Few-shot Object Detection and Viewpoint Estimation for Objects in the Wild

Yang Xiao, Vincent Lepetit, Renaud Marlet

Abstract—Detecting objects and estimating their viewpoints in images are key tasks of 3D scene understanding. Recent approaches have achieved excellent results on very large benchmarks for object detection and viewpoint estimation. However, performances are still lagging behind for novel object categories with few samples. In this paper, we tackle the problems of few-shot object detection and few-shot viewpoint estimation. We demonstrate on both tasks the benefits of guiding the network prediction with class-representative features extracted from data in different modalities: image patches for object detection, and aligned 3D models for viewpoint estimation. Despite its simplicity, our method outperforms state-of-the-art methods by a large margin on a range of datasets, including PASCAL and COCO for few-shot object detection, and Pascal3D+ and ObjectNet3D for few-shot viewpoint estimation. Furthermore, when the 3D model is not available, we introduce a simple category-agnostic viewpoint estimation method by exploiting geometrical similarities and consistent pose labeling across different classes. While it moderately reduces performance, this approach still obtains better results than previous methods in this setting. Last, for the first time, we tackle the combination of both few-shot tasks, on three challenging benchmarks for viewpoint estimation in the wild, ObjectNet3D, Pascal3D+ and Pix3D, showing very promising results.

Index Terms—Few-shot learning, Meta learning, Object detection, Viewpoint estimation

1 Introduction

Detecting objects in 2D images and estimating their 3D pose, as shown in Figure 1, is extremely useful for applications such as 3D scene understanding, augmented reality and robot manipulation. With the emergence of large databases annotated with object bounding boxes and viewpoints, deep-learning-based methods have achieved very good results on both tasks. However these methods, because they rely on rich labeled data, usually fail to generalize to *novel* object categories when only a few annotated samples are available. Additionally, creating 3D annotations is tedious and requires a large amount of expert effort, which slows down the applications of these methods to new objects. *Few-shot learning*, *i.e.*, being able to transfer the knowledge learned from large base categories with abundant annotated images to novel categories with scarce annotated samples is therefore highly desirable in this context.

To address the few-shot learning of object detection, some approaches simultaneously tackle few-shot classification and few-shot localization by disentangling the learning of category-agnostic and category-specific network parameters [1]. Others extract a class-informative feature vector for each class and use these vectors to reweight full-image features [2] or region-of-interest (RoI) features [3]. This reweighting module computes a feature similarity between query images and support classes, which has also been demonstrated to be useful in few-shot instance segmentation [4] and few-shot image classification [5]. However, this reweighting can easily be affected by noisy class-informative features, especially in the few-shot setting where only a few labeled samples are provided for novel categories. Instead, we propose to rely on a slightly more complex combination of query-image features and class-informative features. We show that this more general aggregation module can provide better few-shot object detection performances with smaller variations when experimented with different choices of support images. Besides, it can also be used to exploit class-exemplar 3D models for few-shot viewpoint estimation. Furthermore, we explore the usage of a cosine-similarity-based classifier [6], [7] and find that it slightly improves the detection results.

In parallel to the endeavours made in few-shot object detection, recent work proposes to perform category-agnostic viewpoint estimation that can be directly applied to novel object categories without retraining [8], [9]. However, these methods either require the testing categories to be similar to the training ones [8], or assume the exact CAD model to be provided for each object during inference [9]. Differently, the meta-learning-based method MetaView [10] introduces the category-level few-shot viewpoint estimation problem and addresses it by learning to estimate category-specific keypoints, requiring extra annotations.

While MetaView [10] has achieved significantly improved performance on novel categories for few-shot viewpoint estimation, there are two main disadvantages: 1) specific keypoints have to be designed for different object categories, which requires some expertise and can be difficult to annotate and estimate for tiny and occluded objects; 2) the number of class-specific keypoint estimation branches increases linearly with the number of object classes.

Instead, we rely on a category-agnostic viewpoint estimation network, that directly predicts three Euler angles from an image embedding, without explicit class knowledge. To that end, we exploit the fact that similar classes, e.g., *sofa* and *chair*, often have a consistent canonical pose, with aligned similarities. The reason probably is that many objects are

- *Y. Xiao, V. Lepetit and R. Marlet are with LIGM, Ecole des Ponts, Univ Gustave Eiffel, CNRS, Marne-la-Vallée, France. R. Marlet is also with valeo.ai, Paris, France.*
  E-mail: {yang.xiao, vincent.lepetit, renaud.marlet}@enpc.fr

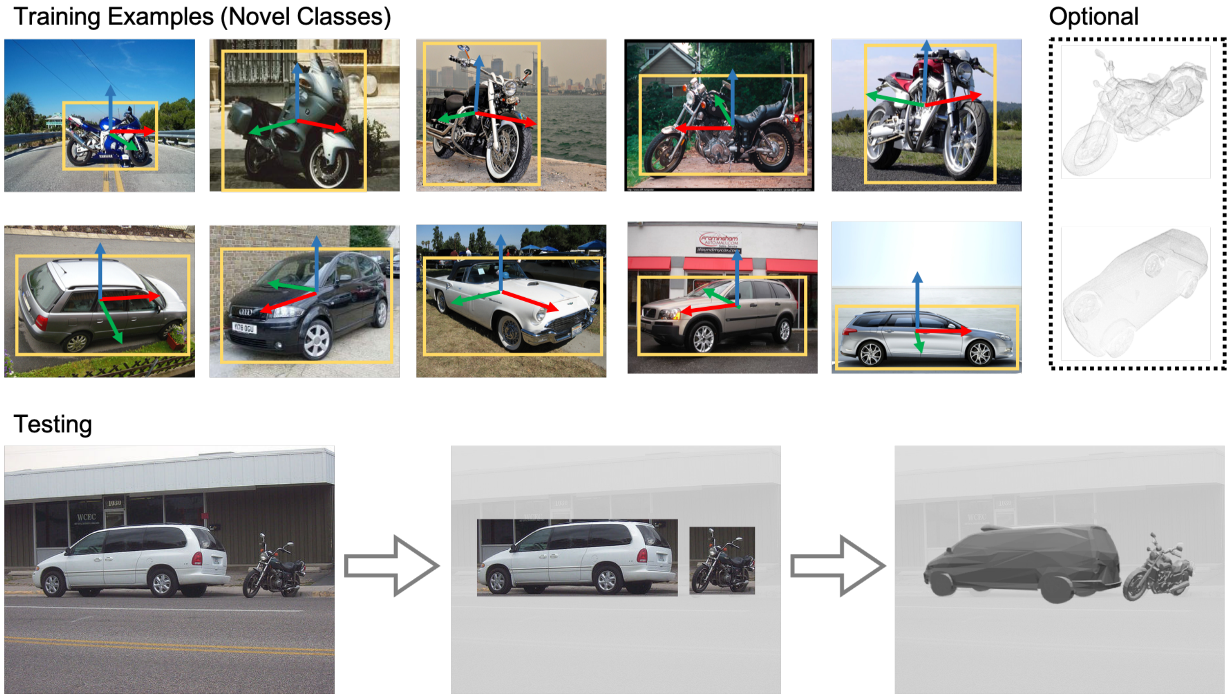


Fig. 1: Few-shot object detection and viewpoint estimation. Starting with images labeled with bounding boxes and viewpoints of objects from base classes, and given only a few similarly labeled images for new categories (top), we predict in a query image the 2D location of objects of new categories, as well as their 3D poses, optionally leveraging just a few arbitrary 3D class models (bottom). To the best of our knowledge, we are the first to conduct this joint task of object detection and viewpoint estimation in the few-shot regime.

consistently oriented with respect to verticality according to their regular usage. Besides, objects often present a main vertical symmetry plane and/or a notion of front and back. This is enough to define a somehow "natural" canonical frame, possibly up to symmetry, even for remotely-related classes such as *chair* and *bed*. Then, leveraging pose consistency across pictured classes, we learn a category-agnostic image feature embedding space by sharing the network weights between all categories. This allows the network to exploit the geometrical similarities shared across different categories. As for the few-shot task, we first train on base classes and then simply operate a balanced fine-tuning with both novel and base classes, which is simpler and more direct than explicit feature-comparison approaches used in other class-agnostic few-shot methods [11], [12], towards segmentation or object counting. Like for any few-shot method, our strategy works all the better when there are more similarities between base and novel classes. Despite its simplicity, we find that this category-agnostic prediction approach does not only outperform the state-of-the-art methods on few-shot viewpoint estimation, but also largely reduces the network complexity.

Moreover, we propose to optionally use 3D models, which we call ”exemplar 3D models”, as additional input of the viewpoint estimation network and to condition the final viewpoint prediction on both the image embeddings and the 3D model embeddings through a feature aggregation module. These 3D models are easy to obtain for many categories [13]. They do not need to correspond exactly to the objects present in the input image—in fact we use the same exemplar 3D model for all the objects of a same category. Their purpose is only to help the viewpoint estimation network generalize better to new classes. The use of these exemplar 3D models for viewpoint estimation is similar to exploiting images annotated with bounding boxes for object detection, from which we extract the task-aware class-specific information. Using this information, we obtain an embedding for each class and condition the network prediction on both the class-informative embeddings and instance-wise query image embeddings through a feature aggregation module. This exploitation of 3D models leads to a clear performance improvement of viewpoint estimation on novel classes under the few-shot learning regime.

Finally, by combining our few-shot object detection with our few-shot viewpoint estimation, we address the joint problem of learning to detect objects in images and to estimate their viewpoints from only a few shots. This corresponds to the real world in contrast with other few-shot viewpoint estimation methods, that only evaluate in the ideal case with ground-truth classes and ground-truth bounding boxes. We demonstrate that our few-shot viewpoint estimation method can achieve very good results even based on the predicted classes and bounding boxes.

To summarize, our contributions are three-fold. **First**, we define a simple yet effective unifying framework that addresses both few-shot object detection and few-shot viewpoint estimation in images, and achieves state-of-the-art performances across various benchmarks. **Second**, we show how the performance of our category-agnostic few-shot viewpoint estimation method is boosted by the additional knowledge at training time of one or a few exemplar 3D models per class, requiring only viewpoint supervision (as opposed to extra annotations such as keypoints), which is a realistic scenario. **Third**, we propose an evaluation of the new few-shot learning task of jointly detecting objects and estimating their viewpoint, for which we provide promising results. Our data and code are available at http://imagine.enpc.fr/~xiaoy/FSDetView/.

This paper is an extended version of our previous work [14], with several improvements:

- introducing a category-agnostic few-shot viewpoint estimation method that predicts viewpoint directly from image embeddings, without relying on any 3D models during training and testing.
- providing a more in-depth explanation of implementation details and a thorough analysis of different components of the method.
- extended evaluation of joint few-shot object detection and viewpoint estimation on Pascal3D+ and Pix3D.

2 Related work

Since there is a vast amount of literature on both object detection and viewpoint estimation, we focus here on recent work that targets these tasks in the case of limited labels.

Few-shot learning. Few-shot learning has been defined for the purpose of transferring the knowledge learned from large base categories with abundant annotated samples to novel categories with only a few annotated samples. Li *et al.* [15] employ Bayesian inference to generalize knowledge from a pre-trained model to perform one-shot learning. While some methods propose to hallucinate additional training examples for the data-starved novel classes [16], [17], [18], [19], recent work is more focused on meta-learning [5], [20], [21], [22], [23], [24], [25], [26] , which we detail below.

Such meta-learning-based methods can be roughly divided into three categories. 1) Metric-learning-based approaches [5], [22], [23], [25], [27], [28], [29], [30] aim to learn an embedding space that is efficiently transferable for scarcely annotated training samples. MatchingNet [5] uses the cosine similarity to find the most similar class for the query image among a small set of labeled images. ProtoNet [22] replaces the weighted nearest neighbor classifier in [5] by a linear classifier where the squared Euclidean distance is used. RelationNet [23] proposes to learn the relation between support data and query data through a neural network, which is similar to CAN [29] and LGM-Net [30]. 2) Optimization-based fast adaptation approaches [21], [24], [31], [32], [33] intend to adjust the optimization algorithm such that the model can quickly converge on the few annotated samples. Ravi and Larochelle [24] train a LSTM-based meta-learner to learn a classifier in new few-shot tasks. Model-Agnostic Meta-Learning (MAML) [31] explicitly optimizes the parameters of the model such that a small number of gradient descents on the novel task will produce good generalization performance. Sun *et al.* [33] propose to adapt a model for few-shot learning tasks by learning scaling and shifting functions of model weights for multiple tasks. 3) Parameter-prediction-based approaches [20], [34], [35] attempt to generate network parameters for new tasks. Bertinetto *et al.* [20] learn the parameters of factorized weight layers based on a single example of each class. Gidaris and Komodakis [34] introduce an attention-based few-shot classification weight generator.

Besides the standard few-shot learning setting, there is also other work focused on different settings. In transductive few-shot learning [26], [36], the unlabeled query set is assumed to be accessible for training and testing. This is highly related to the semi-supervised few-shot learning [37], [38], where an extra unlabeled training set is allowed. These approaches only tackle the problem of few-shot image classification, while we seek to study the more challenging and under-explored problem of few-shot object detection and viewpoint estimation.

Object detection with limited annotations. The general deep-learning models for object detection can be divided into two groups: proposal-based methods and direct methods without proposals. While the R-CNN series [39], [40], [41] and FPN [42] fall into the former line of work, the YOLO series [43], [44] and SSD [45] belong to the latter. All these methods mainly focus on learning from abundant data to improve detection regarding accuracy and speed. Yet, there are also some attempts to solve the problem with limited labeled data.

