Learning Dense Object Patch Encodings from Multiple Views for Low-shot Category Generalization

Stefan Stojanov, Anh Thai, Zixuan Huang, James M. Rehg
Georgia Institute of Technology
{sstojanov, athai6, zixuanh, rehg}@gatech.edu

Abstract

A hallmark of the deep learning era for computer vision is the successful use of large-scale labeled datasets to train feature representations for tasks ranging from object recognition and semantic segmentation to optical flow estimation and novel view synthesis of 3D scenes. In this work, we aim to learn dense discriminative object representations for low-shot category recognition *without* requiring any category labels. To this end, we propose Deep Object Patch Encodings (DOPE), which can be trained from multiple views of object instances without any category or semantic object part labels. To train DOPE, we assume access to sparse depths, foreground masks and known cameras to obtain pixel-level correspondences between views of an object, and use this to formulate a self-supervised learning task to learn discriminative object patches. We find that DOPE can directly be used for low-shot classification of novel categories using local-part matching, and is competitive with and outperforms supervised and self-supervised learning baselines. Code and data available at `https://github.com/rehg-lab/dope_selfsup`

1 Introduction

Achieving high accuracy at object category recognition generally requires deep models with millions of parameters [28, 15] and million-scale datasets [13, 39]. Alleviating this data requirement is a fundamental challenge for computer vision and has driven the development of low-shot [57, 65, 19, 76, 7] and self-supervised [45, 27, 6, 24] methods for learning object representations. Despite the increasing interest in low-shot and self-supervised learning, only a few works tackle the case where multiple unlabelled views of object instances are available for self-supervised learning of discriminative object representations [29, 34]. In addition, their primary focus is not on low-shot generalization. Using multi-view data to learn representations for low-shot categorization in a self-supervised manner is of practical interest, as such data can potentially be obtained from robots moving around and handling objects [21, 25] or by images and videos collected by humans using mobile devices [77, 53].

The goal of this paper is to address this gap by using multiple-views of many object instances for training, combined with some additional geometric information about the views, to learn representations that can be directly used for low-shot object category recognition. We assume access to sparse depth, known camera pose/calibration, and foreground segmentation during training, *with no access to category labels*. We use this information to formulate a self-supervised representation learning task. In practice, the additional data necessary for self-supervision can be obtained in controlled indoor settings using robotic arms equipped with RGB-D sensors, by postprocessing RGB-D sensor data and using forward kinematics to obtain the camera pose [21, 25]. Otherwise, in less constrained settings where the data is videos of objects captured using mobile devices, the data needed for self-supervision can be obtained from a Structure-from-Motion (SfM) and Multi-View Stereo (MVS) pipeline such as COLMAP [55, 56]. Note that we do not assume access to any sort of category-level object or part

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

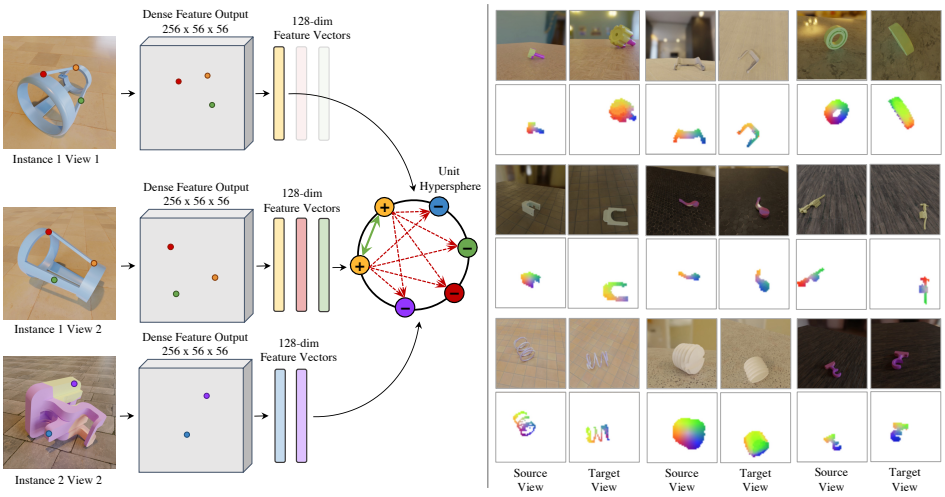

Figure 1: **Left:** Our contrastive learning formulation: We form positives between features at pixel locations of the same 3D point on the object surface, and negatives pixels from different points on the object surface. **Right:** For each feature corresponding to a point on the predicted segmentation mask in the source view we assign a color. For each point in the predicted segmentation for the target views, we assign the color of the highest cosine similarity feature from the source view. We see that our model learns dense object encodings which are consistent across many views of an object.

annotations or any object attribute information for representation learning. At test time, our method only uses single-view RGB images.

To solve this self-supervised learning task, we propose Deep Object Patch Encodings (DOPE). DOPE uses pairs of object views to learn a dense, CNN-based local feature encoding that will map a patch on the surface of the object that is visible in both views to the same feature vector, and different surface patches to different feature vectors (see Figure 1). We define an object patch as a local neighborhood on the 3D surface of a rigid object. Surprisingly, we find that such a representation, trained only on pairs of views of the *same* object instances, *generalizes across novel object categories* (illustrated in Figures 6 and 5). This is demonstrated by the ability to encode the same object parts, e.g. a motorcycle tire, or an airplane wing, with the same learned feature vector across multiple instances of a category. This generalization ability allows us to directly use the DOPE representation (trained only on local patch matches with no category labels) for low-shot object recognition.

DOPE is trained using a local contrastive learning loss which requires correspondences between views to form positive and negative pairs. To find such pairs, we use sparse depth map and camera information. Similar contrastive formulations have been used before to train instance-specific or class-specific dense object descriptors [21, 25] from RGB images for robotics applications, but did not address low-shot recognition. We hypothesize that learning dense surface patch similarity at the local feature level forces the model to learn to encode all visible object parts in two views, and find that such a representation is useful for low-shot object categorization.

We find that DOPE is competitive and in some cases outperforms both self-supervised methods [29] and fully supervised low-shot methods [68, 62, 76] on the synthetic ModelNet [71], ShapeNet [5] and the real-world CO3D [53] datasets. We further find that DOPE can learn a representation using objects from the large, uncurated 3D object dataset ABC [37], that directly transfers to low-shot category recognition. In summary, we make the following contributions:

- Deep Object Patch Encodings (DOPE), a novel self-supervised learning approach based on learning object shape correspondences, generalizes to low-shot object categorization.
- The first study of how a large, unstructured object instance dataset like ABC [37] can be used effectively to learn representations for low-shot category recognition from images.
- Multiple datasets with high photorealism rendered from existing 3D assets that will enable future research in learning representations using multi-view information.

2 Related Work

2.1 Low-shot Learning for Object Recognition

The goal of low-shot learning is to build models that can learn how to classify images based on a few labeled samples. These are typically trained on a large base class dataset, and their low-shot generalization ability is evaluated on many low-shot episodes with a few labeled training images and testing queries from novel, unseen classes. Such models can be categorized based on how the base class data is used, and whether the model's parameters are updated during the low-shot phase. Based on base class data, the *meta learning*-based methods [57, 32, 72, 19] sample low-shot episodes from the base classes to train under the same setting as during evaluation. In contrast, the *whole-classification*-based techniques [68, 62] train the backbone feature representation as a classifier on the base classes, whereas [72, 10, 22, 43] further fine-tune the representation with a meta-learning procedure. Based on whether the model parameters are updated, *optimization*-based techniques [54, 19, 20] aim to learn a representation that can be fine-tuned using a few samples while *metric learning*-based techniques [68, 72, 10, 62] learn a discriminative metric space that will directly transfer to low-shot episodes of novel classes. Our self-supervised representations do not require any labels during training, and like metric learning techniques, can be directly applied to novel classes.

2.2 Self-Supervised Learning

Self-supervised learning aims to formulate tasks that exploit the underlying structure of the data rather than category labels to train deep network-based representations suitable for downstream tasks.

Self-Supervised Learning for Object Classification Early works formulated proxy tasks such as colorization [78], predicting relative positions of patches [14, 45] or predicting rotations [23]. More recently, methods based on instance-level comparisons have led to significant performance improvements. These are trained to map multiple views (usually multiple augmented versions of an image) as the same feature vector under some similarity measure, and different images as different vectors. SimCLR [6] and MoCLR [61] consider augmentations of the same underlying image as positives and other images in the minibatch as negatives, whereas the MoCo-based [8, 27] approaches use a continuously updating queue of negative features. The methods presented in [61, 8, 27, 3, 4] all use a second momentum-updated encoder rather than using the same backbone to encode multiple views of an instance. Our proposed approach makes use of a momentum encoder as well. Further, some works like [24, 9, 3, 4] achieve superior downstream performance for classification without requiring any explicit negatives, while others [26] obtain high performance by revisiting image reconstruction as a proxy task using vision transformers [15].

Self-Supervised Learning for Dense Vision Tasks The main ideas from self-supervised contrastive learning for classification have been extended to learn local, dense, pixel-level representations for tasks like semantic segmentation, keypoint and object part learning. For segmentation and detection, [75, 67, 73, 47] apply contrastive losses at the pixel level, treating features corresponding to individual pixels or pooled regions as positives or negatives, or alternatively use clustering-based [79] losses to fine-tune visual transformers.For keypoint learning, Cheng et al. [12] use a dense contrastive strategy, while Novotny et al. [46] propose a probabilistic matching loss. In addition, SCOPS [33] is a self-supervised technique for object part segmentation of fine grained categories. All of these prior works are trained using pairs of images that have undergone 2D geometric transforms, while we use images corresponding to different camera positions and focus on low-shot categorization.

