


# NOVUM: Neural Object Volumes for Robust Object Classification

Artur Jesslen<sup>1\*</sup>, Guofeng Zhang<sup>2\*</sup>, Angtian Wang<sup>2</sup>, Wufei Ma<sup>2</sup>,  
Alan Yuille<sup>2</sup>, and Adam Kortylewski<sup>1,3</sup>

<sup>1</sup> University of Freiburg

<sup>2</sup> Johns Hopkins University

<sup>3</sup> Max-Planck-Institute for Informatics

**Abstract.** Discriminative models for object classification typically learn image-based representations that do not capture the compositional and 3D nature of objects. In this work, we show that explicitly integrating 3D compositional object representations into deep networks for image classification leads to a largely enhanced generalization in out-of-distribution scenarios. In particular, we introduce a novel architecture, referred to as NOVUM, that consists of a feature extractor and a *neural object volume* for every target object class. Each neural object volume is a composition of 3D Gaussians that emit feature vectors. This compositional object representation allows for a highly robust and fast estimation of the object class by independently matching the features of the 3D Gaussians of each category to features extracted from an input image. Additionally, the object pose can be estimated via inverse rendering of the corresponding neural object volume. To enable the classification of objects, the neural features at each 3D Gaussian are trained discriminatively to be distinct from (i) the features of 3D Gaussians in other categories, (ii) features of other 3D Gaussians of the same object, and (iii) the background features. Our experiments show that NOVUM offers intriguing advantages over standard architectures due to the 3D compositional structure of the object representation, namely: (1) An exceptional robustness across a spectrum of real-world and synthetic out-of-distribution shifts and (2) an enhanced human interpretability compared to standard models, all while maintaining real-time inference and a competitive accuracy on in-distribution data. Code and model can be found at /GenIntel/NOVUM.

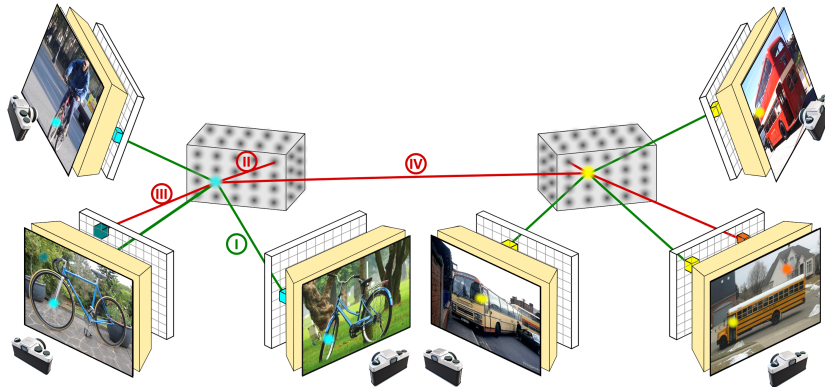
## 1 Introduction

Current deep learning architectures demonstrate advanced capabilities in visual recognition tasks, *e.g.*, object classification, detection, and pose estimation [6, 13, 22, 23, 32]. However, generalization to out-of-distribution (OOD) scenarios remains a fundamental challenge [14, 20, 25, 49]. In contrast, human vision achieves a significantly better robustness under OOD scenarios, *e.g.*, domain

---

\* Equal contribution.

✉ Corresponding author: [jesslen@cs.uni-freiburg.de](mailto:jesslen@cs.uni-freiburg.de).



**Fig. 1:** Schematic overview of how NOVUM is trained. The model consists of a shared backbone (yellow) and one neural object volume for each object class (grey), which are represented as 3D Gaussians on a cuboid shape. During training, the backbone first computes feature maps of the training images. Given the class label and the 3D object pose, the backbone is trained in a contrastive manner using four types of losses: (I) To make features of the same Gaussian similar across instances (green), while at the same time making the features distinct (red) from (II) features of Gaussians from the same object, (III) background features, and (IV) features of Gaussians from other objects.

shift, and occlusions [2,21]. Some cognitive studies hypothesize that human vision relies on a compositional 3D representation of objects while perceiving the world through an analysis-by-synthesis process [27,47]. While object-centric [24,45,46] and compositional representations [7,18,20] show promise in improving sample efficiency and generalization of machine learning algorithms, so far these models are largely ignorant of the 3D nature of our world and mostly limited to simple synthetic domains. This raises the question: Can machines enhance generalization at real-world classification tasks through learning 3D object representations? In this work, we embed a 3D compositional object representation explicitly into the neural network architecture for object classification. We take inspiration from recent advances in neural rendering [19] and prior works on pose estimation [17,37] to design a 3D-aware neural network for image classification. In particular, we propose NOVUM, which is composed of a feature extractor and *neural object volumes* for every object category (Figure 1). Each neural object volume is a spatial composition of Gaussian ellipsoidal kernels that emit feature vectors. During inference, an image is classified by matching the Gaussian features of each category to the input feature map. In this way, object classification is realized using the individual Gaussian features only and without requiring the 3D spatial geometry, hence facilitating a fast and robust classification inference. Each Gaussian contributes to the prediction by focusing individually on different evidences (different parts of objects in our case). When combined together, the lack of evidence from part of the Gaussians can be compensated by the rest, leading to high robustness against outliers or occlusions. Moreover, the inherent 3D representation in our model can also estimate the 3D object pose via inverse

rendering of the feature volume using Gaussian splatting. We use 3D pose annotations of the objects to train the neural features to be distinct from features of other categories’ neural object volumes, as well as spatially apart features of the same neural object volume, and the background clutter. Intuitively, the feature extractor learns to classify every pixel in the image as being either part of the object or as background context.

We evaluate NOVUM on a variety of datasets that contain 3D pose annotations and OOD shifts, such as real-world OOD shifts on the OOD-CV dataset [48], and synthetic OOD shifts on the corrupted PASCAL3D+ [14] and occluded PASCAL3D+ [37]. Our experiments show that NOVUM is exceptionally robust compared to other state-of-the-art architectures (both CNNs and Transformers) at object classification while performing on par with in-distribution data in terms of accuracy and inference speed. Moreover, the 3D pose predictions obtained via inverse rendering are competitive to baseline models which are limited to perform 3D pose estimation only. Finally, we show that the Gaussian matching result provides intuitive human interpretable information about the model prediction, by showing where the model perceives corresponding object parts.

In conclusion, NOVUM introduces a 3D volume representation for classification, achieving robustness through compositionality and 3D-awareness, while offering enhanced human interpretability by visualizing individual Gaussian kernel matches, and still enabling real-time inference.

## 2 Related Work

**Robust Image Classification.** Image classification is a classical task in computer vision. Multiple influential architectures such as ResNet [13], some transformers [23,36] were designed for this task. However, these models have primarily targeted in-distribution data, leaving a significant gap when faced with out-of-distribution (OOD) data such as common synthetic corruptions [14] or real-world OOD data [49]. To bridge this OOD generalization gap, efforts have concentrated mainly on two fronts: data augmentation and architectural design. Data augmentation strategies involve leveraging learned augmentation policies [5], and data mixtures [15] to enhance the diversity of training samples, thereby fostering generalization by generating synthetic OOD data. On the other hand, architectural advances that incorporate general prior knowledge about the world, such as compositionality [7, 18, 20], or object-object centric representations [24, 45, 46], have also shown promising advances in terms of enhancing sample efficiency and generalization in neural networks. Our approach falls into the second category, as we introduce a 3D compositional representation into the architecture of neural networks for object classification. In contrast to standard discriminative methods relying on a single entangled feature representation of an image, our 3D and compositional representation leads to a largely enhanced robustness when faced with occlusions, corruptions, and real-world OOD scenarios.

**Contrastive learning.** Studies have found that, in supervised settings, learning features that are discriminative during training does not guarantee that a model

will generalize, instead, they can have inductive biases towards learning simple “shortcut” features and decision rules [16, 28]. Hence, contrastive learning [11] is an interesting direction to prevent to learn these shortcuts since it is essentially a framework that learns similar/dissimilar representations by optimizing through the similarities of pairs in the representation space. Later the idea of using contrastive pair is extended to Triplet [31]. While traditional contrastive learning has focused on image-level or text-level representations, the extension to feature-level representations has gained lots of attention after InfoNCE [29]. Influential works in various fields include SimCLR [3], CLIP [30], CoKe [1]. In our paper, we adopt a three-level feature contrastive loss that encourages spatially distinct features, categorical specific features, and background specific features.

**3D neural representations for pose estimation.** Explicit 3D object representations can offer significant advantages in terms of generalization over purely image-based representations. Importantly, when using category-level 3D representations for vision tasks, it becomes imperative to not rely on detailed instance-specific features (*e.g.*, object colour or texture), but rather to enable the learning of general category-level features. As a result, employing features that remain invariant to such specificities emerges as a straightforward solution. Initially, [34] delved into computing correspondences between image features and a learned volumetric representation, leveraging HOG features and employing a Proposal-Validation process, which is slow. Later works including [17, 37, 38], have replaced HOG features with neural features capable of encoding richer information. This shift to neural features has facilitated the extension of render-and-compare methods, originally introduced at the pixel-level [4, 41], to the feature level. This extension enhances the models’ ability to generalize to rendering category-level instances. Conceptually, this method embodies an approximate analysis-by-synthesis approach, similar to the principles outlined in [10], which proves to be more robust against out-of-distribution (OOD) data in 3D pose estimation when contrasted with classical discriminative methods [26, 35, 50].