Chen *et al.* [46] propose an approach based on transfer learning to train a network to detect objects of novel classes from just a few annotated images in the target domain. Recent work [2], [3] proposes to integrate a reweighting module in existing detection models such as YOLO or Faster R-CNN, which enables the network to learn generalizable features and automatically adjust them for novel class detection through a set of class-specific coefficient vectors produced from the support samples. Similar to the parameter-prediction-based few-shot learning methods, Wang *et al.* [1] propose to disentangle the learning of category-agnostic and category-specific components in the detection model and learn a weight-generation module to predict category-specific parameters for novel classes. More recently, Wang *et al.* [7] find that a simple fine-tuning detection model can achieve impressive results on novel classes using a category-agnostic box regressor and a cosine-similarity-based box classifier. They also analyze the variance of the detection results obtained with different support samples and show the importance of averaging evaluation results over multiple experimental runs, which has been sometimes disregarded in previous work.

In contrast, we replace the feature reweighting module in [3] by a feature aggregation module that achieves a better detection performance under the few-shot regime. Following [1], [3], [7], we also conduct multiple experiments with randomly selected support samples and report average results to prevent biases in evaluation.

There is also prior work focusing on object detection with limited annotations in different settings. Weakly-supervised detection [47], [48], [49] considers the problem of training a detection model with only image-level labels, but without bounding box annotations that are more difficult to acquire. Semi-supervised detection [50], [51], [52] makes use of a small amount of labeled images per class to generate pseudo labels on a large amount of unlabeled images for training. Zero-shot detection [53], [54], [55] considers there is no available

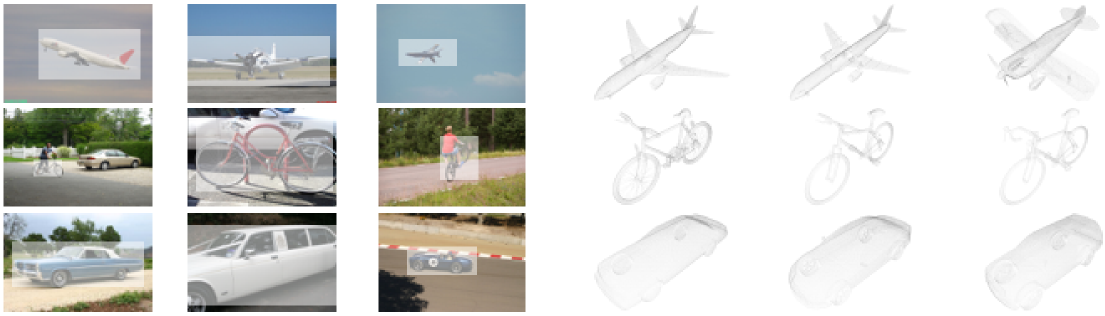


Fig. 2: Examples of class data for object detection (left) & viewpoint estimation (right). While the images with box masks capture the characteristic appearances and the common context for different classes, the point clouds in a canonical object space capture the geometric information such as the principal axis of symmetry and the position of the main object parts.

annotations for the novel categories and relies on external information such as inter-class relation or word embeddings for novel class detection. Since these settings differs from the few-shot object detection setting, they are out of our scope in this work.

Viewpoint estimation with limited annotations. Deep-learning methods for viewpoint estimation follow roughly three different paths: direct estimation of Euler angles [9], [56], [57], [58], [59], template-based matching [60], [61] that encodes images in latent spaces and compares them against a dictionary of pre-defined viewpoints, and keypoint detection relying on 3D bounding box corners [62], [63], [64] or semantic keypoints [8], [65]. Training a viewpoint estimation network requires a large amount of images manually labeled with aligned 3D CAD models or 2D keypoints, which are expensive to obtain in terms of time and human labor. To overcome this limitation, recent works propose to conduct unsupervised viewpoint estimation [66] or predict generic 3D keypoints for all object classes [8]. Alternatively, along with the improvement of image quality and processing speed in rendering methods, abundant synthetic images can be automatically generated for network training [57], [60], [67], [68]. Some work also focuses on training the viewpoint estimation network on a collection of unlabeled images by self-supervised learning [69], [70], [71].

Most of the existing viewpoint estimation methods are designed for known object categories or instances; very little work reports performance on unseen objects [8], [9], [10], [64], [72], [73]. Zhou *et al.* [8] propose a category-agnostic method to learn general keypoints for both seen and unseen objects, while Xiao *et al.* [9] show that better results can be obtained when exact 3D models of the objects are additionally provided. Park *et al.* [73] propose a novel framework for 6D pose estimation of unseen objects by learning a latent 3D representation from a set of reference views for each target object during inference. In contrast to these category-agnostic methods, Tseng *et al.* [10] specifically address the few-shot scenario by training a category-specific viewpoint estimation network for novel classes with limited samples. More recently, Wang *et al.* [74] study the problem of learning to estimate the 3D object pose from a few labeled examples and a collection of unlabeled data, and show promising results in particular on vehicle categories.

Instead of using exact 3D object models as [9], we propose a meta-learning approach to extract a class-informative canonical shape feature vector for each novel class from a few labeled samples, with random object models. Besides, our network can be applied to both base and novel classes without changing the network architecture, while [10] requires a separate meta-training procedure for each class and needs keypoint annotations in addition to the viewpoint.

3 Method

In this section, we first introduce the setup for few-shot object detection and few-shot viewpoint estimation (Section 3.1). Then, we present our network architecture for these two tasks with class data (Section 3.2) and a fine-tuning category-agnostic viewpoint estimation method (Section 3.3). Finally, we describe the learning procedure adopted in both few-shot learning tasks (Section 3.4).

3.1 Few-shot Learning Setup

For both the object detection and viewpoint estimation tasks, we assume we have training samples $(x, y) \in (\mathcal{X}, \mathcal{Y})$. A few 3D shapes may also be available for viewpoint estimation.

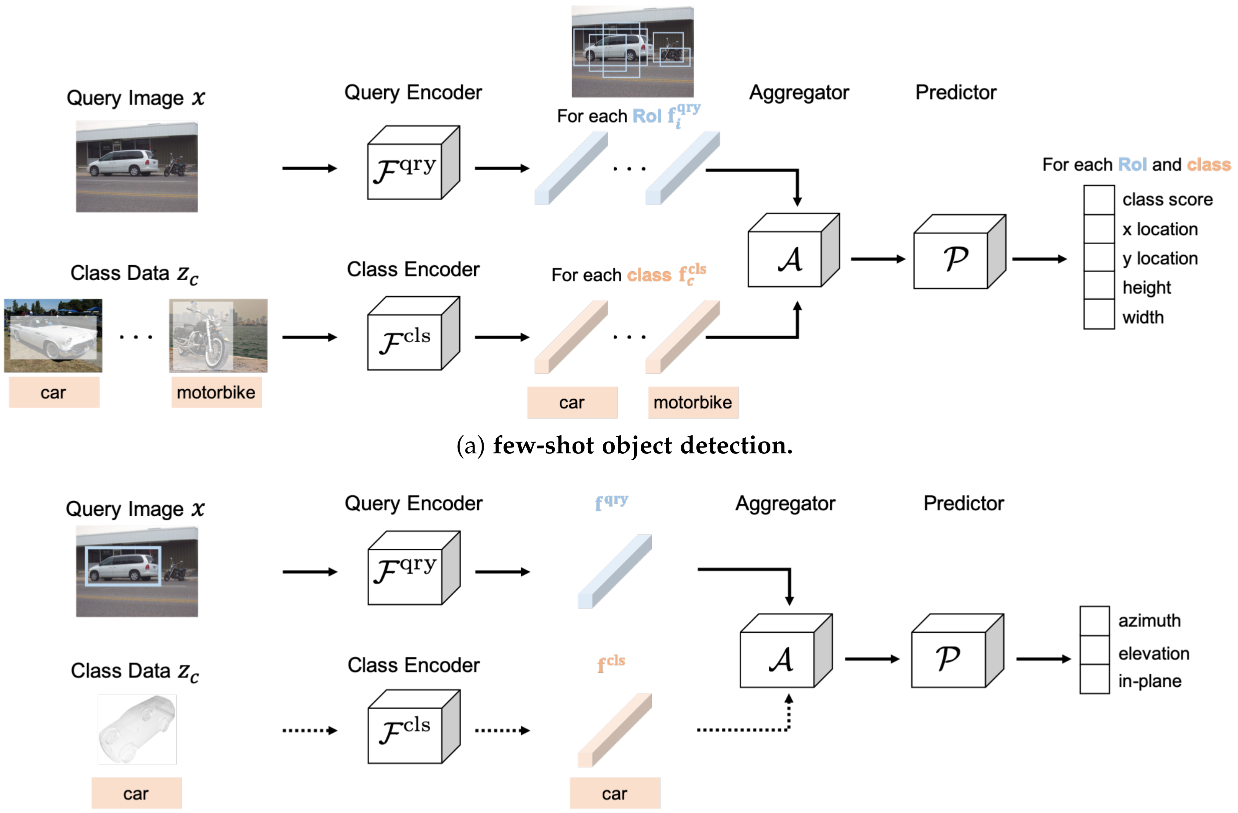
- In the case of object detection, x is an image, $y = \{(\mathsf{cls}_i, \mathsf{box}_i) \mid i \in \mathsf{Obj}_x\}$ indicates the class label cls_i and bounding box box_i of each object i in the image.
- In the case of viewpoint estimation, $x = (\mathsf{cls}, \mathsf{box}, \mathsf{img})$ represents an object of class $\mathsf{cls}(x)$ pictured in bounding box $\mathsf{box}(x)$ of an image $\mathsf{img}(x)$; $y = \mathsf{ang} = (\mathsf{azi}, \mathsf{ele}, \mathsf{inp})$ is the 3D pose (viewpoint) of the object, given by Euler angles (azimuth, elevation, in-plane rotation).

For each class $c \in C = \{\mathsf{cls}_i \mid x \in \mathcal{X}, i \in \mathsf{Obj}_x\}$, we consider a set Z_c of *class data* (see Figure 2) to learn from using meta-learning:

- For object detection, $Z_c = \{(x, \mathsf{mask}_i) \mid x \in \mathcal{X}, i \in \mathsf{Obj}_x\}$ is made of images x plus an extra channel with a binary mask for bounding box box_i of object $i \in \mathsf{Obj}_x$.
- For viewpoint estimation, Z_c is an optional, additional set of 3D models of class c, which is not used in the purely image-based category-agnostic variant.

At each training iteration, class data z_c is randomly sampled in Z_c for each $c \in C$.

In the few-shot setting, we have a partition of the classes $C = C_{\text{base}} \cup C_{\text{novel}}$, with many samples for base classes in C_{base} and only a few samples (possibly also including a few



(a) **few-shot object detection.**

(b) **few-shot viewpoint estimation.**

Fig. 3: **Method overview. (a)** For object detection, we sample for each class c one image x in the training set containing an object j of class c, to which we add an extra channel for the binary mask mask_j of the ground-truth bounding box box_j of object j. Each corresponding vector of class features $\mathrm{f}_c^{\mathrm{cls}}$ (red) is then combined with each vector of query features $\mathrm{f}_i^{\mathrm{qry}}$ (blue) associated to one of the region of interest i in the query image, via an aggregation module. Finally, the aggregated features $\mathrm{f}_{i,c}^{\mathrm{agg}}$ pass through a predictor that estimates a class probability $\mathsf{cls}_{i,c}$ and regresses a bounding box $\mathsf{box}_{i,c}$. **(b)** For few-shot viewpoint estimation, we represent the 3D pose using three Euler angles. We estimate them either directly from the query features extracted from the image or, optionally, indirectly from aggregated features made of both query features and class information extracted from a few point clouds with coordinates in a normalized, canonical object space.

shapes) for novel classes in C_{novel}. The goal is to transfer the knowledge learned on base classes with abundant samples to little-represented novel classes.

3.2 Few-shot Learning with Class Data

Our general approach has three steps illustrated in Figure 3. First, query data x and class-informative data z_c pass respectively through the query encoder $\mathcal{F}^{\mathrm{qry}}$ and the class encoder $\mathcal{F}^{\mathrm{cls}}$ to generate corresponding feature vectors , for each each region of interest (RoI) and each class respectively. Next, a feature aggregation module $\mathcal{A}$ combines a query feature (for a given RoI) with a class feature. Finally, the output of the network is obtained by passing each aggregated feature through a task-specific predictor $\mathcal{P}$:

- For object detection, the predictor estimates a classification score and an object location (i.e.., bounding box) for each region of interest (RoI) and each class.
- For viewpoint estimation, the predictor selects quantized angles by classification, that are refined using regressed angular offsets.