Self-Supervised Learning using 3D Geometry and Multi-View Images Self-supervised learning has been studied in settings where multi-view RGB-D information (images and in-camera dense 3D Point clouds) is available [31, 74, 16, 49, 38, 21]. First, [31, 17, 40, 38, 49] use pixel-level correspondences between views of a scene, where a positive pair of pixels represents the same 3D point in the world projected in each image. Such pixel-level correspondences have been used to align [1] image and depth-based scene representations [31, 17, 40] for point cloud registration [17] and scene segmentation [40, 30], or to learn 3D CNN-based scene representations from multiple views [38, 49]

[1] Here by align we mean make corresponding features between the 2D and 3D representations have high similarity.

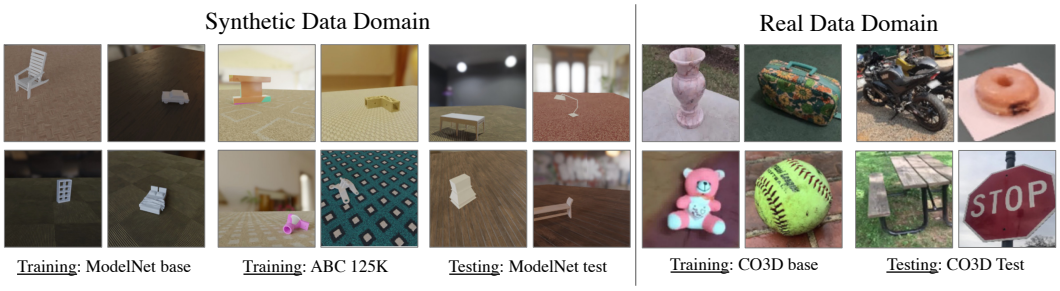

Figure 2: **Left:** Synthetic data rendered from ModelNet and ABC (first and second columns) for self-supervised learning of representations and low-shot testing data from ModelNet (third column). **Right:** Real data from CO3D for self-supervised representation training and low-shot testing.

for self supervised object tracking and few-shot style/shape learning. Florence et al. [21] use RGB images to learn dense object descriptors for robotics applications. In comparison, our goal is to use correspondences between multi-view RGB images of objects to learn representations for low-shot object classification. Other works [74, 30] directly train on 3D point clouds. Last, assuming only access to multiple RGB views, [34] uses an encoder/decoder architecture to predict images of different viewpoints from a single image, whereas VISPE [29] formulates an instance discrimination task. While we share with these works the use of self-supervised representation learning from multiple RGB views for object recognition, we focus on low-shot object classification. We show that our approach outperforms VISPE [29]-like instance level representations on multiple datasets.

2.3 Object Classification on Synthetic Datasets

Object recognition has been studied using 3D datasets such as ModelNet40 [71] converted into point clouds [50, 52, 11, 69], voxels [51, 71], or rendered into images [60, 51, 18, 44, 29, 70, 34]. We use photorealistic rendering to generate data, and conduct the first study to investigate the use of the large unstructured ABC [37] dataset for representation learning in low-shot object recognition.

3 Learning Deep Object Patch Encodings - DOPE

We aim to learn a representation based solely on multi-view images of object instances that can directly generalize to low-shot object category recognition tasks. For training, we assume access to relative pose between views of object instances, foreground/background segmentation and sparse depth. However, for low-shot generalization at test time, we only assume access to single RGB images. We do not make any assumptions about the semantic structure of the data for representation learning, or about the availability of any semantic labels such as explicit parts or attribute information.

In this section we describe our approach for contrastive learning of object patches from multiple views that can generalize to object categories, and present a simple local nearest neighbor-based method to utilize this learned representation for low-shot classification.

3.1 Local Object Representation Learning

Assume we have a dataset $\mathcal{D}$ of object instances o_i, where each object has N calibrated views with masks and known camera intrinsics and extrinsics. Denote the two views of one object as $\mathbf{v}_1^{o_1}$ and $\mathbf{v}_2^{o_1}$. If these two viewpoints are sufficiently close, there will be points on the surface of the object, e.g. one leg of a chair, that will be visible in both views. This means that the same point on the object surface, will get projected to pixel coordinates (u_1, v_1) in $\mathbf{v}^{o_1}$ and (u_2, v_2) in $\mathbf{v}_2^{o_1}$. Let $f : \mathbb{R}^{H \times W \times 3} \rightarrow \mathbb{R}^{h_f \times w_f \times D}$ be a learnable feature map that maps RGB images to a 2D feature grid with D channels and h_f and w_f height and width. Let $\tilde{u}_k$ be the location in the lower spatial dimensional feature output corresponding to u_k. Our goal is to learn f so that the feature vectors for the corresponding pixel locations $\mathbf{z}_1^{o_1} = f(\mathbf{v}_1^{o_1})[\tilde{u}_1, \tilde{v}_1, :]$ and $\mathbf{z}_2^{o_1} = f(\mathbf{v}_2^{o_1})[\tilde{u}_2, \tilde{v}_2, :]$ are so that $\mathbf{sim}(\mathbf{z}_1^{o_1}, \mathbf{z}_2^{o_1}) = 1$, where $\mathbf{sim}$ is some similarity metric, in our case normalized dot product. Conversely, we want $\mathbf{sim}(\mathbf{z}_1^{o_1}, \bar{\mathbf{z}}_2^{o_1}) = 0$, where $\bar{\mathbf{z}}_2^{o_1}$ is a feature from a pixel projected from the object surface in $\mathbf{v}_2^{o_1}$ that is not in correspondence. Further, for a feature $\bar{\mathbf{z}}_2^{o_2}$ extracted from some different object o_2, we want $\mathbf{sim}(\mathbf{z}_1^{o_1}, \bar{\mathbf{z}}_2^{o_2}) = 0$. Note that $\mathbf{z} \in \mathbb{R}^D$. In other words, following the chair leg

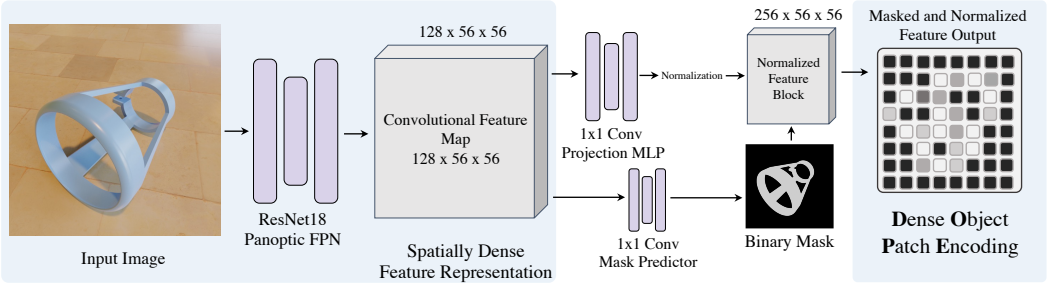

Figure 3: Our architecture consists of a ResNet18 [28] followed by a Panoptic Feature Pyramid network [35] to output a 2D grid of convolutional features. We use an additional branch from these features to predict a binary mask. After applying a series of 1x1 convolution layers, we perform channel-wise unit vector normalization and masking with the binary mask to obtain the final feature output we use for contrastive training and at test time

example, we want f to produce the same feature in the 2D feature grid for a point on the chair leg in any viewpoint where that chair leg is visible (Please refer to Fig. 1 for illustration).

To learn such a representation, we sample a batch of objects $\mathcal{B}$, and for each object $o_k \in \mathcal{B}$ we sample an image pair $\mathbf{v}_1^{o_k}$ and $\mathbf{v}_2^{o_k}$. For each view pair, we sample a set $\mathcal{C}$ of n pixels on the object surface in $\mathbf{v}_1^{o_k}$ using farthest point sampling, and find the corresponding pixels in $\mathbf{v}_2^{o_k}$ [2]. For each pixel in $\mathcal{C}$ we therefore obtain the feature vector $\mathbf{z}_1^{o_k}$ and the corresponding $\mathbf{z}_2^{o_k}$ whose similarity $\mathbf{sim}(\mathbf{z}_1^{o_k}, \mathbf{z}_2^{o_k})$ we want to maximize. Concurrently, we want to minimize: (1) $\mathbf{sim}(\mathbf{z}_1^{o_k}, \bar{\mathbf{z}}_2^{o_k})$ for any feature $\bar{\mathbf{z}}_2^{o_k}$ from a non-corresponding pixel on the object and (2) $\mathbf{sim}(\mathbf{z}_1^{o_k}, \bar{\mathbf{z}}_2^{o_j})$ for any feature $\bar{\mathbf{z}}_2^{o_j}$ from some other object wher $o_j \in \mathcal{B}$, where $j \neq k$. To achieve this we minimize a normalized and temperature-scaled cross entropy loss [58, 8] for each pair of positive correspondences

$$\ell = -\log \frac{\exp\left(\mathbf{sim}(\mathbf{z}_1^{o_k}, \mathbf{z}_2^{o_k})/\tau\right)}{\sum_{\bar{\mathbf{z}}_2^{o_k}} \exp\left(\mathbf{z}_1^{o_k}, \bar{\mathbf{z}}_2^{o_k}\right)/\tau + \sum_{j,j\neq k} \sum_{\bar{\mathbf{z}}_2^{o_j}} \exp\left(\mathbf{z}_1^{o_k}, \bar{\mathbf{z}}_2^{o_j}\right)/\tau}, \tag{1}$$

where τ is a temperature parameter controlling the how peaky the softmax output is. This formulation is commonly used in contrastive learning. We apply this loss for all sampled points in each image pair and refer to it as $\mathcal{L}_{\text{corr}} = \sum_{\mathcal{C}} \ell$.