NOVUM builds on prior work on category-level pose estimation, such as neural mesh models [37, 38]. NOVUM extends this line of work in multiple ways: (1) An architecture with a shared class-agnostic backbone, which enables us to introduce a class-contrastive loss for object classification. (2) A principled, efficient way of performing classification inference that exploits individual 3D Gaussians. (3) A comprehensive mathematical formulation that derives a vMF-based contrastive training loss. (4) A compositional 3D Gaussian representation [19] that can be matched robustly with volume rendering [40] even in strong OOD scenarios. We demonstrate the applicability of compositional 3D representations in classification tasks, hence pioneering a step towards highly advanced capabilities in terms of generalization, interpretability and multi-tasking in vision.

### 3 Method

In this section, we present a deep network with an integrated 3D and compositional volume representation of objects to achieve robust object classification



(Section 3.1). We discuss how the model can be learned (Section 3.2) and how it can be applied to infer the object class as well as the 3D pose at test time (Section 3.3).

### 3.1 NOVUM: A Network Architecture with Neural Object Volumes

**Motivation.** Our aim is to achieve robustness at object classification by devising a neural network architecture that integrates a 3D compositional representation of objects. Specifically, our architecture explicitly uses an object-centric 3D representation to introduce compositionality on two levels of abstraction: (1) *Image-level compositionality*, *i.e.*, a representation that explicitly models an image as a composition of objects and background clutter. (2) *Object-level compositionality*, *i.e.*, a representation that explicitly models objects as a spatial composition of local elements. This compositional representation enhances model robustness by improving the models ability to classify objects even if only few local parts are recognizable (*e.g.*, due to occlusion or due to a novel object topology), or when the object is placed in an unusual context (*e.g.*, a bicycle underwater).

**Neural Object Volumes.** Building on recent advances in Gaussian splatting [19], we represent objects as a 3D density field via a spatial composition of  $K$  Gaussians that are placed on the surface geometry of each object category. Each Gaussian emits an associated feature vector, hence defining the volumetric object representation that we refer to as *neural object volume*. Each neural object volume represents an object category and is learned from feature maps extracted from 2D images by given an annotated 3D pose of the object (Figure 1).

More formally, we define a neural volume density at spatial location  $\mathbf{x} \in \mathbb{R}^3$  as a mixture of three-dimensional Gaussian  $\rho(\mathbf{x}) = \sum_{k=1}^K \rho_k(\mathbf{x})$ . Each Gaussian density is defined as  $\rho_k(\mathbf{x}) = \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ , where  $\boldsymbol{\mu}_k \in \mathbb{R}^3$  is the 3D position of the  $k$ -th Gaussian and  $\boldsymbol{\Sigma}_k \in \mathbb{R}^{3 \times 3}$  is its covariance matrix (defining the direction, shape and size of  $k$ -th kernel). In our setup, we do not assume detailed geometry of the object but simply arrange the Gaussians such that they form a cuboid volume with a pre-defined and diagonal covariance, such that the volume approximately encompasses the variable object instances in the corresponding object category. Each Gaussian is associated with a feature vector, denoted  $C_k \in \mathbb{R}^D$ . For each object category  $y$ , we learn a set of features  $\mathcal{C}_y = \{C_k \in \mathbb{R}^D\}_{k=1}^K$ . We define the set of Gaussian features from all object categories as  $\mathcal{C} = \{\mathcal{C}_y\}_{y=1}^Y$ , where  $Y$  is the total number of object categories. The neural object volume can be rendered into the feature space, using standard volume rendering:

$$\hat{\mathbf{C}}_i(\alpha) = \int_{t_n}^{t_f} T(t) \sum_{k=1}^K \rho_k(\mathbf{r}_\alpha(t)) \mathbf{C}_k dt, \quad \text{where } T(t) = \exp\left(-\int_{t_n}^t \rho(\mathbf{r}_\alpha(s)) ds\right), \quad (1)$$

where the feature  $\hat{\mathbf{C}}_i(\alpha)$  at pixel position  $i$  in the rendered feature map is computed by aggregating the Gaussian features along the ray  $\mathbf{r}_\alpha(t)$ . The ray traverses from the camera center through the pixel  $i$  on the image plane with  $\alpha$  denoting the camera view. Here,  $t$  ranges from the near plane  $t_n$  to the far plane  $t_f$ . The remainder of the image that is not covered by the rendered object

volume is represented as background features  $\mathcal{B} = \{\beta_n \in \mathbb{R}^D\}_{n=1}^{N_b}$  where  $N_b$  is a fixed hyperparameter, and  $\mathcal{B}$  is shared among all object categories.

**NOVUM.** Our model architecture builds on a feature extractor  $\Phi_w$  and a set of neural object volumes, one for each object category. The feature extractor computes a feature map  $F = \Phi_w(I) \in \mathbb{R}^{D \times H \times W}$  from an input image  $I$ , where  $w$  denotes the parameters of the CNN backbone. The feature map  $F$  contains feature vectors  $f_i \in \mathbb{R}^D$  at positions  $i$  on a 2D lattice.

Learning our model requires obtaining correspondences between a Gaussian  $k$  of a neural object volume and a location  $i$  in the feature map of a training image, and we obtain this correspondence by projecting Gaussian into the image feature map  $F$ . With given camera pose  $\alpha$ , we use volume rendering to compute the contribution  $\gamma_{ik}$  of Gaussian feature  $C_k$  to image features  $f_i$ . We estimate a one-to-one correspondence between features and Gaussians by selecting the closest image feature  $f_i$  for each Gaussian, *i.e.*, where  $\gamma_{ik}$  is the largest. Throughout the remaining paper, we denote  $f_{k \rightarrow i}$  to indicate the extracted feature  $f_i$  at location  $i$  that Gaussian  $k$  with mean  $\mu_k$  projects to. We relate the extracted image features to the Gaussians and background features by von-Mises-Fisher (vMF) probability distributions. In particular, we model the probability of generating the feature  $f_i$  from corresponding Gaussian  $C_k$  as  $P(f_{k \rightarrow i}|C_k) = c_M(\kappa)e^{\kappa f_{k \rightarrow i} \cdot C_k}$ , where  $C_k$  represents the mean of each vMF distribution ( $\|f_{k \rightarrow i}\| = 1, \|C_k\| = 1$ ). We also model the probability of generating the feature  $f_i$  from background features as  $P(f_i|\beta_n) = c_M(\kappa)e^{\kappa f_i \cdot \beta_n}$  for  $\beta_n \in \mathcal{B}$ . The concentration parameter  $\kappa$ , which determines the spread of the distribution and can be interpreted as an inverse temperature parameter, is defined as a global hyperparameter. Hence, the normalization constant  $c_M(\kappa)$  is a constant and can be ignored during learning and inference.

### 3.2 Learning Discriminative 3D Volume Representations

Learning NOVUM is challenging because we not only need to maximize the likelihood functions  $P(f_{k \rightarrow i}|C_k)$  and  $P(f_i|\mathcal{B})$ , but also learn a corresponding parameters  $w$  of the backbone. In particular, we maximize the probability that any extracted feature  $f_{k \rightarrow i}$  was generated from  $P(f_{k \rightarrow i}|C_k)$  instead of from any other alternatives. This motivates us to use contrastive learning where we compare the probability that an extracted feature  $f_{k \rightarrow i}$  is generated by the correct Gaussian  $C_k$  or from one of three alternative processes, namely, (i) from the Gaussians of other object classes, (ii) from non-neighboring Gaussians of the same object, and (iii) from the background features (see illustration in Figure 1):

$$\frac{P(f_{k \rightarrow i}|C_k)}{\sum_{\substack{C_l \in \mathcal{C}_y \\ C_l \notin \mathcal{N}_k}} P(f_{k \rightarrow i}|C_l) + \omega_\beta \sum_{\beta_n \in \mathcal{B}} P(f_{k \rightarrow i}|\beta_n) + \omega_{\bar{y}} \sum_{C_m \in \mathcal{C}_{\bar{y}}} P(f_{k \rightarrow i}|C_m)}, \quad (2)$$

where  $\mathcal{N}_k = \{C_r : \|\mu_k - \mu_r\| < \delta, k \neq r\}$  is the neighborhood of  $C_k$  and  $\delta$  is a distance threshold controlling the size of neighborhood.  $y$  is the category of the image and  $\bar{y}$  is a set of all other categories except  $y$ .  $\omega_\beta = \frac{P(\beta_n)}{P(C_k)}$  is the ratio of

the probability that an image feature corresponds to the background instead of the Gaussian  $k$ , and  $\omega_{\bar{y}} = \frac{P(C_m)}{P(C_k)}$  is the ratio of the probability that an image feature corresponds to Gaussians of other categories instead of the Gaussian  $k$ . We compute the final loss  $\mathcal{L}(\mathcal{C}, \mathcal{B}, w)$  by taking the logarithm and summing over all training examples – all sets of features  $\{f_{k \rightarrow i}\}$  from the training set

$$- \sum_y \sum_{k=1}^K o_k \cdot \log \frac{e^{\kappa f_{k \rightarrow i} \cdot C_k}}{\sum_{\substack{C_l \in \mathcal{C}_y \\ C_l \notin \mathcal{N}_k}} e^{\kappa f_{k \rightarrow i} \cdot C_l} + \omega_{\beta} \sum_{\beta_n \in \mathcal{B}} e^{\kappa f_{k \rightarrow i} \cdot \beta_n} + \omega_{\bar{y}} \sum_{C_m \in \mathcal{C}_{\bar{y}}} e^{\kappa f_{k \rightarrow i} \cdot C_m}}, \quad (3)$$

where  $o_k = 1$  if the Gaussian is visible and  $o_k = 0$  otherwise.

