Scaling 3D Compositional Models for Robust Classification and Pose Estimation

Xiaoding Yuan[*,†,1] **Guofeng Zhang**[*,1] **Prakhar Kaushik**[*,1] **Artur Jesslen**[2]
Adam Kortylewski[2,3] **Alan Yuille**[1]
[1]Johns Hopkins University [2]University of Freiburg [3]Max Planck Institute for Informatics

Abstract

Deep learning algorithms for object classification and 3D object pose estimation lack robustness to out-of-distribution factors such as synthetic stimuli, changes in weather conditions, and partial occlusion. Recently, a class of Neural Mesh Models have been developed where objects are represented in terms of 3D meshes with learned features at the vertices. These models have shown robustness in small-scale settings, involving 10 objects, but it is unclear that they can be scaled up to 100s of object classes. The main problem is that their training involves contrastive learning among the vertices of all object classes, which scales quadratically with the number of classes. We present a strategy which exploits the compositionality of the objects, i.e. the independence of the feature vectors of the vertices, which greatly reduces the training time while also improving the performance of the algorithms. We first restructure the per-vertex contrastive learning into contrasting within class and between classes. Then we propose a process that dynamically decouples the contrast between classes which are rarely confused, and enhances the contrast between the vertices of classes that are most confused. Our large-scale 3D compositional model not only achieves state-of-the-art performance on the task of predicting classification and pose estimation simultaneously, surpassing Neural Mesh Models and standard DNNs, but is also more robust to out-of-distribution testing including occlusion, weather conditions, synthetic data, and generalization to unknown classes.

1. Introduction

Large-scale training images and annotations have significantly advanced deep learning, leading to remarkable achievements in various computer vision tasks, including object classification, detection, and pose estimation [19]. Cognitive scientists, however, suggest that human vision is more sophisticated and when classifying objects also recognizes their 3D structure including their shape and pose in a unified way using compositional representations [2, 3, 20]. We hypothesize that endowing computer vision models with 3D representations will improve their performance, particularly in challenging out-of-distribution (OOD) scenarios, including domain shifts due to changes in weather, occlusions, and unfamiliar viewpoints, for which humans show big robustness [40], but where standard deep network models struggle [5, 16, 40]. The key insight is that the 3D structure of objects rarely varies in most OOD settings while deep network features are much more variable.

One promising avenue involves neural mesh models [11, 14, 21, 30, 31]. These models are compositional in the sense that they represent objects by 3D meshes of vertices which are associated with learned vertex features. The vertex features are computed by a DNN feature extractor, CNN or Transformer, which are learned to be independent of each other and be invariant to object viewpoint and instance. Using these compositional models, recent works demonstrated superior performance in generalizing to OOD scenarios for tasks such as image classification[11], 3D pose estimation [14, 31] and 6D pose estimation [21].

However, to date, these neural mesh models have only been demonstrated on small datasets, such as Pascal-3D+ [36] (12 object classes) and OOD-CV[38] (10 object classes), for two reasons. Firstly, because they require datasets with accurate 3D annotations for learning. Secondly, because their learning algorithms scale badly. For example, their contrastive learning includes every vertex from every object class which scales quadratically.

This raises the challenges we address in this work: (I) Can neural mesh models be scaled to a large number of object classes *efficiently*? (II) How will they perform compared to conventional neural networks in independent and identically distributed (IID) testing? (III) Most importantly, will they retain their important robustness properties, e.g., robustness to out-of-distribution (OOD) data, when scaled up?

In this work, we reformulate neural mesh models to allow scaling up to a large number (i.e., 188) of object classes efficiently exploiting the recent availability of 3D annotated data [22, 23], We will refer to it as 3D compositional models in the following sections.

[*]Equal contributions. † Lead author.

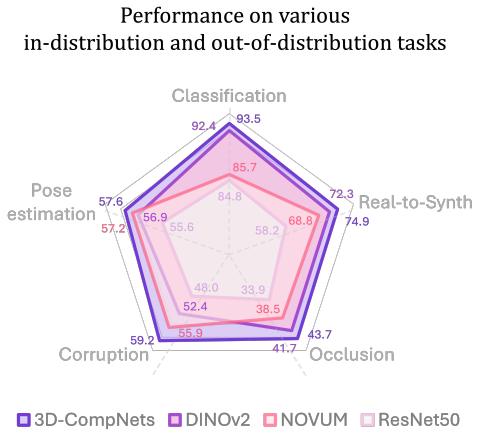


Figure 1. Our model (3D-ComptNets) is able to perform classification and 3D pose estimation simultaneously while being robust to IID and various OOD scenarios (different scales for each axis).

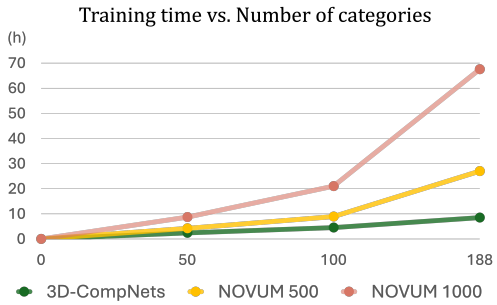


Figure 2. The NOVUM method typically learns an average of 1000 vertex features per class, while both 3D-CompNets and NOVUM 500 learn only a subset of 500 vertex features. Moreover, with our efficient designs, 3D-CompNets requires 7 times less training time compared to the original NOVUM.

Our strategy is to train 3D compositional models with a new algorithm that exploits the compositionality of our objects in terms of their vertices. We demonstrated that only a small portion of the huge number of contrastive pairs is required to optimize the model to achieve strong performance. We first decoupled the full vertex-level contrasting into in-class contrasting and cross-class contrasting for efficiency. Additionally, our algorithm dynamically decouples the contrast between classes that are rarely confused and emphasizes the contrast between the most confused classes. This is similar to classic hard-negative mining [12, 33, 35], differing in that we exploit the compositional structure of our models in a supervised learning manner. These together greatly reduce the number of vertices of the object that need to be contrasted, allowing for a greatly reduced computation.