To avoid learning to encode the background, we learn an additional module $h : \mathbb{R}^{h_f \times w_f \times D} \rightarrow \mathbb{R}^{h_f \times w_f}$ to predict binary masks $\widehat{\mathbf{m}}$ which we train with binary cross entropy $\mathcal{L}_{\text{mask}} = BCE(\widehat{\mathbf{m}}, \mathbf{m})$. We then apply this mask to the final output of f (refer to Fig. 3) to mask out the background. We optimize the sum of both losses with equal weight. Fig. 1 contains qualitative examples of how our proposed approach matches object shape across viewpoints.

3.2 Low-Shot Object Recognition using DOPE

Given that we have the means to encode the same object part from different images (see Figures 5 and 6), we present a simple nearest neighbor-based method to make use of this representation for low-shot category recognition. This method is outlined in Figure 4, and we give a formal description next. Let $\mathbf{Z}_q \in \mathbb{R}^{D \times h_f \times w_f}$ be a feature map of a query image that we wish to classify given a labelled support set of N images. Our goal is to find the image from the support set which is the most similar to the query, and assign its label to the query image.

We first sample k pixel locations from the predicted foreground of the query image using farthest point sampling and extract their corresponding feature vectors $\mathbf{z}_q^1, \ldots, \mathbf{z}_q^k$. We then extract complete feature grids for each of the shot images and flatten them into matrices $\mathbf{Z}_s^1, \ldots, \mathbf{Z}_s^N \in \mathbb{R}^{(h_f \cdot w_f) \times D}$. Note that $\mathbf{z}$ vectors and the columns of the $\mathbf{Z}$ matrices are unit normalized. The product $\mathbf{s}_q^i = \mathbf{Z}_s \mathbf{z}_q^i$ where $\mathbf{s}_q^i \in \mathbb{R}^{h_f \cdot w_f}$ is the similarity of the query feature vector $\mathbf{z_q}$ at the i-th sampled pixel location and all features $\mathbf{Z}_s^n$ from the n-th support image. We compute $\mathbf{s}_q^i$ for all k points in the query image, and use $\sum_{i=1}^{k} \max(\mathbf{s}_q^i)$ as the score for that query/support image pairs. We take the label from the

[2] We obtain these correspondences using the segmentation masks, known camera intrinsics and extrinsics and depths. We also use the depth information to avoid sampling points that are occluded.

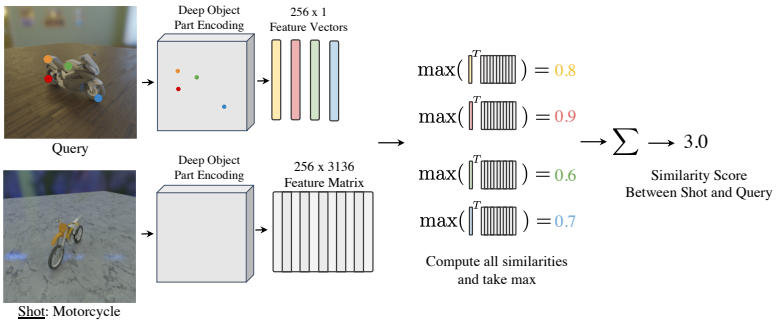

Figure 4: To determine a similarity score between a labelled shot and query image, for each query image, we first use farthest point sampling to randomly choose k pixels on the object surface, based on a predicted segmentation mask (in this example $k = 4$). We then extract features corresponding to those points from the query, and a full feature grid for the shot. After computing the cosine similarity between all query and each shot features, we find the maximum similarity score between each shot and all query features. We sum these scores for the final similarity score.

support image with the highest score and assign it to the query. When there are multiple shot images per category available, we perform a 1-nearest neighbor classification.

4 Implementation Details

The backbone of DOPE is based on a ResNet18 [28] with a Panoptic Feature Pyramid network [35], taking as input a $3 \times 224 \times 224$ image and outputting a $256 \times 56 \times 56$ 2D feature block. This backbone formulation follows the approach in VADeR [48]. The FPN output is then sent through a set of 1x1 convolution layers with dimension 128-1024-1024-256. During training, we use $n = 32$ pixel locations on the object surface for each image pair, and use a batch size of 256 view pairs. DOPE is trained using AdamW [41] and an initial learning rate of 10^{-4} and weight decay of 10^{-2}, for 3000 epochs, using a cosine learning rate annealing schedule. For each view pair, one image is sent to a momentum-encoder backbone as in [27]. At test time, we use $k = 20$ pixel locations for our local nearest neighbor-based classification approach. During training, we use the known segmentation mask to randomly remove the background and perform flipping, color jittering and additional photometric augmentation, building on [21]. We empirically found that this random background removal significantly improves the model's generalization ability, forcing it to learn to encode cues from the foreground object. The self-supervised baselines and low-shot baselines in the Experiments (Section 5) are trained using the same background masking strategy we use for DOPE to ensure a fair comparison. Further implementation details are provided in the Appendix.

5 Experiments

In this section we demonstrate DOPE's *superior* low-shot generalization ability of in comparison to both supervised algorithms and strong self-supervised baselines on synthetic and real datasets. A surprising finding is that our model outperforms few-shot baselines trained on category labels for ModelNet40, and delivers comparable performance for CO3D. For a thorough comparison, all models are evaluated over five different randomly chosen validation and testing splits, where each validation split is used to find the best checkpoint for each test set. In all our experiments, parentheses show confidence intervals based on 2.5K low shot episodes. We include further results and ablation studies in the Appendix.

5.1 Datasets

We use Blender [1] with the Cycles ray-tracing engine to create synthetic data, rendering 20 views per object with varying camera distance, azimuth and elevation. Objects are on top of a plane with a PBR material and illuminated using image-based lighting from an HDRI environment map. These assets were collected from PolyHaven [2].

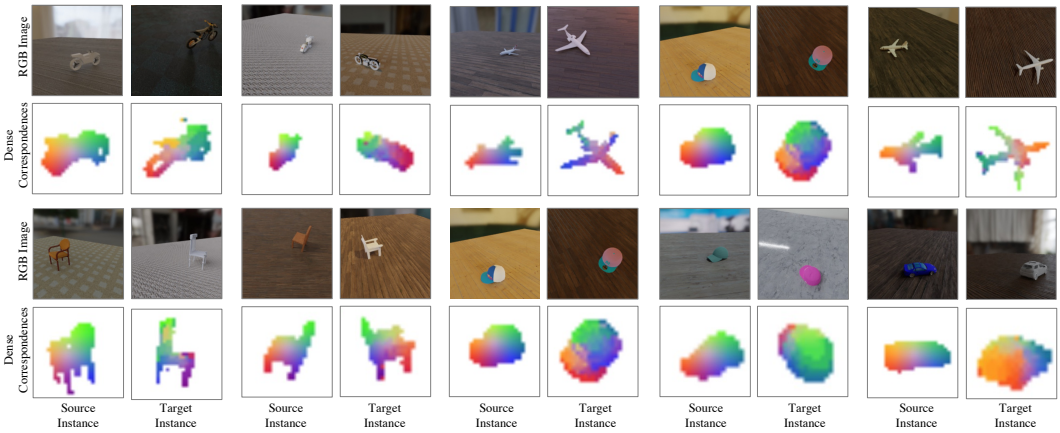

Figure 5: Category-level dense correspondences for ShapeNet [5] emerge from training on the ABC [37] instance dataset. For each feature corresponding to a point on the predicted segmentation mask in the source view we assign a color. For each point in the predicted segmentation for the target views, we assign the color of the highest cosine similarity feature from the source image. We see that our model learns dense object descriptors which transfer across instances of a category. We zoom into the object for the dense visualization images to improve visibility.

Following [34, 29, 59] and convention in low-shot learning we partition ModelNet [71], ShapeNet [5] and CO3D [53] into disjoint sets of base classes—used for representation training, validation classes—used for early stopping and hyperparameter tuning, and test classes—used to test low-shot category recognition. Further, we use the 3D object instance dataset ABC [37] to investigate whether a large and diverse set of object instances without any underlying category structure can be used to learn representations useful for low-shot category generalization. For illustration see Figure 2. Following are additional dataset details:

- **ModelNet40-LS**[71] is a dataset of 3D object categories split into disjoint sets of 20 training, 10 validation and 10 test categories. We use the 20 base classes with labels for training the supervised models, and without labels for the self-supervised models.
- **ShapeNet50-LS**[5] is a dataset of 3D object categories split into 25 training, 10 validation and 20 test categories. We use the 20 base classes with labels for training the supervised models, and without labels for the self-supervised models.
- **ABC**[37] is a dataset of 3D object instances from 3D printing online repositories without any category structure. We randomly select a set of 115K instances from this dataset for training, and directly evaluate transfer to the ModelNet and ShapeNet low-shot test sets.
- **CO3D-LS**[53] (Common Objects in 3D) is a dataset which consists of 20K crowd-sourced videos of objects from 51 categories, where the videos are collected by a person moving around an object using a mobile device. We split CO3D in 31 base, 10 validation and 10 test classes. The dataset is post-processed using COLMAP [56, 55] to estimate the cameras and reconstruct the scene and object geometry. It also includes object segmentation masks estimated using PointRend [36].

