Learning Object Parts from Multiple Views for Low-shot Category Generalization

Anonymous Author(s)
Affiliation
Address
email

Abstract

A hallmark of the deep learning era for computer vision is using large and annotated datasets training to feature representations for tasks ranging from object recognition and semantic segmentation to optical flow estimation and novel view synthesis of scenes. In this work, we aim to learn discriminative object part representations for low-shot category recognition without requiring any category labels. To this end, we propose Deep Object Part 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 object parts. 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.

1 Introduction

Achieving high accuracy at object category recognition generally requires deep models with millions of parameters [27, 15] and million-scale datasets [13, 37]. Alleviating this data requirement is a fundamental challenge for computer vision and has driven the development of low-shot [54, 60, 19, 70, 7] and self-supervised [43, 26, 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-views of object instances without category labels are available for self-supervised learning of discriminative object representations [28, 32]. In addition, their primary focus is not on low-shot generalization. Learning to exploit such multi-view data for self-supervised training of representations capable of low-shot category generalization 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 [71, 50].

The goal of this paper is to use multiple-views of many object instances, combined with some additional geometric information about the views, to learn representations that can be directly used for recognizing categories of objects from a few labeled examples. We assume access to sparse depth, known camera pose/calibration, and foreground segmentation during training, and use this information to formulate a self-supervised object learning task. In practice, this 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 sensors data and using forward kinematics to obtain the camera pose [21, 25]. Otherwise, in less constrained settings where the multi-view data consists of object-centric "turntable-style" videos of objects taken by humans using mobile devices, the self-supervision data can be obtained by postprocessing the videos using a Structure-from-Motion (SfM) and Multi-View Stereo (MVS) pipeline such as COLMAP [52, 53]. Note that we do not assume

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

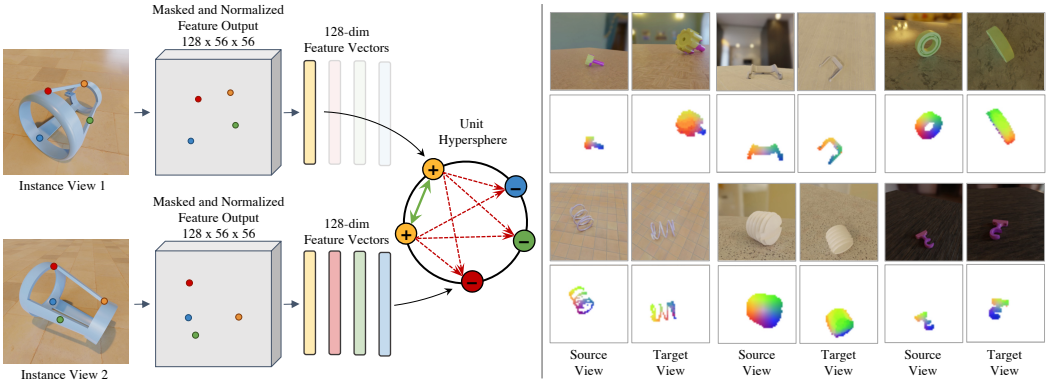

Figure 1: **Left:** Our contrastive learning formulation: We form positives between features at pixel locations of the same 3D point on the object surface, and negatives pixels from different points on the object surface. **Right:** For each feature corresponding to a point on the predicted segmentation 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.

access to any sort of category-level part annotation or object attribute information for representation learning. At test time, our setting only allows for access to RGB images.

To solve this self-supervised learning task, we propose Deep Object Part Encodings (DOPE). DOPE uses pairs of object views to learn a dense, CNN-based local feature encoding that will map an object part that is visible in both views to the same feature vector, and different parts to different feature vectors (for an illustration please see Figure 1). We find that such a representation, trained only on pairs of instance views, *generalizes across instances of novel categories* (illustrated in Figures 6 and 5), as 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 use an approach based on finding local-matches between objects to directly use this representation for low-shot object recognition.

