Scaling 3D Compositional Models for Robust Classification and Pose Estimation

Supplementary Material

7. Grouped Neural Vertex with Dynamically Weighted Compositional Contrastive Learning

In this section, we provide further details about our proposed Grouped Neural Vertex with Dynamically Weighted Compositional Contrastive Learning. How our model samples vertex features for the cross-category loss L_{cross} is outlined in subsection 7.1. Additionally, we include the confusion matrix for the Dynamically Weighted Compositional Contrastive method on the calibration dataset in subsection 7.2.

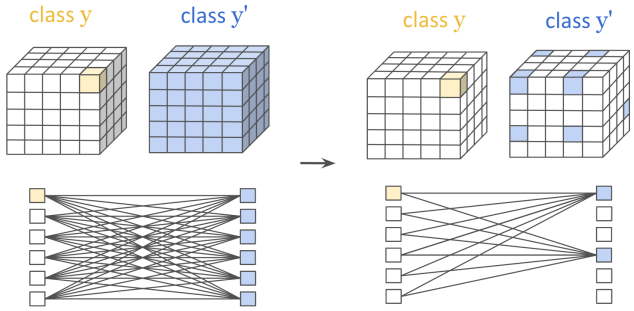


Figure 6. In our grouped cross-category contrasting, we contrast every vertex feature from category y(yellow cube) to only a small subset of the vertex features from each other category y'(blue cube).

7.1. Grouped Neural Vertex Contrasting

As described in subsubsection 3.3.1, we sample a small fixed amount of vertex features $\mathcal{C}_m \sim S(\mathcal{C}_{y'})$ from each category $y' \in Y', Y' = Y \backslash \{y\}$ as negative samples contrasting with vertex feature $\mathcal{C}_k \in \mathcal{C}_y$ of category y. As illustrated in Figure 6, previous contrastive learning in NOVUM conducts a per-vertex contrasting on all the categories. Our Grouped Neural Vertex Contrasting largely reduced the compute by contrast every vertex feature from category y(yellow cube) to only a small subset of the vertex features from each other category y'(blue cubes). The ablation study on how many vertex features from each category y'(blue cubes) are selected can be found in Table 1. We find that 32 are enough for efficient and effective cross-category contrastive learning.

7.2. Dynamically Weighted Compositional Contrasting

We provide the ten most confused categories ranking in the orders of confusion level in Table 5.

Table 5. Most confused categories from the confusion matrix on calibration set.

confusion	True label	Pred label
0.55	jar	pot
0.38	bookshelf	cabinet
0.32	bicycle pump	micrometer
0.32	washing machine	washer
0.26	pencil	pen
0.25	air hammer	power drill
0.25	bumper car	go kart
0.21	bicycle built for two	bicycle
0.21	vending machine	refrigerator

8. Contribution of In-category loss and Cross-category loss

We presents additioanl ablation analysis on the contributions of in-category loss and cross-category loss. Table Table 6 In-category loss focuses on distinguishing between vertices inside an object, thus mainly helping pose estimation by identifying different parts of the object, while the cross-category loss benefits classification because it separates vertices from other object categories.

Table 6. Ablation study of individual loss contributions (accuracy ↑) on in-distribution testing with 188 ImageNet3D categories.

Loss	Classification	Pose estimation
Intra-category only	17.8	56.7
Cross-category only	90.1	1.3
Both (Ours)	**93.5**	**57.6**

9. Error Case Analysis

Through per-category analysis on the IID performance, we found our 3D-compositional model performs less satisfactorily on elongated object classes, see Figure 7 for examples. The reason is that these objects look very similar, and sometimes even identical when facing forward and backwards, left and right, or when rotated along their dominant geometric axis. This ambiguity causes the main difficulty in learning distinct vertex features. Removing the elongated object classes from ImageNet3D+ leads to further improvement by our model. The 10 elongated objects are ”ax”, ”paintbrush”, ”bow”, ”comb”, ”fork”, ”hammer”, ”french horn”, ”knife”, ”pen” and ”pencil”. By removing them from the testing data only, our model performance increases in both classification and 3D pose estimation(see Table 7).