Concisely, our contributions are as follows:

1. We extend 3D-CompNets to an order of magnitude more object classes than previous studies and show they outperform conventional deep networks for both object classification and 3D pose estimation in a unified manner. By comparison, previous studies of 3D-CompNets showed no improvement over conventional deep networks on IID data.
2. We refactor the per-vertex contrastive learning in 3D-CompNets into two levels: in-class and cross-class contrasting, to largely improve learning efficiency.
3. We advance the inter-class contrastive by dynamically decomposing object classes into subgroups and apply dynamic weights on the contrastive loss between classes, enabling more efficient and effective model optimization.
4. We further demonstrate that our model shows robust generalization capabilities on OOD data including occlusion, image corruption, real-to-synthetic and unknown categories.

2. Related Work

Robust Classification and Pose Estimation. Deep Networks have been shown to be non-robust [18, 26, 27] to simple nuisances in tasks like image classification [8–10] and 3D pose estimation [38]. Nuisances like partial occlusion, weather, additive noise, etc. may not have much effect on human visual capabilities however can completely derail deep neural networks outputs. A convincing theory attributes this fragility to lack of 3D compositional knowledge in these models which humans possess [14]. Methods like data augmentations, test time adaptation, noise addition, input masking, etc. have been proposed to make neural models more robust with varying but unsatisfactory levels of success with many arguing that we would need a different architectural approach may be required [17, 32] which incorporates some 3D object knowledge in the models.

Robust Neural Compositional Models. It refers to a family of 2D [13, 17, 32] and 3D models [11, 14, 31] who have shown to be robust to out-of-distribution nuisances like partial occlusion [17, 30], real and synthetic corruptions [11, 13, 14] relative to conventional deep networks and have been utilized to perform robust image classification [11, 17], 3D and 6D pose estimation [21, 30, 31], amodal segmentation [28] and unsupervised domain adaptation [13, 14]. These models focus on learning object-centric, compositional neural representations and often employ the ideas of analysis-by-synthesis [37] in their applications. However, all of these previous works have only been shown to work on small-scale datasets often due to the computationally expensive nature of learning these compositional, object-centric representations. In this work, we build upon ideas introduced by this family of models and scale them up efficiently to work with large datasets.

Contrastive Learning. Contrastive learning was orig-

inally developed for supervised learning [4, 6, 15, 24, 34] but has made its biggest impact when it was modified and applied to self-supervised learning [7] giving state of the art results for many applications. Researchers have tried to adapt the idea of hard-negative mining [12, 33, 35] to improve performance and to improve efficiency but for unsupervised or self-supervised contrastive learning the lack of supervision makes it infeasible to adopt existing negative sampling strategies and motivates the development of other strategies [25]. Although these strategies can be effective they are not always efficient because finding these hard samples takes time. Our approach differs in two respects. Firstly, it is supervised so it is easier to define a hard negative (e.g., two objects that are easily confused with each other). Secondly, we approximate the representation of objects by factorized vertex features on the 3D mesh and we can directly apply contrastive learning on the vertex level with hard negative mining.

3. Method

In this section, we first introduce the prerequisites of 3D Compositional Networks (3D-CompNet) in subsection 3.1, including network architecture and the object-centric 3D neural representation. Then, we explain our motivation and problem statement for scaling up the 3D Compositional Networks in subsection 3.2. We then present our core designs, grouped neural vertex and dynamically weighted compositional contrastive learning (subsection 3.3), to achieve *efficient* scaling up of 3D-CompNet, that can be used to perform robust image classification and 3D pose estimation simultaneously (subsection 3.4).

3.1. Prerequisites: 3D Compositional Network

Our model 3D-CompNet is inspired by and improved from recent advances in Neural Mesh Models [11, 30]. In the following, we describe the individual components of the model in detail.

A 3D Mesh Composed of Vertex Features. We represent objects as a 3D mesh composed of vertices uniformly placed on the surface geometries of distinct object categories. Each vertex also stores a corresponding feature vector, which we refer to as *vertex features*. For the task of classification and pose estimation, we find that cuboid geometries for the mesh suffice [11, 13, 30] but more tightly defined geometries [31] can also be used if available. Each vertex feature is linked to a feature vector $C_k \in \mathbb{R}^D$. We define the feature set for each category y as $\mathcal{C}_y = \{C_k \in \mathbb{R}^D\}_{k=1}^K$, and the collective set across all categories and levels as $\mathcal{C} = \{\mathcal{C}_y\}_{y=1}^Y$, where Y is the number of categories. The remainder of the image that is not covered by the rendered object volume is represented as background features $\mathcal{B} = \{\beta_n \in R^D\}_{n=1}^{N_b}$ where N_b is a fixed hyperparameter and $\mathcal{B}$ is shared across all categories.

Feature Extractor. The second key component of our model is a feature extractor Φ_w, with parameters w, that processes an input image I into a feature map $F = \Phi_w(I) \in \mathbb{R}^{D\times H\times W}$. This map holds feature vectors $f_i \in \mathbb{R}^D$ at each 2D lattice position i. During training, the feature extractor and 3D representation are trained jointly. In particular, our model learns a vertex feature C_k by optimizing its similarity to corresponding image features f_i, given the camera pose α. We establish a simplified one-to-one correspondence of the vertex feature C_k to the closest 2D image feature f_i for each image. For clarity, f_k denotes the 2D image feature corresponding to vertex feature C_k.

Probabilistic model. We model the probability of generating the feature f_k from vertex feature C_k as $P(f_k|C_k) = c_M(\kappa)e^{\kappa f_k \cdot C_k}$, with C_k as the mean of each von Mises–Fisher(vMF) distribution, both f_k and C_k are unit vectors. Similarly, the probability of f_k from background features β_n is $P(f_k|\beta_n) = c_M(\kappa)e^{\kappa f_k \cdot \beta_n}$, where $\beta_n \in \mathcal{B}$. We define the concentration parameter κ, a measure of the spread of the distribution, as a global hyperparameter, allowing us to disregard the normalization constant $c_M(\kappa)$ during learning and inference.