3.2.1 *Few-shot Object Detection.*

We adopt the popular Faster R-CNN [40] approach in our few-shot object detection network (see Figure 3(a)). The query encoder $\mathcal{F}^{\mathrm{qry}}$ includes the backbone, the region proposal network (RPN) and the proposal-level feature alignment module. In parallel, the class encoder $\mathcal{F}^{\mathrm{cls}}$ is the backbone sharing the same weights as $\mathcal{F}^{\mathrm{qry}}$ except for the first convolutional layer, that has an additional fourth channel for extracting class features from RGB images with binary masks of the object bounding boxes [2], [3]. Each extracted vector of query features is aggregated with each extracted vector of class features before being processed for class classification and bounding box regression:

$$\begin{gathered}(\mathsf{cls}_{i,c}, \mathsf{box}_{i,c}) = \mathcal{P}\Big(\mathcal{A}\big(\mathrm{f}_i^{\mathrm{qry}}, \mathrm{f}_c^{\mathrm{cls}}\big)\Big) \\ \text{for } \ \mathrm{f}_i^{\mathrm{qry}} \in \mathcal{F}^{\mathrm{qry}}(x),\ \mathrm{f}_c^{\mathrm{cls}} = \mathcal{F}^{\mathrm{cls}}(z_c),\ c \in C_{\mathrm{train}}\end{gathered} \tag{1}$$

where C_{train} is the set of all training classes, and where $\mathsf{cls}_{i,c}$ and $\mathsf{box}_{i,c}$ are the predicted classification scores and object locations for the i^{th} RoI in query image x and for class c. The prediction branch $\mathcal{P}$ is implemented as two fully-connected layers of size 4096 without activation that output respectively $N_{\mathrm{train}} = |C_{\mathrm{train}}|$ classification scores and

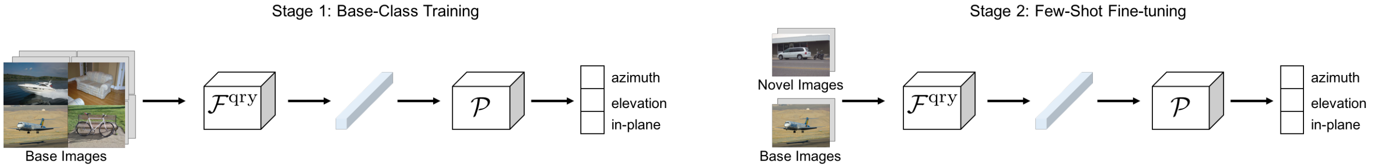

Stage 1: Base-Class Training
Base Images
$\mathcal{F}^{\mathrm{qry}}$
$\mathcal{P}$
azimuth
elevation
in-plane
Stage 2: Few-Shot Fine-tuning
Novel Images
Base Images
$\mathcal{F}^{\mathrm{qry}}$
$\mathcal{P}$
azimuth
elevation
in-plane

Fig. 4: Illustration of our category-agnostic viewpoint estimation approach without using 3D models. The network is first trained on abundant labeled images of base classes (left), then fine-tuned on a balanced set of images containing both base and novel classes (right).

N_{train} box regressions for each RoI. The final predictions are obtained by concatenating all the class-wise network outputs.

Cosine similarity for box classifier. Inspired by Wang *et al.* [7], we use a cosine-similarity-based classifier in the bounding box predictor. We note the weight matrix of the box classifier as $\mathrm{W} = [\mathrm{w}_1, \mathrm{w}_2, \ldots, \mathrm{w}_c]$, where $\mathrm{w}_c \in \mathbb{R}^d$ is the class-wise weight vector and d is the dimension of the aggregated features. Thus, the classification score for the i^{th} RoI and class c can be written as:

$$\mathsf{cls}_{i,c} = \frac{\alpha \mathcal{A}\big(\mathrm{f}_i^{\mathrm{qry}}, \mathrm{f}_c^{\mathrm{cls}}\big)^\top \mathrm{w}_c}{\|\mathcal{A}\big(\mathrm{f}_i^{\mathrm{qry}}, \mathrm{f}_c^{\mathrm{cls}}\big)\| \|\mathrm{w}_c\|}\,, \tag{2}$$

where α is a scaling factor, set to 20 in all experiments. The instance-level feature normalization used in this cosine-similarity-based classifier was found empirically to be helpful in reducing the intra-class variance and improving the detection accuracy of novel classes.

3.2.2 Few-shot Viewpoint Estimation.

For few-shot viewpoint estimation, we rely on the recently proposed PoseFromShape [9] architecture to implement our network. To create class data z_c, we transform the 3D models in the dataset into point clouds by uniformly sampling points on the surface, with coordinates in a normalized, canonical object space. The query encoder $\mathcal{F}^{\mathrm{qry}}$ and class encoder $\mathcal{F}^{\mathrm{cls}}$ (cf. Figure 3(b)) correspond respectively to the image encoder ResNet-18 [75] and shape encoder PointNet [76] in PoseFromShape. By aggregating the query features and class features, we estimate the three Euler angles via the predictor $\mathcal{P}$, which is implemented as a three-layer fully-connected network of sizes 800, 400, 200, each layer being followed by a batch normalization and ReLU activation:

$$\begin{aligned}(\mathsf{azi}, \mathsf{ele}, \mathsf{inp}) &= \mathcal{P}\Big(\mathcal{A}(\mathrm{f}^{\mathrm{qry}}, \mathrm{f}^{\mathrm{cls}})\Big)\\ \text{with}\ \ \mathrm{f}^{\mathrm{qry}} &= \mathcal{F}^{\mathrm{qry}}(\mathsf{crop}(\mathsf{img}(x), \mathsf{box}(x))), \text{ and}\\ \mathrm{f}^{\mathrm{cls}} &= \mathcal{F}^{\mathrm{cls}}(z_c),\ c = \mathsf{cls}(x)\end{aligned} \tag{3}$$

where $\mathsf{crop}(\mathsf{img}(x), \mathsf{box}(x))$ indicates that the query features are extracted from the object-centred crops. Unlike the object detection making a prediction for each class, here we only make the prediction for the object class $\mathsf{cls}(x)$ by passing the corresponding class data through the network. We also use the mixed classification-and-regression viewpoint estimator of [9]: the output consists of angular bin classification scores and within-bin offsets for three Euler angles: azimuth (azi), elevation (ele), and in-plane rotation (inp).

3.2.3 Feature Aggregation.

In recent few-shot object detection methods such as FSRW [2] and Meta R-CNN [3], features are aggregated by reweighting the query features $\mathrm{f}^{\mathrm{qry}}$ according to the output $\mathrm{f}^{\mathrm{cls}}$ of the class encoder $\mathcal{F}^{\mathrm{cls}}$:

$$\mathcal{A}(\mathrm{f}^{\mathrm{qry}}, \mathrm{f}^{\mathrm{cls}}) = \mathrm{f}^{\mathrm{qry}} \odot \mathrm{f}^{\mathrm{cls}}\,, \tag{4}$$

where $\odot$ represents element-wise multiplication (Hadamard product) and $\mathrm{f}^{\mathrm{qry}}$ has the same number of channels as $\mathrm{f}^{\mathrm{cls}}$. By jointly training the query encoder $\mathcal{F}^{\mathrm{qry}}$ and the class encoder $\mathcal{F}^{\mathrm{cls}}$ with this reweighting module, it is possible to learn to generate meaningful reweighting vectors $\mathrm{f}^{\mathrm{cls}}$. $\mathcal{F}^{\mathrm{qry}}$ and $\mathcal{F}^{\mathrm{cls}}$ actually share their weights, except the first layer [3].

We choose to rely on a slightly more complex aggregation scheme. The fact is that feature subtraction is a different but also effective way to measure similarity between image features [77], [78]. The image embedding $\mathrm{f}^{\mathrm{qry}}$ itself, without any reweighting, contains relevant information too. Our aggregation thus concatenates the three forms:

$$\mathcal{A}(\mathrm{f}^{\mathrm{qry}}, \mathrm{f}^{\mathrm{cls}}) = [\mathrm{f}^{\mathrm{qry}} \odot \mathrm{f}^{\mathrm{cls}}, \mathrm{f}^{\mathrm{qry}} - \mathrm{f}^{\mathrm{cls}}, \mathrm{f}^{\mathrm{qry}}]\,, \tag{5}$$

where $[\cdot,\cdot,\cdot]$ represents channel-wise concatenation. The last part of the aggregated features in Eq. (5) is independent of the class data. As observed experimentally in Table 3, this partial disentanglement does not only improve few-shot detection performance, it also reduces the variation introduced by the randomness of support samples.

3.3 Category-agnostic Viewpoint Estimation

We also consider the case where no 3D model is provided. In this case, we bypass the requirement of task-aware class data as mentioned in the previous section and we estimate viewpoints only from the image embeddings. Given a query object x pictured in image $\mathsf{img}(x)$ and its bounding box $\mathsf{box}(x)$, the query encoder generates an image embedding $\mathrm{f}^{\mathrm{qry}}$. Then, given such an embedding, the viewpoint prediction component estimates the three Euler angles:

$$\begin{aligned}(\mathsf{azi}, \mathsf{ele}, \mathsf{inp}) &= \mathcal{P}\Big(\mathrm{f}^{\mathrm{qry}}\Big)\\ \text{with}\ \ \mathrm{f}^{\mathrm{qry}} &= \mathcal{F}^{\mathrm{qry}}(\mathsf{crop}(\mathsf{img}(x), \mathsf{box}(x)))\,.\end{aligned} \tag{6}$$

The feature extraction module is category-agnostic and all object classes share the same prediction module. And the viewpoint predictor $\mathcal{P}$ is implemented in the same way as in Section 3.2.2. Therefore, the network can fully leverage the geometrical similarities between related categories such as *bicycle* and *motorbike*.

As illustrated in Figure 4, we first train on a large base-class dataset and then fine-tune on a balanced dataset consisting of base and novel classes. While not exactly following the general framework of Figure 3, it follows a related pattern, where the 3D branch is removed, as well as, consequently, the aggregation module. This simple yet effective approach outperforms previous methods on few-shot viewpoint estimation (see Section 4.2).

Following previous few-shot approaches, we fine-tune the network on both base and novel categories for "learning without forgetting", which prevents the network to only focus on increasing its performance on novel categories ignoring possible dramatic drops on base categories.

3.4 Learning Procedure

Our learning procedure consists of two phases: *base-class training* on many samples from base classes ($C_{\text{train}} = C_{\text{base}}$), followed by *few-shot fine-tuning* on a balanced small set of samples from both base and novel classes ($C_{\text{train}} = C_{\text{base}} \cup C_{\text{novel}}$). More precisely, in the K-shot fine-tuning stage where only K labeled samples are available for each novel class, we randomly select K samples for each base class to balance the training iterations between base and novel classes. In both phases, we optimize the network using the same loss function.

3.4.1 Loss Function

Detection loss function.

We optimize our few-shot object detection network using the same loss function as Meta R-CNN [3]:

$$\mathcal{L} = \mathcal{L}_{\text{rpn}} + \mathcal{L}_{\text{cls}} + \mathcal{L}_{\text{loc}} + \mathcal{L}_{\text{meta}} \,, \tag{7}$$

where $\mathcal{L}_{\text{rpn}}$ is applied to the output of the RPN to distinguish foreground from background and refine the proposals, $\mathcal{L}_{\text{cls}}$ is a cross-entropy loss for box classification, $\mathcal{L}_{\text{loc}}$ is a Huber loss for box regression, and $\mathcal{L}_{\text{meta}}$ is a cross-entropy loss encouraging class features to be diverse for different classes [3].

Viewpoint loss function. For the task of estimating viewpoints, we discretize each Euler angle with a bin size of 15 degrees and use the same loss function as PoseFromShape [9] to train the network:

$$\mathcal{L} = \sum_{\theta \in \{\text{azi,ele,inp}\}} \mathcal{L}_{\text{cls}}^{\theta} + \mathcal{L}_{\text{reg}}^{\theta} \,, \tag{8}$$

where $\mathcal{L}_{\text{cls}}^{\theta}$ is a cross-entropy loss for angle bin classification of Euler angle θ, and $\mathcal{L}_{\text{reg}}^{\theta}$ is a Huber loss for the regression of offsets relatively to bin centers. Here we remove the meta loss $\mathcal{L}_{\text{meta}}$ used in object detection since we want the network to learn useful inter-class similarities for viewpoint estimation, instead of the inter-class differences for box classification in object detection.

3.4.2 Class Data Construction

For viewpoint estimation, unless otherwise stated, we make use of all the 3D models available for each class (typically less than 10) during both training stages. In contrast, the class data used in object detection requires the information of object class and location, which is limited for novel classes by the number of annotated samples. Therefore, we use a large number of class data for base classes in the base training stage (typically $|Z_c| = 200$, as in Meta R-CNN [3]) and limit the size of Z_c to the number of shots for both base and novel classes in the K-shot fine-tuning stage ($|Z_c| = K$).

For inference, instead of randomly sampling class data from the dataset as done during training, we construct class features once and for all after learning is finished: for each class c, we average all class features used in the few-shot fine-tuning stage:

$$\mathrm{f}_c^{\text{cls}} = \frac{1}{|Z_c|} \sum_{z_c \in Z_c} \mathcal{F}^{\text{cls}}(z_c) \,. \tag{9}$$

This corresponds to the offline computation of all orange feature vectors in Figure 3(a).

4 Experiments

In this section, we first evaluate on few-shot object detection (Section 4.1) and few-shot viewpoint estimation benchmarks (Section 4.2) to empirically assess the effectiveness of our method. For a fair comparison, we use the same splits between base and novel classes as used in previous work [2], [10] and report the performance averaged over multiple runs with different groups of few-shot training examples to obtain a sensible accuracy estimation [7], [10]. Furthermore, we conduct an evaluation of the joint task of few-shot object detection and viewpoint estimation on three datasets to demonstrate the generalization capacity of our method for both tasks in the few-shot regime (Section 4.3). We conclude this empirical study with limitations of our approach (Section 4.4).

4.1 Few-shot Object Detection

We adopt a well-established evaluation protocol for few-shot object detection [1], [2], [3] and report performance on PASCAL VOC [79], [80] (reported in Table 1) and MS-COCO [81] (reported in Table 2).