From the confusion matrix we obtained from the testing

set, we found that confusion always appears between visually similar object categories. The most confused categories are ”air hammer”/”power drill” and ”backpack”/”suitcase”, as shown in Figure 8 More details can be found in appendix.

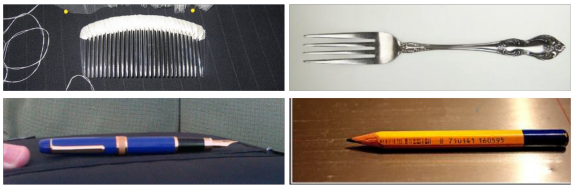


Figure 7. Example images of elongated objects in the ImageNet3D+ dataset. From left to right and top to bottom, the object classes are ”comb”, ”fork”, ”pen” and ”pencil”.

Figure 8. Example images the most confused classes by our model: 25% of ”air hammer” are predicted ”power drill” and 16% of ”backpack” are predicted as ”suitcase”.

Classification

	IID	Occ.	Corr.
All classes	88.2	38.8	57.9
w/o Elongated	**89.3**	**39.7**	**58.5**

3D Pose Estimation

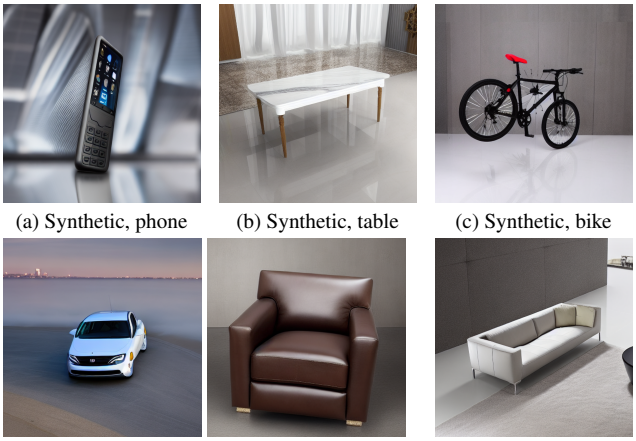
	IID	Occ.	Corr.
All classes	57.6	29.5	45.7
w/o Elongated	**59.3**	**32.8**	**48.3**

Table 7. The classification and pose estimation results by our model on the object classes including and excluding the ten elongated objects. Occlusion and Corruption results are averaged.

10. Visualizations

10.1. Synthetic dataset visualisation

In order to evaluate our method in many different settings, we generated 3D consistent data following [22]. Given some 3D CAD models, we were able to generate data with known objects class an 3D pose annotation. The usage of synthetic data is appealing since it allows to control many parameters during the dataset generation. Benchmark datasets like ImageNet3D can have certain bias (e.g., imbalance in the number of objects per class). Hence, we decided to generate synthetic images to measure our model's capacity to adapt to domain shift (i.e., real-to-synthetic generalization). In order to show the quality of the generated images, we show a subset of the generated data in Fig. 9.



(a) Synthetic, phone (b) Synthetic, table (c) Synthetic, bike

(d) Synthetic, car (e) Synthetic, armchair (f) Synthetic, couch

Figure 9. Visualisation of the generated synthetic data.

(a) Clean image: door (b) L2 Occlusion: fire extinguisher

(c) Fog corrupted image: guitar (d) Pixelate corrupted image: shovel

Figure 10. Qualitative results showing the predictions of our approach for classification and 3D pose estimation

10.2. Qualitative results

We provide a few qualitative results in Fig. 10. We provide an example for the clean images of ImageNet3D+, an example of synthetic occlusion of occluded-ImageNet3D+, and two examples of corrupted images (notably *fog* and *pixelate*). We represent side-by-side the input image along with the input image overlaid by the prediction of our approach. We selected the CAD model of the class that was predicted by our approach and we overlaid the CAD model in the pose predicted by our approach.