To perform inference, we define a binary valued parameter $z_{i,k}$ such that $z_{i,k} = 1$ if the feature vector f_i matches best to any Gaussian feature $\{C_k\} \in \mathcal{C}_y$, and $z_{i,k} = 0$ if it matches best to a background feature. The object likelihood of the extracted feature map $F = \mathbf{\Phi}_w(I)$ can then be computed as:

$$\prod_{f_i \in F} P(f_i|z_{i,k}) = \prod_{f_i \in F} P(f_i|C_k)^{z_{i,k}} \cdot \max_{\beta_n \in \mathcal{B}} P(f_i|\beta_n)^{1-z_{i,k}}. \tag{1}$$

During training and inference, we aim to maximize this score w.r.t. the latent variables.

Training. Similarly to previous approaches [11, 14, 30], we maximize the probability $P(f_k|C_k)$ that any extracted feature f_k was generated from a vertex feature C_k instead of any other alternatives. This is done using a supervised contrastive learning formulation such that the likelihood that an extracted feature f_k is generated by the correct vertex feature C_k is maximized [11] w.r.t (I) distanced vertex features of the same object (II) vertex features of other object classes (III) background features.

3.2. Motivation and Problem Statement

Previous methods [11, 31] learned the *vertex features* by mapping the image feature at each 2D location from a feature extractor to a corresponding vertex in the 3D representation of the object given its 3D pose. The 3D representation for each object class is either in a coarse shape like a cuboid or in an average prototypical shape. During training, the feature extractor is updated using *contrastive loss* between vertex features which ensures that every vertex feature is distinct from one another.

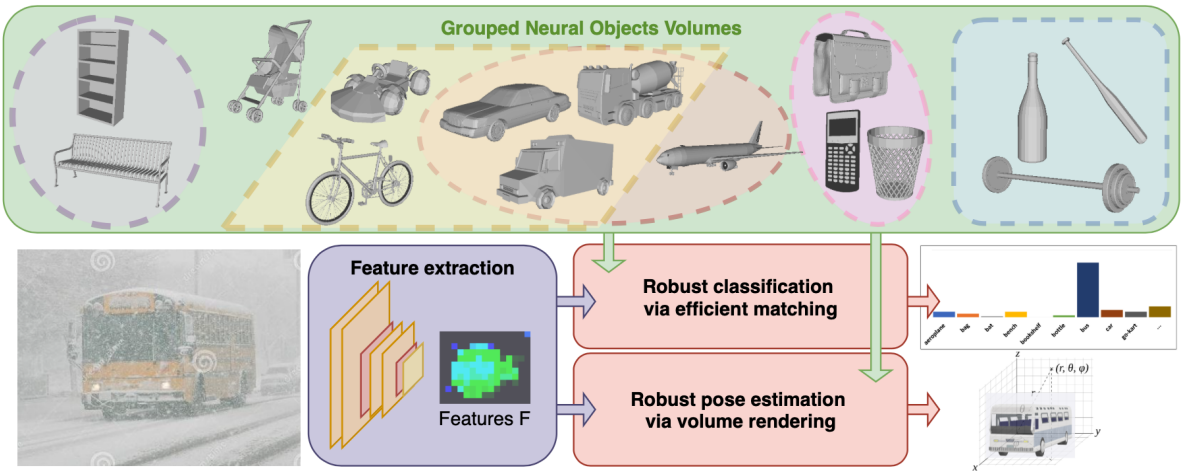


Figure 3. An overview of 3D-CompNet. The top-green box represents the variety of objects (i.e., 188 classes) that we consider and illustrates the grouping of Neural Vertex Features (NVF) (with arbitrary groups for illustrative purposes). The lower part illustrates the inference pipeline. During inference, an image is first processed by the backbone into a feature map F (purple box). Then, by efficiently matching features from F and our NVF, the object class can be predicted (top-red box), or alternatively, given the class label, pose estimation can be performed by leveraging our volumetric representation in a render-and-compare manner (bottom-red box).

However, as we scale up to a large number of object classes Y, we have to learn a large number of these compositional vertex representations. This is problematic for a few reasons:

1. We need to contrast every vertex feature with every other vertex feature of the same object class as well as all other object classes. The calculation/floating point operations grow with a complexity of $\mathcal{O}(Y^2)$.
2. During training, model optimization (using the contrastive loss) becomes more complex as we increase the number of objects, due to the drastic increase in the number of vertex features.
3. During inference, we have to evaluate our data samples against the vertex features of all other classes and an incorrect classification inference may lead to incorrect pose inference.

3.3. Grouped Neural Vertex with Dynamically Weighted Compositional Contrastive Learning

In a departure from previous works, we train vertex features in a *grouped* manner, what we refer to as **Grouped Neural Vertex with Dynamically Weighted Compositional Contrastive Learning**.

We discover that only a small fraction of the vertex feature pairs are necessary for the contrastive learning. We propose to decouple the full per-vertex contrasting into in-class contrasting group and cross-class contrasting group. The cross-class contrasting only happens between a small amount of sampled vertex features from each object class thus largely improving the learning efficiency.

Additionally, we apply dynamic weights on the cross-category contrastive loss in a hard negative mining manner to make the training process even more efficient.

This leads to a sparse and therefore much more efficient contrastive loss calculation as we do not calculate any corresponding loss terms between most vertex feature pairs. This contrastive loss formulation is termed *compositional* since every vertex feature is composed of individual volume features which roughly correspond to object keypoints. Our grouped formulation helps us to ameliorate the drawbacks mentioned in the previous subsection 3.2. The **advantages** include

1. 95% reduction in the number of floating point operations for every contrastive loss calculation as we only calculate the distance between *uniformly-sampled* volume feature pairs of categories with *non-zero* weights.
2. Faster and easier contrastive loss optimization leading to better accuracy.

3.3.1. Grouped Neural Vertex Contrasting

If we try to trivially scale this loss to Y classes, the number of contrastive terms scales by a quadratic (Y^2) factor. In addition, the loss landscape for optimizing over these many parameters further lengthens and complicates the training process. However, we hypothesize that not all of these contrastive loss terms are necessary and that we can make learning more effective by focusing on the most confused vertex feature pairs.

