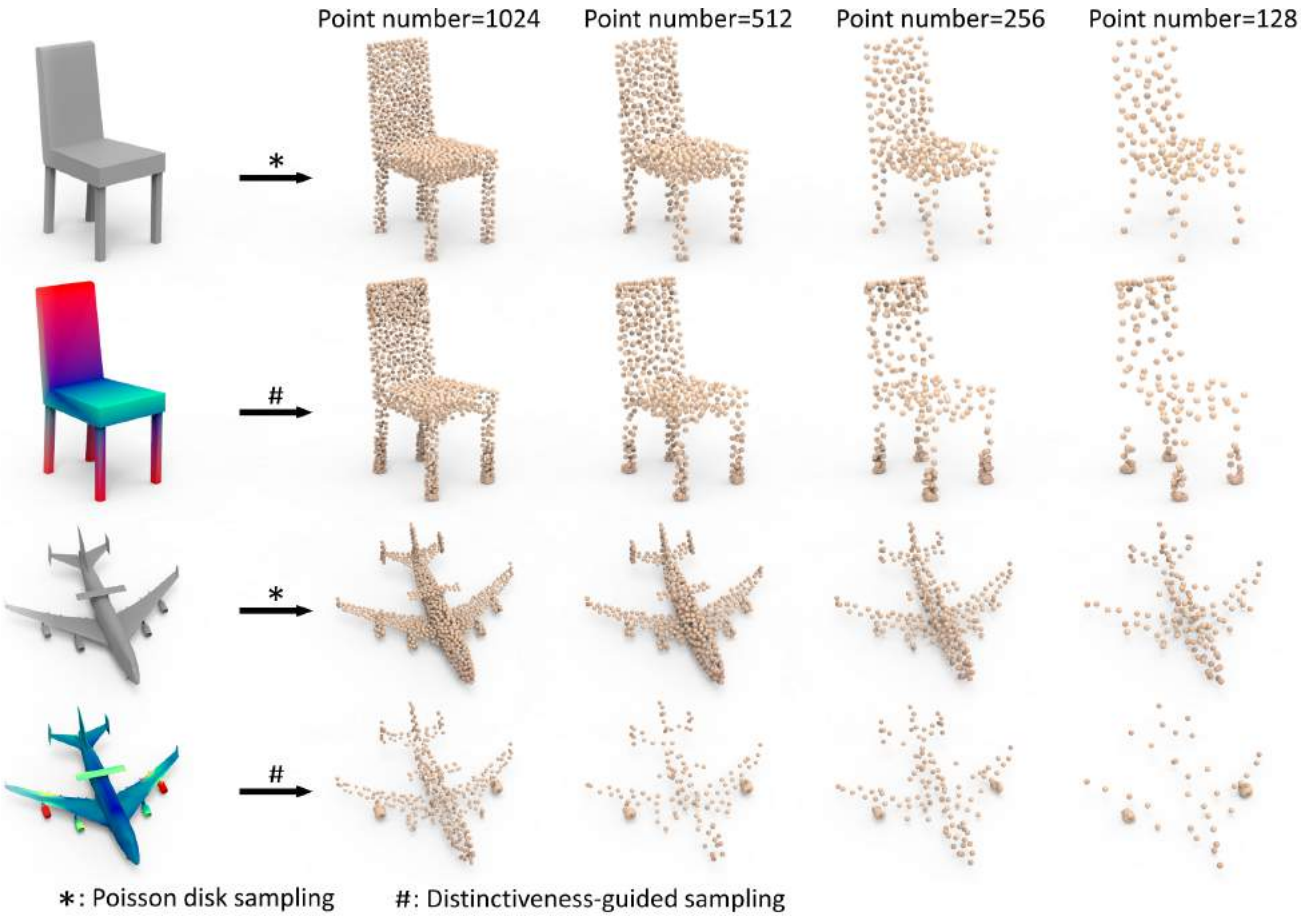

Figure 13. Additional results of distinctiveness-guided sampling vs. Poisson disk sampling.

I. Additional Results of View Selection in 3D Scenes

Figure 14 shows more results of view selection in 3D scenes. Compared with these worst views (bottom row), the automatically selected best views (top row) can reasonably present most distinctive regions.