**Updating Gaussian and Background Features.** The Gaussian and background features  $\mathcal{C}$  and  $\mathcal{B}$  are updated after every gradient-update of the feature extractor. Following [1, 12], we use momentum update for the Gaussian features:

$$C_k \leftarrow C_k \cdot \sigma + f_{k \rightarrow i} \cdot (1 - \sigma), \quad \|C_k\| = 1. \quad (4)$$

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 randomly sample  $T$  new background features  $f_b$  from the feature map, where  $f_b$  is a feature that no Gaussian contributes to and add them into the background feature set  $\mathcal{B}$  (*i.e.*,  $\mathcal{B} \leftarrow \mathcal{B} \cup \{f_b\}$ ). We note that  $\sigma$  and  $T$  are hyper-parameters of our model.

### 3.3 Inference of object category and 3D pose

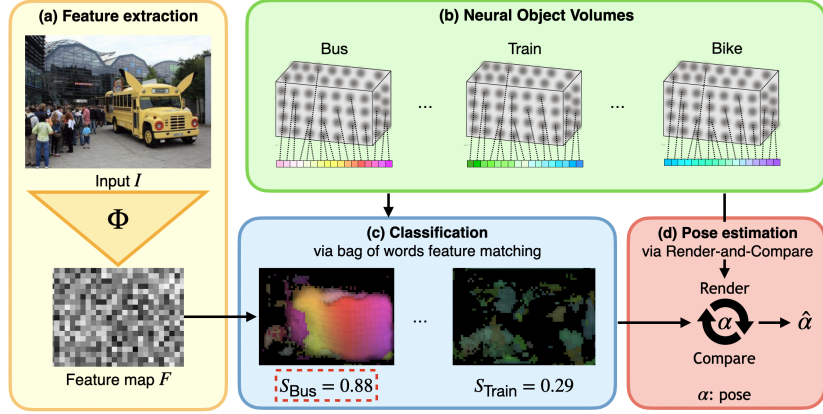
**Object Classification via Feature Matching without Geometry.** Our classification inference pipeline is illustrated in Figure 2. We perform classification in a fast and robust manner via matching the extracted features to the learned Gaussian features and background features. In short, for each object category  $y$ , we compute the foreground likelihood  $P(f_i | \mathcal{C}_y)$  and the background likelihood  $P(f_i | \mathcal{B})$  on all locations  $i$  in the feature map. In this process, we do not take into account the object geometry, which reduces the matching to a simple convolution operation, hence making it very fast. To classify an image, we compare the total likelihood scores of each class average over all locations  $i$ .

In detail, 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  $F = \Phi_w(I)$  can then be computed as:

$$\prod_{f_i \in F} P(f_i | z_{i,k}, y) = \prod_{f_i \in F} P(f_i | C_k)^{z_{i,k}} \prod_{f_i \in F} \max_{\beta_n \in \mathcal{B}} P(f_i | \beta_n)^{1-z_{i,k}}. \quad (5)$$

As described in Section 3.1, the extracted features follow a vMF distribution. Thus the final prediction score of each object category  $y$  is:

$$S_y = \sum_{f_i \in F} \max \left\{ \max_{C_k \in \mathcal{C}_y} f_i \cdot C_k, \max_{\beta_n \in \mathcal{B}} f_i \cdot \beta_n \right\}. \quad (6)$$



**Fig. 2:** Overview of the classification *inference pipeline*. NOVUM is composed of a backbone  $\Phi$  and a set of neural object volumes represented as 3D Gaussians on a cuboid shape (green box) with colored associated features. During inference, an image is first processed by the backbone into a feature map  $F$ . The object class is predicted by *independently* matching the Gaussian features to the feature map (blue box). We color-code the detected Gaussians to highlight the interpretability of our method. Brightness shows the prediction confidence. Note that the model is only confident with the correct class even though the bus is an out-of-distribution sample. The 3D object pose can also be inferred via inverse rendering of the neural object volume (red box).

The final category prediction is  $\hat{y} = \arg \max_{y \in Y} \{S_y\}$ . Figure 2 (blue box) illustrates the matching process for different object classes by color coding the detected Gaussians. For the correct class, the Gaussians can be detected coherently even without taking geometry into account (as can be observed by the smooth color variation), while for wrong classes this is not the case. Our ability to visualize the predicted kernel correspondence demonstrates an advanced interpretability of the decision process compared to standard classifiers.

**Volume Rendering for Pose Estimation.** Given the predicted object category  $\hat{y}$ , we use the Gaussian features  $\mathcal{C}_{\hat{y}}$  to estimate the camera pose  $\alpha$  leveraging the 3D geometrical information of the neural object volumes. Following the vMF distribution, we optimize the pose  $\alpha$  via feature reconstruction:

$$\mathcal{L}(\alpha) = \sum_{f_i \in FG} f_i \cdot \hat{\mathcal{C}}_i(\alpha) + \sum_{f_b \in BG} \max_{\beta_n \in \mathcal{B}} f_b \cdot \beta_n, \quad (7)$$

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  $FG = \{f_i \in F, \sum_{k=1}^K \rho_k(\mathbf{r}_\alpha(t)) > \theta\}$ .  $BG = F \setminus FG$  is the set of features in the background. We estimate the pose by first finding the best initialization of the object pose  $\alpha$  by computing the reconstruction loss (Equation (7)) for a set of pre-defined poses. Subsequently, we start gradient-based optimization using the initial pose that achieved the lowest loss to obtain the final pose prediction  $\hat{\alpha}$ .

## 4 Experiments

In this section, we evaluate NOVUM in terms of generalization on in-distribution and out-of-distribution data, 3D pose estimation and interpretability. We first discuss our experimental setup (Section 4.1), present baselines and results for classification (Section 4.2) and 3D pose estimation (Section 4.3). Additionally, we perform in-depth evaluations of interpretability and prediction consistency, and an ablation study (Section 4.4).

### 4.1 Setup

**Datasets.** We test NOVUM’s robustness using four datasets where 3D pose annotations are available: PASCAL3D+ (P3D+) [43], occluded-P3D+ [39], corrupted-P3D+ [25], and Out-of-Distribution-CV (OOD-CV) [48]. PASCAL3D+ includes 12 object categories. Building on the P3D+ dataset, the occluded-P3D+ dataset is a test benchmark that evaluates robustness under multiple occlusion levels. It simulates realistic occlusion by superimposing occluders on top of the objects with three different levels: L1: 20%-40%, L2: 40%-60%, and L3:60%-80%, where each level has corresponding percent of the objects and the background occluded. P3D+ dataset does not contain significant occlusion and is therefore referred to as occlusion level 0 (L0). The corrupted-P3D+ corresponds to P3D+ where we apply 12 types of corruptions [14, 25] to each image of the original test images, and we choose a severity level of 4 out of 5. The OOD-CV dataset is a benchmark that includes real-world OOD examples of 10 object categories varying in terms of 5 nuisance factors: pose, shape, context, texture, and weather.

**Implementation Details.** Each neural object contains approximately  $K = 1100$  Gaussians that are distributed uniformly on the cuboid. The shared feature extractor  $\Phi$  is a ResNet50 [13] model with two upsampling layers and an input shape of  $640 \times 800$ . All features have a dimension of  $D = 128$  and the size of the feature map  $F$  is  $1/8^{th}$  of the input size. NOVUM is trained as described in Section 3.2, taking around 20 hours on 4 RTX 3090 with 200 epochs. During training, we use  $N = 2560$  background features. For each gradient step, we use  $\sigma = 0.9$  for momentum update of the Gaussian features and sample  $T = 5$  new background features to update  $\mathcal{B}$ . We set  $\kappa = 1$  (see Appendix 9 for more details). We predict the object class as described in Section 3.3. The feature matching for classification takes around **0.01s** per sample on 1 RTX 3090, which is comparable to state-of-the-art feed-forward classification models. NOVUM can also infer the 3D object pose via inverse rendering by first evaluating the reconstruction loss on 144 initial poses (12 azimuth angles, 4 elevation angles, 3 in-plane rotations) and subsequently starting a gradient-based inverse rendering (Equation (7)) starting with lowest feature reconstruction loss as initialization. The pose inference pipeline takes around 0.21s per sample on 1 RTX 3090. We note that this inference might be further optimized *e.g.*, by caching intermediate matching results efficiently.