To reduce the number of contrastive pairs, we first decouple the loss into an in-category loss L_{in} and a cross-category loss L_{cross}. For each vertex feature C_k, we formulate these two loss terms as:

Models	N	GFLOPS ↓	Time ↓	Accuracy
NOVUM*	full	61.2	66.8h	85.7
3D-CompNets	64	4.31 (-93%)	13.3h	86.3
3D-CompNets	**32**	**2.71 (-96%)**	**9.3h**	**86.5**

Table 1. Efficiency Improvement: Our scalable 3D-CompNet achieves up to 96% reduction in GFLOPs for contrastive loss computation, and over 7× faster training speed. N refers to the sampled features for cross-category loss computation. All backbones are ResNet50. Bold row indicates our final model configuration.
* For full convergence, NOVUM requires 100 epochs instead of 12.

Models	Backbone	IID	Synthetic
ResNet50	resnet50	84.8	58.2
NOVUM	resnet50	85.7	68.8
3D-CompNets	resnet50	86.5	69.3
DINOv2	vit-b-14	92.4	72.3
3D-CompNets	**vit-b-14**	**93.5**	**74.9**

Table 2. Classification Results on Imagenet3D+. We evaluate our proposed 3D-CompNets with different backbone feature extractors. Bold row indicates our final model configuration.

$$L_{in}(k) = \frac{P(f_k|C_k)}{\sum_{C_l \in \mathcal{C}_y} P(f_k|C_l) + \sum_{\beta_n \in \mathcal{B}} P(f_k|\beta_n)} \tag{2}$$

$$L_{cross}(k) = \frac{P(f_k|C_k)}{P(f_k|C_k) + \sum_{y' \in Y'} \sum_{C_m \sim S(\mathcal{C}_{y'})} P(f_k|C_m)}, \tag{3}$$

where $Y' = Y \backslash \{y\}$ refers to the categories excluding the image class y.

The in-category loss L_{in} contrasts every vertex feature $C_k \in \mathcal{C}_y$ with every other vertex feature $\{C_l\} \in \mathcal{C}_y$ of the same category y and the background features $\{\beta_n\} \in \mathcal{B}$.

The cross-category loss L_{cross}, on the other hand, contrasts a vertex feature $C_k \in \mathcal{C}_y$ with a set of vertex features $\{C_m\} \in \mathcal{C}'$, where $\mathcal{C}' = \mathcal{C} \backslash \mathcal{C}_y$, that belongs to all other categories. In the cross-category loss, we uniformly sample a small fixed amount N of vertex features $C_m \sim S(\mathcal{C}_{y'})$ from each category y'. Experiments in Table 1 ablates the number of vertex features sampled from each category and we find $N = 32$ is enough for the cross-category contrasting and shows the best classification performance. Figure 2 shows the training time reduction compared to the original NOVUM model and the trend when scaling up with the number of categories.

3.3.2. Dynamically Weighted Compositional Contrasting

To further improve the learning efficiency, we propose a dynamically weighted cross-category contrastive learning.

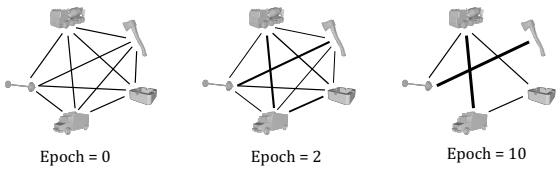


Figure 4. Illustrative example of our Dynamically Weighted Contrastive Learning. Weights applied on the cross-category contrastive loss terms change dynamically during training. They are determined by the confusion matrix on the calibration data split every two epochs.

Every 2 epochs, we validate the model's performance on held-out calibration data. The confusion matrix is normalized over the groundtruth dimension (image class y). Using the confusion matrix from the calibration data split, we weigh the pairwise cross-category contrastive loss term between category y and category y' by the confusion level between y and y', $\omega_{y,y'}$, where $\omega_{y,y'} \in [0, 1]$. Weight $\omega_{y,y'}$ is set to 0 when the confusion level is below 0.05 between object classes y and y', which means we don't calculate the contrastive loss between these classes anymore. This weighting changes dynamically throughout the training, and in the end, will be sparse with the majority of the vertex feature pairs not being *grouped* together (i.e., 0 weight), as illustrated in Figure 4. Note that the weights in Figure 4 are shown at the category level, but in practice, they are applied at the vertex level, since our model learns through vertex-level contrastive learning.

We formulate the new dynamically weighted cross-category loss as follows:

$$L_{cross}(k) = \frac{P(f_k|C_k)}{P(f_k|C_k) + \sum_{y' \in Y'} \omega_{y,y'} \sum_{C_m \sim S(\mathcal{C}_{y'})} P(f_k|C_m)}, \tag{4}$$

where $\omega_{y,y'}$ is the *grouping weight* which is calculated using the confusion matrix between object categories of the calibration dataset.

We compute the final loss $\mathcal{L}(\mathcal{C}, \mathcal{B})$ for each training example by taking the neg-logarithm and summing over all sets of features $\{f_k\}$ as:

$$\begin{aligned} \mathcal{L} &= \mathcal{L}_{in} + \mathcal{L}_{cross} \\ &= -\sum_{k}^{K} o_k \cdot (\log \frac{e^{\kappa f_k \cdot C_k}}{\sum_{C_l \in \mathcal{C}_y} e^{\kappa f_k \cdot C_l} + \sum_{\beta_n \in \mathcal{B}} e^{\kappa f_k \cdot \beta_n}} \\ &+ \log \frac{e^{\kappa f_k \cdot C_k}}{e^{\kappa f_k \cdot C_k} + \sum_{y' \in Y'} \omega_{y,y'} \sum_{C_m \sim S(\mathcal{C}_{y'})} e^{\kappa f_k \cdot C_m}}), \end{aligned} \tag{5}$$

Models	tricycle	unicycle	laptop	go kart	RV	suitcase	teapot	filing cabinet	sofa	projector	**Average**
ResNet50	17.1	5.2	15.5	6.1	20.5	2.9	32.4	11.6	11.4	23.1	15.0
DINOv2	29.6	9.0	18.2	51.5	41.7	17.5	36.7	**75.3**	**59.6**	41.2	38.9
3D-CompNets	**42.5**	**30.1**	**51.7**	**69.6**	**46.1**	**47.6**	**44.0**	62.6	32.9	**46.4**	**45.5**

Table 3. Generalization performance of pose estimation on unseen object categories under accuracy $\pi/6$ $\uparrow$. Both DINOv2 and our 3D-CompNets have a ViT-B/14 backbone with the same pretraining but with different object representation methods and learning objectives.

where $o_k = 1$ if the vertex is visible in the image and $o_k = 0$ otherwise.