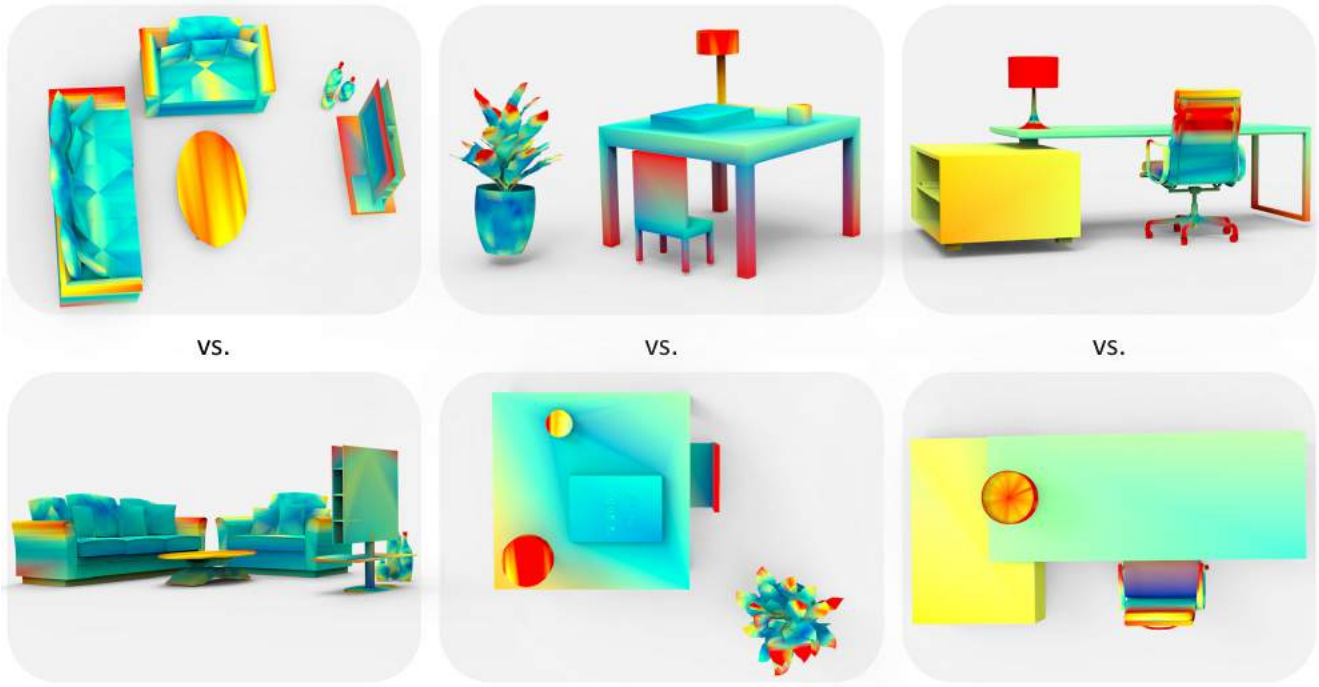

Figure 14. Additional results of view selection in 3D scenes.

J. Discussions on Hyper-parameters

In this part, we explore the distinctiveness detection performance by setting different hyper-parameter values on λ in Eq.(6), and α and β in Eq.(7). Please refer to our main paper for the details of each parameter.

(i) λ in Eq.(6). Figure 15 shows the distinction results by setting different values of λ. Note that, the parameter λ serves like a margin. Hence, when the per-shape global features produced for a negative pair are distant enough, i.e., $D(\mathbf{g}_j, \mathbf{g}_j^-) \geq \lambda$, no efforts are wasted on enlarging that distance. In our default setting, we set $\lambda = 2.0$, meaning that we encourage all the negative samples to be far away from the anchor $\mathbf{g}_j$. If we set $\lambda = 0$, it means that the negative pairs have no effect, and the triplet inputs will degenerate to consider only the anchor & positive samples. Therefore, the distinctive results become the worst; see the first row in Figure 15. If we set $\lambda = 1.0$, it means that we penalize only these negative pairs with $D(\mathbf{g}_j, \mathbf{g}_j^-) < 1.0$. The corresponding results shown in the second row are acceptable, but not as good (or localized) as the results produced by our default setting; see the bottom row.