DOPE is trained using with a local contrastive learning loss, where positive feature pairs correspond to feature encodings of the same object part in a pair of views, and negatives to different parts. To find such positive and negative correspondence 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 and did not pertain to low-shot recognition. In addition, VISPE [28] proposes using multiple views of instances to learn global, view invariant feature vectors of objects. We hypotheize that such global encodings are not necessarily suitable for category-level generalization, as they only need to encode the minimum amount of information to tell object instances apart. In contrast, learning part similarity at the local level forces the model to learn how to encode all visible object parts between two views, which we hypothesize is useful for low-shot object categorization.

We find that DOPE is competitive and in some cases outperforms both self-supervised methods [28] and fully supervised low-shot approaches [62, 57, 70, 41] on the synthetic ModelNet [65] and the real CO3D [50] datasets. We further find that DOPE can learn a representation using objects from the large, uncurated 3D object dataset ABC [35], that directly transfers to low-shot category recognition. In summary, we make the following contributions:

- Deep Object Part Encodings (DOPE), a novel self-supervised learning approach based on learning object shape correspondences that can directly generalize to low-shot object category learning on both synthetic datasets.
- The first study of how a large, unstructured object instance dataset like ABC [35] can be used effectively to learn representations for low-shot category recognition from images.
- Multiple datasets with high photorealism generated from existing 3D object datasets that will enable future research in learning representations using multi-view information for low-shot category recognition. We plan a full release of code and data upon acceptance

2 Related Work

2.1 Low-shot Learning for Object Recognition

The goal of low-shot learning is to develop 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 how the base class data is used, the *meta learning*-based category of techniques such as [54, 31, 66, 19] samples low-shot episodes from the large base class dataset to effectively train under the same setting as during evaluation. In contrast, the *whole-classification*-based techniques [62, 57] train the backbone feature representation via standard cross-entropy classification on the whole base class dataset, whereas [66, 10, 22, 41] further fine-tune the representation with a meta-learning procedure. Based on whether the model parameters are updated, *optimization*-based techniques [51, 19, 20] aim to learn a representation that can be effectively fine-tuned using a few samples while *metric learning*-based techniques [62, 66, 10, 57] aim to learn a discriminative metric space that will directly transfer to low-shot episodes of 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 [72], determining the relative position of image patches [14, 43] or predicting rotations [23]. More recently, methods based on instance-level comparisons have led to significant performance improvements on downstream tasks. Such methods are trained to map multiple views (usually multiple augmented versions of an image) as the same feature vector under some similarity measure, usually cosine distance of vectors normalized to a hypersphere, and different images as different vectors. SimCLR [6] and MoCLR [56] consider augmentations of the same underlying image as positives and other images in the minibatch as negatives, whereas the MoCo-based [8, 26] approaches use a continuously updating queue of negative features as negatives. The methods presented in [56, 8, 26, 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.

Self-Supervised Learning for Dense Vision Tasks The main ideas from self-supervised contrastive learning for classification tasks have been applied to learn local, dense, pixel-level representations for tasks like semantic segmentation and unsupervised keypoint learning. Such works [69, 61, 67, 44] apply contrastive learning at the pixel level, treating features corresponding to individual pixels or pooled regions of images as positives or negatives, and show improvements over supervised pre-training for segmentation and detection. Cheng et al. [12] on the other hand use such a strategy for unsupervised keypoint learning. Our approach uses a Panoptic FPN-based 2D CNN architecture similar to these works, but focus on self-supervised learning of object representations from multiple views and directly applying them for low-shot categorization.

Self-Supervised Learning using 3D Geometry and Multi-View Images Self-supervised learning has also been studied in settings where multi-view RGB-D information (images and in-camera dense 3D Point clouds) is available [30, 68, 16, 46, 36, 21]. The first body of works [30, 17, 38, 36, 46] use pixel-level correspondences between pairs of views of a scene, where a positive pair of pixels corresponds to the same 3D point in the world projected to each image, to align [1] image-based and depth-based scene representations [30, 17, 38] for point cloud registration [17] and scene segmentation [38, 29], or directly align 3D CNN-based scene representations from multiple views [36, 46] for self supervised object tracking and few-shot style/shape learning. Florence et