5.2 Baseline Methods

We provide a brief overview of the supervised and self-supervised baseline algorithms we evaluate against. All baselines are implemented using the same ResNet18[28] backbone and are tuned to our setting (details are included in the Appendix).

- **SimpleShot**[68] is a simple and competitive low-shot baseline. A CNN-based backbone is trained with cross-entropy on the base classes, and tested with nearest class mean classification.
- **RFS**[62] is a strong but simple baseline, trained with cross-entropy on the base classes, and a logistic regression-based classifier trained on the support data in each low-shot episode.
- **FEAT**[76] finetunes a transformer-based [64] set-to-set function on top of a cross-entropy-trained backbone. Methods that outperform FEAT on miniImageNet like COSOC [42] do so by eliminating learning classification by focusing on the background. As there is no correlation between classes and backgrounds in our synthetic setting, FEAT represents the state of the art in low-shot learning in our synthetic data comparisons.

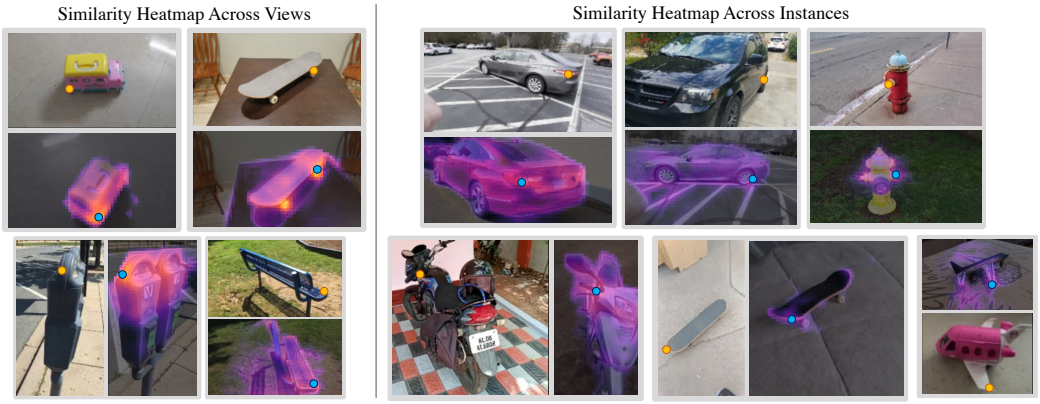

Figure 6: Similarity heatmap for the query point (yellow) **Left:** across different views of the same object and **Right:** across different object instances from CO3D. The blue point on the heatmap-overlaid image indicates the predicted correspondence of the query point. Note that these objects are from validation or test categories, which are different from the training categories.

Table 1: Results on image-only low-shot recognition on the **ModelNet40-LS**. DOPE trained on ABC outperforms supervised baselines and the global self supervised learning approaches. Dashed line separates supervised and self-supervised methods.

	5-way		10-way	
Episode Setup →	1-shot	5-shot	1-shot	5-shot
SimpleShot [68]	56.55 ±0.42	69.87 ±0.32	41.27 ±0.24	54.84 ±0.18
RFS [62]	57.31 ±0.40	73.77 ±0.33	42.22 ±0.34	59.97 ±0.18
FEAT [76]	57.46 ±0.39	71.73 ±0.32	41.72 ±0.24	57.84 ±0.18
SupMoCo [43]	55.32 ±0.40	71.82 ±0.33	39.87 ±0.23	57.15 ±0.17
VISPE [29]	56.27 ±0.44	67.76 ±0.35	40.41 ±0.26	51.97 ±0.18
VISPE++- SimSiam [9]	53.83 ±0.29	69.86 ±0.25	39.84 ±0.17	54.34 ±0.13
VISPE++ - MoCoV2 [27]	57.05 ±0.42	71.81 ±0.35	43.23 ±0.25	59.78 ±0.18
DOPE (ours)	57.51 ±0.44	70.44 ±0.36	42.73 ±0.26	55.52 ±0.19
VISPE++ - SimSiam [9] - ABC	60.24 ±0.28	**76.55 ±0.22**	47.02 ±0.18	**64.51 ±0.13**
VISPE++ - MoCoV2 [27] - ABC	61.07 ±0.41	75.96 ±0.32	47.67 ±0.25	63.27 ±0.18
DOPE (ours) - ABC	**62.76 ±0.43**	**76.86 ±0.33**	**49.39 ±0.26**	**64.77 ±0.18**
DOPE (ours) - ABC - [66] pred. mask	57.63 ±0.40	71.62 ±0.31	43.43 ±0.24	57.72 ±0.18

- **SupMoCo**[43] uses a labeled base-class dataset to perform contrastive training, and trains a cosine-classifier [22] for each low-shot episode. We finetune a logistic regression-based classifier as in RFS due to the smaller episode sizes in our setting, upon consulting the authors. We found that it results in improved performance. SupMoCo is the state of the art for low-shot learning on MetaDataset [63].
- **VISPE**[29] is a self-supervised baseline that uses a temperature-scaled cross-entropy to learn global features so that two views of the same object are positives and two views of different objects are negatives. We use a nearest class mean classifier in the learned global feature space for low-shot classification as we found that it gives the highest performance.
- **VISPE++** We improve on the original VISPE implementation by applying more recent contrastive learning techniques based on either MoCoV2 [8] or SimSiam [9], resulting in stronger baselines. We use a 1-nearest neighbor classifier in the learned global feature space for low-shot classification as we found that it gives the highest performance.

5.3 DOPE trained on ABC outperforms supervised and self-supervised baselines on ModelNet and ShapeNet

We evaluate DOPE's ability to directly perform low-shot classification without any category supervision during representation learning on synthetic data, and demonstrate its improved low-shot generalization ability over both supervised and self-supervised baselines when trained on the ABC dataset. The results are presented in Table 1 and Table 2. We observe that when trained on ABC without any category labels, DOPE outperforms the self-supervised baselines. Note that both the SimSiam [9] and MoCoV2 [27] versions of VISPE++ and our proposed DOPE outperform supervised baselines by a large margin when trained on ABC.

Table 2: Results on image-only low-shot recognition on the **ShapeNet-LS**. DOPE trained on ABC outperforms supervised baselines and outperforms the global self supervised learning approaches. Dashed line separates supervised and self-supervised methods.

	5-way		10-way	
Episode Setup →	1-shot	5-shot	1-shot	5-shot
SimpleShot [68]	58.07 ±0.45	70.06 ±0.39	43.00 ±0.29	55.96 ±0.25
RFS [62]	57.93 ±0.46	73.23 ±0.37	43.08 ±0.28	59.71 ±0.25
FEAT [76]	57.83 ±0.45	71.81 ±0.37	42.92 ±0.29	58.95 ±0.25
VISPE	57.69 ±0.47	68.65 ±0.39	42.43 ±0.29	54.00 ±0.24
VISPE++ - SimSiam [9]	53.22 ±0.30	66.73 ±0.31	39.12 ±0.19	52.25 ±0.17
VISPE++ - MoCoV2 [27]	55.83 ±0.42	69.11 ±0.36	41.14 ±0.28	54.86 ±0.25
DOPE (ours)	58.64 ±0.48	70.43 ±0.38	44.26 ±0.30	56.07 ±0.26
VISPE++ - SimSiam [9] - ABC	58.48 ±0.32	72.19 ±0.34	44.59 ±0.20	59.17 ±0.18
VISPE++ - MoCoV2 [27] - ABC	58.99 ±0.46	72.26 ±0.39	44.96 ±0.30	59.02 ±0.26
DOPE (ours) - ABC	**62.00 ±0.48**	**73.55 ±0.38**	**47.93 ±0.31**	**60.72 ±0.27**

Table 3: Results on image-only low-shot recognition on **CO3D-LS**. DOPE outperforms the global self supervised learning-only approaches. Dashed line separates supervised and self-supervised methods.

	5-way		10-way	
Episode Setup →	1-shot	5-shot	1-shot	5-shot
SimpleShot [68]	62.44 ±0.41	79.18 ±0.29	49.43 ±0.26	68.32 ±0.17
RFS [62]	62.59 ±0.41	**81.50 ±0.26**	50.16 ±0.26	**71.89 ±0.16**
FEAT [76]	**64.48 ±0.43**	79.04 ±0.29	**51.58 ±0.26**	69.66 ±0.16
SupMoCo [43]	62.34 ±0.43	76.86 ±0.44	48.79 ±0.27	67.84 ±0.16
VISPE	54.68 ±0.43	70.12 ±0.35	40.85 ±0.25	56.46 ±0.20
VISPE++ (SimSiam)	56.02 ±0.27	74.29 ±0.21	43.45 ±0.17	62.88 ±0.12
VISPE++ (MoCoV2)	60.25 ±0.41	76.30 ±0.30	47.09 ±0.26	64.98 ±0.18
DOPE (ours)	61.77 ±0.44	75.16 ±0.32	48.07 ±0.27	62.97 ±0.17

We also investigate removing the assumption of known foreground masks for training DOPE, by extracting foreground masks for the ABC dataset using the off-the-shelf unsupervised instance segmentation method FreeSOLO [66]. The result is presented in the bottom row of Table 1. We observe that even when the assumption of known foreground masks is removed, DOPE's performance drops but remains competitive with supervised baselines when trained on ABC.