**Evaluation.** We evaluate our approach on two tasks: classification and 3D pose estimation. 3D pose estimation involves predicting azimuth, elevation, and rota-

tions of an object with respect to a camera. Following [50], the pose estimation error is calculated between the predicted rotation matrix  $R_{\text{pred}}$  and the ground truth rotation matrix  $R_{\text{gt}}$  as  $\Delta(R_{\text{pred}}, R_{\text{gt}}) = \|\log m(R_{\text{pred}}^T R_{\text{gt}})\|_F / \sqrt{2}$ . We measure accuracy using two thresholds  $\frac{\pi}{18}$  and  $\frac{\pi}{6}$ .

**Classification baselines.** We compare the performance of our approach to four competitive baseline architectures (*i.e.*, Resnet50, Swin-T, Convnext, and ViT-b-16) for the classification task. During training, these baselines are trained with a classification and a pose estimation head. Hence, 3D information is leveraged for these classification evaluations. For each baseline, we use a classification head for which the output is the number of classes in the dataset (*i.e.*, 12 for (occluded, corrupted)-P3D+; 10 for OOD-CV). We finetune each baselines during 200 epochs. In order to make baselines more robust, we apply standard data augmentation (*i.e.*, scale, translation, rotation, and flipping) for each during training. More details about baselines can be found in the appendix.

**3D-Pose estimation.** We compare the performance of our approach to five other baselines for the 3D pose estimation task. For Resnet50, Swin-T, Convnext, and ViT-b-16, we consider the pose estimation problem as a classification problem (following [50]) by using 42 intervals of  $\sim 8.6^\circ$  for each parameter that needs to be estimated (azimuth and elevation angle, and in-plane rotation). We fine-tune each baseline for 200 epochs. Similarly to classification, we apply standard data augmentation during training. We further evaluate against NeMo [37] that was explicitly designed for robust 3D pose estimation. We following the publicly available code and train a NeMo model for each class.

## 4.2 Robust Object Classification

We first evaluate the performance on IID data. As the L0 column of Table 1 shows, our approach achieves 99.5% for classification, which is comparable to other baselines. Furthermore, our approach manages to robustly classify images in various out-of-distribution scenarios. From Table 1, we can see that our representation allows to outperform all other traditional baselines with around 6% accuracy on average for different levels of occlusions and with up to 33% accuracy boost for images under five different types of nuisances in OOD-CV. For corrupted data, our approach also outperforms the baselines on average. In summary, NOVUM achieves **significant improvements in OOD generalization** while **maintaining state-of-the-art accuracy for IID** data for classification. Finally, it is worth noting that our approach is **more consistent** than all baselines. Our approach consistently outperforms baselines over all nuisances, which indicates the intrinsic robustness in our architecture.

## 4.3 Robust 3D Pose Estimation

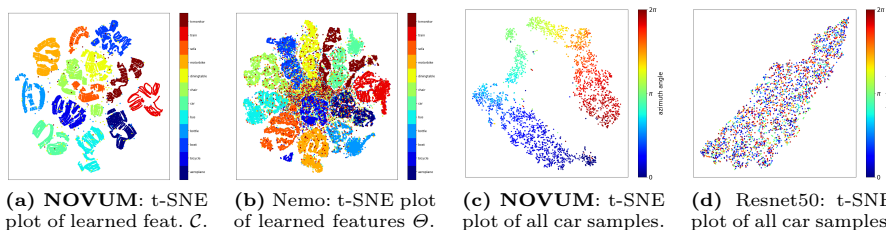
According to the results in Table 2, our approach outperforms all baselines significantly across IID and OOD scenarios, while also exceeding the performance of NeMo [37], the current state-of-the-art method for robust 3D pose estimation. This higher performance can be attributed to the better optimization of volume

**Table 1:** Classification accuracy results on P3D+, occluded-P3D+, OOD-CV and corrupted-P3D+ datasets. L0 corresponds to unoccluded images from Pascal3D+, and occlusion levels L1-L3 are from occluded-P3D+ dataset with occlusion ratios stated in Section 4.1. Our approach performs similarly in IID scenarios, while steadily outperforming all baselines in OOD scenarios. First is highlighted in **bold**, second is underlined. Higher is better. Full results in Appendix.

Dataset	P3D+	occluded-P3D+				OOD-CV	corrupted-P3D
Nuisance	L0	L1	L2	L3	Mean	Mean	Mean
Resnet50	99.3	93.8	77.8	45.2	72.3	51.4	78.7
Swin-T	99.4	93.6	77.5	46.2	72.4	<u>64.2</u>	78.9
Convnext	<u>99.4</u>	<u>95.3</u>	81.3	50.9	<u>75.8</u>	<u>56.0</u>	85.6
ViT-b-16	99.3	94.7	80.3	49.4	74.8	59.0	<u>87.6</u>
NOVUM	<b>99.5</b>	<b>97.2</b>	<b>88.3</b>	<b>59.2</b>	<b>81.6</b>	<b>85.2</b>	<b>91.3</b>

rendering which gives more stable gradients compared to mesh-based differentiable rendering.

#### 4.4 Comprehensive assessment of our representation



**Fig. 3:** (a-b) t-SNE plots comparing (a) the learned features  $\mathcal{C}$  of our approach and (b) the learned vertex features  $\Theta$  of NeMo. As can be seen, our contrastive loss allows a much clearer distribution of the space while keeping Gaussian features from different classes far from each other (while the low-quality clustering observed in (b) may likely originates from the ImageNet pretraining). (c-d) t-SNE plots of the mean extracted feature for each car image of the test dataset. We observe a very clear organization of the samples according to the azimuth angle for (c) our approach while this organization is completely absent in (d) other feed-forward baselines (*e.g.*, Resnet50).

**Interpretability.** Our explicit volumetric representation can be leveraged to predict the object class and its pose in an image. Insightful information also lies in the Gaussian feature matching between image features and the neural object volumes. Visualizing the matching results (see Figure 4 and more results including videos in Appendix 10 and Figure S2), reveals which object parts are perceived by the model at any given location in the image. As illustrated in Fig-

**Table 2:** 3D-Pose Estimation results for different datasets. A prediction is considered correct if the angular error is lower than threshold (*i.e.*,  $\frac{\pi}{6}$ , and  $\frac{\pi}{18}$ ). Higher is better. Our approach shows it is capable of robust 3D pose estimation that performs similarly to the current state-of-the-art. Note that models marked with a “\*” possess an explicit 3D representation.

Dataset	P3D+	occ-P3D+	cor-P3D+	OOD-CV	P3D+	occ-P3D+	cor-P3D+	OOD-CV
Threshold	$\pi/6$				$\pi/18$			
Resnet50	82.2	53.8	33.9	51.8	39.0	15.8	15.8	18.0
Swin-T	81.4	48.2	34.5	50.9	46.2	16.6	15.6	19.8
Convnext	82.4	49.3	37.1	50.7	38.9	14.1	24.1	19.9
ViT-b-16	82.0	50.8	38.0	48.0	38.0	15.0	21.3	21.5
NeMo*	86.1	62.2	48.0	51.6	<b>61.0</b>	31.8	<b>43.4</b>	21.9
NOVUM *	<b>88.2</b>	<b>63.1</b>	<b>49.1</b>	<b>52.9</b>	<u>59.1</u>	<b>32.7</b>	<b>43.4</b>	<b>22.8</b>

ure 4, the occluded parts of the objects are not colored, meaning that no Gaussian has been matched with these image features. Figures 3a and 3c moreover show that our learned features are disentangled and encode useful information in terms of object classes and 3D pose. When comparing Figures 3a and 3b, we observe different cluster for each category while the low-quality clustering observed in the latter Figure may likely originates from the ImageNet pretraining. When comparing Figures 3c and 3d, we observe a consistent distribution of car instances depending of their pose in the former while in the latter, features don’t have any explicit organization in terms of pose.

**Table 3:** Consistency of predictions for classification and 3D pose estimation. The last line shows the accuracy of those images with both object pose (threshold:  $\frac{\pi}{6}$ ) and class correctly predicted. "cls." stands for classification.