Updating vertex features and Background Features. The vertex features and background features $\mathcal{C}$ and $\mathcal{B}$ are updated after every gradient update of the feature extractor. Following [1, 6], we use momentum update for the vertex features:

$$C_k \leftarrow C_k \cdot \sigma + f_k \cdot (1 - \sigma), \quad \|C_k\| = 1. \tag{6}$$

The background features are simply resampled from the newest batch of training images. In particular, we remove the oldest features in $\mathcal{B}$, i.e. $\mathcal{B} = \{\beta_n\}_{n=1}^{N} \setminus \{\beta_n\}_{n=1}^{T}$. Next, we sample T new background features f_b from the feature map, ensuring f_b is not influenced by any vertex feature, and update $\mathcal{B}$ as $\mathcal{B} \leftarrow \mathcal{B} \cup \{f_b\}$. Note that σ and T are model hyperparameters.

3.4. Inference of Class Label and 3D Pose

Fast Robust Classification. Image classification is performed swiftly and robustly by matching extracted features to learned vertex features of all vertex features and background features. For each category y, we compute both foreground $P(f_i|\mathcal{C}_y)$ and background $P(f_i|\mathcal{B})$ likelihoods across all lattice locations i on the feature map. Ignoring object geometry simplifies this to a fast convolution operation. Image classification involves comparing average total likelihood scores across all locations for each class.

As described in subsection 3.3, the extracted features follow a vMF distribution. Thus we define the final classification score of an object class y as:

$$S_y = \sum_{f_i \in F} \max\{\max_{C_k \in \mathcal{C}_y} f_i \cdot C_k, \max_{\beta_n \in \mathcal{B}} f_i \cdot \beta_n\}. \tag{7}$$

The final category prediction is $\hat{y} = argmax_{y \in Y}\{S_y\}$.

Volume Rendering for Pose Estimation. Given the predicted object category $\hat{y}$, we use the vertex feature $\mathcal{C}_{\hat{y}}$ to estimate the camera pose α leveraging the 3D geometrical information of the neural object volumes. Following the vMF distribution, we optimize our pose prediction α via feature reconstruction loss [11, 14, 21, 31] during inference:

$$\mathcal{L}(\alpha) = -\sum_{f_i \in FG} f_i \cdot \hat{\boldsymbol{C}}_i(\alpha) - \sum_{f_b \in BG} \max_{\beta_n \in \mathcal{B}} f_b \cdot \beta_n, \tag{8}$$

where FG is the set of foreground features that are covered by the rendered neural object, i.e. those features for which the aggregated volume density is bigger than a threshold $FG = \{f_i \in F, \sum_{k=1}^{K} \rho_k(\boldsymbol{r}_\alpha(t)) > \theta\}$. $BG = F \setminus FG$ is the set of features in the background. 144 evenly spaced (12 for azimuth, 4 for elevation, 3 for theta) candidate poses in the 3D space are predefined as a set of initial poses. Pose estimation starts from the optimal initial pose through computation of the reconstruction loss (Equation 8) across predefined poses, followed by gradient-based optimization to determine the final pose prediction $\hat{\alpha}$.

4. Experimental Details

4.1. Datasets

We use two different types of data in our experiments, notably real and synthetic data.

Real Data We train and evaluate our method on real data using the ImageNet3D dataset [23], a large dataset for 3D understanding with class and 6D pose annotation. We selected a total of 188 classes with enough images for a total of 61 230 images divided in 30 630 training images and 30 600 test images. We then create occluded-Imagenet3D following [29] by placing occlusion on both object and background in three levels: L1, L2, and L3. In L1, around 10% of the object and 30% of the background will be occluded, and 30%, 50% for L2 and 50%, 70% for L3. We also test on corruptions following [8] for 4 kinds of common types of corruptions in natural environment on level 4.

Synthetic Data For out-of-distribution testing, we also test our method on synthetic data generated following the approach outlined by [22]. This method enables precise 3D geometry control of diffusion models, allowing us to obtain detailed 3D annotations for the generated images. We generate the synthetic data for a subset of the object classes that exist in our real dataset. Hence, we have 50 synthetic classes and 500 images for each class. We included visualizations of the generated synthetic data in the appendix.

4.2. Baselines

We compare the performance of our approach to 2 competitive standard baseline methods, ResNet50 and DINOv2 Vit-B-14 model, for classification and 3D pose estimation. To perform multiple tasks, ResNet50 and DINOv2 are trained with a dual regression head: one for classification and one

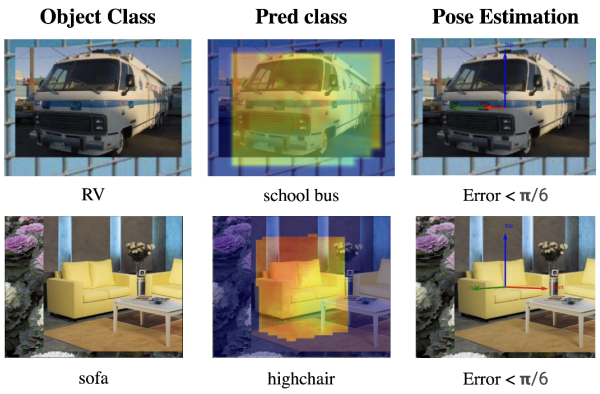


Figure 5. Examples of generalization performance of our approach on *unknown* object categories. The second column shows the feature activations using the vertex features from the predicted class. The third column show the 3D pose estimation result using the predicted class.

for pose estimation. The classification head has an output size of the number of object classes. Pose estimation produces an output size of 3, representing elevation, in-place rotation, and azimuth, respectively. Each baseline uses standard cross-entropy loss and is optimized to best performance on the validation set.