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

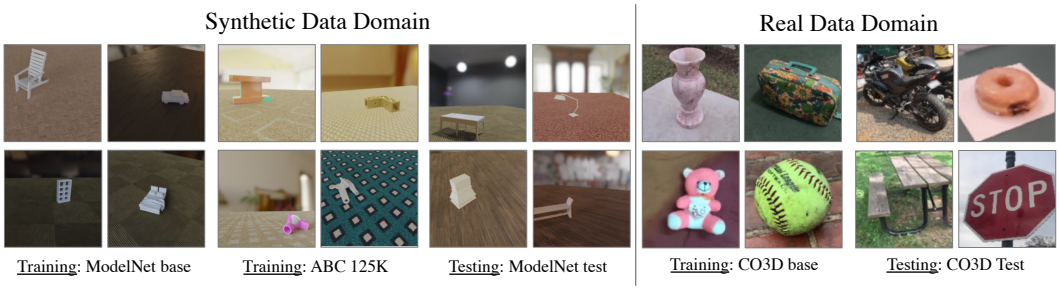

Figure 2: **Left:** Synthetic data rendered from ModelNet and ABC (first and second columns) for self-supervised learning of representations and low-shot testing data from ModelNet (third column). **Right:** Real data from CO3D for self-supervised representation training and low-shot testing.

al. [21] use only image-based representations to learn dense object descriptors for object instances. In comparison to these works, our goal is to use geometry-derived correspondences between multi-view 2D images of objects to learn representations that can directly be applied for low-shot object category classification. Another line of works [68, 29] directly applies contrastive training to 3D point clouds. Last, assuming only access to multiple RGB views without any depth information, [32] uses an encoder/decoder architecture to predict different viewpoints of an object from a single image. Further, VISPE [28] uses contrastive learning to train an instance-level discriminative representation by mapping two views of the same object as similar and two views of different objects as dissimilar. While we share with these works the use of self-supervised representation learning from multiple RGB views to perform recognition, we focus on the low-shot object category classification task. We show that our approach outperforms instance-level representation on multiple datasets.

2.3 Object Classification on Synthetic Datasets

Object recognition has been extensively studied on datasets of 3D meshes such as ModelNet40 [65] rendered into various representations such as point clouds [47, 49, 11, 63], voxels [48, 65] and images [55, 48, 18, 42, 28, 64, 32]. Our work extends these prior works in two ways: First, we use the most photorealistic rendering pipeline to generate synthetic data for studying object recognition using these datasets to date and conduct the first study in how large unstructured datasets like ABC [35] can be used to learn representations suitable for low-shot object recognition.

3 Multi-View Self-Supervised Instance Learning for Low-Shot Generalization

We aim to learn a representation using 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, at test time for low-shot generalization, 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 labelling such as explicit parts or attribute information.

Following [32, 28] and convention in low-shot learning we partition ModelNet [65] and CO3D [50] 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 unstructured 3D object instance dataset ABC [35] 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 lease refer to Figure 2.

4 DOPE: Deep Object Part Encodings

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

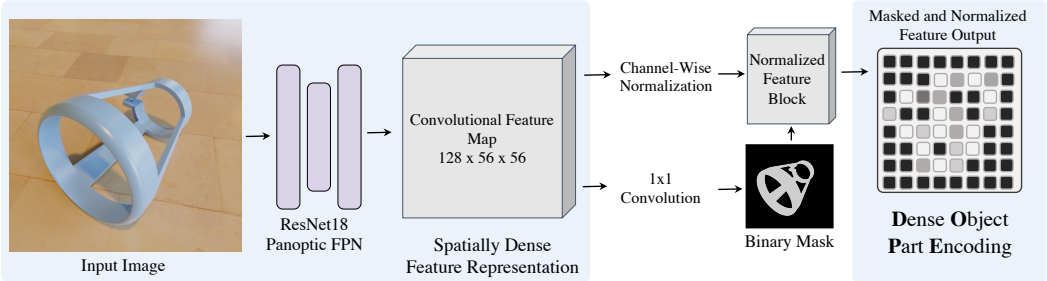

Figure 3: Our architecture consists of a ResNet18 [27] followed by a Panoptic Feature Pyramid network [33] to output a 2D grid of convolutional features. We use an additional branch from these features to predict a binary mask. After performing channel-wise unit vector normalization and masking with the binary mask, we obtain the final feature output we use for contrastive training and at test time