(ii) α & β in Eq.(7). To explore the effect of setting different values of α and β, we implement the same quantitative evaluation on the distinction results here, as in Section 4.2 of our main paper. Table 1 below shows the shape classification accuracy on the ModelNet40 test dataset under different values of α and β, in terms of decreasing the number of downsampled points. Note that, the distinctiveness detection results are not very sensitive to the values of α and β. If we set values in the same order of magnitude, e.g., $\alpha = 2.0$ vs. $\alpha = 3.0$ vs. $\alpha = 4.0$, the distinctive results will be almost the same. Hence, to clearly show the difference, we set values on different orders of magnitude. From the table, we can observe that our default settings, i.e., $\alpha = 3.0$ and $\beta = 10^{-5}$, leads to the highest classification accuracy.

Table 1. Comparing the overall shape classification accuracy on ModelNet40 when downsampling the test shapes with more points on the distinctive regions detected with different values of α and β in Eq.(7).

# points	$\beta = 10^{-5}$			$\alpha = 3.0$		
	$\alpha = 0.1$	$\alpha = 3.0$	$\alpha = 10.0$	$\beta = 10^{-4}$	$\beta = 10^{-5}$	$\beta = 10^{-6}$
1024	0.876	**0.881**	0.877	0.873	**0.881**	0.874
512	0.866	**0.874**	0.867	0.865	**0.874**	0.864
256	0.825	**0.829**	0.821	0.811	**0.829**	0.813