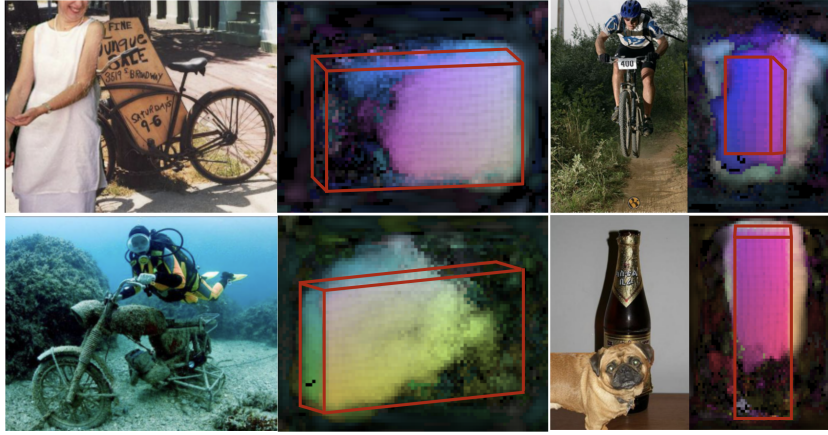
Dataset	P3D+	occ-P3D+			
Task	L0	L1	L2	L3	
classification	99.5	97.2	88.3	59.2	
3D pose (th: $\frac{\pi}{6}$ )	90.1	82.6	71.3	52.7	
cls. & 3D pose	89.8	81.9	70.2	50.2	

**Table 4:** Ablation studies of the background features and the shape of the representation on P3D+ and occluded-P3D+. We ablate both the background features and the object shape.

Components		P3D+	occ-P3D+			
$\mathcal{B}$	Shape	L0	L1	L2	L3	
	single feature	93.2	90.3	80.4	44.0	
✓	sphere	99.3	97.0	87.9	59.0	
	cuboid	99.3	97.0	85.7	53.0	
✓	cuboid	<b>99.5</b>	<b>97.2</b>	<b>88.3</b>	<b>59.2</b>	

**Consistency.** Another valuable characteristics of our approach lies in the fact that it is consistent between the different tasks. In Table 3, we observe that the accuracy for samples that have correct class and pose prediction are limited by the pose estimation itself since they are fairly similar. In IID scenarios, the





**Fig. 4:** Four qualitative results that were misclassified by ViT-b-16. We show for each: (left) the input image and (right) the extracted feature map and the predicted 3D pose overlaid. We color coded the features by encoding the color as a function of  $\mu_k$  of the matched Gaussian  $C_k$  (as done in NOCS [42]). Hence, a smooth color gradient shows a high quality matching. In the extracted features, the brightness illustrates the confidence of matching with the Gaussian features.

difference is only of 0.3%, while in OOD scenarios the difference is around 1% on average. We believe that this consistency comes from the common explicit volumetric representation that is shared for all tasks. Such a behavior would not be expected between different task-specific models that are trained separately.

**Efficiency.** For classification, NOVUM matches real-time performance as other CNN or transformer-based baselines, reaching an inference speed of 50FPS. Despite variations in parameter numbers among baselines (Swin-T: 28M, ViT-b-16: 86M, NOVUM: 83M), we find no correlation of the parameter count with OOD robustness. For pose estimation, compared to the render-and-compare-based method, NeMo [37], our model uses a significantly lower parameter count (NOVUM: 83M, NeMo: 996M, *i.e.*, an impressive 12x reduction) due to our class-agnostic backbone. We also observe that our model converges faster in the render-and-compare optimization compared to NeMo (NOVUM: 30 steps, NeMo: 300 steps), which can be attributed to our class-contrastive representation and the neural object volumes which yields better gradient during the optimization. More detailed comparisons can be found in Appendix 8.5.

**Ablations.** We ablate the geometry selection and background features during training. In NOVUM, we form a cuboid with Gaussians. An alternative could be to choose to (1) employ a finer-grained representation, (2) utilize a single Gaussian to represent the volume, or (3) adopt a spherical geometric representation instead of a cuboid. Opting for the first alternative would necessitate either establishing a deformable geometry or treating each sub-category as a distinct class, which is beyond the scope of this work. In Table 4 (more details in Appendix 8.6), we ablate the shape of our 3D representation. As expected,

the choice of the representation’s shape does not have a pronounced influence on performance, as was already underlined for meshes [37]. However, we observe a slight advantage in favor of the cuboid shape which may be attributed to the cuboid’s closer approximation of the true shape. Additionally, we considered a *single feature vector* approach, where a single Gaussian per class is employed during training using contrastive learning. We observe a performance drop by up to 15%, which highlights the importance of the geometry during training. These findings corroborate the classification results of our method: by selectively omitting some geometric information (*i.e.*, the 3D structure), we can reach similar outcomes while significantly enhancing computational efficiency. Ultimately, we note that the background model  $\mathcal{B}$  is beneficial during training since it promotes greater dispersion among neural features. This proves to be useful for inference in cases marked by occlusions but does not have visible effect in IID scenarios.

## 5 Limitations and Future Work

Despite the strong generalization performance, multi-task capabilities and interpretability, the proposed NOVUM model also has limitations. For example, the current necessity for pose annotation during training is suboptimal. However, we maintain an optimistic outlook regarding future advancements that may mitigate this requirement. Notably, recent studies [8, 9, 44] are investigating the demanding task of aligning objects with minimal examples, indicating a promising research trajectory. Currently, the geometry of the neural object volumes is fixed, which is suboptimal for object categories with large intra-class variability. Moving forward, exploring methods for flexible deformations would likely enhance the performance of the model, while also enabling the accurate segmentation of objects, thus enriching the scope of the proposed framework.

## 6 Conclusion

In this work, we introduced NOVUM, an architecture for object classification with an explicit compositional 3D object representation. We presented a framework for learning neural object volumes with associated features across various categories using annotated 3D object poses. Our experimental results shows that NOVUM achieves much higher robustness compared to other state-of-the-art architectures under OOD scenarios, *e.g.*, occlusion and corruption, with competitive performance in in-distribution scenarios, and similar inference speed for image classification. Further, we demonstrate NOVUM can also estimate 3D pose accurately by following an inverse rendering approach, achieves an enhanced human interpretability and consistency of multi-task predictions. In conclusion, our results showcase NOVUM as a pioneering step towards highly advanced generalization capabilities in vision models.

## Acknowledgements

Adam Kortylewski acknowledges support via his Emmy Noether Research Group funded by the German Reserach Foundation (DFG) under Grant No. 468670075. Alan Yuille acknowledges support from Army Research Laboratory award W911NF2320008 and Office of Naval Research N00014-21-1-2812.

## References

1. Bai, Y., Wang, A., Kortylewski, A., Yuille, A.: Coke: Localized contrastive learning for robust keypoint detection (2022)
2. Bengio, Y., Lecun, Y., Hinton, G.: Deep learning for ai. *Communications of the ACM* **64**(7), 58–65 (2021)
3. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations (2020)
4. Chen, X., Dong, Z., Song, J., Geiger, A., Hilliges, O.: Category level object pose estimation via neural analysis-by-synthesis. In: *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVI* 16. pp. 139–156. Springer (2020)
5. Cubuk, E.D., Zoph, B., Mane, D., Vasudevan, V., Le, Q.V.: Autoaugment: Learning augmentation policies from data. *arXiv preprint arXiv:1805.09501* (2018)
6. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929* (2020)
7. George, D., Lehrach, W., Kansky, K., Lázaro-Gredilla, M., Laan, C., Marthi, B., Lou, X., Meng, Z., Liu, Y., Wang, H., et al.: A generative vision model that trains with high data efficiency and breaks text-based captchas. *Science* **358**(6368), eaag2612 (2017)
8. Goodwin, W., Havoutis, I., Posner, I.: You only look at one: Category-level object representations for pose estimation from a single example. *arXiv preprint arXiv:2305.12626* (2023)
9. Goodwin, W., Vaze, S., Havoutis, I., Posner, I.: Zero-shot category-level object pose estimation. In: *European Conference on Computer Vision*. pp. 516–532. Springer (2022)
10. Grenander, U.: A unified approach to pattern analysis. In: *Advances in computers*, vol. 10, pp. 175–216. Elsevier (1970)
11. Hadsell, R., Chopra, S., LeCun, Y.: Dimensionality reduction by learning an invariant mapping. In: *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’06)*. vol. 2, pp. 1735–1742 (2006). <https://doi.org/10.1109/CVPR.2006.100>
12. He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum contrast for unsupervised visual representation learning. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. pp. 9729–9738 (2020)
13. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. pp. 770–778 (2016)
14. Hendrycks, D., Dietterich, T.: Benchmarking neural network robustness to common corruptions and perturbations. In: *International Conference on Learning Representations* (2019)