4.3. Hyper-parameter choices

The input image size is 640×800 for ResNet50 backbone and there are two upsampling layers to integrate the output from the last three layers of the feature extractor. The size of the feature map F is $1/8^{th}$ of the input size. For the ViT-B-14 backbone, the input image size is 644×812 and the output feature map F is $1/14^{th}$ of the input size. All output features are projected to a dimension of $D = 128$.

Our method is trained as described in subsection 3.3. For each class, the corresponding vertex feature is composed of approximately $K = 500$ vertex features for each object class. To model the background, we use $N = 2560$ background features. We use momentum update for the vertex features using $\sigma = 0.9$ and sample $T = 5$ new background features from the image background to update $\mathcal{B}$ at each gradient step. Our model converges after only 12 epochs.

4.4. Evaluation

We evaluate all methods on two different tasks: image classification and 3D pose estimation. Image classification consists of estimating the object category of the main object in the image. The classification performance is evaluated by accuracy over all the object categories. The 3D pose estimation task requires predicting the azimuth, elevation, and in-plane rotation of an object to a fixed camera. The pose estimation error is calculated between the predicted rotation matrix R_{p} and the ground truth rotation matrix R_{gt} as $e = \left\| \log m \left(R_{\text{p}}^T R_{\text{gt}} \right) \right\|_F / \sqrt{2}$, following [39]. We define the accuracy of 3D pose estimation using a threshold where a prediction is considered correct if $e < \frac{\pi}{6}$.

5. Results

In this section, we demonstrate the largely improved efficiency and effectiveness of our approach (subsection 5.1) and evaluate our approach and baselines on classification and 3D pose estimation using real-world data in in-distribution (subsection 5.2) when scaling up with the number of objects. Additionally, we show the generalization ability of our approach testing on unknown object categories (subsection 5.3). Finally, we provide results testing on out-of-distribution (OOD) scenarios including occlusion, image corruption and generalization to large-scale synthetic-to-real data (subsection 5.4). Figure 1 demonstrates the overall performance of 3D-CompNets across multiple tasks in both in-domain and out-of-distribution scenarios.

5.1. Training Time Efficiency

We report quantitative results about the drastic decrease in loss computations and the training time by our model in Table 1. Our model uses 96% less loss FLOPS and converges 7 times faster than NOVUM, but still outperforms it and other standard neural networks thanks to our simple yet novel training methodology changes. Particularly, our model can converge with only 12 training epochs, and we can outperform NOVUM performances which only converges after 100 epochs.

Also, the training time of our model increases approximately linearly with the number of categories Y, while the original NOVUM scales quadratically. We compared our model with two NOVUM settings: NOVUM with 1000 vertices and with 500 vertices per category. We report the training time for each model to best converge on different numbers of categories in Figure 2. Considering more 3D data available in the future, an algorithm that scales up linearly is crucial both theoretically and practically.

5.2. Classification and 3D Pose Estimation

Table 2 and Table 4 show classification and 3D pose estimation performance on the base Imagenet3D dataset [23], synthetic data generated using [22], its corrupted version using corruptions like fog, snow, etc. from the Imagenet-C dataset [8], and partial occlusion with levels ranging from $20 - 80\%$. All our baselines have 3D information incorporated in them during training. NOVUM [11] is our ablative baseline, which is learned without our *Grouped neural Vertex with Dynamically weighted Compositional contrastive Learning*. All model performances reported here are trained till full convergence. In Table 2, we show comparisons of classification task between our 3D-aware model and the same backbones with standard classification heads. Our

	Model	Backbone	IID	Occlusion				Corruption				
				L1	L2	L3	**Average**	brightness	frost	snow	fog	**Average**
Class.	Resnet50	resnet50	84.8	58.8	34.7	11.2	33.9	71.4	37.5	19.2	63.9	48.0
	NOVUM	resnet50	85.7	64.6	37.6	13.4	38.5	75.1	46.1	30.1	72.6	55.9
	DINOv2	vit-b-14	92.4	68.3	40.4	16.5	41.7	71.0	49.6	22.9	66.2	52.4
	3D-CompNets	vit-b-14	**93.5**	**69.3**	**42.8**	**19.1**	**43.7**	**82.6**	**50.2**	**30.4**	**73.6**	**59.2**
Pose Est.	Resnet50	resnet50	55.6	40.4	27.5	14.4	27.4	50.8	29.1	38.7	51.3	42.5
	NOVUM	resnet50	57.2	42.6	28.8	15.6	29.0	51.9	32.5	41.0	52.7	44.5
	DINOv2	vit-b-14	56.9	42.7	28.2	15.7	28.9	52.2	30.4	40.9	51.6	43.8
	3D-CompNets	vit-b-14	**57.6**	**43.4**	**29.2**	**16.0**	**29.5**	**54.1**	**32.7**	**42.2**	**53.9**	**45.7**

Table 4. Classification and 3D pose estimation on clean(IID), occluded, and corrupted ImageNet3D+ dataset. Different occlusion levels (L1, L2, L3) and different corruption types applied. 3D pose estimation results are reported under accuracy $\pi/6 \uparrow$. Our approach outperforms standard neural network models and the baseline NMMs(NOVUM) in both in-distribution and out-of-distribution testing.

model outperforms both the standard classification DNNs by 1.7% - 2.0% and the NOVUM baseline by 2.5% under the IID testing. Moreover, in Table 4, our model also shows the strongest performance on 3D pose estimation.

5.3. Generalization to Unknown Categories

In this section, we report the pose estimation performance on unknown object categories to demonstrate our model's zero-shot generalization ability. We trained our proposed 3D-CompNets and the baseline models only on 178 object categories from the ImageNet3D dataset and tested pose estimation on the other 10 unknown categories.

To perform pose estimation on unknown categories, our approach uses the category $\hat{y}$ with the lowest feature reconstruction loss as described in subsection 3.4 as the predicted category and its corresponding vertex features $\mathcal{C}_{\hat{y}}$ to estimate the camera pose. ResNet and ViT baselines use their pose estimation heads to output the pose prediction directly. Table 3 shows the generalization performance of pose estimation when testing on unknown categories. Figure 5 visualizes the feature reconstruction and pose estimation. We also found that our model tends to predict 3D poses using seen categories that are visually similar to the unknown category, e.g., 68% of the unknown category "teapot" uses "kettle" features and 95% of the unknown category "unicycle" use "bicycle" features for 3D pose estimation.