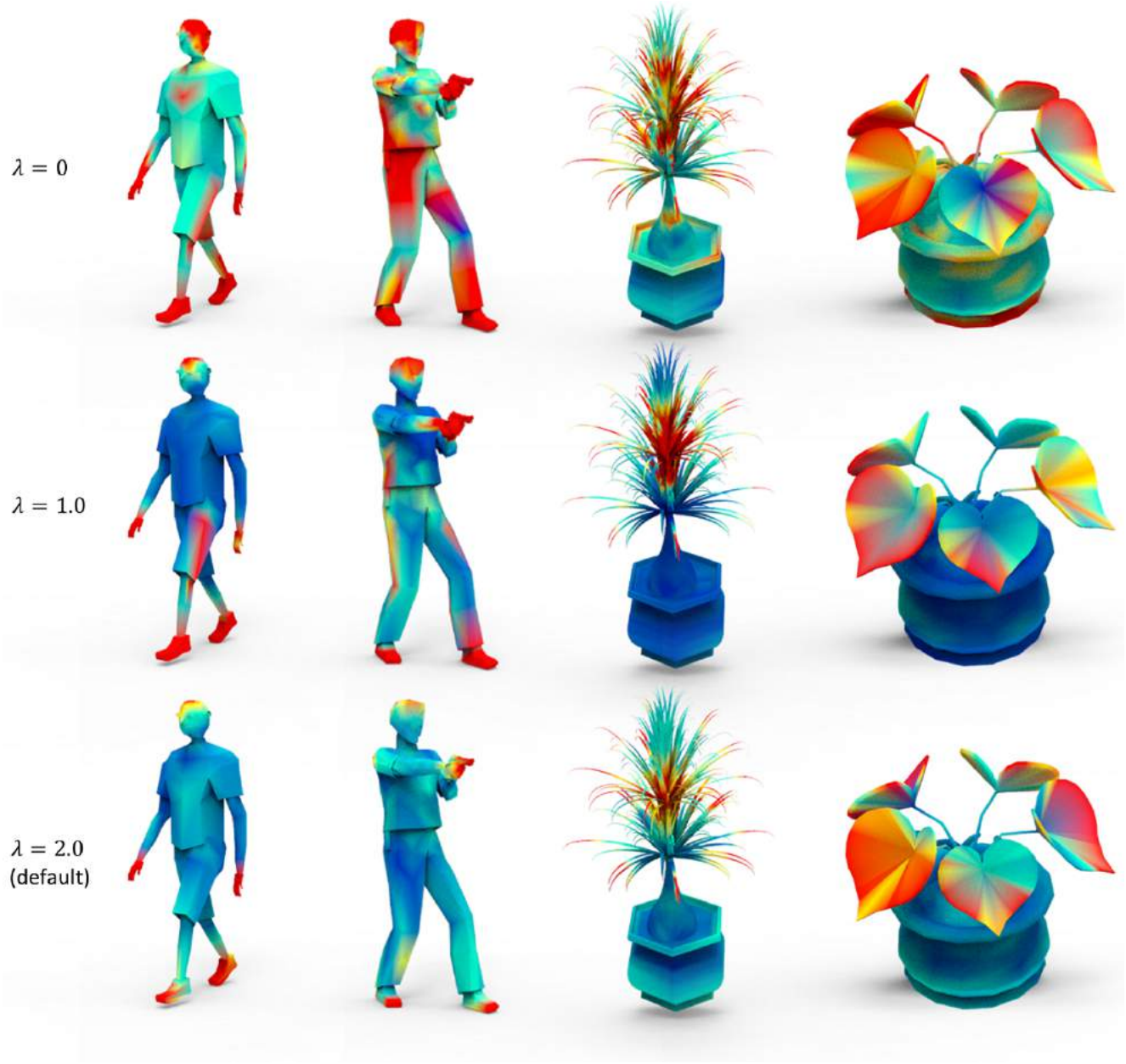

Figure 15. The distinction results when setting different values of λ. For the details of λ, please refer to Eq.(6) in our main paper.

K. Additional Distinctiveness Detection Results based on Random Sampling

Further, we performed a larger scale random sampling over the ModelNet40 test set. For every set of 15 objects in the test set, we randomly picked one object out of the set for distinctiveness detection. Since there are totally 2,468 testing objects in the ModelNet40 test set, we thus picked 166 objects altogether.

Figures 16-20 below show the distinctiveness detection results on these randomly-picked models using the network model trained on the ModelNet40 training set. From the figures, we can see that our method achieves good detections for most shapes.

Figure 16. Additional distinctiveness detection results #1.

Figure 17. Additional distinctiveness detection results #2.

Figure 18. Additional distinctiveness detection results #3.

Figure 19. Additional distinctiveness detection results #4.

Figure 20. Additional distinctiveness detection results #5.

L. Additional Results of using Different Training sets

Figure 21 shows the distinctive regions on the Car shapes (top two rows) and Chair shapes (bottom two rows) detected using different training sets. For the Car shapes, when trained using the inter-class dataset (i.e., the whole ModelNet40), the tires, rearview mirrors, top, and four corners of the car are detected as distinctive, since they are common and unique by comparing the Car category against other categories. However, when trained using the Car intra-class dataset, since all the cars have similar tires and rearview mirrors, hence they are not detected as distinctive. Similarly, for the Chair shapes, when trained using the same inter-class dataset, the legs, armrest, and back regions are detected as distinctive, while, when trained using the Chair intra-class dataset, the back region is no longer distinctive, since the chairs in the dataset mostly have a back.