4.1.1 Experimental Setup

Datasets. PASCAL VOC [79], [80] is a small-scale object detection dataset containing 20 object categories. Following the common protocol [39], [40], [44], we use the test set of VOC 2007 [79] for testing and the train-val set of VOC 07-12 [80] for training, which results in 16,551 training images and 4,952 testing images. Among the 20 object categories, [2] introduces three few-shot splits by randomly selecting 5 classes as the novel ones while keeping the remaining 15 ones as the base: (*bird, bus, cow, motorbike, sofa / rest*); (*aeroplane, bottle, cow, horse, sofa / rest*); (*boat, cat, motorbike, sheep, sofa / rest*). We evaluate on these 3 different base/novel splits assuming that only K annotated bounding boxes are provided for each novel class during training, where K equals 1, 2, 3, 5 or 10.

MS-COCO [81] is a large-scale object detection dataset containing 80 object categories. We follow [1], [2], [3] to use 5,000 images from the mini-val set for testing and use the remaining 118,287 images in train-val set for training. Among the 80 object categories, we select the 20 classes common to PASCAL VOC as novel classes and consider the remaining 60 classes as base classes. For this dataset, the evaluation

TABLE 1: **Few-shot object detection evaluation on PASCAL VOC.** We report the Average Precision with a single IoU threshold at 0.5 ($AP^{0.5}$) under 3 different splits for 5 novel classes [2] with a small number of shots. *Results computed over single experimental run with a fix set of support images.

	Novel Set 1					Novel Set 2					Novel Set 3				
Method \ Shots	1	2	3	5	10	1	2	3	5	10	1	2	3	5	10
LSTD [46]*	8.2	1.0	12.4	29.1	38.5	11.4	3.8	5.0	15.7	31.0	12.6	8.5	15.0	27.3	36.3
FSRW [2]*	14.8	15.5	26.7	33.9	47.2	15.7	15.2	22.7	30.1	40.5	21.3	25.6	28.4	42.8	45.9
MetaDet [1]	18.9	20.6	30.2	36.8	49.6	**21.8**	23.1	27.8	31.7	43.0	20.6	23.9	29.4	43.9	44.1
Meta R-CNN [3]	19.9	25.5	35.0	45.7	51.5	10.4	19.4	29.6	34.8	45.4	14.3	18.2	27.5	41.2	48.1
TFA w/fc [7]	22.9	34.5	40.4	46.7	52.0	16.9	26.4	30.5	34.6	39.7	15.7	27.2	34.7	40.8	44.6
TFA w/cos [7]	25.3	**36.4**	42.1	47.9	52.8	18.3	**27.5**	30.9	34.1	39.5	17.9	27.2	34.3	40.8	45.6
Ours w/fc	24.2	35.3	42.2	**49.1**	57.4	21.6	24.6	**31.9**	37.0	**45.7**	21.2	30.0	**37.2**	**43.8**	49.6
Ours w/cos	**26.9**	35.7	**42.3**	48.9	**57.8**	21.2	26.7	30.6	**37.7**	45.1	**24.3**	**30.4**	36.3	41.6	**50.1**

TABLE 2: **Few-shot object detection evaluation on MS-COCO.** We report the standard MS-COCO evaluation metrics on the 20 novel classes of COCO. *Results computed over single experimental run with a fix set of support images.

		Average Precision						Average Recall					
Shots	Method	0.5:0.95	0.5	0.75	S	M	L	1	10	100	S	M	L
10	LSTD [46]*	3.2	8.1	2.1	0.9	2.0	6.5	7.8	10.4	10.4	1.1	5.6	19.6
	FSRW [2]*	5.6	12.3	4.6	0.9	3.5	10.5	10.1	14.3	14.4	1.5	8.4	28.2
	MetaDet [1]	7.1	14.6	6.1	1.0	4.1	12.2	11.9	15.1	15.5	1.7	9.7	30.1
	Meta R-CNN [3]	8.7	19.1	6.6	2.3	7.7	14.0	12.6	17.8	17.9	7.8	15.6	27.2
	FSOD [82]*	11.1	20.4	10.6	–	–	–	–	–	–	–	–	–
	MPSR [83]*	9.8	17.9	9.7	**3.3**	9.2	16.1	15.7	21.2	21.2	4.6	19.6	34.3
	TFA w/fc [7]	9.1	17.3	8.5	–	–	–	–	–	–	–	–	–
	TFA w/cos [7]	9.1	17.1	8.8	–	–	–	–	–	–	–	–	–
	Ours w/fc	12.5	27.3	9.8	2.5	13.8	19.9	20.0	25.5	25.7	7.5	27.6	38.9
	Ours w/cos	**13.6**	**28.6**	**11.3**	2.6	**14.6**	**22.1**	**20.6**	**26.8**	**27.0**	**7.9**	**28.8**	**41.3**
30	LSTD [46]*	6.7	15.8	5.1	0.4	2.9	12.3	10.9	14.3	14.3	0.9	7.1	27.0
	FSRW [2]*	9.1	19.0	7.6	0.8	4.9	16.8	13.2	17.7	17.8	1.5	10.4	33.5
	MetaDet [1]	11.3	21.7	8.1	1.1	6.2	17.3	14.5	18.9	19.2	1.8	11.1	34.4
	Meta R-CNN [3]	12.4	25.3	10.8	2.8	11.6	19.0	15.0	21.4	21.7	8.6	20.0	32.1
	MPSR [83]*	14.1	25.4	14.2	**4.0**	12.9	23.0	17.7	24.2	24.3	5.5	21.0	39.3
	TFA w/fc [7]	12.0	22.2	11.8	–	–	–	–	–	–	–	–	–
	TFA w/cos [7]	12.1	22.0	12.0	–	–	–	–	–	–	–	–	–
	Ours w/fc	14.7	30.6	12.2	3.2	15.2	23.8	22.0	28.2	28.4	8.3	30.3	42.1
	Ours w/cos	**16.4**	**32.6**	**14.7**	3.5	**17.1**	**26.2**	**23.3**	**29.7**	**29.9**	**8.8**	**31.9**	**44.7**

protocol used in previous work is to test on $K = 10$ or 30 annotated bounding boxes for each novel class.

Evaluation metrics. We measure the Average Precision (AP) of detections as the area under a precision-recall curve. For few-shot object detection on PASCAL VOC, we classically report $AP^{0.5}$, that computes AP with a single Intersection over Union (IoU) threshold at 0.5. For evaluation on MS-COCO, we use the standard MS-COCO evaluation metrics [40], [44]: mAP, $AP^{0.5}$, $AP^{0.75}$, AP^{S}, AP^{M}, AP^{L}, AR^{1}, AR^{10}, AR^{100}, AR^{S}, AR^{M}, AR^{L}. While $AP^{0.5}$ and $AP^{0.75}$ represent respectively the AP with a single IoU threshold at 0.5 and 0.75, mAP is the averaged AP over multiple IoU thresholds from 0.5 to 0.95 with a step of 0.05. Average Recall (AR) computed with the N most confident predictions per image is noted as AR^{N}, where N equals 1, 10 or 100. Moreover, we report the detection performance across different object scales: S (small: $\text{area} < 32^2$ square pixels), M (medium: $32^2 \leq \text{area} < 96^2$) and L (large: $96^2 \leq \text{area}$).

Training details. We employ the same learning scheme as [3], which uses the SGD optimizer with an initial learning rate of 10^{-3} and a batch size of 4. Weight decay and momentum are set to 0.0005 and 0.9, respectively. In the first training stage, we train for 20 epochs and divide the learning rate by 10 after each 5 epochs. In the second stage, we train for 5 epochs with a learning rate of 10^{-3} and another 4 epochs with a learning rate of 10^{-4}. For anchor scales, we use three scales $(128^2, 256^2, 512^2)$ for PASCAL VOC and add a fourth scale of 64^2 for MS-COCO. The three aspect ratios of anchors are set to 1:2, 1:1, 2:1. We augment the data with horizontal flipping. Training on a single Titan-X GPU takes around one day for PASCAL VOC and ten days for MS-COCO.

4.1.2 Few-shot Detection Results

Cosine similarity vs. dot product. We first compare the cosine-similarity-based box classifier (Ours w/cos) with the normal FC-based classifier (Ours w/fc) that uses a simple dot product between feature representations and weight vectors to compute the classification scores. Indeed, in few-shot learning tasks, features learned with a cosine-similarity-based classifier have been found empirically to generalize better to novel categories compared to features learned with FC-based classifier [7], [34]. As observed in Table 1 and Table 2, even though the improvement is not systematic on novel classes of PASCAL VOC, cosine similarity does bring a consistent performance boost on novel classes of COCO, compared to FC-based classifier with direct dot product.

Different feature aggregations. We analyze the impact of different feature aggregation schemes. For this purpose, we evaluate K-shot object detection on PASCAL VOC with $K = 3$ or 10. Here, we compare results obtained by models with an FC-based classifier. The results are reported in Table 3. We can see that our feature aggregation scheme

TABLE 3: **Ablation study on the feature aggregation scheme.** Using the same class splits of PASCAL VOC as in Table 1, we measure the performance of few-shot object detection on the novel classes for 3 shots and 10 shots. We report the average and standard deviation of the AP50 metric over ten runs. $\mathrm{f}^{\mathrm{qry}}$ is the query features and $\mathrm{f}^{\mathrm{cls}}$ is the class features.

	Novel Set 1		Novel Set 2		Novel Set 3	
Method \ Shots	3	10	3	10	3	10
$[\mathrm{f}^{\mathrm{qry}} \odot \mathrm{f}^{\mathrm{cls}}]$	35.0 ± 3.6	51.5 ± 5.8	29.6 ± 3.5	45.4 ± 5.5	27.5 ± 5.2	48.1 ± 5.9
$[\mathrm{f}^{\mathrm{qry}} \odot \mathrm{f}^{\mathrm{cls}}, \mathrm{f}^{\mathrm{qry}}]$	36.6 ± 7.1	49.6 ± 4.3	27.5 ± 5.7	41.6 ± 3.7	28.7 ± 5.9	44.0 ± 2.7
$[\mathrm{f}^{\mathrm{qry}} \odot \mathrm{f}^{\mathrm{cls}}, \mathrm{f}^{\mathrm{qry}}, \mathrm{f}^{\mathrm{cls}}]$	37.6 ± 7.2	54.2 ± 4.9	30.0 ± 2.9	41.0 ± 5.3	33.6 ± 5.0	47.5 ± 2.3
$[\mathrm{f}^{\mathrm{qry}} \odot \mathrm{f}^{\mathrm{cls}}, \mathrm{f}^{\mathrm{qry}} - \mathrm{f}^{\mathrm{cls}}]$	39.2 ± 4.5	55.5 ± 3.9	31.7 ± 6.2	45.2 ± 3.3	35.6 ± 5.6	48.9 ± 3.3
$[\mathrm{f}^{\mathrm{qry}} \odot \mathrm{f}^{\mathrm{cls}}, \mathrm{f}^{\mathrm{qry}} - \mathrm{f}^{\mathrm{cls}}, \mathrm{f}^{\mathrm{qry}}]$	$\mathbf{42.2 \pm 2.1}$	$\mathbf{57.4 \pm 2.7}$	$\mathbf{31.9 \pm 2.7}$	$\mathbf{45.7 \pm 1.8}$	$\mathbf{37.2 \pm 3.5}$	$\mathbf{49.6 \pm 2.2}$

$[\mathrm{f}^{\mathrm{qry}} \odot \mathrm{f}^{\mathrm{cls}}, \mathrm{f}^{\mathrm{qry}} - \mathrm{f}^{\mathrm{cls}}, \mathrm{f}^{\mathrm{qry}}]$ yields the best precision. In particular, although the difference $\mathrm{f}^{\mathrm{qry}} - \mathrm{f}^{\mathrm{cls}}$ could in theory be learned from the individual feature vectors $[\mathrm{f}^{\mathrm{qry}}, \mathrm{f}^{\mathrm{cls}}]$, the network performs better when explicitly provided with their subtraction. Moreover, our aggregation scheme significantly reduces the variance introduced by the random sampling of few-shot support data, which is a major issues in few-shot learning (although sometimes neglected).

Comparison with the state of the art. Tables 1 and 2 show the comparison with previous few-shot object detection methods. On the PASCAL VOC dataset, our method achieves the best performance in most cases, in particular when the number of shots tends to be large. This indicates that our method can better leverage the task-relevant information from novel classes when more labeled examples are provided. Moreover, it significantly improves results on MS-COCO for all evaluation metrics, which validates again the effectiveness of our approach.

4.2 Few-shot Viewpoint Estimation

Following the few-shot viewpoint estimation protocol proposed in [10], we evaluate our method in two settings: *intra*-dataset on ObjectNet3D [84] (cf. Table 4) and *inter*-dataset between ObjectNet3D and Pascal3D+ [85] (cf. Table 5).

4.2.1 Experimental Setup

Datasets. Pascal3D+ [85] is a standard evaluation benchmark used in 3D pose estimation. Unlike 6D pose estimation datasets [86], [87], [88] that usually focus on dozens of objects with limited environment variations, Pascal3D+ contains 12 man-made object categories with 2k to 4k images per category,allowing the benchmarking of object pose estimation in the wild. ObjectNet3D [84], extended from Pascal3D+, features 100 object categories, with 90,127 images and 201,888 objects in total. In both datasets, only a small number of roughly-aligned 3D models are provided for each category.