5.4. Domain Shift

We also test the generalization ability to out-of-distribution data. During testing only, we created occlusion data by randomly covering the original testing images with out-of-interest objects are occluders. We also applied different types of natural corruption to the original testing images. Table 4 shows that our method outperforms all other baselines with a large margin on both classification and 3D pose estimation under occlusion and image corruption. We also report real-to-synthetic generalization performances for classification in Table 2. We demonstrate our neural vertex features are strongly robust to various OOD scenarios under drastic domain shifts, including occlusion, unusual weather environments and domain shifts from real to synthetic.

6. Conclusion and Discussion

In this work, we argue that endowing computer vision object models with 3D representations will improve their performance, particularly in challenging out-of-distribution (OOD) scenarios. To demonstrate this, we scaled up 3D-CompNets to 188 object categories taking advantage of a recent dataset with 3D annotation. We design *Grouped neural Vertex with Dynamically weighted Compositional contrastive Learning* (GVDComp) to greatly increase the learning speed and improved performance, resulting in an effective and efficient scaling with the number of object classes. Note that 3D-CompNets is trained to distinguish between vertex features, but outperformed the traditional DNN design testing on object classification and 3D pose estimation simultaneously, in both IID and challenging OOD scenarios.

Note that we use the cuboid model as a rough approximation to represent objects across different shapes. Factorization on the vertex features also simplifies the representation and learning. Excitingly, we already achieved good results with all those approximations. We believe in future works, replacing the cuboid model and the factorized vertices can help lead to an even better object representation.

Acknowledgements

AK acknowledges support via his Emmy Noether Research Group funded by the German Research Foundation (DFG) under Grant No. 468670075. AY acknowledges support from ARL award W911NF2320008, ONR: N00014-21-1-2690 and National Eye Institute (NEI) with Award ID: R01EY037193.

References

[1] Yutong Bai, Angtian Wang, Adam Kortylewski, and Alan Yuille. Coke: Localized contrastive learning for robust keypoint detection. In *WACV*, 2023. 6

[2] Irving Biederman. Recognition-by-components: a theory of human image understanding. *Psychological review*, 94(2): 115, 1987. 1

[3] Irving Biederman. Recognizing depth-rotated objects: A review of recent research and theory. *Spatial vision*, 13:241–53, 2000. 1

[4] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, pages 1597–1607, 2020. 3

[5] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*. MIT Press, 2016. 1

[6] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, pages 9729–9738, 2020. 3, 6

[7] Olivier Henaff. Data-efficient image recognition with contrastive predictive coding. In *ICML*, pages 4182–4192. PMLR, 2020. 3

[8] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv*, 2019. 2, 6, 7

[9] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8340–8349, 2021.

[10] Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. *CVPR*, pages 15262–15271, 2021. 2

[11] Artur Jesslen, Guofeng Zhang, Angtian Wang, Wufei Ma, Alan Yuille, and Adam Kortylewski. Novum: Neural object volumes for robust object classification. In *ECCV*, pages 264–281, 2024. 1, 2, 3, 6, 7

[12] Yannis Kalantidis, Mert Bulent Sariyildiz, Noe Pion, Philippe Weinzaepfel, and Diane Larlus. Hard negative mixing for contrastive learning. In *Adv. Neural Inform. Process. Syst.*, pages 21798–21809. Curran Associates, Inc., 2020. 2, 3

[13] Prakhar Kaushik, Adam Kortylewski, and Alan Yuille. A bayesian approach to ood robustness in image classification. *arXiv*, 2024. 2, 3

[14] Prakhar Kaushik, Aayush Mishra, Adam Kortylewski, and Alan Yuille. Source-free and image-only unsupervised domain adaptation for category level object pose estimation. *arXiv*, 2024. 1, 2, 3, 6

[15] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *Adv. Neural Inform. Process. Syst.*, 33:18661–18673, 2020. 3

[16] Gregor Koporec and Janez Perš. Deep learning performance in the presence of significant occlusions - an intelligent household refrigerator case. In *IEEE/CVF International Conference on Computer Vision Workshop*, pages 2532–2540, 2019. 1

[17] Adam Kortylewski, Ju He, Qing Liu, and Alan Loddon Yuille. Compositional convolutional neural networks: A deep architecture with innate robustness to partial occlusion. In *CVPR*, pages 8940–8949, 2020. 2

[18] Adam Kortylewski, Qing Liu, Angtian Wang, Yihong Sun, and Alan Yuille. Compositional convolutional neural networks: A robust and interpretable model for object recognition under occlusion. In *IJCV*, pages 736–760. Springer, 2021. 2

[19] Yann LeCun, Y. Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521:436–44, 2015. 1

[20] Charles Leek, Irene Reppa, and Martin Arguin. The structure of three-dimensional object representations in human vision: Evidence from whole-part matching. *Journal of experimental psychology. Human perception and performance*, 31:668–84, 2005. 1

[21] Wufei Ma, Angtian Wang, Alan Yuille, and Adam Kortylewski. Robust category-level 6d pose estimation with coarse-to-fine rendering of neural features. In *ECCV*, pages 492–508. Springer, 2022. 1, 2, 6

[22] Wufei Ma, Qihao Liu, Jiahao Wang, Xiaoding Yuan, Angtian Wang, Yi Zhang, Zihao Xiao, Guofeng Zhang, Beijia Lu, Ruxiao Duan, Yongrui Qi, Adam Kortylewski, Yaoyao Liu, and Alan Yuille. Adding 3d geometry control to diffusion models. *arXiv*, 2023. 1, 6, 7, 2

[23] Wufei Ma, Guofeng Zhang, Qihao Liu, Guanning Zeng, Letian Zhang, Adam Kortylewski, Yaoyao Liu, and Alan Yuille. Imagenet3d: Towards general-purpose object-level 3d understanding. *Adv. Neural Inform. Process. Syst.*, 38, 2024. 1, 6, 7

[24] Ishan Misra and Laurens van der Maaten. Self-supervised learning of pretext-invariant representations. In *CVPR*, pages 6706–6716, 2020. 3