4.1 Local Object Representation Learning

Assume we have a dataset of object instances with N calibrated views per object with masks and known camera intrinsics and extrinsics. Denote the two views of one object as $\mathbf{v}_1$ and $\mathbf{v}_2$, and a feature representation $f : \mathbb{R}^{H \times W \times 3} \rightarrow \mathbb{R}^{h_f \times w_f \times D}$ as 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. 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 $\mathbf{v}_1$ and (u_2, v_2) in $\mathbf{v}_2$. Our goal is to learn f so that the feature vectors for the corresponding pixel locations $\mathbf{z_1} = f(\mathbf{v}_1)[\bar{u}_1, \bar{v}_1, :]$ and $\mathbf{z_2} = f(\mathbf{v}_2)[\bar{u}_2, \bar{v}_2, :]$, where $\bar{u}_i$ is the location in the lower spatial dimensional feature output corresponding to u_i, are so that $\mathbf{sim}(\mathbf{z}_1, \mathbf{z}_2) = 1$, where $\mathbf{sim}$ is some similarity metric, in our case normalized dot product. Conversely, $\mathbf{sim}(\mathbf{z}_1, \mathbf{z}_2) = 0$ for different $\mathbf{z}_1$ and $\mathbf{z}_2$. Note that $\mathbf{z} \in \mathbb{R}^D$. In other words, following the chair leg 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, for each image pair we first sample a set $\mathcal{C}$ of n pixels on the object surface in $\mathbf{v}_1$ using farthest point sampling, and find the corresponding pixels in $\mathbf{v}_2$ [2]. For a point $i \in \mathcal{C}$ we therefore obtain feature vectors $\mathbf{z}_1^i$ and $\mathbf{z}_2^i$ whose similarity $\mathbf{sim}(\mathbf{z}_1^i, \mathbf{z}_2^i)$ we want to maximize, and at the same time for any feature $\mathbf{z}_2^j$ corresponding to some other point on the object we want to minimize $\mathbf{sim}(\mathbf{z}_1^i, \mathbf{z}_2^j)$. To achieve this we minimize a normalized and temperature-scaled cross entropy loss [6] for each pair of positive correspondences

$$\mathcal{L}(i) = -\log \frac{\exp\left(\mathbf{sim}(\mathbf{z}_1^i, \mathbf{z}_2^i)/\tau\right)}{\sum_{j \in \mathcal{C}, j \neq i} \exp\left(\mathbf{sim}(\mathbf{z}_1^i, \mathbf{z}_2^j)/\tau\right)} \tag{1}$$

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 $\mathcal{L}_{corr}$.

To ensure that f does not use any capacity to learn how to encode the background, which is not useful for encoding object representation, 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 $\hat{\mathbf{M}}$ which we train with binary cross entropy $\mathcal{L}_{mask} = BCE(\hat{\mathbf{M}}, \mathbf{M})$. We then apply this mask to the final output of f (refer to Fig. 3) to mask out the background. Fig. 1 contains qualitative examples of how our proposed approach matches object shape across viewpoints. During training we optimize the sum of both losses with equal weight.

4.2 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

[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.

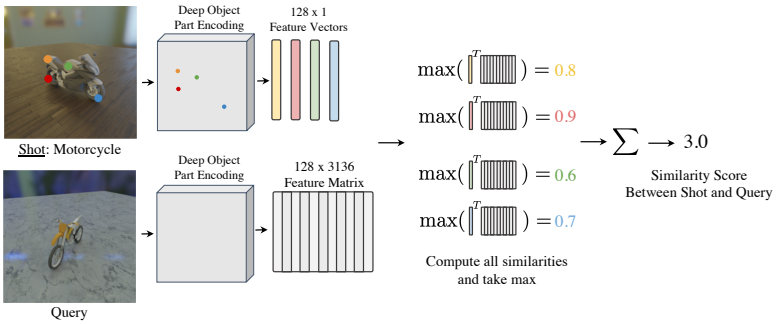

Figure 4: To determine a similarity score between a labelled shot and query image, for each shot image, we first use farthest point sampling to randomly choose k pixels on the object surface, based on a predicted segmentation mask (in this example $k = 4$). We then extract features corresponding to those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

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 and a labelled support set of N images. The following is a procedure to find the most similar image from the support set 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 assign the label to the query from the support image with the highest score. When there are multiple shot images per category available, we perform a 1-nearest neighbor classification.