15. Hendrycks, D., Mu, N., Cubuk, E.D., Zoph, B., Gilmer, J., Lakshminarayanan, B.: Augmix: A simple data processing method to improve robustness and uncertainty. arXiv preprint arXiv:1912.02781 (2019)
16. Hermann, K.L., Lampinen, A.K.: What shapes feature representations? exploring datasets, architectures, and training (2020)
17. Iwase, S., Liu, X., Khirodkar, R., Yokota, R., Kitani, K.M.: Repose: Fast 6d object pose refinement via deep texture rendering. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 3303–3312 (October 2021)
18. Jin, Y., Geman, S.: Context and hierarchy in a probabilistic image model. In: 2006 IEEE computer society conference on computer vision and pattern recognition (CVPR’06). vol. 2, pp. 2145–2152. IEEE (2006)
19. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G.: 3d gaussian splatting for real-time radiance field rendering. ACM Transactions on Graphics **42**(4) (2023)
20. Kortylewski, A., He, J., Liu, Q., Yuille, A.L.: Compositional convolutional neural networks: A deep architecture with innate robustness to partial occlusion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (June 2020)
21. Kortylewski, A., Liu, Q., Wang, H., Zhang, Z., Yuille, A.: Combining compositional models and deep networks for robust object classification under occlusion. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 1333–1341 (2020)
22. LeCun, Y., Bengio, Y., et al.: Convolutional networks for images, speech, and time series. The handbook of brain theory and neural networks **3361**(10), 1995 (1995)
23. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 10012–10022 (2021)
24. Locatello, F., Weissenborn, D., Unterthiner, T., Mahendran, A., Heigold, G., Uszkoreit, J., Dosovitskiy, A., Kipf, T.: Object-centric learning with slot attention. Advances in Neural Information Processing Systems **33**, 11525–11538 (2020)
25. Michaelis, C., Mitzkus, B., Geirhos, R., Rusak, E., Bringmann, O., Ecker, A.S., Bethge, M., Brendel, W.: Benchmarking robustness in object detection: Autonomous driving when winter is coming. arXiv preprint arXiv:1907.07484 (2019)
26. Mousavian, A., Anguelov, D., Flynn, J., Kosecka, J.: 3d bounding box estimation using deep learning and geometry. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. pp. 7074–7082 (2017)
27. Neisser, U., et al.: Cognitive Psychology. Appleton-Century-Crofts (1967)
28. Nguyen, T., Raghu, M., Kornblith, S.: Do wide and deep networks learn the same things? uncovering how neural network representations vary with width and depth (2021)
29. van den Oord, A., Li, Y., Vinyals, O.: Representation learning with contrastive predictive coding (2019)
30. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., Sutskever, I.: Learning transferable visual models from natural language supervision (2021)
31. Schroff, F., Kalenichenko, D., Philbin, J.: Facenet: A unified embedding for face recognition and clustering. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 815–823 (2015). <https://doi.org/10.1109/CVPR.2015.7298682>

32. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
33. Sra, S.: A short note on parameter approximation for von mises-fisher distributions: and a fast implementation of  $\text{is}(x)$ . Computational Statistics **27**, 177–190 (2012)
34. Stark, M., Goesele, M., Schiele, B.: Back to the future: Learning shape models from 3d cad data. In: British Machine Vision Conference (BMVC). pp. 1–11 (01 2010). <https://doi.org/10.5244/C.24.106>
35. Tulsiani, S., Malik, J.: Viewpoints and keypoints. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1510–1519 (2015)
36. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017)
37. Wang, A., Kortylewski, A., Yuille, A.: Nemo: Neural mesh models of contrastive features for robust 3d pose estimation. In: International Conference on Learning Representations (2021)
38. Wang, A., Mei, S., Yuille, A.L., Kortylewski, A.: Neural view synthesis and matching for semi-supervised few-shot learning of 3d pose. Advances in Neural Information Processing Systems **34**, 7207–7219 (2021)
39. Wang, A., Sun, Y., Kortylewski, A., Yuille, A.L.: Robust object detection under occlusion with context-aware compositionalsnets. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12645–12654 (2020)
40. Wang, A., Wang, P., Sun, J., Kortylewski, A., Yuille, A.: Voge: a differentiable volume renderer using gaussian ellipsoids for analysis-by-synthesis. In: The Eleventh International Conference on Learning Representations (2022)
41. Wang, H., Sridhar, S., Huang, J., Valentin, J., Song, S., Guibas, L.J.: Normalized object coordinate space for category-level 6d object pose and size estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2642–2651 (2019)
42. Wang, H., Sridhar, S., Huang, J., Valentin, J., Song, S., Guibas, L.J.: Normalized object coordinate space for category-level 6d object pose and size estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (June 2019)
43. Xiang, Y., Mottaghi, R., Savarese, S.: Beyond pascal: A benchmark for 3d object detection in the wild. In: IEEE winter conference on applications of computer vision. pp. 75–82. IEEE (2014)
44. Yang, H., Shi, J., Carlone, L.: TEASER: Fast and Certifiable Point Cloud Registration. IEEE Trans. Robotics (2020)
45. Yi, K., Gan, C., Li, Y., Kohli, P., Wu, J., Torralba, A., Tenenbaum, J.B.: Clevrer: Collision events for video representation and reasoning. arXiv preprint arXiv:1910.01442 (2019)
46. Yi, K., Wu, J., Gan, C., Torralba, A., Kohli, P., Tenenbaum, J.: Neural-symbolic vqa: Disentangling reasoning from vision and language understanding. Advances in neural information processing systems **31** (2018)
47. Yuille, A., Kersten, D.: Vision as bayesian inference: analysis by synthesis? Trends in cognitive sciences **10**(7), 301–308 (2006)
48. Zhao, B., Wang, J., Ma, W., Jesslen, A., Yang, S., Yu, S., Zendel, O., Theobalt, C., Yuille, A., Kortylewski, A.: Ood-cv-v2: An extended benchmark for robustness to out-of-distribution shifts of individual nuisances in natural images. arXiv preprint arXiv:2304.10266 (2023)

- 49. Zhao, B., Yu, S., Ma, W., Yu, M., Mei, S., Wang, A., He, J., Yuille, A., Kortylewski, A.: Ood-cv: A benchmark for robustness to individual nuisances in real-world out-of-distribution shifts. In: Proceedings of the European Conference on Computer Vision (ECCV) (2022)
- 50. Zhou, X., Karpur, A., Luo, L., Huang, Q.: Starmap for category-agnostic keypoint and viewpoint estimation. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 318–334 (2018)

## Supplementary material

We provide additional results and discussions to support the experimental results in the main paper.

## 7 NeMo baseline extension

In the following, we introduce in more details how Nemo [37] can be naïvely extended to perform classification and how does it perform for this new task.

### 7.1 Extension procedure

NeMo [37] is originally designed to perform 3D-pose estimation and the class of the object is considered as known. We leverage one class-specific trained NeMo model for each object category (12 for Pascal3D+; 10 for OOD-CV). We use the exact same procedure as in the original paper [37] for training the class-specific NeMo models and for inference. Since NeMo relies on a render-and-compare approach, we can obtain the reconstruction loss from the final prediction for each candidate class which we leverage to assess the quality of the predicted 3D-pose. Finally, the class corresponding to the lowest loss corresponds to the predicted class.

### 7.2 Classification

In this section, we compare results between the previously described naïve extension of NeMo (*i.e.*, ext.-NeMo) and our approach for classification. Table S1 shows that for classification, our approach considerably outperforms the naïve extension of NeMo for Pascal3D+ and OOD-CV. In the most challenging scenarios (e.g., L3 occlusion and weather), our approach even performs more than 2x better. Besides this substantial performance improvement, the computation requirements are notably lower (both in terms of memory and temporal requirements). Hence, this demonstrates that learning the neural textures in a discriminative manner between classes is crucial to observe good classification performances.

### 7.3 3D Pose Estimation

3D Pose estimation is not affected by any changes of the extension procedure. Hence, results reported in the main paper for NeMo still stand for the extended version of NeMo.

## 8 Additional results

In this section, we supplement the findings outlined in the main paper and show results for  $\frac{\pi}{18}$  and  $\frac{\pi}{6}$  thresholds. Then we show the full results for all tasks for all corruptions of the corrupted-PASCAL3D+ dataset.

**Table S1:** Classification accuracy results on PASCAL3D+, occluded-PASCAL3D+ and OOD-CV datasets. Higher is better. We observe a significant performance improvement between the naïve NeMo extension (referred to as "ext.-Nemo") and our approach. Our approach performs better in all scenarios. Abbreviations: "cont." stands for context, "text." stands for texture, and "weat." stands for weather.

Dataset	P3D+	occluded-P3D+					OOD-CV					
Nuisance	L0	L1	L2	L3	mean	cont.	pose	shape	text.	weat.	mean	
ext.-NeMo	88.0	72.5	49.3	22.3	48.0	52.2	43.2	54.8	45.5	40.4	46.3	
NOVUM	<b>99.5</b>	<b>97.2</b>	<b>88.3</b>	<b>59.2</b>	<b>81.6</b>	<b>85.3</b>	<b>88.1</b>	<b>83.6</b>	<b>90.1</b>	<b>82.8</b>	<b>85.2</b>	

### 8.1 Classification

Tables S2 and S4 show complementary results to the ones shown in the paper. We observe similar trends for all metrics than the ones presented in Sections 1. Our approach shows much higher performances compared to current state-of-the-art (SOTA) for the classification task.

**Table S2:** Classification accuracy results on PASCAL3D+, occluded-PASCAL3D+ and OOD-CV datasets. First is highlighted in **bold**, second is underlined. L0 corresponds to unoccluded images from Pascal3D+, and occlusion levels L1-L3 are from occluded-PASCAL3D+ dataset with occlusion ratios stated in 4.1. NOVUM performs similarly in IID scenarios, while steadily outperforming all baselines in OOD scenarios. Abbreviations: "cont." stands for context, "text." stands for texture, and "weat." stands for weather.