[25] Joshua Robinson, Ching-Yao Chuang, Suvrit Sra, and Stefanie Jegelka. Contrastive learning with hard negative samples. *ICLR*, 2021. 3

[26] Evgenia Rusak, Steffen Schneider, Peter Gehler, Oliver Bringmann, Wieland Brendel, and Matthias Bethge. Adapting imagenet-scale models to complex distribution shifts with self-learning. *arXiv*, 2021. 2

[27] Steffen Schneider, Evgenia Rusak, Luisa Eck, Oliver Bringmann, Wieland Brendel, and Matthias Bethge. Improving robustness against common corruptions by covariate shift adaptation. *Adv. Neural Inform. Process. Syst.*, 2020. 2

[28] Yihong Sun, Adam Kortylewski, and Alan Yuille. Weakly-supervised amodal instance segmentation with compositional priors. *arXiv*, 2020. 2

[29] Angtian Wang, Yihong Sun, Adam Kortylewski, and Alan L Yuille. Robust object detection under occlusion with context-aware compositionalnets. In *CVPR*, pages 12645–12654, 2020. 6

[30] Angtian Wang, Adam Kortylewski, and Alan Yuille. Nemo: Neural mesh models of contrastive features for robust 3d pose estimation. In *ICLR*, 2021. 1, 2, 3

[31] Angtian Wang, Wufei Ma, Alan Yuille, and Adam Kortylewski. Neural textured deformable meshes for robust analysis-by-synthesis. *WACV*, 2024. 1, 2, 3, 6

[32] Jianyu Wang, Zhishuai Zhang, Cihang Xie, Yuyin Zhou, Vittal Premachandran, Jun Zhu, Lingxi Xie, and Alan Yuille. Visual concepts and compositional voting. *arXiv*, 2017. 2

[33] Jinqiang Wang, Tao Zhu, Liming Luke Chen, Huansheng Ning, and Yaping Wan. Negative selection by clustering for contrastive learning in human activity recognition. *IEEE Internet of Things Journal*, 10(12):10833–10844, 2023. 2, 3

[34] Zhirong Wu, Yuanjun Xiong, Stella X. Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. *CVPR*, pages 3733–3742, 2018. 3

[35] Jun Xia, Lirong Wu, Ge Wang, Jintao Chen, and Stan Z. Li. Progcl: Rethinking hard negative mining in graph contrastive learning. In *ICML*, pages 24332–24346. PMLR, 2022. 2, 3

[36] Yu Xiang, Roozbeh Mottaghi, and Silvio Savarese. Beyond pascal: A benchmark for 3d object detection in the wild. In *IEEE Winter Conference on Applications of Computer Vision*, pages 75–82, 2014. 1

[37] Alan Yuille and Daniel Kersten. Vision as bayesian inference: analysis by synthesis? *Trends in Cognitive Sciences*, 10(7): 301–308, 2006. 2

[38] Bingchen Zhao, Jiahao Wang, Wufei Ma, Artur Jesslen, Siwei Yang, Shaozuo Yu, Oliver Zendel, Christian Theobalt, Alan Yuille, and Adam Kortylewski. Ood-cv-v2: An extended benchmark for robustness to out-of-distribution shifts of individual nuisances in natural images. In *IEEE TPAMI*, 2024. 1, 2

[39] Xingyi Zhou, Arjun Karpur, Linjie Luo, and Qixing Huang. Starmap for category-agnostic keypoint and viewpoint estimation. In *ECCV*, 2018. 7

[40] Hongru Zhu, Peng Tang, Jeongho Park, Soojin Park, and Alan Yuille. Robustness of object recognition under extreme occlusion in humans and computational models. *arXiv*, 2019. 1

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.

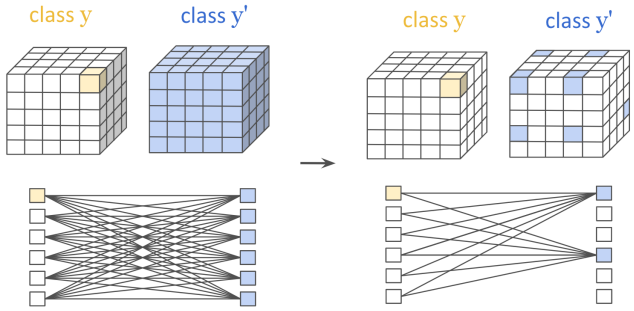


Figure 6. In our grouped cross-category contrasting, we contrast every vertex feature from category y(yellow cube) to only a small subset of the vertex features from each other category y'(blue cube).

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.

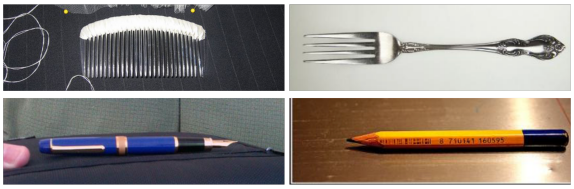


Figure 7. Example images of elongated objects in the ImageNet3D+ dataset. From left to right and top to bottom, the object classes are ”comb”, ”fork”, ”pen” and ”pencil”.

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

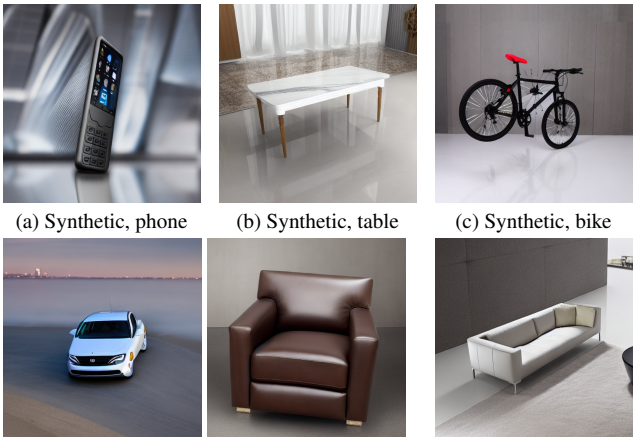
	IID	Occ.	Corr.
All classes	57.6	29.5	45.7
w/o Elongated	**59.3**	**32.8**	**48.3**

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.