5.4 DOPE trained CO3D outperforms other self-supervised baselines

We also evaluate DOPE's low-shot generalization ability on real data, and observe that it has improved low-shot generalization ability over other self-supervised baselines. The results are presented in Table 3. We observe that DOPE can successfully be trained on real data with estimated scene geometry, and that it outperforms self-supervised learning baselines that do not perform any local feature learning in the 1-shot scenario by up to 1.5%-points.

5.5 DOPE Qualitative Results

We present qualitative results for dense object correspondences on ShapeNet [5] in Figure 5 and for CO3D [53] in Figure 6. DOPE learns to map the same object parts across different views of the same instance and generalizes further to different object instances in different viewpoints (the front and back legs of the chairs are correctly matched in Figure 5 and the back light of the cars in Figure 6). Note that for CO3D the local representations were trained on a different set of object categories, and were trained on the ABC dataset for ShapeNet. Further, we observe an intriguing property that the model is uncertain for object parts that are similar, as shown in the similarity heatmap in Figure 6 (e.g. the wheels of the skateboard). This highlights the ability of DOPE to successfully encode local features of objects and generalize to unseen data.

5.6 Negative Sampling Strategy

We present an ablation study in Table 4 to understand the effect of how negatives are sampled on low-shot classification performance. Recall from the left side of Figure 1 and Section 3.1 that

Table 4: Ablation study over different strategies for obtaining negative samples for contrastive training. We observe that is essential to sample from the second object view and from other object instances in the batch to train the best representation. All models are trained on ABC and evaluated on ModelNet40-LS.

	5-way		10-way	
Episode Setup →	1-shot	5-shot	1-shot	5-shot
DOPE - ABC - 2nd view + other objects	**62.76** ±0.43	**76.86** ±0.33	**49.39** ±0.26	**64.77** ±0.18
DOPE - ABC - other object only	61.51 ±0.42	75.54 ±0.32	47.95 ±0.26	62.65 ±0.18
DOPE - ABC - 2nd view only	43.50 ±0.39	56.58 ±0.28	30.58 ±0.25	39.06 ±0.17

Table 5: We investigate the tradeoff between using two views of an object corresponding to two different camera positions in the 3D world against two "views" generated by augmenting a single image twice. We find that using two physically different viewpoints during training has a high impact on generalization ability.

	5-way		10-way	
Episode Setup →	1-shot	5-shot	1-shot	5-shot
DOPE - ABC - multi-view augmented image pairs	**62.76 ±0.43**	**76.86 ±0.33**	**49.39 ±0.26**	**64.77 ±0.18**
DOPE - ABC - single-view augmented image pairs	38.36 ±0.43	51.68 ±0.33	25.78 ±0.26	37.19 ±0.18

while positives can only be obtained by comparing corresponding pixels in two views of one object, negatives can come either from non-corresponding points in the second view of an object, or from other objects. We observe that it is essential to both sample negatives from the second view and from the other objects in the batch to obtain best representation.

5.7 Training with Multiple 3D Views of Objects is Essential

While prior dense contrastive learning works [48, 67, 73] generate training pairs by augmenting a single 2D image of an object, our method is trained using two images corresponding to two different camera placements in 3D space. To understand the importance of different 3D viewpoints of an object, we train DOPE using the same loss, but generate image pairs by applying two augmentations of the same image instead. We find that augmenting the same image to generate view pairs is insufficient for low-shot generalization.

6 Limitations and Discussion

The main limitation of our proposed approach is its requirement of estimated scene geometry in order to derive pixel-level correspondences between multiple views of object instances. While this information allows us to train better self-supervised representations that do not require category labels, it comes at a computational cost when applied to real datasets. Future work may include developing means to apply contrastive learning at the local level but without requiring explicit pixel level correspondences, potentially making use of the temporal contiguity of videos.

Negative Societal Impact Contrastive training generally requires long training runs with multiple GPUs, which has potential for negative environmental impact. This may be addressed in the future through improvements in chip design and optimization of deep models.

7 Conclusion

Progress in few-shot and self-supervised learning is essential for overcoming the current requirements for large labeled datasets. This paper presents a self-supervised method that learns from multiple views of object instances and can recognize novel categories based on a few labels. Our findings are validated on both synthetic and real datasets. Qualitatively, we observe that our proposed approach can match semantic elements on the object surface (e.g. tires, chair legs) in a way that generalizes to novel object categories. Developing means to perform local contrastive learning without explicit multi-view pixel-level correspondences is an exciting direction for future work.

8 Acknowledgement

This work was supported by the NIH R01HD104624-01A1 and partly funded by a gift from Facebook. We thank Miao Liu and Wenqi Jia for helpful feedback and discussion.

References

[1] Blender, https://blender.org/.

[2] Poly haven, https://polyhaven.com/.

[3] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in Neural Information Processing Systems*, 33:9912–9924, 2020.

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

[5] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015.

[6] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[7] Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, and Jia-Bin Huang. A closer look at few-shot classification. In *International Conference on Learning Representations*, 2018.

[8] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020.

[9] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15750–15758, 2021.

[10] Yinbo Chen, Zhuang Liu, Huijuan Xu, Trevor Darrell, and Xiaolong Wang. Meta-baseline: Exploring simple meta-learning for few-shot learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9062–9071, October 2021.

[11] Yunlu Chen, Vincent Tao Hu, Efstratios Gavves, Thomas Mensink, Pascal Mettes, Pengwan Yang, and Cees GM Snoek. Pointmixup: Augmentation for point clouds. In *Proceedings of the European Conference on Computer Vision (ECCV) 2020*.

[12] Zezhou Cheng, Jong-Chyi Su, and Subhransu Maji. On equivariant and invariant learning of object landmark representations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9897–9906, 2021.

[13] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[14] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE international conference on computer vision*, pages 1422–1430, 2015.

[15] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[16] Mohamed El Banani and Justin Johnson. Bootstrap your own correspondences. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6433–6442, 2021.

[17] Mohamed El Banani and Justin Johnson. Bootstrap your own correspondences. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6433–6442, October 2021.

[18] Yifan Feng, Zizhao Zhang, Xibin Zhao, Rongrong Ji, and Yue Gao. Gvcnn: Group-view convolutional neural networks for 3d shape recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 264–272, 2018.

[19] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1126–1135, 2017.

[20] Chelsea Finn, Kelvin Xu, and Sergey Levine. Probabilistic model-agnostic meta-learning. In *Advances in Neural Information Processing Systems*, pages 9516–9527, 2018.

[21] Peter R Florence, Lucas Manuelli, and Russ Tedrake. Dense object nets: Learning dense visual object descriptors by and for robotic manipulation. In *Conference on Robot Learning*, pages 373–385. PMLR, 2018.

[22] Spyros Gidaris and Nikos Komodakis. Dynamic few-shot visual learning without forgetting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4367–4375, 2018.

[23] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*, 2018.

[24] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. *arXiv preprint arXiv:2006.07733*, 2020.

[25] Denis Hadjivelichkov and Dimitrios Kanoulas. Fully self-supervised class awareness in dense object descriptors. In *Conference on Robot Learning*, pages 1522–1531. PMLR, 2022.

[26] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16000–16009, 2022.

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

[28] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[29] Chih-Hui Ho, Bo Liu, Tz-Ying Wu, and Nuno Vasconcelos. Exploit clues from views: Self-supervised and regularized learning for multiview object recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9090–9100, 2020.

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

[31] Ji Hou, Saining Xie, Benjamin Graham, Angela Dai, and Matthias Nießner. Pri3d: Can 3d priors help 2d representation learning? In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5693–5702, 2021.

[32] Han Hu, Jiayuan Gu, Zheng Zhang, Jifeng Dai, and Yichen Wei. Relation networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3588–3597, 2018.

[33] Wei-Chih Hung, Varun Jampani, Sifei Liu, Pavlo Molchanov, Ming-Hsuan Yang, and Jan Kautz. Scops: Self-supervised co-part segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 869–878, 2019.

[34] Dinesh Jayaraman, Ruohan Gao, and Kristen Grauman. Shapecodes: self-supervised feature learning by lifting views to viewgrids. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 120–136, 2018.

[35] Alexander Kirillov, Ross Girshick, Kaiming He, and Piotr Dollár. Panoptic feature pyramid networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6399–6408, 2019.

[36] Alexander Kirillov, Yuxin Wu, Kaiming He, and Ross Girshick. Pointrend: Image segmentation as rendering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9799–9808, 2020.

[37] Sebastian Koch, Albert Matveev, Zhongshi Jiang, Francis Williams, Alexey Artemov, Evgeny Burnaev, Marc Alexa, Denis Zorin, and Daniele Panozzo. Abc: A big cad model dataset for geometric deep learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9601–9611, 2019.

[38] Shamit Lal, Mihir Prabhudesai, Ishita Mediratta, Adam W Harley, and Katerina Fragkiadaki. Coconets: Continuous contrastive 3d scene representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12487–12496, 2021.

[39] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[40] Yunze Liu, Li Yi, Shanghang Zhang, Qingnan Fan, Thomas Funkhouser, and Hao Dong. P4contrast: Contrastive learning with pairs of point-pixel pairs for rgb-d scene understanding. *arXiv preprint arXiv:2012.13089*, 2020.

[41] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2018.

[42] Xu Luo, Longhui Wei, Liangjian Wen, Jinrong Yang, Lingxi Xie, Zenglin Xu, and Qi Tian. Rectifying the shortcut learning of background for few-shot learning. *Advances in Neural Information Processing Systems*, 34, 2021.