Evaluation metrics. We use the most common metrics for evaluation: Acc30, which is the percentage of estimations with a rotational error smaller than 30°, and MedErr, which is the median rotational error measured in degrees. We compute the rotational error as $\Delta(R_{\text{pred}}, R_{\text{gt}}) = \frac{\| \log(R_{\text{pred}}^{\top}, R_{\text{gt}}) \|_F}{\sqrt{2}}$, where $\| \cdot \|_F$ is the Frobenius norm. Following previous work [8], [10], we only use the non-occluded and non-truncated objects for evaluation, and assume in this subsection, for all methods, that the ground-truth classes and ground-truth bounding boxes are provided at test time.

Training details. We resize the object image crops into 224×224 pixels as the input for our viewpoint estimation networks, with (Ours w/ 3D) or without (Ours w/o 3D) using exemplar 3D models. Both networks are trained using the Adam optimizer with a batch size of 16. Weight decay is set to 0.0005. During the base-class training stage, we train for 150 epochs with a learning rate of 10^{-4}. For few-shot fine-tuning, we train for 50 epochs with learning rate of 10^{-4} and another 50 epochs with a learning rate of 10^{-5}. Standard data augmentation is applied during training, such as random rotation, random flipping and color jittering. The training is done in about one day on a single Titan-X GPU.

Compared methods. For few-shot viewpoint estimation, we compare our method to MetaView [10] and to two adaptations of StarMap [8]. More precisely, the authors of MetaView [10] re-implemented StarMap with one stage of ResNet-18 as the backbone, and trained the network with MAML [31] for a fair comparison in the few-shot regime (StarMap+M). They also provided StarMap results by just fine-tuning it on the novel classes using the scarce labeled data (StarMap+F). We consider the two variants of our method, with (Ours w/ 3D) or without 3D data (Ours w/o 3D) at training time.

4.2.2 Few-shot Viewpoint Estimation Results

Intra-dataset evaluation. We follow the protocol of [9], [10] to split the 100 categories of ObjectNet3D into 80 base classes and 20 novel classes. As shown in Table 4, our model outperforms the recently proposed meta-learning-based method MetaView [10] by a very large margin in overall performance: +16 points in Acc30 and half MedErr (from 31.5° down to 15.6°). Besides, keypoint annotations are not available for some object categories such as door, pen and shoe in ObjectNet3D. This lack of annotations limits the generalization of keypoint-based approaches [8], [10] as they require a set of manually labeled keypoints for network training. In contrast, our model can be trained and evaluated on all object classes of ObjectNet3D as we only rely on the viewpoint annotations. More importantly, our model can be directly deployed on different classes using the same architecture, while MetaView learns a set of separate category-specific semantic keypoint detectors for each class. This flexibility suggests that our approach is likely to exploit the similarities between different categories (e.g., bicycle and motorbike) and has more potentials for applications to robotics and augmented reality.

Inter-dataset evaluation. To further evaluate our method in a more practical scenario, we use a source dataset for base

TABLE 4: Intra-dataset 10-shot viewpoint estimation evaluation. We report Acc30(↑) / MedErr(↓) on the same 20 novel classes of ObjectNet3D for each method, while 80 are used as base classes. All models are trained and tested on ObjectNet3D.

Method	bed	bookshelf	calculator	cellphone	computer	door	f_cabinet
StarMap+F [8]	0.32 / 47.2	0.61 / 21.0	0.26 / 50.6	0.56 / 26.8	0.59 / 24.4	- / -	0.76 / 17.1
StarMap+M [8]	0.32 / 42.2	0.76 / 15.7	0.58 / 26.8	0.59 / 22.2	0.69 / 19.2	- / -	0.76 / 15.5
MetaView [10]	0.36 / 37.5	0.76 / 17.2	**0.92** / 12.3	0.58 / 25.1	0.70 / 22.2	- / -	0.66 / 22.9
Ours w/o 3D	0.53 / 26.8	0.82 / 9.4	0.76 / 11.6	0.54 / 24.0	0.82 / 11.8	0.86 / 3.1	0.83 / 11.1
Ours w/ 3D	**0.64** / **14.8**	**0.90** / **7.8**	0.90 / **8.2**	**0.61** / **13.2**	**0.86** / **10.3**	**0.90** / **0.8**	**0.86** / **10.2**

Method	guitar	iron	knife	microwave	pen	pot	rifle
StarMap+F [8]	0.54 / 27.9	0.00 / 128	0.05 / 120	0.82 / 19.0	- / -	0.51 / 29.9	0.02 / 100
StarMap+M [8]	0.59 / 21.5	0.00 / 136	0.08 / 117	0.82 / 17.3	- / -	0.51 / 28.2	0.01 / 100
MetaView [10]	0.63 / 24.0	0.20 / 77	0.05 / **98**	0.77 / 17.9	- / -	0.49 / 31.6	0.21 / **81**
Ours w/o 3D	0.60 / 21.5	0.08 / 118	0.21 / 137	0.91 / 8.9	0.39 / 63.2	0.64 / 17.5	0.15 / 91
Ours w/ 3D	**0.68** / **19.4**	**0.34** / **60**	**0.27** / 137	**0.93** / **7.4**	**0.47** / **36.4**	**0.76** / **11.8**	**0.28** / 87

Method	shoe	slipper	stove	toilet	tub	wheelchair	All
StarMap+F [8]	- / -	0.08 / 128	0.80 / 16.1	0.38 / 36.8	0.35 / 39.8	0.18 / 80.4	0.41 / 41.0
StarMap+M [8]	- / -	0.15 / 128	0.83 / 15.6	0.39 / 35.5	0.41 / 38.5	0.24 / 71.5	0.46 / 33.9
MetaView [10]	- / -	0.07 / 115	0.74 / 21.7	0.50 / 32.0	0.29 / 46.5	0.27 / **55.8**	0.48 / 31.5
Ours w/o 3D	0.35 / 47.2	0.19 / 125	0.86 / 11.3	0.49 / 30.2	0.50 / 32.0	**0.36** / 57.8	0.56 / 22.0
Ours w/ 3D	**0.49** / **30.6**	**0.28** / **93**	**0.91** / **9.5**	**0.69** / **17.8**	**0.65** / **16.4**	0.35 / 61.2	**0.65** / **15.6**

TABLE 5: Inter-dataset 10-shot viewpoint estimation evaluation. We report Acc30(↑) / MedErr(↓) on the 12 novel classes of Pascal3D+, while the 88 base classes are in ObjectNet3D. All models are trained on ObjectNet3D and tested on Pascal3D+.

Method	aero	bike	boat	bottle	bus	car	chair
StarMap+F [8]	0.03 / 102	0.05 / 98.8	0.07 / 99	0.48 / 31.9	0.46 / 33.0	0.18 / 80.8	0.22 / **74.6**
StarMap+M [8]	0.03 / 99	0.08 / 88.4	0.11 / 92	0.55 / 28.0	0.49 / 31.0	0.21 / 81.4	0.21 / 80.2
MetaView [10]	0.12 / 104	0.08 / 91.3	0.09 / 108	0.71 / 24.0	0.64 / 22.8	0.22 / 73.3	0.20 / 89.1
Ours w/o 3D	0.14 / 88	0.30 / 67.8	0.20 / 83	0.81 / 12.1	0.73 / 9.6	0.43 / 53.8	0.30 / 78.8
Ours w/ 3D	**0.21** / **73**	**0.33** / **64.7**	**0.25** / **78**	**0.91** / **11.6**	**0.74** / **9.0**	**0.49** / **32.8**	**0.32** / 79.1

Method	table	mbike	sofa	train	tv	All
StarMap+F [8]	0.46 / 31.4	0.09 / 91.6	0.32 / 44.7	0.36 / 41.7	0.52 / 29.1	0.25 / 64.7
StarMap+M [8]	0.29 / 36.8	0.11 / 83.5	0.44 / 42.9	0.42 / 33.9	0.64 / 25.3	0.28 / 60.5
MetaView [10]	0.39 / 36.0	0.14 / 74.7	0.29 / 46.2	0.61 / 23.8	0.58 / 26.3	0.33 / 51.3
Ours w/o 3D	0.51 / 31.2	0.36 / 49.8	0.49 / 34.6	0.62 / 16.1	0.77 / **18.7**	0.46 / 38.3
Ours w/ 3D	**0.59** / **20.9**	**0.44** / **37.2**	**0.58** / **23.9**	**0.72** / **12.1**	**0.79** / 19.0	**0.51 / 29.1**

classes and another target dataset for novel (disjoint) classes. Following the same split as MetaView [10], we use all 12 categories of Pascal3D+ as novel categories and the remaining 88 categories of ObjectNet3D as base categories. Distinct from the previous intra-dataset experiment that focuses more on the cross-category generalization capacity, this inter-dataset setup also reveals the cross-domain generalization ability.

As shown in Table 5, our approach again significantly outperforms StarMap and MetaView. Our overall improvement in inter-dataset evaluation is even larger than in intra-dataset evaluation: we gain +19 points in Acc30 and again divide MedErr by about 2 (from 51.3° down to 28.3°). This indicates that our approach, by leveraging viewpoint-relevant 3D information, not only helps the network generalize to novel classes from the same domain, but also addresses the domain shift issues when trained and evaluated on different datasets.

Visual results. We illustrate on Figure 5 viewpoint estimation for novel objects in ObjectNet3D and Pascal3D+. We show both success (green boxes) and failure cases (red boxes) to help analyze possible error types. We visualize categories giving large rotational errors: *iron*, *knife*, *rifle* and *slipper* for ObjectNet3D, *aeroplane*, *bicycle*, *boat* and *chair* for Pascal3D+. The most common failure cases come from objects with similar appearances in different poses, *e.g.*, *iron* and *knife* in ObjectNet3D, *aeroplane* and *boat* in Pascal3D+. It seems that more complex methods based on keypoints [8], [10] perform a bit better on this kind of objects, although being nevertheless grossly wrong too. Other failure cases include heavy clutter cases (*bicycle*) and large shape variations between training objects and testing objects (*chair*).

4.2.3 Ablation Study

Different 3D model representations, if any. In Table 6, we analyze the impact of different 3D model representations in our few-shot viewpoint estimation approach using exemplar 3D models. Besides using a point cloud (Point Cloud), we can also represent 3D shapes using a group of depth images (Depth) or non-textured rendered images (Rendering) captured in a set of camera locations defined on the upper hemisphere. We also use the normalized, canonical object space [89], [90], [91] to represent the 3D models by transforming the 3D coordinates into RGB values (Object Coord.). For these variants that consider 2D inputs rather than a 3D point cloud, we implement the class encoder $\mathcal{F}^{\text{cls}}$ using a ResNet-18 to extract features from images.

We find that using point clouds (with PointNet encoding) provides the best overall performance compared to training with the other 3D representations. This demonstrates the effectiveness of 3D model embedding with point clouds for viewpoint estimation. By comparing the performance gap between our methods using 3D models, regardless of the choice of 3D representation, and our method without using 3D models (first row in Table 6), we note again that the 3D models can indeed help improve the viewpoint estimation

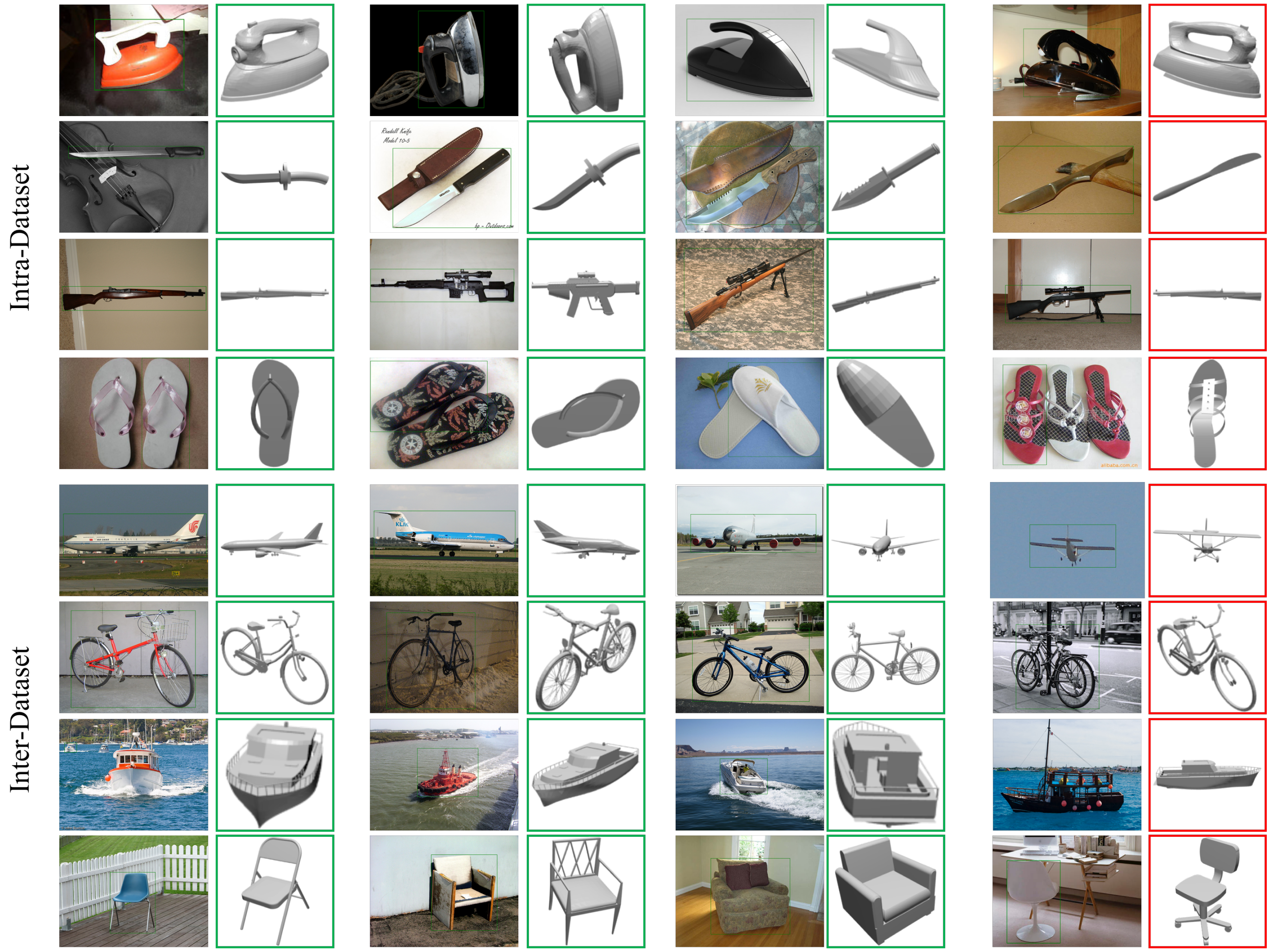


Fig. 5: Qualitative results of few-shot viewpoint estimation using ground-truth 2D bounding boxes (and classes). We visualize results on ObjectNet3D and Pascal3D+. For each category, we show three success cases (first six columns) and one failure case (last two columns). CAD models are shown here only for the purpose of illustrating the estimated viewpoint. Failure cases usually result from appearance ambiguities of a same object in different poses, or from heavily cluttered scenes.