5 Implementation Details

The backbone of DOPE is based on a ResNet18 [27] with a Panoptic Feature Pyramid network [33], taking as input a $3 \times 224 \times 224$ image and outputting a $128 \times 56 \times 56$ 2D feature block. This backbone formulation follows the approach in VADeR [45]. During training, we use $n = 16$ pixel locations on the object surface for each image pair, and use a batch size of 256 pairs for each parameter update. DOPE is trained using AdamW [39] and a learning rate of 10^{-4} and weight decay of 10^{-2}. For each view pair, one image is sent to a momentum-encoder backbone as in [26]. 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. 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 in the Experiments section 6 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.

6 Experiments

In this section we demonstrate the low-shot generalization ability of DOPE's compared with both supervised algorithms and strong self-supervised baselines on synthetic and real datasets. We include further results and ablation studies in the Appendix.

6.1 Datasets

We briefly describe the 3D object datasets we use for evaluation. We use Blender [1] with the Cycles ray-tracing engine to create synthetic data with high-photo realism, rendering 20 views per object

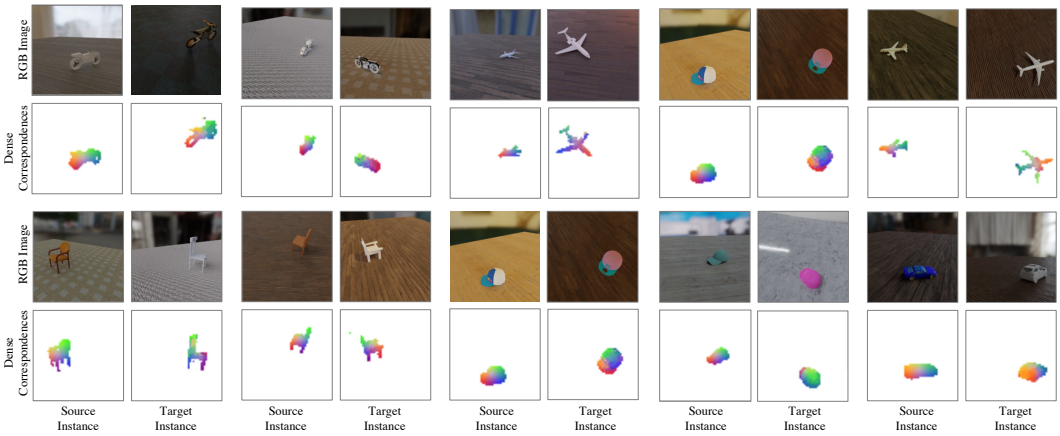

Figure 5: Category-level dense correspondences for ShapeNet [5] emerge from training on the ABC [35] instance dataset. For each feature corresponding to a point on the predicted segmentation mask in the source view we assign a color. For each point in the predicted segmentation for the target views, 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.

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-only approach VISPE++. Dashed line separates supervised and self-supervised methods. Parenthesis show confidence intervals based on 2.5K low shot episodes.