[43] Orchid Majumder, Avinash Ravichandran, Subhransu Maji, Alessandro Achille, Marzia Polito, and Stefano Soatto. Supervised momentum contrastive learning for few-shot classification. *arXiv preprint arXiv:2101.11058*, 2021.

[44] Federico Monti, Davide Boscaini, Jonathan Masci, Emanuele Rodola, Jan Svoboda, and Michael M Bronstein. Geometric deep learning on graphs and manifolds using mixture model cnns. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5115–5124, 2017.

[45] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European conference on computer vision*, pages 69–84. Springer, 2016.

[46] David Novotny, Samuel Albanie, Diane Larlus, and Andrea Vedaldi. Self-supervised learning of geometrically stable features through probabilistic introspection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3637–3645, 2018.

[47] Pedro O O Pinheiro, Amjad Almahairi, Ryan Benmalek, Florian Golemo, and Aaron C Courville. Unsupervised learning of dense visual representations. *Advances in Neural Information Processing Systems*, 33:4489–4500, 2020.

[48] Pedro O O Pinheiro, Amjad Almahairi, Ryan Benmalek, Florian Golemo, and Aaron C Courville. Unsupervised learning of dense visual representations. *Advances in Neural Information Processing Systems*, 33:4489–4500, 2020.

[49] Mihir Prabhudesai, Shamit Lal, Darshan Patil, Hsiao-Yu Tung, Adam W Harley, and Katerina Fragkiadaki. Disentangling 3d prototypical networks for few-shot concept learning. *arXiv preprint arXiv:2011.03367*, 2020.

[50] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017.

[51] Charles R Qi, Hao Su, Matthias Nießner, Angela Dai, Mengyuan Yan, and Leonidas J Guibas. Volumetric and multi-view cnns for object classification on 3d data. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5648–5656, 2016.

[52] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in neural information processing systems*, pages 5099–5108, 2017.

[53] Jeremy Reizenstein, Roman Shapovalov, Philipp Henzler, Luca Sbordone, Patrick Labatut, and David Novotny. Common objects in 3d: Large-scale learning and evaluation of real-life 3d category reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10901–10911, 2021.

[54] Andrei A Rusu, Dushyant Rao, Jakub Sygnowski, Oriol Vinyals, Razvan Pascanu, Simon Osindero, and Raia Hadsell. Meta-learning with latent embedding optimization. *arXiv preprint arXiv:1807.05960*, 2018.

[55] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[56] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision (ECCV)*, 2016.

[57] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in neural information processing systems*, pages 4077–4087, 2017.

[58] Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. *Advances in neural information processing systems*, 29, 2016.

[59] Stefan Stojanov, Anh Thai, and James M. Rehg. Using shape to categorize: Low-shot learning with an explicit shape bias. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1798–1808, June 2021.

[60] Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik Learned-Miller. Multi-view convolutional neural networks for 3d shape recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 945–953, 2015.

[61] Yonglong Tian, Olivier J Henaff, and Aäron van den Oord. Divide and contrast: Self-supervised learning from uncurated data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10063–10074, 2021.

[62] Yonglong Tian, Yue Wang, Dilip Krishnan, Joshua B Tenenbaum, and Phillip Isola. Rethinking few-shot image classification: a good embedding is all you need? In *European Conference on Computer Vision (ECCV) 2020*, August 2020.

[63] Eleni Triantafillou, Tyler Zhu, Vincent Dumoulin, Pascal Lamblin, Utku Evci, Kelvin Xu, Ross Goroshin, Carles Gelada, Kevin Swersky, Pierre-Antoine Manzagol, et al. Meta-dataset: A dataset of datasets for learning to learn from few examples. In *International Conference on Learning Representations*, 2019.

[64] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[65] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *Advances in neural information processing systems*, pages 3630–3638, 2016.

[66] Xinlong Wang, Zhiding Yu, Shalini De Mello, Jan Kautz, Anima Anandkumar, Chunhua Shen, and Jose M Alvarez. FreeSOLO: Learning to segment objects without annotations. *arXiv preprint arXiv:2202.12181*, 2022.

[67] Xinlong Wang, Rufeng Zhang, Chunhua Shen, Tao Kong, and Lei Li. Dense contrastive learning for self-supervised visual pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3024–3033, 2021.

[68] Yan Wang, Wei-Lun Chao, Kilian Q Weinberger, and Laurens van der Maaten. Simpleshot: Revisiting nearest-neighbor classification for few-shot learning. *arXiv preprint arXiv:1911.04623*, 2019.

[69] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *Acm Transactions On Graphics (tog)*, 38(5):1–12, 2019.

[70] Xin Wei, Yifei Gong, Fudong Wang, Xing Sun, and Jian Sun. Learning canonical view representation for 3d shape recognition with arbitrary views. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 407–416, 2021.

[71] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1912–1920, 2015.

[72] Yongqin Xian, Tobias Lorenz, Bernt Schiele, and Zeynep Akata. Feature generating networks for zero-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5542–5551, 2018.

[73] Tete Xiao, Colorado J Reed, Xiaolong Wang, Kurt Keutzer, and Trevor Darrell. Region similarity representation learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10539–10548, 2021.

[74] Saining Xie, Jiatao Gu, Demi Guo, Charles R Qi, Leonidas Guibas, and Or Litany. Pointcontrast: Unsupervised pre-training for 3d point cloud understanding. In *European conference on computer vision*, pages 574–591. Springer, 2020.

[75] Zhenda Xie, Yutong Lin, Zheng Zhang, Yue Cao, Stephen Lin, and Han Hu. Propagate yourself: Exploring pixel-level consistency for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16684–16693, 2021.

[76] Han-Jia Ye, Hexiang Hu, De-Chuan Zhan, and Fei Sha. Few-shot learning via embedding adaptation with set-to-set functions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[77] Lin Yen-Chen, Pete Florence, Jonathan T Barron, Tsung-Yi Lin, Alberto Rodriguez, and Phillip Isola. Nerf-supervision: Learning dense object descriptors from neural radiance fields. *arXiv preprint arXiv:2203.01913*, 2022.

[78] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *European conference on computer vision*, pages 649–666. Springer, 2016.

[79] Adrian Ziegler and Yuki M Asano. Self-supervised learning of object parts for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14502–14511, 2022.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] We provide qualitative and quantitative evidence for our claims in Section 5.
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] In Section 6
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes] In Section 6
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] We report this in the Appendix
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] We report 95% confidence intervals for each of our experiments over many low-shot evaluation episodes. As our experiments are computationally expensive, we do not include results over multiple runs at this stage but would include them for the final version.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] We report this in the Appendix
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [Yes] We provide details in the appendix.
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes] We discuss this in the Appendix.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A Appendix Overview

This appendix is structured as follows: We first present an ablation study for our model in Section B; In Section C we provide additional qualitative results on the CO3D [53] dataset; In Section D we provide additional details about the datasets used in our experiments and their licenses; In Section E we provide details on the baselines we use, their implementations and hyperparameters; In Section F we describe the compute resources used in our research.

B Ablation Study

We present empirical results for ablating different elements of our model. Using the ModelNet [71] dataset, we train models without randomly removing the background of the input as a data augmentation step during training (denoted as DOPE w/o random background remove) and without predicting the object mask during training and multiplying it with the local feature encoding (denoted as DOPE w/o mask prediction). The results on the first ModelNet validation set are presented in Table 6. We observe that removing either of these two elements significantly reduces the performance of our model. The reduction in performance because of not randomly removing the backgrounds potentially indicates that without this augmentation, our model uses background texture/geometry information to learn features which do not generalize across instances.

Table 6: Ablation study over two elements of our proposed approach: without randomly masking the foreground objects in input images during training, and without predicting the object mask and multiplying it with the local feature encoding. We observe significant reductions in performance in both cases. Evaluation is done on the ModelNet [71] validation classes.

	1-shot 5-way
DOPE	61.78
DOPE w/o random background remove	54.24
DOPE w/o mask prediction	50.06

C Additional Qualitative Results on CO3D

In Figure 7 we present additional qualitative results on the CO3D [53] dataset. We observe that our proposed approach can find correspondences between similar object parts across different instances of the same category.

D Dataset Details

For all our synthetic datasets we render 20 views of each object randomly positioned on a plane with a physically based rendering (PBR) surface material that is randomly chosen. Lighting comes from a set of high dynamic range imaging (HDRI) lighting environments that are also randomly chosen. Rendering is done using the ray-tracing renderer Cycles in Blender [1]. We use 25 PBR materials and 46 HDRI maps with CC0 licenses sourced from PolyHaven [2]. We provide sample images from all synthetic datasets in Figure 8.

Deriving Pixel-Level Correspondences For the synthetic datasets ModelNet [71], ShapeNet [5], and ABC [37], we have ground truth camera instrinstics, extrinsics, depth maps and segmentation masks, which allows us to extract pixel-level correspondences between two views of an object. For the real CO3D dataset [53], each object video has camera instrinsics and extrinsics estimated using COLMAP [56, 55] and masks estimated with PointRend [36], which allows us to extract estimated pixel-level correspondences between two views of an object.

Data Augmentation For all self-supervised and low-shot learning algorithms we perform color jittering, gamma and contrast augmentations. In addition, we also randomly remove the background using the provided foreground mask. When the background is masked, we also randomly translate and rotate the foreground in the image, and randomize the background as in [21] (for examples please see Figure 9).