Dataset	P3D+	occluded-P3D+					OOD-CV					
Nuisance	L0	L1	L2	L3	mean	cont.	pose	shape	text.	weat.	mean	
Resnet50	99.3	93.8	77.8	45.2	72.3	45.1	61.2	55.2	48.3	47.3	51.4	
Swin-T	<u>99.4</u>	93.6	77.5	46.2	72.4	63.0	71.4	<u>65.9</u>	61.4	<u>59.6</u>	<u>64.2</u>	
Convnext	<u>99.4</u>	95.3	81.3	<u>50.9</u>	75.8	<u>53.6</u>	61.2	<u>60.8</u>	<u>57.2</u>	<u>47.1</u>	<u>56.0</u>	
ViT-b-16	99.3	94.7	80.3	49.4	74.8	57.8	67.3	61.0	54.7	54.5	59.0	
NOVUM	<b>99.5</b>	<b>97.2</b>	<b>88.3</b>	<b>59.2</b>	<b>81.6</b>	<b>85.3</b>	<b>88.1</b>	<b>83.6</b>	<b>90.1</b>	<b>82.8</b>	<b>85.2</b>	

### 8.2 Additional results for $\frac{\pi}{18}$ and $\frac{\pi}{6}$ thresholds

Table S3 shows complementary results to the ones shown in the paper. We observe similar trends for all metrics than the ones presented in Sections 4.3. Our approach shows equivalent performances to current state-of-the-art (SOTA) for the 3D-pose estimation task even though it is not specifically designed to perform this task.



**Table S3:** Pose Estimation results on (occluded)-PASCAL3D+, and OOD-CV dataset. Pose accuracy is evaluated for error under two thresholds:  $\frac{\pi}{6}$  and  $\frac{\pi}{18}$  separately. Noticeably, our approach has equivalent performances to current SOTA for 3D-pose estimation event though it has not been specifically designed for this task. Abbreviations: "cont." stands for context, "text." stands for texture, and "weat." stands for weather.

Threshold: $\frac{\pi}{18}$											
Dataset	P3D+	occluded-P3D+				OOD-CV					
Nuisance	L0	L1	L2	L3	mean	cont.	pose	shape	text.	weat.	mean
Resnet50	33.8	22.4	15.8	9.1	15.8	15.5	12.6	15.7	22.3	23.4	18.0
Swin-T	29.7	23.3	15.6	10.8	16.6	18.3	14.4	16.9	21.1	26.3	19.8
Convnext	38.9	22.8	12.8	6.6	14.1	18.1	<b>14.5</b>	16.5	21.7	26.6	19.9
ViT-b-16	38.0	23.9	13.7	7.4	15.0	24.7	13.8	15.6	25.0	28.3	21.5
NeMo	61.0	45.0	30.7	14.6	31.8	21.9	6.9	19.5	<b>34.0</b>	30.4	21.9
NOVUM	<b>69.5</b>	<b>47.5</b>	<b>31.6</b>	<b>16.2</b>	<b>32.7</b>	<b>26.2</b>	12.1	<b>25.8</b>	32.5	<b>34.5</b>	<b>22.8</b>

Threshold: $\frac{\pi}{6}$											
Dataset	P3D+	occluded-P3D+				OOD-CV					
Nuisance	L0	L1	L2	L3	mean	cont.	pose	shape	text.	weat.	mean
Resnet50	82.2	66.1	42.1	53.8	57.8	34.5	50.5	53.1	<b>61.5</b>	60.0	51.8
Swin-T	81.4	58.5	47.3	38.8	48.2	52.3	41.1	45.7	50.1	64.9	50.9
Convnext	82.4	63.7	47.9	36.4	49.3	51.7	<b>43.4</b>	44.8	48.0	<b>65.9</b>	50.7
ViT-b-16	82.0	65.4	49.5	37.6	50.8	54.7	34.0	49.5	59.1	59.0	51.3
NeMo	86.1	75.9	63.9	45.6	62.2	50.3	35.3	49.6	57.5	52.2	51.6
NOVUM	<b>90.1</b>	<b>82.6</b>	<b>71.3</b>	<b>52.7</b>	<b>63.1</b>	<b>54.9</b>	24.8	<b>54.6</b>	<b>61.5</b>	55.3	<b>52.9</b>

### 8.3 Full corrupted-PASCAL3D+ results

Table S4 shows corrupted-PASCAL3D+ results for classification and for all corruptions. Similarly to what have been discussed previously, NOVUM significantly outperforms all baselines.

**Table S4:** Classification results for (corrupted)-PASCAL3D+. 3D-pose estimation is evaluated for error under two thresholds:  $\frac{\pi}{6}$  and  $\frac{\pi}{18}$  separately. Our approach significantly outperforms all baselines. Abbreviations: "bright." stands for brightness.

Dataset		P3D+	corrupted-P3D+												
Nuisance		L0	defocus blur	glass blur	motion blur	zoom blur	snow	frost	fog	bright.	contrast	elastic transform	pixelate	jpeg	mean
classification: $ACC \uparrow$	Resnet50	99.3	67.6	41.4	73.5	87.5	84.4	84.3	93.9	98.0	90.0	46.4	82.1	95.5	78.7
	Swin-T	99.4	60.7	37.1	70.9	81.3	88.5	91.6	95.4	97.9	92.1	56.3	79.2	95.3	78.9
	Convnext	99.4	70.1	58.7	76.5	<b>90.0</b>	<b>92.3</b>	92.9	<b>98.5</b>	<b>99.2</b>	<b>98.4</b>	67.6	84.2	<b>98.7</b>	85.6
	ViT-b-16	99.3	64.5	<b>78.1</b>	80.3	88.2	91.2	<b>94.1</b>	90.5	98.7	85.1	84.8	96.9	<b>98.7</b>	87.6
	NOVUM	<b>99.5</b>	<b>90.5</b>	65.7	<b>86.4</b>	84.2	91.2	89.5	98.4	98.4	97.1	<b>97.2</b>	<b>97.1</b>	98.4	<b>91.3</b>

#### 8.4 3D-pose initialization results

To initiate the render-and-compare process, we require an initial pose denoted as  $\alpha_{init}$ . We achieve this by pre-sampling 144 distinct feature maps and subsequently calculating the similarity between these extracted features from the image and the pre-rendered maps. The initial pose  $\alpha_{init}$  is then determined as the pose corresponding to the highest similarity between the rendered map and the image feature map.

Importantly, by utilizing  $\alpha_{init}$  as a coarse 3D pose prediction, we can achieve a remarkable computation speed of approximately 0.04 seconds per sample on a single RTX 3090 GPU. This represents an 80% reduction in computation time compared to the full pipeline of our approach. We observe in Table S5 that  $\frac{\pi}{6}$  results are consistent with the full pipeline. However, the  $\frac{\pi}{18}$  results suffer from the coarse prediction importantly. The coarse prediction performs quite well in terms of  $\frac{\pi}{6}$  but drops significantly compared to our full pipeline.

**Table S5:** 3D-Pose Estimation results for different datasets. A prediction is considered correct if the angular error is lower than a given threshold (*i.e.*,  $\frac{\pi}{6}$  and  $\frac{\pi}{18}$ ). The coarse pose initialization is quite accurate (*i.e.*, around 2% drop in performance for P3D+) when looking at the coarse metric of accuracy below  $\frac{\pi}{6}$  but is significantly lower when looking at the latter metric (*i.e.*,  $\frac{\pi}{18}$ ). Abbreviations: "Pose init." stands for Pose initialization.

Dataset	P3D+	occ- P3D+	cor- P3D+	OOD-CV	P3D+	occ- P3D+	cor- P3D+	OOD-CV
Threshold	$\pi/6$				$\pi/18$			
Pose init.	84.3	52.8	34.2	43.7	29.8	17.0	21.1	18.8
NOVUM	<b>88.2</b>	<b>63.1</b>	<b>49.1</b>	<b>52.9</b>	<b>59.1</b>	<b>32.7</b>	<b>43.4</b>	<b>22.8</b>

#### 8.5 Efficiency

For classification, our method attains real-time performance comparable to other CNN or transformer-based baselines, consistently handling over 50FPS, as detailed in Table S6. Despite variations in parameter numbers among these baselines, we find no discernible correlation between parameter count and OOD robustness. Notably, Table S6 illustrates that our approach outperforms all baselines significantly at OOD generalization, even with similar inference speeds.

In the realm of pose estimation, our model demonstrates a substantial reduction in parameter count when compared to the render-and-compare-based method NeMo [37], as indicated in Table S6. This reduction is mostly due to our class-agnostic backbone. Additionally, our inference speed surpasses NeMo by approximately 2 times. This acceleration is primarily due to the fewer steps

involved in our render-and-compare process (Ours: 30 steps, NeMo: 300 steps), driven by the observed quicker convergence towards local optima facilitated by our class-contrastive representation and better gradient during the optimization. Although CNN and transformer-based baselines exhibit higher inference speeds for pose estimation, their performance in OOD scenarios is notably lower.