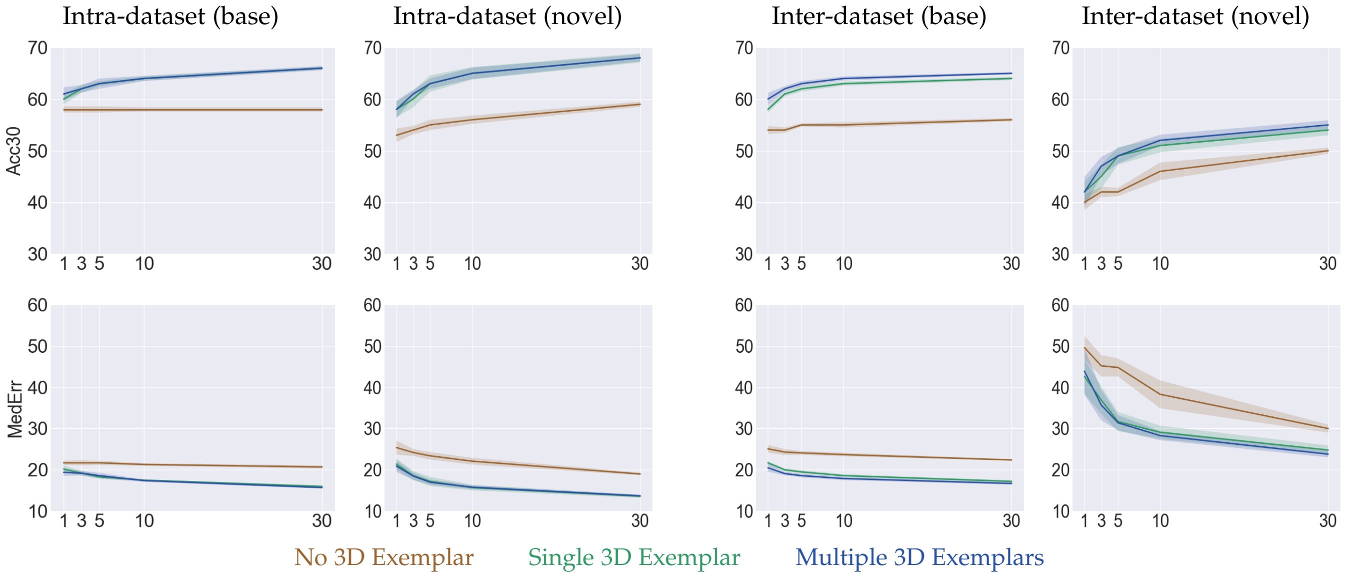


Fig. 6: Few-shot viewpoint estimation evaluation using different number of shots. For each metric, we report the average and standard deviation computed over 10 random experiments.

TABLE 6: Efficacy of different 3D representations, if any. We show few-shot viewpoint estimation results on the 20 novel classes of ObjectNet3D. The first row represents our approach without using any form of 3D information, while other rows correspond to our method using exemplar 3D models with different representations. We also plot the four different 3D representations of an example CAD model on the bottom.

	Acc30(↑) / MedErr(↓)	
3D exemplar	Base	Novel
None	0.58 ± 0.01 / 21.3 ± 0.31	0.56 ± 0.01 / 22.1 ± 0.80
Depth	0.61 ± 0.01 / 22.0 ± 0.97	0.57 ± 0.02 / 24.3 ± 1.52
Object Coord.	0.61 ± 0.01 / 22.0 ± 0.54	0.59 ± 0.02 / 23.7 ± 1.09
Rendering	0.61 ± 0.01 / 21.7 ± 0.92	0.60 ± 0.01 / 22.9 ± 0.77
Point Cloud	$\mathbf{0.64} \pm 0.01$ / $\mathbf{17.5} \pm 0.18$	$\mathbf{0.65} \pm 0.01$ / $\mathbf{15.6} \pm 0.38$

accuracy on novel classes and reduce the variance introduced by different support training samples.

Number of exemplars. We show detailed evaluation of few-shot viewpoint estimation with different number of shots in Figure 6. For both the intra-dataset and inter-dataset evaluations, we compute the accuracies and median errors on base and novel classes. We report the average results and the standard deviations computed over 10 experimental runs with different support training samples.

We first note that all variants of our viewpoint estimation approach can achieve better results when more annotated samples are provided. Secondly, we find that our approach using only one 3D exemplar model per class clearly improves the performance on both base and novel classes compared to results without using 3D models. Moreover, adding 3D information also reduces the variance on novel classes, which can clearly be seen in the inter-dataset evaluation. This shows that our method without 3D models, which relies on geometrical similarities and consistent labeling between different categories, can already learn a good image embedding space for few-shot viewpoint estimation. Yet, adding 3D models can certainly provide a more direct guidance for a better generalization towards novel categories.

On the other hand, we note that the performance gap between our approach using a single 3D exemplar per class or using multiple 3D exemplars per class is negligible compared to the gap between using or not 3D models. It demonstrates that even a single 3D model is sufficient to obtain a good 3D-aware class embedding for viewpoint estimation. It is possible that extra 3D exemplars could prove useful to generate more informative class embeddings, but it would probably require a more sophisticated feature combination than just feature averaging.

4.3 Joint Detection and Viewpoint Estimation

To further demonstrate the generality of our approach in real-world scenarios, we consider the *joint* problem of detecting objects of novel classes in images and estimating their viewpoints. The fact is that evaluating a viewpoint estimator on ground-truth classes and ground-truth bounding boxes is a toy setting [8], [10], that is not representative of actual needs. On the contrary, estimating viewpoints based on predicted detection is much more realistic and challenging. Note that our object detection model and our viewpoint estimation model were trained separately.

4.3.1 Experimental Setup

Datasets. As introduced in Section 4.2, Pascal3D+ [85] and ObjectNet3D [84] are two common viewpoint estimation benchmarks that have already been used in a number of previous publications. Apart from these two datasets, we also evaluate our method on a more recent benchmark: Pix3D [92]. This is a large-scale dataset of 10,069 image-shape pairs with accurate 2D-3D alignment. It contains 395 3D shapes of 9 object categories. Each shape is associated with a set of images capturing the exact object in various environments.

Evaluation metric. As we are considering the joint evaluation of object detection and viewpoint estimation in this section, the metric should reflect the performance of both tasks. We thus compute the percentage of objects for which the intersection over union between the ground-truth bounding box and the predicted bounding box (with the right class) is larger than 0.5 *and* the rotational error between the ground-truth viewpoint and the predicted viewpoint is smaller than 30°. This metric corresponds to the $Acc_{R\frac{\pi}{6}}$ proposed in [93], which is used to evaluate a joint focal length and 3D pose estimation approach.

Compared methods. We compare our approach to the other viewpoint estimation methods, namely MetaView [10] and StarMap+M, which is the best performing adaptation of StarMap [8] (cf. Tables 4-5). However, these methods are only evaluated on perfect detections, *i.e.*, ground-truth classes and ground-truth bounding boxes, and no code is available to rerun them on other inputs. Regarding our approach, we consider the case of imperfect detections, where classes and bounding boxes are predicted by our object detector. Note that the object class is only useful for our viewpoint estimation variant that exploits exemplar 3D models (Ours w/ 3D), as the method variant without 3D information (Ours w/o 3D) is category-agnostic.

4.3.2 Results

Intra-dataset evaluation on ObjectNet3D. To experiment with this scenario, we split ObjectNet3D into 80 base classes and 20 novel classes as done in Section 4.2, and train the object detector and viewpoint estimator using the abundant annotated samples of base classes and scarce labeled samples of novel classes. In this setting, both training and testing samples are from the same dataset, *i.e.* ObjectNet3D.

As recalled in the left part of Table 7, our few-shot viewpoint estimation outperforms other methods by a large margin when evaluated using ground-truth classes and ground-truth bounding boxes in the 10-shot setting. When using predicted classes and predicted bounding boxes, accuracy drops for most categories. One explanation is that viewpoint estimation becomes difficult when the objects are truncated by imperfect predicted bounding boxes, especially

TABLE 7: Evaluation of joint few-shot detection and viewpoint estimation. We first recall viewpoint estimation results assuming perfect detection, *i.e.*, using the ground-truth classes and ground-truth bounding boxes (cf. Tables 4-5). Then we use as input predicted classes and estimated bounding boxes given an object detector. As no code is available to evaluate StarMap+M and MetaView in this setting, we can only evaluate our viewpoint estimation method, for which we used our own detections as input. (Ours w/o 3D actually does not need to know the class as it is category-agnostic.) We report the percentage of objects that are correctly detected (right class) with IoU threshold at 0.5, and a rotational error less than 30°.

	Intra-dataset evaluation on ObjectNet3D																					Inter-dataset evaluation on Pascal3D+												
Method	bed	bshelf	calc	cphone	comp	door	fcabin	guit	iron	knife	micro	pen	pot	rifle	shoe	slipper	stove	toilet	tub	wchair	All	aero	bike	boat	bottle	bus	car	chair	table	mbike	sofa	train	tv	All
Evaluated using ground-truth classes and ground-truth bounding boxes (viewpoint estimation)																																		
StarMap+M [8]	32	76	58	59	69	–	76	59	0	8	82	–	51	1	–	15	83	39	41	24	46	3	8	11	55	49	21	21	29	11	44	42	64	28
MetaView [10]	36	76	92	58	70	–	66	63	20	5	77	–	49	21	–	7	74	50	29	27	48	12	8	9	71	64	22	20	39	14	29	61	58	33
Ours w/o 3D	53	82	76	54	82	86	83	60	8	21	91	39	64	15	35	19	86	49	50	36	56	14	30	20	81	73	43	30	51	36	49	62	77	46
Ours w/ 3D	64	90	90	61	86	90	86	68	34	27	93	47	76	28	49	28	91	69	65	35	65	21	33	25	91	74	49	32	59	44	58	72	79	51
Evaluated using predicted classes and predicted bounding boxes (detection + viewpoint estimation)																																		
Ours w/o 3D	44	73	57	43	48	60	65	60	7	5	55	17	46	4	16	12	76	41	48	19	40	14	14	10	12	73	34	19	0	20	41	64	74	31
Ours w/ 3D	56	75	70	47	53	64	65	75	39	8	57	22	57	15	36	24	82	64	58	24	50	15	22	15	15	74	42	16	0	30	54	70	74	35

for tiny objects (*shoes*) and ambiguous objects with similar appearances in different poses (*knives*, *rifles*). Yet, by comparing the performance gap between, on the one hand, our method when tested using predicted classes and predicted boxes, and, on the other hand, MetaView when tested using ground-truth classes and ground-truth boxes, we find that our approach is able to reach a better accuracy: 50% against 48%. This improvement is a strongly encouraging achievement since we free the viewpoint estimation approach from requiring the perfect ground-truth bounding boxes (and classes) without degrading the performance.

Inter-dataset evaluation on Pascal3D+. Here, we consider all 12 object categories of Pascal3D+ as novel classes, while the base classes are a set of disjoint object categories from ObjectNet3D and COCO for viewpoint estimation and object detection, respectively. We use the same split as in the inter-dataset few-shot viewpoint estimation (Section 4.2), that divides the 100 ObjectNet3D categories into 12 novel ones that intersect with Pascal3D+ and 88 remaining base classes. Besides, the 12 classes of Pascal3D+ are completely included in the 20 PASCAL VOC object categories, which are set to be the novel classes in the few-shot object detection on MS-COCO (Section 4.1). Therefore, we first use the 10-shot object detection network trained on MS-COCO to detect the novel objects on Pascal3D+, and then, using the predicted 2D bounding boxes, the 10-shot viewpoint estimation network trained on ObjectNet3D. Unlike the intra-dataset evaluation on ObjectNet3D, our networks are trained and tested on different datasets in this part.