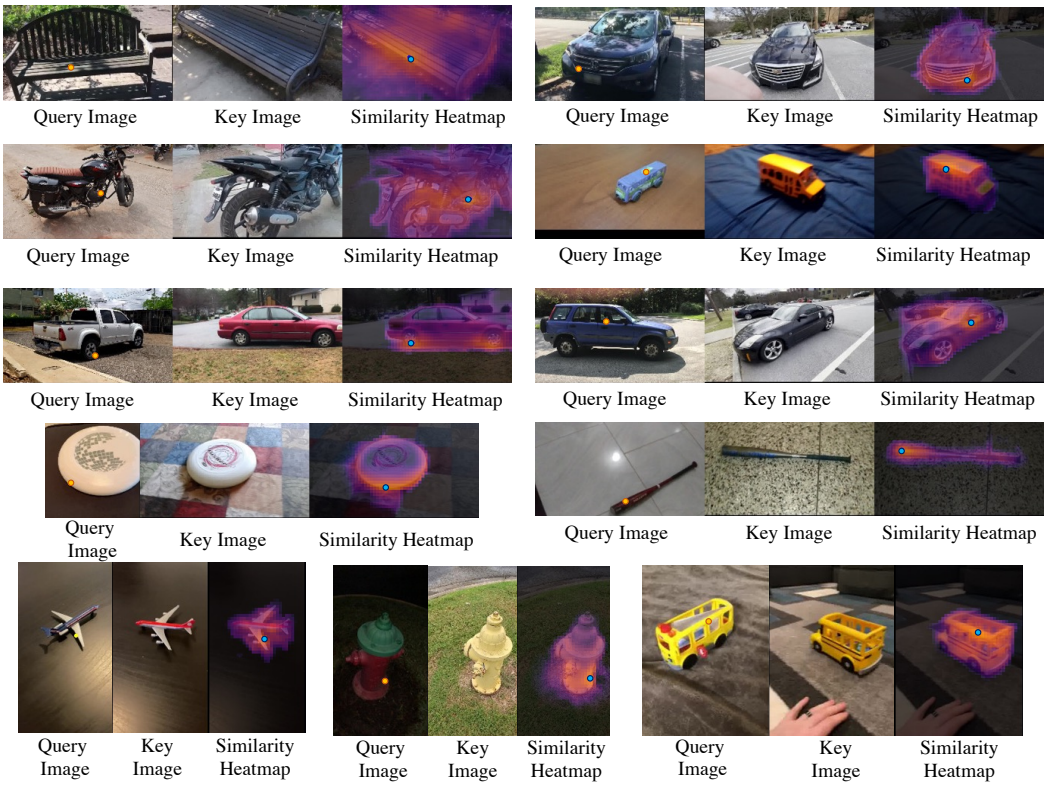

Figure 7: Additonal qualitative results on the CO3D [53] dataset. We present a similarity heatmap for the query point (yellow) in the query image over the entire query image. The blue point on the heatmap overlaid image indicates the predicted correspondence of the query point. Note that these objects are from validation or test categories, which are different from the training categories.

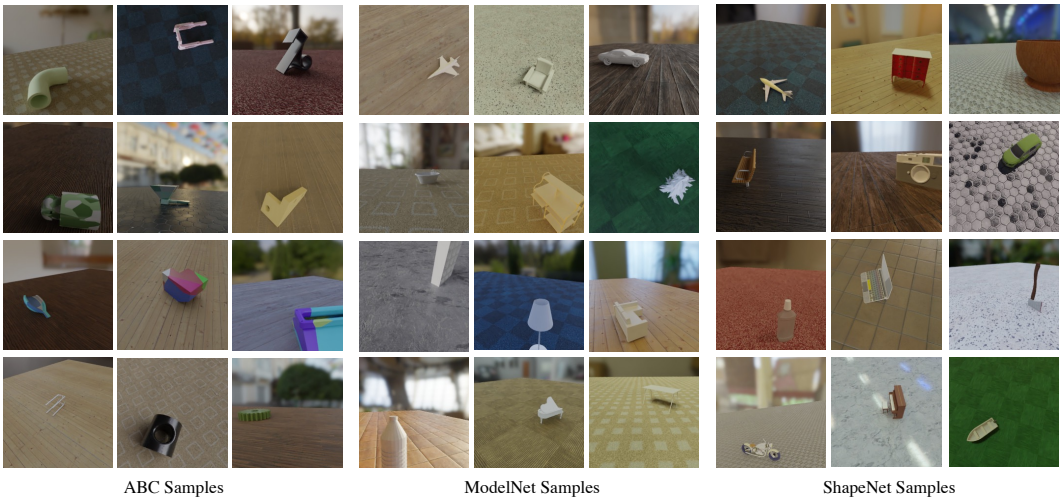

Figure 8: Visualization of our synthetic data rendered from ABC [37], ModelNet [71] and ShapeNet [5]. Note the high environment and viewpoint variability across the images.

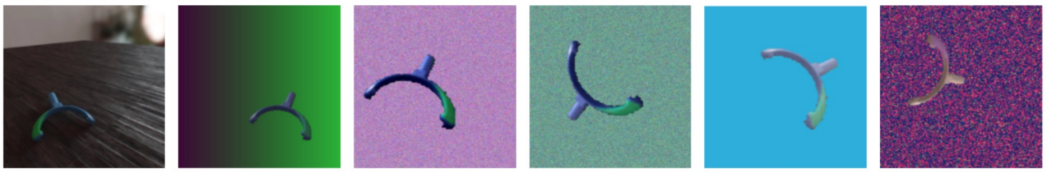

Figure 9: Illustration of our augmentation strategy by removing the background and applying random geometric transformations to the foreground object.

D.1 ModelNet40-LS [71]

The training and validation splits for ModelNet40-LS are shown in Table 7, the first of which which we adopt from [59]. We use 15 queries for low-shot validation and testing. The dataset copyright information is available at `https://modelnet.cs.princeton.edu/`.

D.2 ABC [37]

We randomly sample 115K objects for training and 10K for validation from the total set of downloadable objects. Originally the ABC objects do not come with any surface materials. We generate materials with random colors and random voronoi patterns when rendering the objects. The licensing information for this dataset is available at `https://deep-geometry.github.io/abc-dataset/#license`.

D.3 ShapeNet [5]

The training and validation splits for ShapeNet55-LS are shown in Table 8, the first of which we adopt from [59]. We use 15 queries for low-shot validation and testing. The dataset terms of use are available at `https://shapenet.org/terms`.

D.4 CO3D-LS [53]

The training and validation splits for ShapeNet55-LS are shown CO3D is shown in Table 8. We select the training set by taking the 31 categories with the most data, and randomly sample two sets of 10 without replacement or validation and testing from the remaining 20 classes. For each object video clip in the dataset, we subsample every 3rd frame. We use 15 queries for low-shot validation and testing. The terms and conditions of the CO3D dataset are available at `https://ai.facebook.com/datasets/co3d-downloads/`.

E Baseline Algorithm Implementation

All algorithms are implemented in PyTorch [17] 1.8.2 LTS where possible following released codebases from the original papers.

E.1 SimpleShot [68]

We follow the original implementation of SimpleShot [1]. We train SimpleShot with the AdamW [41] optimizer, with a batch size of 256, a learning rate of 0.001, weight decay of 0.0001 as we found it gives improved results over using SGD with momentum. We train SimpleShot for 500 epochs on ShapeNet and 1000 epochs on CO3D and ModelNet with a 0.1 learning rate decay at 0.7 and 0.9 of the total number of epochs.

E.2 RFS [62]

We follow the original implementation of RFS from [2]. RFS requires first training a backbone with cross-entropy on the training classes. To do this we follow the same training procedure as SimpleShot,

as it also consists of training with cross-entropy on the base classes. Like in the original codebase, we use Scikit-Learn [18] to train a logistic classifier for each low-shot episode.

E.3 FEAT [76]

We follow the original implementation of FEAT [3] in our implementation. We train a separate model for each n-way m-shot episode configuration as in the original paper. We use the original optimizer and hyperparameters, but halve the softmax temperature values for ShapeNet and ModelNet, and quarter them for CO3D as we find that it results in improved low-shot generalization.

E.4 SupMoCo [43]

We follow the pseudocode in the Appendix of the original paper to implement SupMoCo. We use a queue of size $K = 4096$ because of our smaller datasets and SGD with cosine learning rate decay for 2000 epochs, with a batch size of 256, an initial learning rate of 0.05 and weight decay of 0.0001.

E.5 VISPE [29]

We use the original VISPE implementation [4] in our experiments. We use AdamW [41] with a batch size of 32, learning rate of 0.0001, weight decay of 0.01 for 1000 epochs as we found it improves the low-shot generalization performance over the original hyperparameters.

E.6 VISPE++ [27]

For our MoCo-based VISPE++ baseline, we follow the original MoCo codebase [5]. Rather than the standard application of augmentations to a single view of an object to obtain two positive images, we give two views of the same object as two positive images, and two views of different objects as negatives. We use a two layer 1024-dim MLP as the projection head. When training on ABC we use a queue of size $K = 16348$ and when training on other datasets we use a smaller queue of size $K = 4096$. We train this model using the AdamW [41] optimizer for for 3500 epochs with a learning rate of 0.0001 and weight decay of 0.01.

For our SimSiam-based VISPE++ baseline, we use the implementation from [9]. We use the same random masking augmentations as DOPE, but the original color jittering parameters of SimSiam because we found that results in improved low-shot generalization.