	5-way			10-way		
Episode Setup →	1-shot	5-shot	10-shot	1-shot	5-shot	10-shot
SimpleShot [62]	50.84 ±0.41	63.67 ±0.33	66.94 ±0.31	36.45 ±0.22	49.89 ±0.16	52.94 ±0.15
RFS [57]	50.85 ±0.40	66.48 ±0.35	71.06 ±0.32	37.11 ±0.33	53.37 ±0.16	58.18 ±0.14
FEAT [70]	49.44 ±0.40	64.46 ±0.34	68.01 ±0.31	36.21 ±0.22	51.26 ±0.16	55.38 ±0.16
SupMoCo [41]	46.49 ±0.38	62.14 ±0.33	66.56 ±0.31	32.93 ±0.22	48.03 ±0.17	52.66 ±0.16
VISPE [28]	51.49 ±0.44	62.33 ±0.34	65.23 ±0.32	37.39 ±0.24	48.09 ±0.16	50.87 ±0.15
VISPE++	53.67 ±0.21	67.44 ±0.17	71.54 ±0.16	40.47 ±0.12	54.16 ±0.09	58.66 ±0.08
DOPE (ours)	53.87 ±0.21	66.99 ±0.17	70.94 ±0.14	40.49 ±0.22	53.43 ±0.17	57.92 ±0.16
VISPE++ — ABC	56.50 ±0.43	69.75 ±0.32	74.13 ±0.29	42.97 ±0.24	56.90 ±0.18	61.61 ±0.16
DOPE (ours) — ABC	**57.50 ±0.42**	**71.56 ±0.31**	**76.11 ±0.29**	**44.14 ±0.23**	**59.14 ±0.23**	**64.25 ±0.16**

with varying camera distance, azimuth and elevation. Objects are rendered 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].

- **ModelNet40-LS**[65] 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.
- **ABC**[35] 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 test set.
- **CO3D-LS**[50] (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 the dataset into 31 base, 10 validation and 10 test categories. The dataset is post-processed using COLMAP [53, 52] to estimate the cameras and reconstruct the scene and object geometry. It also includes object segmentation masks estimated using PointRend [34].

6.2 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[27] backbone and are hyperparameter tuned to our dataset setting (details are included in the Supplement).

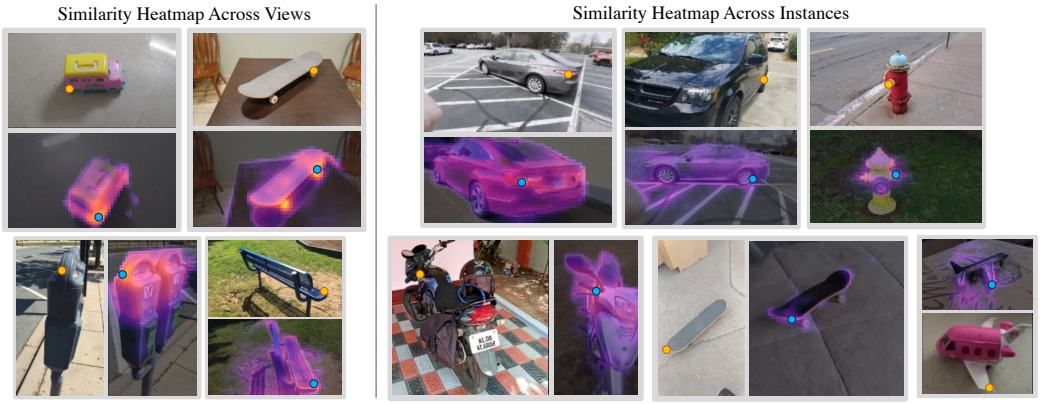

Figure 6: Similarity heatmap for the query point (yellow) **Left:** across different views of the same object and **Right:** across different object instances from CO3D. The blue point on the heatmap-overlaid image indicates the predicted correspondence of the query point. Note that these objects are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

Table 2: Results on image-only low-shot recognition on the **CO3D-LS**. DOPE is competitive with supervised baselines and outperforms the global self supervised learning-only approach VISPE++. Dashed line separates supervised and self-supervised methods. Parenthesis show confidence intervals based on 2.5K low shot episodes.

	5-way		10-way	
Episode Setup →	1-shot	5-shot	1-shot	5-shot
SimpleShot [62]	58.24 ±0.43	**76.15 ±0.29**	44.18 ±0.25	61.07 ±0.16
RFS [57]	58.24 ±0.43	72.48 ±0.33	44.18 ±0.25	61.07 ±0.16
FEAT [70]	**59.88 ±0.43**	75.53 ±0.31	**46.05 ±0.25**	**65.51 ±0.16**
VISPE	55.08 ±0.42	69.59 ±0.33	40.82 ±0.24	56.19 ±0.18
VISPE++	56.82 ±0.40	72.61 ±0.31	44.02 ±0.25	61.22 ±0.17
DOPE (ours)	58.31 ±0.40	73.40 ±0.15	44.91 ±0.15	61.40 ±0.08