In terms of floating-point operations (FLOPs), Table S6 reveals a stark contrast, with NeMo requiring a significantly higher number of FLOPs at 3619 GFLOPs, whereas our approach demands only 301 GFLOPs. The slightly increased number of operations in our approach in comparison to other CNN and transformer-based methods can be mainly attributed to the final feature matching procedure.

**Table S6:** Overview of the parameter counts and computation time and cost for each method as well as performances in OOD scenarios (*i.e.*, mean over all nuisances of the OOD-CV dataset). NeMo refers to [37]. For our approach, we show differnet values for the GFlops, for the classification and pose estimation pipeline, respectively. Abbreviations: "class." stands for classification, "pose est." stands for pose estimation.

Method	#Parameter	GFlops	Inference class. (s)	Inference pose est. (s)	Accuracy class.	Accuracy pose est.
Resnet50	84M	84.5	0.01	0.01	51.4	51.8
Swin-T	28M	60.8	0.01	0.01	64.2	50.9
Convnext	30M	91.1	0.01	0.01	56.0	50.7
ViT-b-16	86M	22.6	0.01	0.01	59.0	48.0
NeMo	996M	3619.2	NA	0.41	NA	48.0
NOVUM	83M	98.0 - 301.6	0.02	0.21	<b>85.2</b>	<b>52.9</b>

## 8.6 Ablation study

In the main paper, we conducted ablation experiments exclusively on two datasets, namely Pascal3D+ and occluded-P3D+. This choice was necessitated by computational resource constraints. It is important to note that we do not possess any compelling rationale to believe that the outcomes observed in the aforementioned datasets would significantly differ had we examined the remaining two datasets.

In the main paper, we only evaluated ablations for two datasets (*i.e.*, Pascal3D+ and occluded-P3D+) due to computation resources. We do not have any reason to think that findings made on the aforementioned datasets would be any different if studied on the two remaining datasets.

**Shape ablation** In order to evaluate our approach with a different shape, we studied in more details the performances using a spherical shape. We used the

same neural object volume for all object classes. Each neural object volume is composed of a mixture of 1281 gaussians. We followed exactly the same approach as the one described in the main paper. We trained our model for 200 epochs using contrastive learning, including the background features. On top of the results provided in the main paper, we provide some additional 3D pose estimation results in Table S7 to compare the performances of both our approach with cuboid and spherical neural object volumes.

**Table S7:** 3D-Pose Estimation results. A prediction is considered correct if the angular error is lower than a given threshold (*i.e.*,  $\frac{\pi}{6}$ , and  $\frac{\pi}{18}$ ). Higher is better. We observe that a spherical neural object volume performs well in IID scenarios but does not generalize as well as the cuboid in more difficult OOD scenarios.

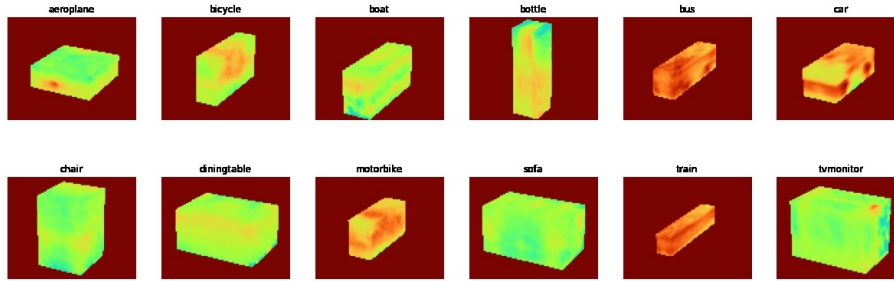
Threshold	$\pi/6$					$\pi/18$				
Dataset	P3D+	occluded-P3D+				P3D+	occluded-P3D+			
Nuisance	L0	L1	L2	L3	Mean	L0	L1	L2	L3	Mean
NOVUM-sphere	90.2	75.0	62.1	40.1	59.1	65.9	47.0	30.1	13.9	30.3
NOVUM-cuboid	90.1	82.6	71.3	52.7	69.1	69.5	55.5	40.6	22.2	39.7

**"single feature" ablation** To assess the efficacy of our 3D representation, we conducted an ablation study focused on the 3D representation itself. This involved replacing the neural object volume with a single feature vector. Hence, at the difference of Section 3, we do not define any neural object volume for this case, instead, for each class  $y$  we define the feature vector  $C'_y \in \mathbb{R}^D$ . Subsequently, we executed a contrastive learning process, randomly selecting positive features from the object in the image. We omitted the background model from consideration since the number of feature vectors aligns with the number of classes (*e.g.*, 12 for P3D+). Consequently, these features are already suitably distributed within the feature space and the background clutter model was not necessary. During the classification inference stage (where predicting the 3D pose is not feasible within the current setup), we computed features matching with the 12 feature vectors at our disposal. The predicted class corresponds to the class for which the feature vector exhibits the most significant matching with the image features.

## 9 Estimation of concentration parameters

There is no closed form solution to estimating the the concentration ( $\kappa$ ) parameters. Therefore, we set it to  $\kappa = 1$ . To ensure that setting the values to a constant, we tested the effect of approximating the parameter using a standard

method as proposed in [33]. In Figure S1, we can observe that the concentration of the learned features is slightly higher, where the object variability is low (*e.g.*, wheels, bottle), whereas it is lower for objects with very variable appearance or shape (*e.g.*, airplanes, chairs, sofa). When integrating the learned concentration parameter into the classification and pose estimation inference we observe almost no effect on the results (see Table S8). Importantly, estimating these parameters is non-trivial because of the **lack of a closed-form solution and potential imbalances among the visibility of different vertices**. This needs to be studied more thoroughly, but our hypothesis is that the learned representation compensates for the mismatch of the cuboid to the object shape and therefore the weighting of the concentration parameter does not have a noticeable effect.



**Fig. S1:** Plot of the concentration  $\kappa$  estimates for each vertex using the training dataset (range shown is 0.5 (blue) - 1 (red)).

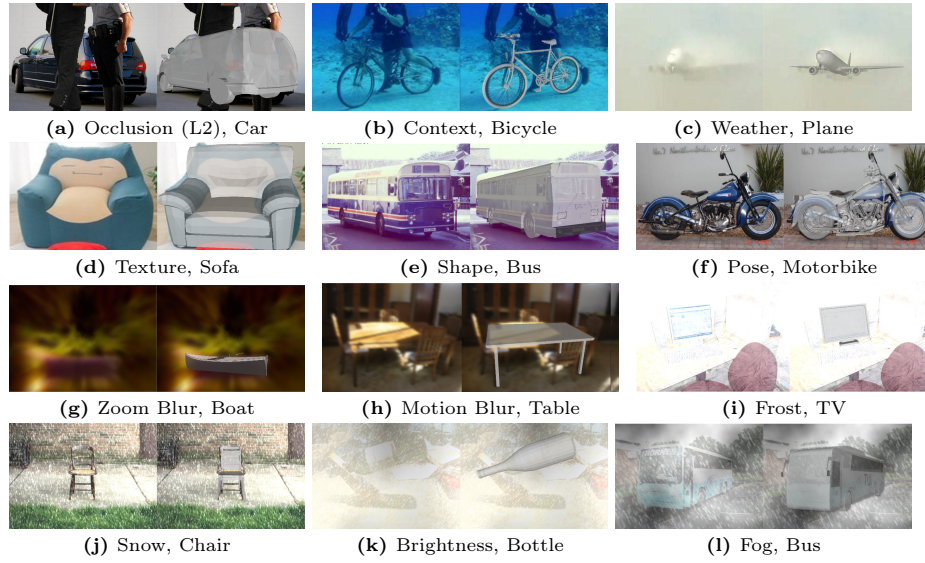
**Table S8:** Classification accuracy results on PASCAL3D+ and occluded-PASCAL3D+.

Dataset	P3D+	occluded-P3D+				
Nuisance	L0	L1	L2	L3	mean	
Ours with estimated $\kappa$	99.5	97.2	88.4	59.2	81.6	
Ours with $\kappa = 1$	99.5	97.2	88.3	59.2	81.6	

## 10 Additional visualizations

We have generated **supplementary videos** showcasing the feature matching between image features and the neural object volume features which are available at <https://genintel.github.io/NOVUM>.

Additionally, in Figure S2, we provide qualitative results. Every image is OOD data with different nuisances. We can see that these scenarios are very likely to be encountered by classification models in the real world. Given the class and 3D pose predictions originating from our approach, we overlaid a 3D CAD model on the input image. We can see how our approach successfully predicts both the object category and object pose for these challenging images.



**Fig. S2:** Qualitative results of our approach on Occluded PASCAL3D+ and OOD-CV (a-f), and on Corrupted PASCAL3D+ (g-l). Captions follow the format  $\{nuisance\}$ ,  $\{class\}$ . We illustrate the predicted 3D pose using a CAD model. Note that the CAD model is not used in our approach. All images were correctly classified by our approach but incorrectly classified by at least one baseline.