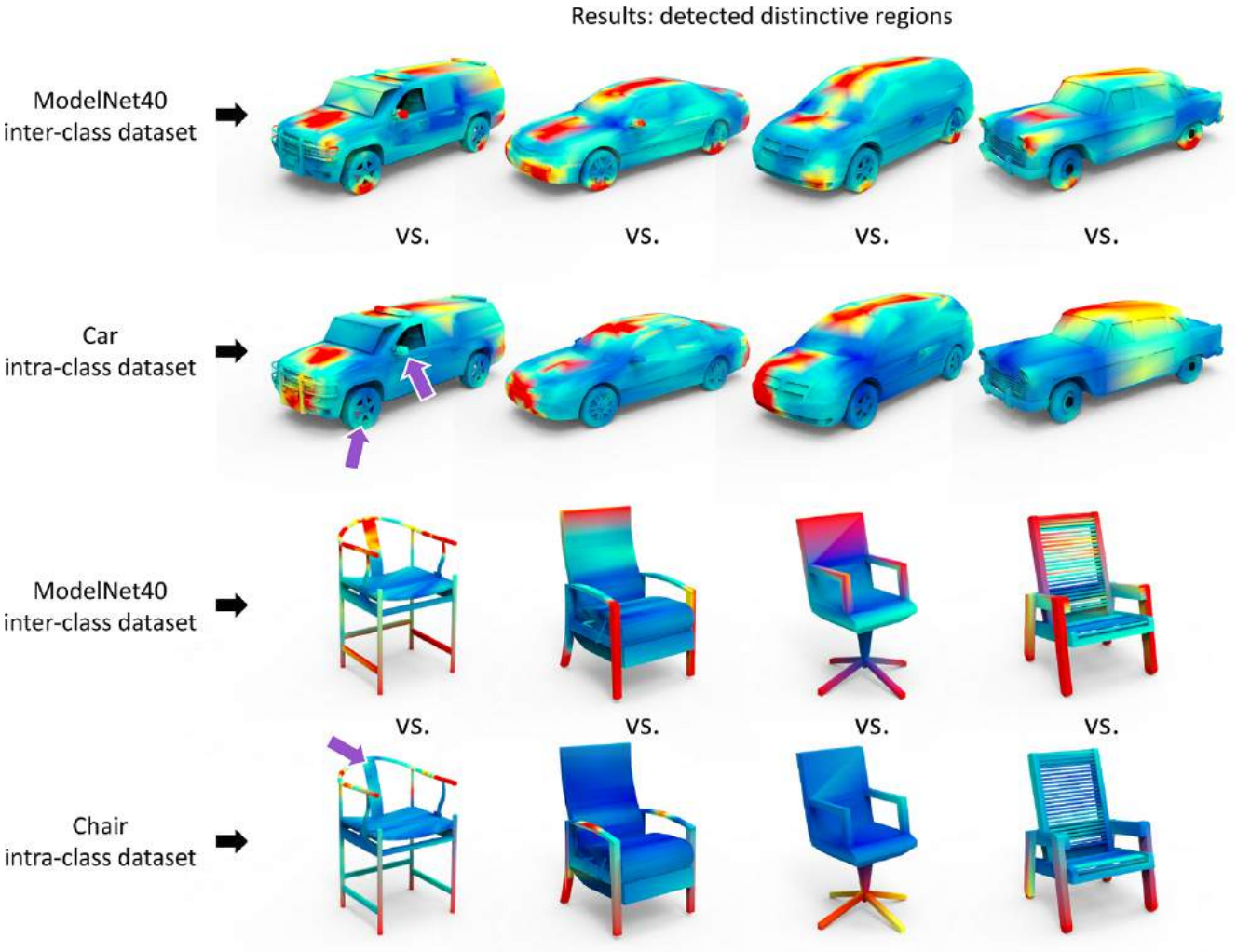

Figure 21. Additional distinctiveness detection results using different training sets.

M. Additional Non-extreme Distinctiveness Detection Results

Figure 22 below shows additional distinctiveness detection results, which are not simply extreme regions. Also, not all extreme regions are detected as distinctive.