- **SimpleShot**[62] is a strong baseline that outperforms many more complex low-shot learning methods. In SimpleShot, a CNN-based backbone is trained with cross-entropy on the base classes, and tested with nearest class mean classification on the test classes.
- **RFS**[57] is a strong but simple baseline, trained with cross-entropy on the base classes, with a logistic regression-based classifier trained on data the support data in each low-shot episode.
- **FEAT**[70] finetunes a transformer-based [59] set-to-set function on top of a cross-entropy-trained backbone, to further refine the backbone embeddings. Methods that outperform FEAT on miniImageNet like COSOC [40] 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.
- **SupMoCo**[41] 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 [58].
- **VISPE**[28] is a self-supervised baseline that uses a temperature-scaled cross-entropy to train using the global backbone features of two views of the same instances as positives and two views of different instances as 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 a more recent contrastive learning technique based on MoCoV2[8], resulting in a stronger baseline. We use a one nearest neighbor classifier in the learned global feature space for low-shot classification as we found that it gives the highest performance.

6.3 DOPE trained on ABC outperforms supervised and self-supervised baselines on ModelNet

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. We observe that despite being trained on ABC without any category labels, DOPE outperforms the second best baseline by up to 2.4%-points. Note that both VISPE++ and our proposed DOPE outperform supervised baselines by a large margin when trained on ABC.

6.4 DOPE trained CO3D outperforms self-supervised baselines and is competitive with supervised baselines

We also evaluate DOPE's ability to directly perform low-shot classification without any category supervision during representation learning on real data, and demonstrate its improved low-shot generalization ability over both supervised and self-supervised baselines. The results are presented in Table 2. 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 by up to 1.5%-points. We also observe that DOPE is close in performance to supervised baselines SimpleShot, RFS and FEAT.

6.5 DOPE Qualitative Results

We present qualitative results for object part learning for ShapeNet [5] in Figure 5 and for CO3D [50] on the validation and test categories 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 are the chairs are correctly matched in Figure 5 and the back light of the cars in Figure 6). Note that local representations were trained on a different set of object categories (for CO3D) and a completely different dataset (for ShapeNet training is done on ABC). 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 object parts and generalize to unseen data.

7 Limitation 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 like DOPE 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 can potentially have a significant negative environmental impact. This may be addressed in the future through improvements in chip design and optimization of deep models.

8 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 labelled on par with category supervised low-shot learning methods. Our findings are validated on both synthetic and real datasets. Qualitatively, we observe that our proposed approach can learn object parts 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.

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, 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.

[27] 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.

[28] 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.

[29] 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.

[30] 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.

[31] 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.

[32] 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.

[33] 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.

[34] 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.

[35] 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.

[36] 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.

[37] 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.

[38] 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.

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

[40] 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.

[41] 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.

[42] 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.

[43] 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.

[44] 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.

[45] 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.

[46] 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.

[47] 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.

[48] 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.

[49] 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.

[50] 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.

[51] 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.

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

[53] 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.

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

[55] 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.

[56] 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.

[57] 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.

[58] 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.

[59] 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.

[60] 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.

[61] 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.

[62] 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.

[63] 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.

[64] 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.

[65] 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.

[66] 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.

[67] 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.

[68] 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.

[69] 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.

[70] 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.

[71] 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.

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

Checklist

The checklist follows the references. Please read the checklist guidelines carefully for information on how to answer these questions. For each question, change the default **[TODO]** to [Yes] , [No] , or [N/A] . You are strongly encouraged to include a **justification to your answer**, either by referencing the appropriate section of your paper or providing a brief inline description. For example:

- Did you include the license to the code and datasets? [Yes] See Section **??**.
- Did you include the license to the code and datasets? [No] The code and the data are proprietary.
- Did you include the license to the code and datasets? [N/A]

Please do not modify the questions and only use the provided macros for your answers. Note that the Checklist section does not count towards the page limit. In your paper, please delete this instructions block and only keep the Checklist section heading above along with the questions/answers below.

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 6.
   (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 7
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes] In Section 7
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)? [No] Code and data will be made available after publication. Implementation details are included in the appendix
   (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]