We report the results in the right part of Table 7. Again, our few-shot viewpoint estimation network outperforms other methods by a large margin when evaluated using ground-truth classes and ground-truth bounding boxes in the 10-shot setting. Even though a performance drop appears when replacing the ground-truth bounding boxes by the predicted ones, our method using exemplar 3D models still outperforms other methods: 35% against 33%. This improvement is especially impressive considering the fact that our object detection and viewpoint estimation networks are both tested on a new dataset that is different from the training datasets, which is a big step towards realistic scenarios and industrial applications.

Visual results. We provide in Figure 7 some qualitative results of few-shot object detection and viewpoint estimation of novel objects on ObjectNet3D and Pascal3D+. For each sample we show the predicted bounding boxes on the left and the estimated viewpoints on the right (visualized by the projected CAD models). Besides the appearance ambiguities causing major viewpoint estimation errors, we note that the principal failure cases result from the target objects being missed by our object detector (iron and knife) or the objects being wrongly classified (car and motorbike). Another error is that only one bounding box is predicted for multiple objects of the same class, which usually occurs in cluttered scenes (pen). These detection errors contribute considerably to the performance drop between evaluating using ground-truth bounding boxes and evaluating using predicted bounding boxes, especially for categories mainly containing tiny objects such as knife in ObjectNet3D and bottle in Pascal3D+.

TABLE 8: Inter-dataset few-shot detection and viewpoint estimation evaluation on Pix3D. †Detection network from [93]. ‡Our few-shot object detection and viewpoint estimation networks trained and tested on different datasets.

Method	bed	chair	sofa	table	Mean
Detection + Viewpoint Estimation					
Fine-grained [94]	95	88	95	73	88
GP2C [93]	98	91	97	77	91
Detection† + Few-shot Viewpoint Estimation‡					
Ours w/o 3D	81	47	88	53	67
Ours w/ 3D	86	51	92	58	72
Few-shot Detection‡ + Few-shot Viewpoint Estimation‡					
Ours w/o 3D	68	34	81	13	49
Ours w/ 3D	71	36	87	14	52

Additional results on Pix3D. To further demonstrate the effectiveness of our few-shot object detection network and few-shot viewpoint estimation network, we follow GP2C [93] and conduct evaluation on four object categories of Pix3D: *bed*, *chair*, *sofa* and *table*. As these four classes are completely included in the 12 Pascal3D+ object categories that are considered as novel categories in the inter-dataset evaluation described before, we use the same object detector trained on MS-COCO and viewpoint estimator trained on ObjectNet3D to perform an inter-dataset evaluation on Pix3D.

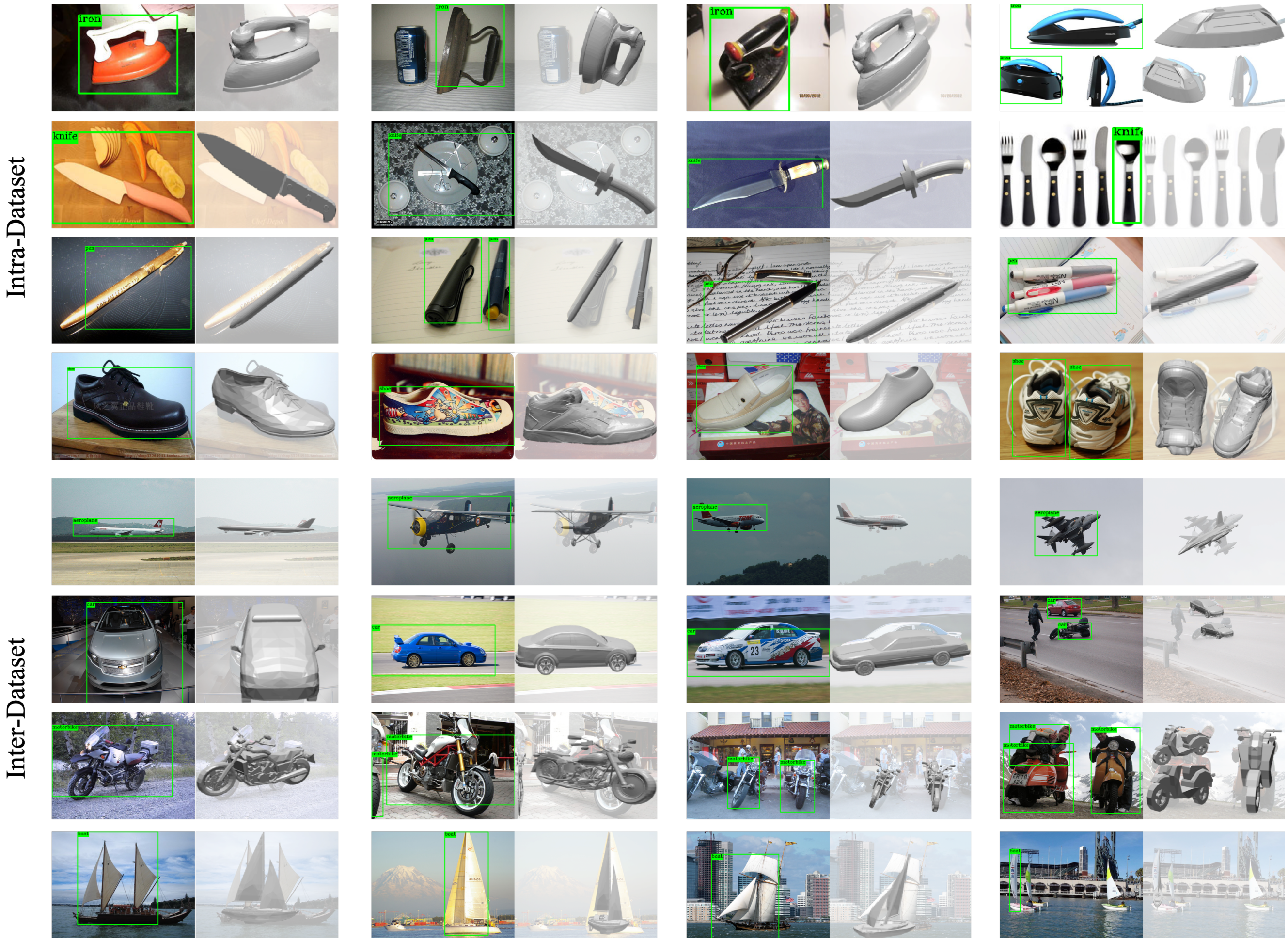


Fig. 7: Qualitative results of joint few-shot object detection and viewpoint estimation using the predicted 2D bounding boxes given by our object detection model. We visualize results on ObjectNet3D and Pascal3D+. For each category, we show three success cases (the first six columns) and one failure case (the last two columns). For each testing image, we project the CAD model of the corresponding class into the predicted 2D bounding box and rotate it according to the estimated viewpoint. Error cases include: missing target objects (iron, knife, boat); failed classification (motorbike, car); cluttered objects being detected as one (pen); successful detection but failed viewpoint estimation (shoe and airplane).

We first report our results evaluated using the 2D bounding boxes predicted by GP2C in the middle of Table 8. Even though the performance drops from 91% to 72%, this result is very encouraging since our viewpoint estimation network has only trained on 10 annotated samples for each testing category while previous methods has trained on thousands of annotated samples. Besides using only a small number of annotated training samples of the target classes, our viewpoint estimation network is trained on ObjectNet3D images and directly tested on Pix3D images, while Fine-grained [94] and GP2C [93] use images from the same dataset for training and testing. Therefore, our setting is much harder compared to [93], [94]. We then report our results evaluated using predicted bounding boxes given by our few-shot object detector at the bottom of Table 8. The overall performance drops around 20% points compared to the evaluation using bounding boxes predicted by GP2C, where the detection network is pre-trained on all 80 object categories of MS-COCO and fine-tuned on the 4 categories of Pix3D. In both cases, our viewpoint estimation method using 3D models performs better than our method without 3D models. This consistent improvement demonstrates again the benefits of adding 3D information in viewpoint estimation.

4.4 Limitations

Our work shares a common limitation with other work on viewpoint estimation in that it does not handle very well small objects, which have less visible cues, and objects that are nearly symmetrical, such as *knives*. In the latter case, a wrong prediction of the front-back orientation can result in a very large prediction error, although the rendered views can be very similar to the actual images. It is even more so in the few-shot setting, where only a few labeled samples are provided for the novel categories. Preventing such failure cases could require a specific treatment of almost-symmetries.

Also, as discussed in the introduction regarding the case where we do not use 3D model information for viewpoint estimation but a class-agnostic approach instead, we rely on the fact that objects of different but related classes often are consistently oriented, with aligned similarities. While it is the case for all datasets we know of, this fact is not guaranteed.

Yet, in case of orientation discrepancies between classes, a dataset can somehow be "normalized" before training by applying systematic rotations of ground-truth viewpoints.

5 Conclusion and Perspectives

In this work, we presented an approach to few-shot object detection and viewpoint estimation that can tackle both tasks in a coherent and efficient framework. We demonstrated the benefits of this approach in terms of accuracy, and significantly improved the state of the art on several standard benchmarks for few-shot object detection and few-shot viewpoint estimation. Moreover, we showed that our few-shot viewpoint estimation model can achieve promising results on the novel objects detected by our few-shot detection model, compared in an adversarial setting to other existing methods tested on perfect detection, *i.e.*, ground-truth classes and ground-truth bounding boxes.

This is of particular interest for scene understanding in weakly-controlled environments, such as robotic manipulation with various objects in the wild. In future work, we are interested in developing category-agnostic models that can detect arbitrary objects and estimate their poses without seeing them during training. We will also expand our approach to perform 3D model retrieval and estimation refinement by selecting the 3D candidate that best agrees with the measured visual evidence, which might include RGB images, depth maps, and deep features extracted by a neural network. The exploitation of multiple views and additional inputs such as depth maps could also be considered.

References

[1] Y.-X. Wang, D. Ramanan, and M. Hebert, "Meta-learning to detect rare objects," in *IEEE International Conference on Computer Vision (ICCV)*, 2019.

[2] B. Kang, Z. Liu, X. Wang, F. Yu, J. Feng, and T. Darrell, "Few-shot object detection via feature reweighting," in *IEEE International Conference on Computer Vision (ICCV)*, 2019.

[3] X. Yan, Z. Chen, A. Xu, X. Wang, X. Liang, and L. Lin, "Meta R-CNN : Towards general solver for instance-level low-shot learning," in *IEEE International Conference on Computer Vision (ICCV)*, 2019.

[4] Z. Fan, J.-G. Yu, Z. Liang, J. Ou, C. Gao, G. Xia, and Y. Li, "Fgn: Fully guided network for few-shot instance segmentation," *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[5] O. Vinyals, C. Blundell, T. P. Lillicrap, K. Kavukcuoglu, and D. Wierstra, "Matching networks for one shot learning," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2016.

[6] W.-Y. Chen, Y.-C. Liu, Z. Kira, Y.-C. Wang, and J.-B. Huang, "A closer look at few-shot classification," in *International Conference on Learning Representations (ICLR)*, 2019.

[7] X. Wang, T. E. Huang, T. Darrell, J. E. Gonzalez, and F. Yu, "Frustratingly simple few-shot object detection," in *International Conference on Machine Learning (ICML)*, 2020.

[8] X. Zhou, A. Karpur, L. Luo, and Q. Huang, "StarMap for category-agnostic keypoint and viewpoint estimation," in *European Conference on Computer Vision (ECCV)*, 2018.

[9] Y. Xiao, X. Qiu, P. Langlois, M. Aubry, and R. Marlet, "Pose from shape: Deep pose estimation for arbitrary 3D objects," in *British Machine Vision Conference (BMVC)*, 2019.

[10] H.-Y. Tseng, S. D. Mello, J. Tremblay, S. Liu, S. Birchfield, M.-H. Yang, and J. Kautz, "Few-shot viewpoint estimation," in *British Machine Vision Conference (BMVC)*, 2019.

[11] C. Zhang, G. Lin, F. Liu, R. Yao, and C. Shen, "CANet: Class-agnostic segmentation networks with iterative refinement and attentive few-shot learning," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[12] S.-D. Yang, H.-T. Su, W. H. Hsu, and W.-C. Chen, "Class-agnostic few-shot object counting," in *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2021.

[13] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu, "ShapeNet: An Information-Rich 3D Model Repository," Stanford University — Princeton University — Toyota Technological Institute at Chicago, Tech. Rep. arXiv:1512.03012 [cs.GR], 2015.

[14] Y. Xiao and R. Marlet, "Few-shot object detection and viewpoint estimation for objects in the wild," in *European Conference on Computer Vision (ECCV)*, 2020.

[15] F.-F. Li, R. Fergus, and P. Perona, "One-shot learning of object categories," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2006.

[16] B. Hariharan and R. B. Girshick, "Low-shot visual recognition by shrinking and hallucinating features," in *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[17] E. Schwartz, L. Karlinsky, J. Shtok, S. Harary, M. Marder, A. Kumar, R. Feris, R. Giryes, and A. Bronstein, "Delta-encoder: an effective sample synthesis method for few-shot object recognition," in *Advances in Neural Information Processing Systems (NeuIPS)*, 2018.

[18] Y.-X. Wang, R. B. Girshick, M. Hebert, and B. Hariharan, "Low-shot learning from imaginary data," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[19] Y. Xian, S. Sharma, B. Schiele, and Z. Akata, "F-vaegan-d2: A feature generating framework for any-shot learning," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[20] L. Bertinetto, J. F. Henriques, J. Valmadre, P. H. S. Torr, and A. Vedaldi, "Learning feed-forward one-shot learners," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2016.