F Compute Details

To train our models we use an 8 GPU server with Titan Xp GPUs. Training our proposed approach requires 4 GPUs using Distributed Data Parallel in PyTorch [17].

Training	# samples	Validation + Test	# samples	Split assignment
bookshelf	672	door	129	v_0, t_1, t_2, t_3, t_4
chair	989	keyboard	165	v_0, t_1, t_2, t_3, t_4
plant	340	flower_pot	169	v_0, t_1, v_2, t_3, t_3
bed	615	curtain	158	v_0, t_1, v_2, t_3, t_4
monitor	565	person	108	v_0, v_1, t_2, v_3, t_4
piano	331	cone	187	v_0, v_1, t_2, t_3, v_4
mantel	384	xbox	123	v_0, v_1, v_2, t_3, t_4
car	297	cup	99	v_0, t_1, t_2, v_3, v_4
table	492	bathtub	156	v_0, v_1, v_2, v_3, t_4
bottle	435	wardrobe	107	v_0, t_1, t_2, t_3, v_4
airplane	726	lamp	144	t_0, v_1, t_2, v_3, v_4
sofa	780	stairs	144	t_1, v_1, v_2, t_3, v_4
toilet	444	laptop	169	t_0, t_1, t_2, v_3, v_4
vase	575	tent	183	t_0, v_1, t_2, t_3, t_4
dresser	286	bench	193	t_0, t_1, v_2, t_3, v_4
desk	286	range_hood	215	t_0, t_1, t_2, v_3, v_4
night_stand	286	stool	110	t_0, t_1, t_2, v_3, v_4
guitar	255	sink	148	t_0, v_1, t_2, v_3, t_4
glass_box	271	radio	124	t_0, v_1, v_2, v_3, t_4
tv_stand	367	bowl	84	t_0, v_1, v_2, t_3, t_4
Total				
20 classes	9396	20 classes	2915	

Table 7: Split composition of ModelNet40-LS. Rightmost column indicates the assignment of the class to each of the 5 validation/testing splits.

Training	# samples	Validation + Test	# samples	Split assignment
chair	500	stove	218	v_0, v_1, t_2, t_3, v_4
table	495	microwave	152	v_0, t_1, t_2, t_3, v_4
bathtub	499	microphone	67	v_0, t_1, v_2, t_3, t_4
cabinet	499	cap	56	v_0, v_1, t_2, v_3, v_4
lamp	500	dishwasher	93	v_0, t_1, t_2, t_3, v_4
car	525	keyboard	65	v_0, t_1, t_2, t_3, t_4
bus	500	tower	133	v_0, v_1, t_2, t_3, t_4
cellular	500	helmet	162	v_0, t_1, t_2, v_3, t_4
guitar	500	birdhouse	73	v_0, t_1, v_2, t_3, v_4
bench	499	can	108	v_0, t_1, t_2, t_3, t_4
bottle	498	piano	239	t_0, v_1, t_2, v_3, t_4
laptop	460	train	389	t_0, t_1, v_2, v_3, t_4
jar	499	file	298	t_0, t_1, t_2, v_3, t_4
loudspeaker	496	pistol	307	t_0, t_1, t_2, v_3, t_4
bookshelf	452	motorcycle	337	t_0, t_1, v_2, v_3, t_4
faucet	500	printer	166	t_0, t_1, v_2, t_3, v_4
vessel	864	mug	214	t_0, v_1, t_2, t_3, t_4
clock	496	rocket	85	t_0, v_1, v_2, t_3, t_4
airplane	500	skateboard	152	t_0, v_1, v_2, v_3, v_4
pot	500	bed	233	t_0, t_1, t_2, t_3, v_4
rifle	498	ashcan	343	t_0, t_1, t_2, t_3, v_4
display	498	washer	169	t_0, t_1, t_2, t_3, t_4
knife	423	bowl	186	t_0, t_1, v_2, t_3, t_4
telephone	498	bag	83	t_0, v_1, v_2, v_3, t_4
sofa	499	mailbox	94	t_0, v_1, t_2, t_3, t_4
		pillow	96	t_0, t_1, t_2, t_3, t_4
		earphone	73	t_0, t_1, v_2, t_3, t_4
		camera	113	t_0, t_1, t_2, t_3, t_4
		basket	113	t_0, v_1, t_2, v_3, v_4
		remote	66	t_0, t_1, t_2, t_3, t_4
Total				
25 classes	12698	30 classes	4883	

Table 8: Split composition of ShapeNet55-LS. Rightmost column indicates the assignment of the class to each of the 5 validation/testing splits.

Training	# samples		Validation + Test	# samples	Split assignment
wineglass	453		car	210	v_0, t_1, t_2, t_3, v_4
keyboard	638		bottle	277	v_0, v_1, v_2, t_3, t_4
mouse	431		baseballglove	84	v_0, v_1, t_2, t_3, v_4
bowl	660		frisbee	121	v_0, t_1, t_2, t_3, t_4
broccoli	379		tv	29	v_0, v_1, v_2, v_3, v_4
chair	675		toyplane	225	v_0, v_1, v_2, t_3, t_4
handbag	749		baseballbat	71	v_0, t_1, t_2, v_3, t_4
toytrain	272		pizza	134	v_0, v_1, t_2, v_3, v_4
carrot	740		hydrant	307	v_0, t_1, t_2, v_3, v_4
bicycle	340		hotdog	69	v_0, v_1, v_2, v_3, t_4
cellphone	416		parkingmeter	21	t_0, t_1, v_2, t_3, t_4
ball	542		banana	198	t_0, v_1, v_2, t_3, v_4
teddybear	734		motorcycle	267	t_0, t_1, t_2, t_3, v_4
cake	348		bench	250	t_0, v_1, t_2, t_3, v_4
backpack	832		donut	193	t_0, t_1, v_2, v_3, t_4
hairdryer	503		microwave	50	t_0, v_1, t_2, v_3, t_4
couch	223		stopsign	193	t_0, t_1, v_2, t_3, v_4
toilet	355		skateboard	82	t_0, t_1, v_2, v_3, t_4
remote	392		toybus	141	t_0, v_1, t_2, v_3, t_4
toaster	299		kite	150	t_0, t_1, v_2, v_3, v_4
vase	647				
laptop	501				
toytruck	466				
umbrella	498				
suitcase	482				
plant	563				
apple	391				
cup	169				
book	658				
sandwich	244				
orange	479				
Total					
21 classes	15079		10 classes	3072	

Table 9: Split composition of CO3D. Rightmost column indicates the assignment of the class to each of the 5 validation/testing splits.

Appendix References

[1] https://github.com/mileyan/simple_shot.

[2] https://github.com/WangYueFt/rfs/.

[3] https://github.com/Sha-Lab/FEAT.

[4] https://github.com/chihhuiho/VISPE.

[5] https://github.com/facebookresearch/moco.

[6] Blender, https://blender.org/.

[7] Poly haven, https://polyhaven.com/.

[8] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015.

[9] MMSelfSup Contributors. MMSelfSup: Openmmlab self-supervised learning toolbox and benchmark. https://github.com/open-mmlab/mmselfsup, 2021.

[10] Peter R Florence, Lucas Manuelli, and Russ Tedrake. Dense object nets: Learning dense visual object descriptors by and for robotic manipulation. In *Conference on Robot Learning*, pages 373–385. PMLR, 2018.

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

[12] Chih-Hui Ho, Bo Liu, Tz-Ying Wu, and Nuno Vasconcelos. Exploit clues from views: Self-supervised and regularized learning for multiview object recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9090–9100, 2020.

[13] Alexander Kirillov, Yuxin Wu, Kaiming He, and Ross Girshick. Pointrend: Image segmentation as rendering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9799–9808, 2020.

[14] Sebastian Koch, Albert Matveev, Zhongshi Jiang, Francis Williams, Alexey Artemov, Evgeny Burnaev, Marc Alexa, Denis Zorin, and Daniele Panozzo. Abc: A big cad model dataset for geometric deep learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9601–9611, 2019.

[15] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2018.

[16] Orchid Majumder, Avinash Ravichandran, Subhransu Maji, Alessandro Achille, Marzia Polito, and Stefano Soatto. Supervised momentum contrastive learning for few-shot classification. *arXiv preprint arXiv:2101.11058*, 2021.

[17] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

[18] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

[19] Jeremy Reizenstein, Roman Shapovalov, Philipp Henzler, Luca Sbordone, Patrick Labatut, and David Novotny. Common objects in 3d: Large-scale learning and evaluation of real-life 3d category reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10901–10911, 2021.

[20] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[21] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision (ECCV)*, 2016.

[22] Stefan Stojanov, Anh Thai, and James M. Rehg. Using shape to categorize: Low-shot learning with an explicit shape bias. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1798–1808, June 2021.

[23] Yonglong Tian, Yue Wang, Dilip Krishnan, Joshua B Tenenbaum, and Phillip Isola. Rethinking few-shot image classification: a good embedding is all you need? In *European Conference on Computer Vision (ECCV) 2020*, August 2020.

[24] Yan Wang, Wei-Lun Chao, Kilian Q Weinberger, and Laurens van der Maaten. Simpleshot: Revisiting nearest-neighbor classification for few-shot learning. *arXiv preprint arXiv:1911.04623*, 2019.

[25] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1912–1920, 2015.

[26] Han-Jia Ye, Hexiang Hu, De-Chuan Zhan, and Fei Sha. Few-shot learning via embedding adaptation with set-to-set functions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.