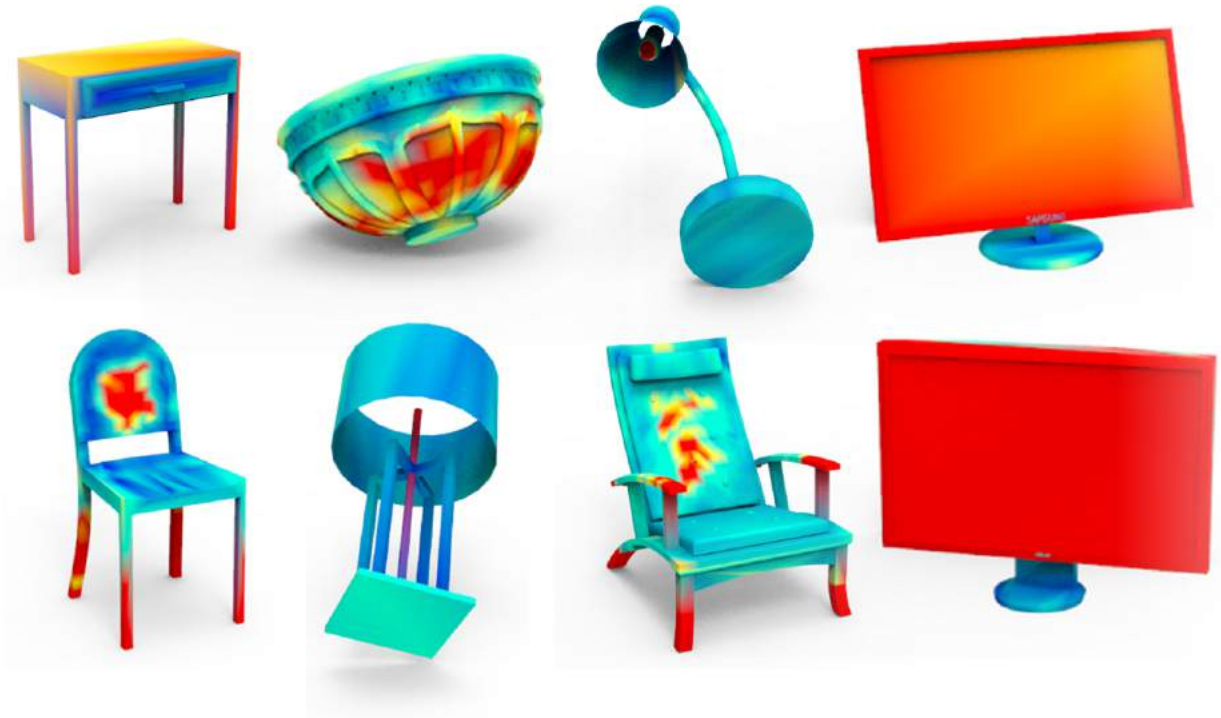

Figure 22. Additional non-extreme distinctiveness detection results.

N. Hierarchical Analysis on our Method

To analyze our method in a hierarchical manner, we gradually simplified our full pipeline to produce the following two versions:

- *version (a)*: we removed the designed attention unit (see Figure 4 in our main paper for details) from the full pipeline; and
- *version (b)*: we further removed the contrastive loss from version (a), so version (b) is the simplest pipeline with only the clustering-based non-parametric softmax loss.

We then separately trained each version using the ModelNet40 dataset. After training, we followed the same quantitative evaluation procedure as in Section 4.2 to evaluate how helpful the distinctive regions detected by each version are to shape classification. Table A below summarizes the overall classification accuracy for each version. From the results, we can see that the accuracy decreases as we gradually remove the major components

Table 2. Comparing the shape classification accuracy on ModelNet40 when downsampling the test shapes with more points on the distinctive regions detected by different versions.

# points	full pipeline	version (a)	version (b)
		remove attention	remove attention + contrastive loss
1024	**0.881**	0.877	0.872
512	**0.874**	0.871	0.865
256	**0.829**	0.806	0.801

References

[1] D. Arthur and S. Vassilvitskii. k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1027–1035. Society for Industrial and Applied Mathematics, 2007. 2

[2] Y. Li, R. Bu, M. Sun, and B. Chen. PointCNN. *Int. Conf. on Neural Information Processing Systems (NIPS)*, 2018. 6

[3] F. Murtagh and P. Legendre. Ward's hierarchical agglomerative clustering method: which algorithms implement Ward's criterion? *Journal of classification*, 31(3):274–295, 2014. 2

[4] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. PointNet: Deep learning on point sets for 3D classification and segmentation. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 652–660, 2017. 6

[5] C. R. Qi, L. Yi, H. Su, and L. J. Guibas. PointNet++: Deep hierarchical feature learning on point sets in a metric space. In *Int. Conf. on Neural Information Processing Systems (NIPS)*, 2017. 6

[6] R. B. Rusu, N. Blodow, and M. Beetz. Fast point feature histograms (FPFH) for 3D registration. In *IEEE Int. Conf. on Robotics and Automation*, pages 3212–3217. IEEE, 2009. 10

[7] X. Y. Stella and J. Shi. Multiclass spectral clustering. In *IEEE Int. Conf. on Computer Vision (ICCV)*, page 313. IEEE, 2003. 2

[8] F. Tombari, S. Salti, and L. Di Stefano. Unique signatures of histograms for local surface description. In *European Conf. on Computer Vision (ECCV)*, pages 356–369. Springer, 2010. 10

[9] Y. Yang, C. Feng, Y. Shen, and D. Tian. FoldingNet: Point cloud auto-encoder via deep grid deformation. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 206–215, 2018. 10