[21] M. Andrychowicz, M. Denil, S. G. Colmenarejo, M. W. Hoffman, D. Pfau, T. Schaul, B. Shillingford, and N. de Freitas, "Learning to learn by gradient descent by gradient descent," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2016.

[22] J. Snell, K. Swersky, and R. S. Zemel, "Prototypical networks for few-shot learning," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[23] H. Hu, J. Gu, Z. Zhang, J. Dai, and Y. Wei, "Relation networks for object detection," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[24] S. Ravi and H. Larochelle, "Optimization as a model for few-shot learning," in *International Conference on Learning Representations (ICLR)*, 2017.

[25] K. Lee, S. Maji, A. Ravichandran, and S. Soatto, "Meta-learning with differentiable convex optimization," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[26] S. X. Hu, P. Moreno, Y. Xiao, X. Shen, G. Obozinski, N. Lawrence, and A. Damianou, "Empirical Bayes transductive meta-learning with synthetic gradients," in *International Conference on Learning Representations (ICLR)*, 2020.

[27] B. N. Oreshkin, P. R. López, and A. Lacoste, "Tadam: Task dependent adaptive metric for improved few-shot learning," in *Advances in Neural Information Processing Systems (NeuIPS)*, 2018.

[28] H. Li, D. Eigen, S. F. Dodge, M. D. Zeiler, and X. Wang, "Finding task-relevant features for few-shot learning by category traversal," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[29] R. Hou, H. Chang, B. Ma, S. Shan, and X. Chen, "Cross attention network for few-shot classification," in *Advances in Neural Information Processing Systems (NeuIPS)*, 2019.

[30] H. Li, W. Dong, X. Mei, C. Ma, F. Huang, and B. Hu, "Lgm-net: Learning to generate matching networks for few-shot learning," in *International Conference on Machine Learning (ICML)*, 2019.

[31] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *International Conference on Machine Learning (ICML)*, 2017.

[32] C. Finn, K. Xu, and S. Levine, "Probabilistic model-agnostic meta-learning," in *Advances in Neural Information Processing Systems (NeuIPS)*, 2018.

[33] Q. Sun, Y. Liu, T.-S. Chua, and B. Schiele, "Meta-transfer learning for few-shot learning," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[34] S. Gidaris and N. Komodakis, "Dynamic few-shot visual learning without forgetting," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[35] S. Qiao, C. Liu, W. Shen, and A. Yuille, "Few-shot image recognition by predicting parameters from activations," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[36] Y. Liu, J. Lee, M. Park, S. Kim, E. Yang, S. Hwang, and Y. Yang, "Learning to propagate labels: Transductive propagation network

for few-shot learning," in *International Conference on Learning Representations (ICLR)*, 2019.

[37] M. Ren, E. Triantafillou, S. Ravi, J. Snell, K. Swersky, J. B. Tenenbaum, H. Larochelle, and R. S. Zemel, "Meta-learning for semi-supervised few-shot classification," in *International Conference on Learning Representations ICLR*, 2018.

[38] V. Garcia and J. Bruna, "Few-shot learning with graph neural networks," in *International Conference on Learning Representations ICLR*, 2018.

[39] R. Girshick, "Fast R-CNN," in *IEEE International Conference on Computer Vision (ICCV)*, 2015.

[40] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems (NeuIPS)*, 2015.

[41] K. He, G. Gkioxari, P. Dollár, and R. B. Girshick, "Mask R-CNN," in *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[42] T.-Y. Lin, P. Dollár, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie, "Feature pyramid networks for object detection," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[43] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[44] J. Redmon and A. Farhadi, "Yolo9000: Better, faster, stronger," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[45] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. E. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *European Conference on Computer Vision (ECCV)*, 2016.

[46] C. Hao, W. Yali, W. Guoyou, and Q. Yu, "LSTD: A low-shot transfer detector for object detection," in *AAAI Conference on Artificial Intelligence (AAAI)*, 2018.

[47] H. O. Song, Y. J. Lee, S. Jegelka, and T. Darrell, "Weakly-supervised discovery of visual pattern configurations," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2014.

[48] H. Bilen and A. Vedaldi, "Weakly supervised deep detection networks," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[49] A. Diba, V. Sharma, A. M. Pazandeh, H. Pirsiavash, and L. V. Gool, "Weakly supervised cascaded convolutional networks," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[50] I. Misra, A. Shrivastava, and M. Hebert, "Watch and learn: Semi-supervised learning of object detectors from videos," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[51] Y.-X. Wang and M. Hebert, "Model recommendation: Generating object detectors from few samples," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[52] X. Dong, L. Zheng, F. Ma, Y. Yang, and D. Meng, "Few-example object detection with model communication," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2019.

[53] A. Bansal, K. Sikka, G. Sharma, R. Chellappa, and A. Divakaran, "Zero-shot object detection," in *European Conference on Computer Vision (ECCV)*, 2018.

[54] S. Rahman, S. H. Khan, and F. M. Porikli, "Zero-shot object detection: Learning to simultaneously recognize and localize novel concepts," in *Asian Conference on Computer Vision (ACCV)*, 2018.

[55] P. Zhu, H. Wang, and V. Saligrama, "Zero shot detection," *IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)*, 2019.

[56] S. Tulsiani and J. Malik, "Viewpoints and keypoints," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[57] H. Su, C. R. Qi, Y. Li, and L. J. Guibas, "Render for CNN: Viewpoint estimation in images using CNNs trained with rendered 3D model views," in *IEEE International Conference on Computer Vision (ICCV)*, 2015.

[58] A. Mousavian, D. Anguelov, J. Flynn, and J. Kosecka, "3D bounding box estimation using deep learning and geometry," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[59] Y. Xiao, Y. Du, and R. Marlet, "Posecontrast: Class-agnostic object viewpoint estimation in the wild with pose-aware contrastive learning," in *International Conference on 3D Vision (3DV)*, 2021.

[60] M. Sundermeyer, Z.-C. Marton, M. Durner, M. Brucker, and R. Triebel, "Implicit 3D orientation learning for 6D object detection from RGB images," in *European Conference on Computer Vision (ECCV)*, 2018.

[61] M. Sundermeyer, M. Durner, E. Y. Puang, Z.-C. Márton, and R. Triebel, "Multi-path learning for object pose estimation across domains," *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[62] M. Rad and V. Lepetit, "BB8: A scalable, accurate, robust to partial occlusion method for predicting the 3D poses of challenging objects without using depth," in *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[63] A. Grabner, P. M. Roth, and V. Lepetit, "3D pose estimation and 3D model retrieval for objects in the wild," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[64] G. Pitteri, S. Ilic, and V. Lepetit, "CorNet: Generic 3D corners for 6D pose estimation of new objects without retraining," in *IEEE International Conference on Computer Vision Workshops (ICCVw)*, 2019.

[65] G. Pavlakos, X. Zhou, A. Chan, K. G. Derpanis, and K. Daniilidis, "6-DoF object pose from semantic keypoints," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2017.

[66] S. Suwajanakorn, N. Snavely, J. Tompson, and M. Norouzi, "Discovery of latent 3d keypoints via end-to-end geometric reasoning," *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[67] M. Rad, M. Oberweger, and V. Lepetit, "Feature mapping for learning fast and accurate 3d pose inference from synthetic images," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[68] Y. Labbé, J. Carpentier, M. Aubry, and J. Sivic, "Cosypose: Consistent multi-view multi-object 6d pose estimation," in *European Conference on Computer Vision (ECCV)*, 2020.

[69] X. Deng, Y. Xiang, A. Mousavian, C. Eppner, T. Bretl, and D. Fox, "Self-supervised 6d object pose estimation for robot manipulation," in *International Conference on Robotics and Automation (ICRA)*, 2020.

[70] S. K. Mustikovela, V. Jampani, S. De Mello, S. Liu, U. Iqbal, C. Rother, and J. Kautz, "Self-supervised viewpoint learning from image collections," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[71] G. Wang, F. Manhardt, J. Shao, X. Ji, N. Navab, and F. Tombari, "Self6d: Self-supervised monocular 6d object pose estimation," *European Conference on Computer Vision (ECCV)*, 2020.

[72] S. Tulsiani, J. Carreira, and J. Malik, "Pose induction for novel object categories," in *IEEE International Conference on Computer Vision (ICCV)*, 2015.

[73] K. Park, A. Mousavian, Y. Xiang, and D. Fox, "Latentfusion: End-to-end differentiable reconstruction and rendering for unseen object pose estimation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[74] A. Wang, S. Mei, A. Yuille, and A. Kortylewski, "Neural view synthesis and matching for semi-supervised few-shot learning of 3d pose," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[75] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[76] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "PointNet: deep learning on point sets for 3D classification and segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[77] P. Ammirato, C.-Y. Fu, M. Shvets, J. Kosecka, and A. C. Berg, "Target driven instance detection," 2018, arXiv preprint arXiv:1803.04610.

[78] W. Kuo, A. Angelova, J. Malik, and T.-Y. Lin, "ShapeMask: Learning to segment novel objects by refining shape priors," in *IEEE International Conference on Computer Vision (ICCV)*, 2019.

[79] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL visual object classes (voc) challenge," *International Journal of Computer Vision (IJCV)*, 2010.

[80] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL visual object classes challenge: A retrospective," *International Journal of Computer Vision (IJCV)*, 2015.

[81] T.-Y. Lin, M. Maire, S. J. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *European Conference on Computer Vision (ECCV)*, 2014.

[82] Q. Fan, W. Zhuo, C.-K. Tang, and Y.-W. Tai, "Few-shot object detection with attention-RPN and multi-relation detector," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[83] J. Wu, S. Liu, D. Huang, and Y. Wang, "Multi-scale positive sample refinement for few-shot object detection," in *European Conference on Computer Vision (ECCV)*, 2020.

[84] Y. Xiang, W. Kim, W. Chen, J. Ji, C. Choy, H. Su, R. Mottaghi, L. Guibas, and S. Savarese, "ObjectNet3D: A large scale database for 3D object recognition," in *European Conference on Computer Vision (ECCV)*, 2016.

[85] Y. Xiang, R. Mottaghi, and S. Savarese, "Beyond PASCAL: A benchmark for 3D object detection in the wild," in *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2014.

[86] S. Hinterstoißer, V. Lepetit, S. Ilic, S. Holzer, G. R. Bradski, K. Konolige, and N. Navab, "Model based training, detection and pose estimation of texture-less 3D objects in heavily cluttered scenes," in *Asian Conference on Computer Vision (ACCV)*, 2012.

[87] T. Hodan, P. Haluza, S. Obdrzálek, J. Matas, M. I. A. Lourakis, and X. Zabulis, "T-less: An rgb-d dataset for 6d pose estimation of texture-less objects," *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2017.

[88] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox, "PoseCNN: A convolutional neural network for 6D object pose estimation in cluttered scenes," in *Robotics: Science and Systems (RSS)*, 2018.

[89] E. Brachmann, A. Krull, F. Michel, S. Gumhold, J. Shotton, and C. Rother, "Learning 6d object pose estimation using 3d object coordinates," in *European Conference on Computer Vision (ECCV)*, 2014.

[90] H. Wang, S. Sridhar, J. Huang, J. Valentin, S. Song, and L. J. Guibas, "Normalized object coordinate space for category-level 6D object pose and size estimation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[91] A. Grabner, P. Roth, and V. Lepetit, "Location field descriptor: Single image 3D model retrieval in the wild," in *International Conference on 3D Vision (3DV)*, 2019.

[92] X. Sun, J. Wu, X. Zhang, Z. Zhang, C. Zhang, T. Xue, J. B. Tenenbaum, and W. T. Freeman, "Pix3d: Dataset and methods for single-image 3d shape modeling," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[93] A. Grabner, P. Roth, and V. Lepetit, "Gp2c: Geometric projection parameter consensus for joint 3d pose and focal length estimation in the wild," in *IEEE International Conference on Computer Vision (ICCV)*, 2019.

[94] Y. Wang, X. Tan, Y. Yang, X. Liu, E. Ding, F. Zhou, and L. S. Davis, "3d pose estimation for fine-grained object categories," in *European Conference on Computer Vision Workshop (ECCVw)*, 2018.

Yang Xiao is a Ph.D. candidate in Computer Science at Ecole des Ponts ParisTech, France. He received his M.S. in Signal and Image Processing from University Paris-Saclay and B.S. in Optical and Electronic Information from Huazhong University of Science and Technology. His research interests include object pose estimation and 3D scene understanding in computer vision.

Vincent Lepetit is a research director at ENPC ParisTech, France. Before that, he was a full professor at the Institute for Computer Graphics and Vision, Graz University of Technology, and before that, a senior researcher at CVLab, EPFL, Switzerland. He currently focuses on 3D scene understanding from images, with application to 3D hand and object tracking, 3D reconstruction, and camera localization. With his co-authors, he received the Koenderick 'test of time' award for the BRIEF local descriptor in 2020.

Renaud Marlet is a Senior Researcher at École des Ponts ParisTech (ENPC) and a Principal Scientist at valeo.ai, France. He has held positions both in academia (researcher at Inria) and in the software industry (expert at Simulog, deputy CTO of Trusted Logic). He was the head of the IMAGINE group at LIGM/ENPC (2010-2019). He is currently interested in scene understanding and semantized 3D reconstruction, with applications to robotics, autonomous driving and civil engineering.